

ตารางสมการ

ลำดับที่	สมการ	ความหมาย และที่มา
2.1	$n_1 \sin \theta_1 = n_2 \sin \theta_2$	ดัชนีการหักเหของแสงของวัสดุ 1 และ 2 ที่มา : เทคนิคการส่องสว่าง หน้า 1-18
2.2	Solid angle (ω) = A / R² steradian	อัตราส่วนพื้นที่ผิวทรงกลมต่อรัศมีของทรงกลมกำลังสอง มีหน่วยเป็น สเตอเรเดียน (Steradian) ที่มา :Mechanical and Electrical Equipment for building
2.3	$E = I / d^2$	ปริมาณความส่องสว่างบนพื้นผิวที่พิจารณา (ลักซ์) ที่มา :Mechanical and Electrical Equipment for building
2.4	$FL = Fc \times \rho$	ปริมาณความจ้า หน่วยเป็น ฟุตแลมเบิร์ต ที่มา :Mechanical and Electrical Equipment for building
2.5	$FL = Fc \times l$	ปริมาณการส่องสว่าง หน่วยเป็น ฟุตแคนเดิล ที่มา :Mechanical and Electrical Equipment for building
2.6	$C = (L2 - L1) / L1$	ความเปรียบต่าง (Contrast) ที่มา : เทคนิคการส่องสว่าง หน้า 1-1
2.7	$ET = 0.170 \sin(4\pi (J - 80)/373)$ $- 0.129 \sin (2\pi (J - 8) / 355)$	เวลาเทียบเท่า (Equation of time) ที่มา : 1989 ASHRAE Handbook Fundamentals. I-P Edition, page 27.3
2.8	$t_s = t_d - 1$	เวลามาตรฐาน (หน่วยเป็นทศนิยม) ที่มา : 1989 ASHRAE Handbook Fundamentals. I-P Edition, page 27.3
2.9	$t = t_s + ET + (12 (SM - L)) / \pi$	เวลาคงอาทิตย์ (หน่วยเป็นทศนิยม) ที่มา : 1989 ASHRAE Handbook Fundamentals. I-P Edition.
2.10	$\delta = 0.4093 \sin (2\pi (J - 81) / 368)$	มุมคล้อยต่ำของดวงอาทิตย์ (หน่วยเรเดียน) ที่มา : 1989 ASHRAE Handbook Fundamentals. I-P Edition.
2.11	$a_1 = \arcsin(\sin l \sin \delta - \cos l \cos \delta \cos(\pi\tau/12))$	มุมอัลติจูดดวงอาทิตย์ (หน่วยเรเดียน) ที่มา : 1989 ASHRAE Handbook Fundamentals. I-P Edition.
2.12	$a_s = \arctan ((- [\cos \delta \sin (\pi\tau/12)] /$ $- [\cos l \sin + \sin l \cos \delta \cos(\pi\tau/12)]))$	มุมอัลซิมูทดวงอาทิตย์ (หน่วยเรเดียน) ที่มา : 1989 ASHRAE Handbook Fundamentals. I-P Edition.
2.13	$a_z = a_s - a_o$	ระดับมุมอัลซิมูทดวงอาทิตย์ (หน่วยเรเดียน) ที่มา : 1989 ASHRAE Handbook Fundamentals. I-P Edition.
2.14	$a_1 = \arccos (\cos a_1 / \cos a_2)$	ค่ามุมอาร์คคอร์ส (หน่วยเรเดียน) ที่มา : 1989 ASHRAE Handbook Fundamentals. I-P Edition.

ลำดับที่	สมการ	ความหมาย และที่มา
2.15	$a_p = \arctan (\sin a_i / \cos a_z)$	มุมโพไรไฟล์ (หน่วยเรเดียน) ที่มา : 1989 ASHRAE Handbook Fundamentals. I-P Edition.
2.16	$a_p = \arctan (\tan a_i / \cos a_z)$	มุมโพไรไฟล์ (หน่วยเรเดียน) ที่มา : 1989 ASHRAE Handbook Fundamentals. I-P Edition.
2.17	$E_{sc} = Km \int_{300}^{770} G_\lambda V_\lambda d\lambda$	ค่าคงที่ความส่องสว่างดวงอาทิตย์ หน่วยกิโวลต์ (klx) ที่มา : Daylighting , page 363
2.18	$E_{xt} = E_{sc}(1 + 0.034\cos(2\pi(J-2)/365))$	ค่าความส่องสว่างของแสงอาทิตย์นอกบรรยากาศของโลก หน่วยเป็น กิโวลต์ (klx) ที่มา : Daylighting , page 363
2.19	$E_{dn} = E_{xt} e^{-cm}$	ค่าความส่องสว่างของแสงจากดวงอาทิตย์ ที่ระดับน้ำทะเล หน่วยเป็น klx ที่มา : Daylighting , page 363
2.20	$m = 1 / \sin a_i$	ค่ามวลอากาศ (Optical air mass , Dimensionless) ที่มา : Daylighting , page 363
2.21	$Edh = Edn \sin a_i$	ค่าความสว่างของแสงอาทิตย์ในแนวราบ หน่วยกิโวลต์ ที่มา : Daylighting , page 363
2.22	$Edv = Edn \cos a_i$	ค่าความสว่างของแสงอาทิตย์ในแนวตั้ง หน่วยกิโวลต์ ที่มา : Daylighting , page 363
2.23	$E_{nh} = A + B \sin^c a_i$	ค่าความส่องสว่างในแนวราบของท้องฟ้า ที่ปราศตาสีงปก คลุม หน่วยเป็นกิโวลต์ (klx) ที่มา : Daylighting , page 364
2.24	$L_z = E_{nh} ZL$	ค่าความส่องสว่างที่เซนติ (กิโวลต์เดลาต่อตารางเมตร) ที่มา : Daylighting , page 364
2.25	$Z_0 = \pi/2 - at$	ค่ามุมเซนติ ที่ระดับต่างๆ ที่มา : Daylighting , page 364
2.26	$L_{\zeta,\alpha} = L_z (0.91+10e^{-3\gamma}+0.45\cos^2\gamma)(1-e^{-0.32/\cos\zeta})$ $/ ((0.91+10e^{-3Z_0}+0.45\cos^2Z_0)(1-e^{-0.32}))$	ความสว่างของท้องฟ้า ครั้งทรงกลมที่เกิดจากมุม ζ และ α (หน่วยกิโวลต์เดลาต่อตารางเมตร ,kcd/m ²) ที่มา : Daylighting , page 364
2.27	$\gamma = \arccos(\cos Z_0 \cos \zeta + \sin Z_0 \sin \zeta \cos \alpha)$	ค่ามุมระหว่างดวงอาทิตย์ และตำแหน่ง P บนท้องฟ้า ที่มา : Daylighting , page 364
2.28	$L_{\zeta,\alpha} = L_z (0.526 + 5e^{-1.5\gamma})(1 - e^{-0.80/\cos\gamma})$ $/ ((0.526 + 5e^{-1.5Z_0}) + (1 - e^{-0.80}))$	ค่ามุมระหว่างดวงอาทิตย์ และตำแหน่ง P บนท้องฟ้า ที่มา : Daylighting , page 364
2.29	$L_{\zeta,\alpha} = L_z ((0.864(e^{-0.52\cos\zeta} / e^{-0.52}))$ $+ 0.136(1 - e^{-0.82/\cos\zeta}) / (1 - e^{-0.52}))$	ค่ามุมระหว่างดวงอาทิตย์ และตำแหน่ง P บนท้องฟ้า ที่มา : Daylighting , page 364
2.30	$L_{\zeta,\alpha} = (L_z / 3) (1 + 2\cos\zeta)$	ค่าความสว่างของท้องฟ้า หน่วยเป็นหน่วย เป็นกิโวลต์เดลาต่อตารางเมตร (kcd/m ²) ที่มา : Daylighting , page 364

ลำดับที่	สมการ	ความหมาย และที่มา
2.31	$dE_{kh} = L_{\zeta,\alpha} \cos\zeta d\omega = L_{\zeta,\alpha} \cos\zeta \sin\zeta d\zeta d\alpha$	ค่าความส่องสว่างบนระนาบราบ หน่วยกิโวลต์ (klx) ที่มา : Daylighting , page 365
2.32	$E_{kh} = 1/\pi \iint L_{\zeta,\alpha} \sin\zeta \cos\zeta d\zeta d\alpha$	ค่าความส่องสว่างบนระนาบราบ หน่วยกิโวลต์ (klx) ที่มา : Daylighting , page 365
2.33	$E_{kh} = 1/\pi \int_0^{2\pi} \int_0^{\pi/2} L_{\zeta,\alpha} \sin\zeta \cos\zeta d\zeta d\alpha$	ค่าความส่องสว่างบนระนาบราบ กำหนดขอบเขต ครึ่งท้องฟ้า หน่วยกิโวลต์ (klx) ที่มา : Daylighting , page 365
2.34	$E_{kv} = 1/\pi \int_{\alpha_z + \pi/2}^{\alpha_z + \pi/2} \int_0^{\pi/2} L_{\zeta,\alpha} \sin\zeta \cos\zeta d\zeta d\alpha$	ค่าความส่องสว่างบนระนาบตั้ง กำหนดขอบเขต ครึ่งท้องฟ้า หน่วยกิโวลต์ (klx) ที่มา : Daylighting , page 365
2.35	$LA = LZ (1 + 2 \sin A) / 3$	ความสว่างของท้องฟ้า ที่ตำแหน่งมุม A องศาเหนือ ตามแนวระนาบของดวงอาทิตย์ ที่มา : Daylighting design & analysis, 1986
2.36	$E_h = 300 + 21,000 \sin A \text{ (lux)}$	ความสว่างของท้องฟ้าจากแสงตรงของดวงอาทิตย์ ที่มา : คมกฤช ชูเกียรติมัน, วิทยานิพนธ์ 2540
2.37	$E_h = 1345 + 14,795 \sin A \text{ (lux)}$	ความสว่างของท้องฟ้าจากแสงกระจายของท้องฟ้า ที่มา : คมกฤช ชูเกียรติมัน, วิทยานิพนธ์ 2540
2.38	$E_{HP} = 570 \text{ A}$	ความสว่างภายนอกที่ระดับระนาบ ภายใต้สภาพท้องฟ้า มีเมฆปกคลุมที่บ มีหน่วยเป็นกิโวลต์ ที่มา : คมกฤช ชูเกียรติมัน, วิทยานิพนธ์ 2540
2.39	$E_H = 0.35 E_s + 0.89 E_c$	ความส่องสว่างภายนอกที่ระดับแนวระนาบ ภายใต้ ท้องฟ้าแบบเมฆปกคลุมบางส่วน (มีหน่วยลักซ์) ที่มา : คมกฤช ชูเกียรติมัน, วิทยานิพนธ์ 2540
2.40	$L_A = L_z (1 + 2 \sin A) / 3$	ความสว่างของท้องฟ้าที่ตำแหน่งมุม A องศา เหนือ ระดับในแนวระนาบ (Horizon) ในทิศทางต่างๆ ที่มา : คมกฤช ชูเกียรติมัน, วิทยานิพนธ์ 2540
2.41	$E_H = 300 + 21,000 \sin A \text{ (lux)}$	ความส่องสว่างภายนอกที่ระดับแนวระนาบ ภายใต้ ท้องฟ้ามีเมฆปกคลุมบางส่วน (มีหน่วยลักซ์) ที่มา : คมกฤช ชูเกียรติมัน, วิทยานิพนธ์ 2540
2.42	$E_{sp} = (E_e) (A_g) (T_g) (LLF) (CU)$	ค่าระดับความส่องสว่าง ณ บริเวณจุดที่ต้องการพิจารณา ที่มา: Mechanical and Electrical Equipment for building
2.43	$E_{sp} = [(E_{sv}) (A_g) (T_g) (Cs) (Ks)]$ $+ [(E_{gv}) (A_g) (T_g) (Cg) (Kg)]$	ค่าระดับความส่องสว่าง ณ บริเวณจุดที่ต้องการพิจารณา เมื่อคำนวณด้วยวิธีลูเมนเมตรอด ที่มา: Mechanical and Electrical Equipment for building
2.44	$E_{s,o} = E_{gv,o} - [E_{gh,o} * 0.1]$	ค่าความสว่างจากท้องฟ้ามีเมฆปกคลุมบางส่วน ที่ตกกระทบพื้นผิวในแนวตั้ง ที่มา : Daylighting design & analysis, 1986
2.45	$E_{g,o} = E_{gh,o} * 0.1$	ค่าความสว่างจากการสะท้อนของพื้นดิน ภายใต้สภาพท้องฟ้า มีเมฆปกคลุมบางส่วน ที่ตกกระทบผิวระนาบตั้ง ที่มา : Daylighting design & analysis, 1986

ลำดับที่	สมการ	ความหมาย และที่มา
2.46	$E_{s,c} = E_{dv,c}$	ค่าความสว่างของแสงกระจายจากท้องฟ้าโปร่ง ที่ตกกระทบพื้นผิวในแนวตั้ง ที่มา : Daylighting design & analysis, 1986
2.47	$E_{g,c} = E_{GH,c}$	ค่าความสว่างรวมที่ตกกระทบพื้นผิวในแนวระนาบ ที่มา : Daylighting design & analysis, 1986
2.48	$E_{s,c} = E_{GV,c} - [E_{GH,c} * 0.1]$	ค่าความสว่างรวมที่ตกกระทบพื้นผิวระนาบ ที่มา : Daylighting design & analysis, 1986
2.49	$E_{s,u} = E_{gu}$	ปริมาณความสว่างที่ตกกระทบพื้นผิวระนาบตั้ง จากท้องฟ้า (E_{vk}) และพื้นดิน (E_{vg}) (หน่วยฟุตแคนเดิล) ที่มา : Daylighting design & analysis, 1986
2.50	$E_{s,u} = 0.5 * E_{GV}$	ปริมาณความสว่างที่ตกกระทบพื้นผิวระนาบตั้ง เมื่อสภาพท้องฟ้ามีเมฆปกคลุมบางส่วน (หน่วยฟุตแคนเดิล) ที่มา : Daylighting design & analysis, 1986
2.51	$DF_o = (E_i / E_e) \times 100$	ค่าปริมาณความส่องสว่างเฉลี่ยไลท์แฟกเตอร์ ตามอัตราส่วนระหว่าง ความส่องสว่างภายใน และภายนอกอาคาร ที่มา: Mechanical and Electrical Equipment for building
2.52	$DF(\%) = \frac{\text{ความสว่างภายใน}}{\text{ความสว่างภายนอกอาคาร}} \times 100$	ค่าปริมาณความส่องสว่างเฉลี่ยไลท์แฟกเตอร์ ตามอัตราส่วนระหว่าง ความส่องสว่างภายใน และภายนอกอาคาร ในกรณี ที่ค่าความส่องสว่างคิดเป็นเปอร์เซ็นต์ ที่มา: Mechanical and Electrical Equipment for building
2.53	$DF_o = SC_o + ERC_o + IRC_o$	ค่าเฉลี่ยไลท์แฟกเตอร์ จากองค์ประกอบแสงตรงดวงอาทิตย์, องค์ประกอบแสงสะท้อนภายนอกอาคาร และองค์ประกอบแสงสะท้อนภายในอาคาร ที่มา : Daylighting design & analysis, 1986
2.54	$DF_o = SC_o + ERC_o + [(MF)(IRC_o)]$	ค่าเฉลี่ยไลท์แฟกเตอร์ เมื่อเพิ่มสัมประสิทธิ์ ค่าความสกปรกภายในอาคาร ซึ่งจะมีผลต่อปริมาณค่าความส่องสว่างของการสะท้อนแสงภายในอาคาร ที่มา : Daylighting design & analysis, 1986
4.1	$E_{d,c} = E_{o,c} \cos \theta$	ความสว่างรังสีตรงจากดวงอาทิตย์ ภายใต้ท้องฟ้าโปร่ง ที่มา : Daylighting design & analysis, 1986
4.2	$E_{DH,c} = E_{DN} \cos \theta_0$	ความสว่างรังสีตรงจากดวงอาทิตย์ บนระนาบนอน ภายใต้ท้องฟ้าโปร่ง ที่มา : Daylighting design & analysis, 1986
4.3	$E_{DN} = E_{sc} e^{-\alpha_n}$	ปริมาณรังสีตรงดวงอาทิตย์ เมื่อ E_{sc} คือปริมาณคงที่รังสีดวงอาทิตย์ = 128 klux ที่มา : Daylighting design & analysis, 1986
4.4	$\alpha_1 = (\tau) \sec \theta_0$	ความสกปรกของท้องฟ้าต่ำ (Low turbidity) ที่มา : Daylighting design & analysis, 1986

ลำดับที่	สมการ	ความหมาย และที่มา
4.5	$\alpha_2 = BmT$	ความสกปรกของท้องฟ้าสูง (High turbidity) ที่มา : Daylighting design & analysis, 1986
4.6	$m = 1 / (\cos \theta_0 + 0.158(93.885 - \theta_0)^{-1.253})$	ค่ามวลอากาศ (Air mass) ที่มา : Daylighting design & analysis, 1986
4.7	$T = [(h + 85)/(39.5e^{-w} + 47.4) + 0.1] + (16 + 0.22w)\beta$	ค่าความสกปรกของอากาศ (Atmosphere Turbidity) ที่มา : Daylighting design & analysis, 1986
4.8	$Ed_{h,c} = Ed_{back} + Ed_{circsol}$	ความส่องสว่างกระจาย บนพื้นระนาบนอน ภายใต้สภาพท้องฟ้าโปร่ง (Prof. Littlefair, 1965) ที่มา : Daylighting design & analysis, 1986
4.9	$Ed_{h,c} = (r) Ed_n / (CNd)^2$	ความส่องสว่างกระจาย บนพื้นระนาบนอน ภายใต้สภาพท้องฟ้าโปร่ง (Prof. Robins-Hunter) ที่มา : Daylighting design & analysis, 1986
4.10	$E_{g,c} = (E_{gh,c}) (F_g) (\rho_g)$	ค่าการสะท้อนแสงจากพื้นดิน ภายใต้สภาพท้องฟ้าโปร่ง ที่มา : Daylighting design & analysis, 1986
4.11	$F_g = (1 - \cos \theta) / 2$	ค่าของมุมระหว่างพื้นผิวและระนาบพื้นดิน ที่มา : Daylighting design & analysis, 1986
4.12	$E_{GV,c} = E_{DV,c} + Ed_{V,c} + E_{g_{V,c}}$	แสงสว่างรวมบนระนาบตั้ง ภายใต้ท้องฟ้าโปร่ง ที่มา : Daylighting design & analysis, 1986
4.13	$Ed_{v, orient,c} = (b_0 + b_1h + b_2h^2 + b_3h^3)$	สมการถดถอยเพื่อคาดการณ์แสงสว่างในระนาบตั้งในทิศทางต่างๆ ที่มา : Daylighting design & analysis, 1986
4.14	$E_{gv} = 0.1 (E_{DN} \cos \theta_0 + Ed_{h,c})$	ค่าการสะท้อนแสงของพื้นผิวในระนาบตั้ง ที่มา : Daylighting design & analysis, 1986
4.15	$E_{G,O} = E_{D,O} + Ed_{,o} + E_{g,o}$	ความสว่างบนระนาบในทิศทางต่างๆ ภายใต้สภาพท้องฟ้า ที่มีเมฆปกคลุมทึบ ที่มา : Daylighting design & analysis, 1986
4.16	$E_{GH,O} = a(C_1h + C_2h^2 + C_3h^3 + C_4h^4 + C_5h^5)$	ความสว่างรวม ในระนาบนอน(Global illuminance, $E_{GH,O}$) ที่มา : Daylighting design & analysis, 1986
4.17	$E_{GV,O} = [a(E_1h + E_2h^2 + E_3h^3 + E_4h^4 + E_5h^5)] + [(0.5)(\rho_g)(E_{GH,O})]$	ความสว่างรวม ในระนาบตั้ง(Global illuminance, $E_{GV,O}$) ที่มา : Daylighting design & analysis, 1986
4.18	$E_{Dk,c} = [(E_{GH,c})(\cos k) + (E_{GV,south,c})(\sin k)(\cos l)]$	ค่าความสว่างรังสีตรง ที่ตกกระทบบนพื้นผิวที่มีความลาดชัน ในทิศทางต่างๆ ที่มา : Daylighting design & analysis, 1986
4.19	$E_{Dk,c} = [(E_{GH,c})(\cos k) + (E_{GV,orient,c})(\sin k)]$	ความสว่างในระนาบตั้ง เมื่อกำหนดทิศทางในการคำนวณ ที่มา : Daylighting design & analysis, 1986
4.20	$Ed_{k,c} = [(E_{GH,c})(\cos k) + (E_{GV,orient,c})(\sin k)]$	ความสว่างกระจาย ที่ตกกระทบพื้นผิวที่มีความลาดชัน ภายใต้ท้องฟ้าโปร่ง ที่มา : Daylighting design & analysis, 1986

ลำดับที่	สมการ	ความหมาย และที่มา
4.21	$E_{gk,o} = [(E_{gh,c})(\cos k) + (E_{gv,orient,c})(\sin k)]$	ความสว่างกระจาย ที่ตกกระทบพื้นผิวที่มีระนาบลาดชัน ภายใต้ท้องฟ้าโปร่ง ที่มา : Daylighting design & analysis, 1986
4.22	$Ma = Exvu(Bua) + Exvk(Bka) + Exvg(Bga)$	สัมประสิทธิ์การบังแดดของอุปกรณ์บังแดดในระนาบนอน สำหรับด้านที่โดนบังแดด ที่มา : Daylighting design & analysis, 1986
4.23	$Mu = Exvu(Buu) + Exvk(Bku) + Exvg(Bgu)$	สัมประสิทธิ์การบังแดดของอุปกรณ์บังแดดในระนาบนอน สำหรับด้านที่ไม่โดนบังแดด ที่มา : Daylighting design & analysis, 1986
4.24	$Ma = Exvu(Bua) + 0.5(Exvk + Exvg)(Bka + Bga)$	สัมประสิทธิ์การบังแดดของอุปกรณ์บังแดดในระนาบตั้ง สำหรับด้านที่โดนบังแดด ที่มา : Daylighting design & analysis, 1986
4.25	$Mu = Exvu(Buu) + 0.5(Exvk + Exvg)(Bku + Bgu)$	สัมประสิทธิ์การบังแดดของอุปกรณ์บังแดดในระนาบตั้ง สำหรับด้านที่ไม่โดนบังแดด ที่มา : Daylighting design & analysis, 1986
4.26	$\int_{S_2} ((\cos\theta_1 \cos\theta_2) / \pi r^2) dS_2$	องค์ประกอบจากท้องฟ้าของแสงสว่าง ที่มา : Daylighting
4.27	(Illumination at P due to dS2 / Illumination at P due to complete hemisphere) = (Configuration factor / 1)	ความสว่าง บนพื้นผิวที่ 1 ของตำแหน่ง P อันเนื่องมาจากพื้นที่ dS ₂ ของพื้นผิวที่ 2 ที่มา : Daylighting
4.28	$\frac{FR}{4\pi R^2} = \frac{1}{(1 - R)}$	ผลรวมของปริมาณความสว่าง บนพื้นผิวทรงกลม อันเนื่องมาจากการสะท้อนแสงที่มีความต่อเนื่องไม่รู้จัก (Infinite) ที่มา : Daylighting
4.29	$\frac{\text{First reflected flux from interior surface}}{\text{Total area of internal surface} \times (1 - R)}$	ผลรวมของปริมาณความสว่าง บนพื้นผิวทรงกลม อันเนื่องมาจากการสะท้อนแสงที่มีความต่อเนื่องไม่รู้จัก ที่มา : Daylighting
4.30	$\text{Avg internally reflected} = \frac{E_w}{E_h} \frac{WR}{A(1 - R)}$	การสะท้อนแสงภายในอาคาร ที่มา : Daylighting
4.31	$FR = \sum (F_1R_1 + F_2R_2 + F_3R_3 + \dots)$	ค่าการสะท้อนแสงที่มีความต่อเนื่องกัน ที่มา : Daylighting
4.32	$\int_{x=1}^X \int_{y=0}^Y [Y / (1 + X^2 + Y^2)^{3/2}] dx dy$	ความส่องสว่างแสงตรงช่องเปิด ที่มา : Daylighting
4.33	$\frac{1}{2} [\tan^{-1}X - (1/\sqrt{1+Y^2}) \tan^{-1}(X/\sqrt{1+Y^2})]$	ความส่องสว่างแสงตรงช่องเปิด เมื่ออินทิเกรตสมการ 4.32 ที่มา : Daylighting
4.34	$EI = 1/2\pi [\tan^{-1}X - (1/\sqrt{1+Y^2}) \tan^{-1}(X/\sqrt{1+Y^2})]$	ความส่องสว่างแสงตรงจากช่องเปิด ที่มา : Daylighting

ลำดับที่	สมการ	ความหมาย และที่มา
4.35	$E_i = 50/\pi [\tan^{-1}X - (1/\sqrt{(1+Y^2)})\tan^{-1}\{X/\sqrt{(1+Y^2)}\}]%$	ความส่องสว่างแสงตรงจากช่องเปิด เมื่อคิดเป็นร้อยละ ที่มา : Daylighting
4.36	$E_i = 1/360 [\tan^{-1}X - (1/\sqrt{(1+Y^2)})\tan^{-1}\{X/\sqrt{(1+Y^2)}\}]$	ความส่องสว่างแสงตรงจากช่องเปิด กรณีใช้ค่ามุมเป็นเรเดียน ที่มา : Daylighting
4.37	$E_i = 1/3.6 [\tan^{-1}X - (1/\sqrt{(1+Y^2)})\tan^{-1}\{X/\sqrt{(1+Y^2)}\}]%$	ความส่องสว่างแสงตรงจากช่องเปิด เป็นร้อยละ กรณีใช้ค่ามุมเป็นเรเดียน ที่มา : Daylighting
4.38	$1/2\pi [\tan^{-1}W/D - (D/\sqrt{(H^2+D^2)})\tan^{-1}\{W/\sqrt{(H+D^2)}\}]$	ความส่องสว่างแสงตรงจากช่องเปิด กรณีที่แทนค่า X,Y ด้วยขนาดของช่องเปิด (W,D,H) ที่มา : Daylighting
4.39	$50/\pi [\tan^{-1}W/D - (D/\sqrt{(H^2+D^2)})\tan^{-1}\{W/\sqrt{(H+D^2)}\}]%$	ความส่องสว่างแสงตรงจากช่องเปิด เป็นร้อยละ กรณีที่แทนค่า X,Y ด้วยขนาดของช่องเปิด (W,D,H) ที่มา : Daylighting
4.40	$Sky\ Factor\ (SF) = 1/2\pi (\phi - \phi_1 \cos\theta)$ $= 50/\pi (\phi - \phi_1 \cos\theta)%$	ความส่องสว่างแสงตรงจากช่องเปิด กรณีที่แทนค่า X,Y ด้วยขนาดมุมเรเดียน ที่มา : Daylighting
4.41	$Sky\ Factor = 100\ WH^2/\pi D(D^2 + H^2)%$	ความส่องสว่างแสงตรงจากช่องเปิด เป็นร้อยละ กรณีที่แทนค่าต้องการผลการคำนวณอย่างคร่าวๆ ที่มา : Daylighting
4.42	$Sky\ Factor = 30\ WH^2/D(D^2 + H^2)%$	ความส่องสว่างแสงตรงจากช่องเปิด เป็นร้อยละ กรณีที่แทนค่าต้องการผลการคำนวณอย่างคร่าวๆ ที่มา : Daylighting
4.43	$Sky\ factor\ (SF) = 400\ LB/\pi(L^2 + D^2)%$	ความส่องสว่างแสงตรงจากช่องเปิด เป็นร้อยละ กรณีช่องเปิดอยู่บนระนาบนอน ที่มา : Daylighting
4.44	$Sky\ factor\ (SF) = 120\ LB/(L^2 + D^2)%$	ความส่องสว่างแสงตรงจากช่องเปิด เป็นร้อยละ กรณีช่องเปิดอยู่บนระนาบนอน ที่มา : Daylighting
4.45	$Sky\ factor = 50(1 - \cos A)%$	ความส่องสว่างแสงตรงจากช่องเปิด เป็นร้อยละ กรณีช่องเปิดทรงกลม ที่มา : Daylighting
4.46	$D^2 = 4LB/\pi$	เส้นผ่าศูนย์กลางของทรงกลม ที่มา : Daylighting
4.47	$Avg.\ I.R.C = \frac{W}{A(1-R)}(C1\ Rfw + C2\ Rcw)$	ค่าความส่องสว่างกระจายภายในอาคาร (Indirect Component) ที่มา : Daylighting
4.48	$Avg.\ I.R.C = \frac{W}{A(1-R)}(C1\ Rfw + 5Rcw)$	ค่าความส่องสว่างกระจายภายในอาคาร กรณีปราศจากสิ่งกีดขวางภายนอกอาคาร ที่มา : Daylighting

ลำดับที่	สมการ	ความหมาย และที่มา
4.49	$\text{Avg Ei} = \frac{(E_1 \times R_1) + (E_2 \times R_2) + \dots + (E_n \times R_n)}{(R_1 + R_2 + \dots + R_n)}$	ค่าเฉลี่ยพื้นที่ความส่องสว่างที่ต้องการ ที่มา : กนกวรรณ อุสันโน , วิทยานิพนธ์ 2540
4.50	$E = N \times L \times MF \times CU / A$	ความส่องสว่าง (ลักซ์ หรือ ฟุตแคนเดิล) ที่มา : เทคนิคการส่องสว่าง
4.51	$RCR = 5 \times H \times (L \times W) / (L \times W)$	ค่า Room Index (แบบ อเมริกา) ที่มา : เทคนิคการส่องสว่าง
4.52	$K = (L \times W) / (H \times (L + W))$	ค่า Room Index (แบบ อังกฤษ และ ญี่ปุ่น) ที่มา : เทคนิคการส่องสว่าง
4.53	$\text{Total watt of Lamp} = \frac{E \times \text{Area}}{\text{Efficacy} \times (CU \times LLF)}$	ความส่องสว่างภายใน จากแสงประดิษฐ์ด้วยวิธีลูเมน ที่มา : คมกฤษ ชูเกียรติมัน, วิทยานิพนธ์ 2540
4.54	$\text{Energy Consumption} = \text{Watt of Lamp} + \text{Ballast loss}$	การใช้พลังงานของหลอดไฟแสงประดิษฐ์ รวมการสูญเสียของบัลลาสต์ ที่มา : คมกฤษ ชูเกียรติมัน, วิทยานิพนธ์ 2540
4.55	$\text{Total Energy Consumption} = \text{No. of lamp} (\text{Watt of Lamp} + \text{Ballast loss})$	การใช้พลังงานทั้งหมดของหลอดไฟแสงประดิษฐ์ รวมการสูญเสียของบัลลาสต์ ที่มา : คมกฤษ ชูเกียรติมัน, วิทยานิพนธ์ 2540
4.56	$\text{Total Energy Consumption} = \frac{\text{Total Watt of Lamp} (1 + \text{Watt of Ballast Loss})}{\text{Watt of Lamp}}$	การใช้พลังงานทั้งหมดของหลอดไฟแสงประดิษฐ์ รวมการสูญเสียของบัลลาสต์ (เมื่อจัดรูปสมการใหม่) ที่มา : คมกฤษ ชูเกียรติมัน, วิทยานิพนธ์ 2540
5.1	$\text{Illuminance(D)} = \text{ABCD} - \text{AB} - \text{BC} + \text{B}$	ความส่องสว่างช่องเปิด D เมื่อตำแหน่งที่ต้องการทราบความส่องสว่างไม่ได้อยู่ในแนวเดียวกับช่องเปิด ที่มา : Daylighting
5.2	$\text{Illuminance(AD)} = (\text{AB} - \text{B}) + (\text{DC} - \text{C})$	ความส่องสว่างช่องเปิด AD เมื่อตำแหน่งที่ต้องการทราบความส่องสว่างไม่ได้อยู่ในแนวเดียวกับช่องเปิด ที่มา : Daylighting

บรรณานุกรม

ภาษาไทย

- กนกวรรณ อุสันโน. รูปแบบของอุปกรณ์บังแดดที่เหมาะสมสำหรับห้องเรียน. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต ภาควิชาสถาปัตยกรรมศาสตร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2539.
- คมกฤษ ชูเกียรติมั่น. การใช้แสงธรรมชาติเสริมเพื่อลดการใช้พลังงาน. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต ภาควิชาสถาปัตยกรรมศาสตร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2540.
- ชานาญ ท่อเกียรติ. เทคนิคการส่องสว่าง. กรุงเทพมหานคร : มหาวิทยาลัยเกษตรศาสตร์, 2540.
- บุญเลิศ เอี่ยมทัศน. เริ่มเรียนเขียนโปรแกรมบนวินโดวส์ด้วยเคลไฟล์. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2539.
- วิช วรรณประเสริฐ. ประสิทธิภาพในการนำแสงธรรมชาติมาใช้ในอาคารสำนักงานทั่วไป.
วิทยานิพนธ์ปริญญาโทมหาบัณฑิต ภาควิชาสถาปัตยกรรมศาสตร์
บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2537.
- วิทยาศาสตร์ เทคโนโลยีและสิ่งแวดล้อม, กระทรวง. พัฒนาและส่งเสริมพลังงานทดแทน. อนุรักษ์พลังงาน,
กอง. คู่มือการอนุรักษ์พลังงานในอาคาร. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : กองอนุรักษ์พลังงาน
กรมพัฒนาและส่งเสริมพลังงาน, 2536.
- สุนทร บุญญาธิการ, รศ. เทคนิคการออกแบบบ้านประหยัดพลังงาน เพื่อคุณภาพชีวิตที่ดีกว่า.
กรุงเทพมหานคร : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2542.
- สุนทร บุญญาธิการ, รศ. กรณีศึกษาอาคารอนุรักษ์พลังงานเฉลิมพระเกียรติการวิจัยประยุกต์เพื่อการ
ออกแบบอาคารประหยัดพลังงาน. วารสารอาษา (สิงหาคม 2539) : 60 – 120.
- สุรพงศ์ จิระรัตนานนท์. EFFICIENT LIGHTING AND DAYLIGHTING VOLUME 2. กรุงเทพมหานคร :
แผนกวิจัยพลังงาน สถาบันเทคโนโลยีแห่งเอเชีย, 2539.
- สุรพงศ์ จิระรัตนานนท์. DAYLIGHTING FOR BUILDINGS IN THE TROPIC II. บรรยายที่ห้อง ET238
แผนกวิจัยพลังงาน สถาบันเทคโนโลยีแห่งเอเชีย, 29 สิงหาคม 2541. (เอกสาร)

ภาษาอังกฤษ

- America Society of Heating, Refrigerating and Air Conditioning Engineerings.
1989 ASHRAE Handbook of Fundamental. I-P Edition. Atlanta Georgia, 1989.
- Benjamin Stein and John S.Reynolds. Mechanical and Electrical Equipment for Building 8th Edition.
Singapore : John Wiley & Sons, Inc., 1992.
- Claude L.Robbins. Daylighting design & analysis. New York : Van Nostrand Rein Hold company, 1986.
- Don A. York, Eva F. Tucker and Charlene C. Cappello. DOE-2 REFERENCE MANUAL (Version 2.1A)
Part 1. California : Lawrence Berkeley Laboratory, May 1981.
- Fuller Moore. Concept and Practice of Architectural Daylighting. New York : Van Nostrand Reinhold, 1984.
- Henry J.Cowan. Handbook of Architectural Technology. New York : Van Nostrand Rein Hold, 1991.
- IES. Illuminating Engineering Society of North America. IES Lighting Handbook 1981 Reference Volume I.
New York, 1981.

- IES. Illuminating Engineering Society of North America. IES Lighting Handbook 1981 Reference Volume II.
New York, 1981.
- Jan F.Kreider and Ari Rabl. HEATING AND COOLING OF BUILDINGS.
Singapore : McGraw-Hill,Inc, 1993.
- J.F. Waymouth and R.E. Levin. Designers Handbook light source applications.
Denver : GTE Products Corp., 1980.
- Lawrence Berkeley Laboratory Building Technologies Program Energy & Environment Division.
SUPERLITE 2.0 (User's Manual). California : Lawrence Berkeley Laboratory, March 1994.
- M.Santamoris and D.Asimakopoulos. Passive Cooling of Buildings. London : Antony Rowe Ltd., 1996.
- Olgay, V. Design with Climate : Bioclimatic Approach to Architectural Regionalism.
Fourth Printing. New Jersey : Printon University Press, 1961.
- RG.Hopkinson. DAYLIGHTING. London : Heinemann co.,ltd., 1966.
- William T.Meyer,AIA. Energy Economic and Building Design. New York : McGraw-Hill Book Company, 1993.
- Xavier Pacheco and Steve Teixeira. Delphi Developer's Guide. Indiana : Sams Publishing. 1995.

ภาคผนวก

ภาคผนวก ก. ตารางแสดงตำแหน่งของดวงอาทิตย์ มุมโพรไฟล์ (Profile Angle) มุม (Azimuth Angle)

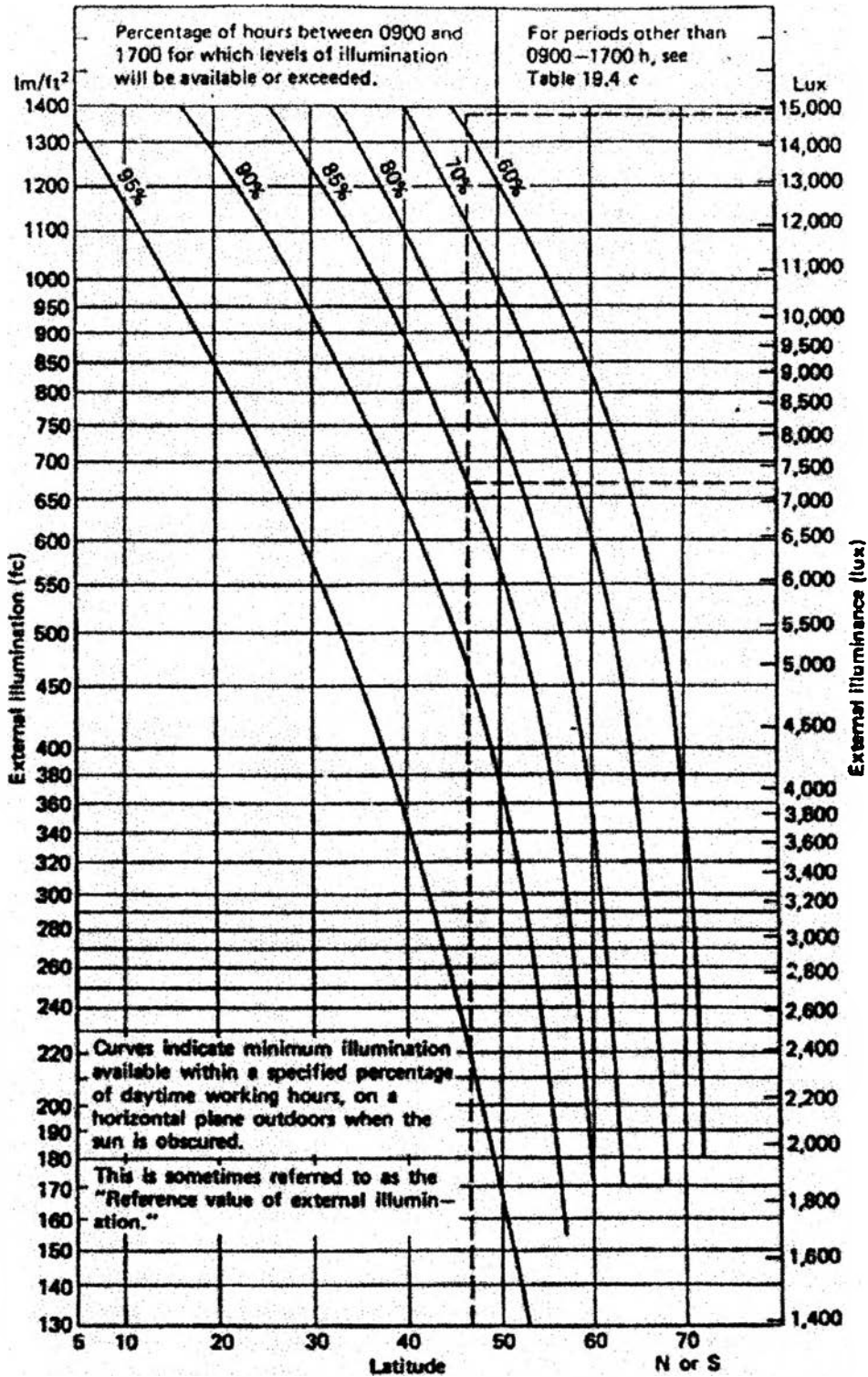
วันที่ 21 ของทุกเดือน ณ เวลาต่างๆ สำหรับเส้นรุ้งที่ 14 องศาเหนือ

(ที่มา : เทคนิคการออกแบบบ้านประหยัดพลังงาน เพื่อชีวิตที่ดีกว่า , หน้า 240)

Month	Solar Time	Solar Position		Profile (Shadow Line) Angle																Solar Position			Solar Time		
	Time	ALT	AZ	N	NNE	NE	ENE	E	ESE	SE	SSE	S	SSW	SW	WSW	W	WNW	NW	NNW	HOR	AZ	ALT	Time		
DEC	7	8	63			23	11	9	8	8	10	17	62								82	63	8	7	
	8	20	58			59	33	24	21	21	25	35	66								70	58	20	8	
	9	32	50			82	54	39	33	32	36	44	64								58	50	32	9	
	10	43	38				74	56	46	43	44	49	62	83							48	38	43	10	
	11	50	22					73	59	53	50	52	59	72							40	22	50	11	
	12	53	0					90	74	62	55	53	55	62	74						37	0	53	12	
	13	50	22									52	50	53	59	73					40	22	50	13	
	14	43	38									49	44	43	46	56	74				48	38	43	14	
	15	32	50									44	36	32	33	39	54	82			58	50	32	15	
	16	20	58									35	25	21	21	24	33	59			70	58	20	16	
	17	8	63									17	10	8	8	9	11	23			82	63	8	17	
	JAN + NOV	7	9	67			23	12	10	9	10	12	21									81	67	9	7
		8	22	62			55	33	25	22	23	27	40	75								68	62	22	8
		9	34	53			78	53	40	35	34	39	49	70								56	53	34	9
		10	45	42				72	57	48	45	47	53	66								46	42	45	10
		11	53	24				89	73	61	55	53	55	62	75							37	24	53	11
		12	56	0					90	75	65	58	56	58	65	75						34	0	56	12
13		53	24									55	53	55	61	73	89				37	24	53	13	
14		45	42									53	47	45	48	57	72				46	42	45	14	
15		34	53									49	39	34	35	40	53	78			56	53	34	15	
16		22	62									40	27	23	22	25	33	55			68	62	22	16	
17		9	67									21	12	10	8	9	11	23			81	67	9	17	
FEB + OCT		7	12	76		56	22	15	12	12	14	19	40									79	76	12	7
		8	26	71		84	48	33	27	26	28	35	55									65	71	26	8
		9	39	64			69	51	43	39	40	47	61									51	64	39	9
		10	51	52			85	69	58	52	51	55	64	78								39	52	51	10
		11	61	32				85	74	66	62	65	72	83								29	32	61	11
		12	65	0					90	80	72	67	65	67	72	80						25	0	65	12
	13	61	32									65	62	62	66	74	85				29	32	61	13	
	14	51	52									64	55	51	52	58	69	85			39	52	51	14	
	15	39	64									61	47	40	39	43	51	69			51	64	39	15	
	16	26	71									55	35	28	26	27	33	48	84		65	71	26	16	
	17	12	76									40	19	14	12	12	15	22	56		79	76	12	17	
	MAR + SEP	7	14	87		39	21	16	14	15	19	31	76									76	87	14	7
		8	29	82		66	43	33	29	30	35	47	76									61	82	29	8
		9	43	77		81	61	50	44	44	48	58	76									47	77	43	9
		10	57	67			76	66	59	57	59	66	76									33	67	57	10
		11	69	49				81	74	71	70	72	76	83								21	49	69	11
		12	76	0					90	85	80	77	76	77	80	85						14	0	76	12
13		69	49									76	72	70	71	74	81				21	49	69	13	
14		57	67									76	66	59	57	58	66	76			33	67	57	14	
15		43	77									76	58	48	44	43	50	61	81		47	77	43	15	
16		29	82									76	47	35	30	29	33	43	66		61	82	29	16	
17		14	87									76	31	19	15	14	16	21	39		76	87	14	17	
APR + AUG		6	3	101	14	5	4	3	3	4	5	14										87	101	3	6
		7	17	98	66	31	22	18	18	20	27	51										73	98	17	7
		8	32	95	82	53	39	33	32	35	44	64										58	95	32	8
		9	46	92	88	68	55	48	46	49	57	72										44	92	46	9
		10	61	89		78	69	63	61	63	68	78										29	89	61	10
		11	75	83		86	81	77	75	76	78	83										15	83	75	11
	12	86	0					90	86	85	85	65									4	0	86	12	
	13	75	83									65	65	65	66	90					15	83	75	13	
	14	61	89									83	78	68	63	61	63	69	78		29	89	61	14	
	15	46	92	88								72	57	49	46	48	55	68			44	92	46	15	
	16	32	95	82								64	44	35	32	33	39	53			58	95	32	16	
	17	17	98	66								51	27	20	18	18	22	31			73	98	17	17	
	18	3	101	14								14	5	4	3	3	4	5			87	101	3	18	
	APR + AUG	6	5	109	14	7	5	5	5	6	11	56										86	109	5	6
		7	19	197	49	28	21	19	20	24	36	73										72	197	19	7
		8	33	105	68	47	36	33	34	39	52	79										58	105	33	8
		9	47	104	77	61	51	47	48	53	64	83										44	104	47	9
10		61	106	81	71	64	61	62	67	75											30	106	61	10	
11		74	114	83	78	76	74	76	80	85											16	114	74	11	
12		84	180	84	84	86	87	89								89	87	86	84		6	180	84	12	
13		74	114	83											85	80	76	74	76	78	16	114	74	13	
14		61	106	81											75	67	62	61	64	71	30	106	61	14	
15		47	104	77											83	64	53	48	47	51	44	104	47	15	
16		33	105	68											79	52	39	34	33	36	47	105	33	16	
17		19	107	49											73	36	24	20	19	21	28	107	19	17	
18		5	109	14											56	11	6	5	5	5					

ภาคผนวก ข. ตารางแสดงค่าความสว่างท้องฟ้า
เมื่อพิจารณาจากเส้นรุ้ง (Latitude) ของตำแหน่งที่ตั้ง

(ที่มา : Daylight International Recommendations for the Calculation of Natural Daylight ,1970)



ภาคผนวก ข. ความสัมพันธ์ระหว่างปริมาณความสว่าง และปริมาณรังสีจากดวงอาทิตย์

ในสภาพท้องฟ้าที่มีความแตกต่างกันพลังงานแสงอาทิตย์จะประกอบด้วย อินฟราเรด, อัลตราไวโอเลต, และรังสีของคลื่นแสง ซึ่งค่าต่าง ๆ เหล่านี้สามารถที่จะแปลเปลี่ยนเป็นแสงสว่างจากดวงอาทิตย์ได้ โดยเทียบหนึ่งต่อหนึ่งหน่วยของแสงสว่าง ต่อหน่วยของปริมาณ รังสีจากดวงอาทิตย์ ประสิทธิภาพความส่องสว่าง ของดวงอาทิตย์ จะแปรผันตามปริมาณพลังงานจากรังสีดวงอาทิตย์ ที่ประกอบด้วยช่วงคลื่น ที่มีความยาวคลื่นระหว่างช่วงคลื่นแสงที่มองเห็นและไม่เห็น (อินฟราเรด และ อัลตราไวโอเลต) และตัวแปรที่มีอิทธิพลของลักษณะท้องฟ้าโปร่ง ,ท้องฟ้ามีเมฆปกคลุมทึบและค่าเฉลี่ยจากท้องฟ้า (clear, overcast, average sky) และมุมอัลติจูด , มุมอัลซิมูทของดวงอาทิตย์ ซึ่งในส่วนใหญ่ประสิทธิภาพของความสว่างจะแยกออกเป็น จากดวงอาทิตย์เพียงอย่างเดียว, จากท้องฟ้าเพียงอย่างเดียว และจากปริมาณรังสีรวมของดวงอาทิตย์และท้องฟ้า (Global Radiation Sun Plus Sky)

จากข้อมูลที่ได้อธิบาย จะพบว่าประสิทธิภาพความสว่างของแสงอาทิตย์ ที่ได้จากการตรวจวัดจะมีความส่องสว่าง และปริมาณรังสีของดวงอาทิตย์ในแนวทางเดียวกัน เมื่อทดสอบโดยการวัดประสิทธิภาพ ความส่องสว่างในหน่วย ลูเมนต่อวัตต์ (lumen per watt) ในลักษณะของปริมาณความสว่าง ต่อหนึ่งหน่วยพื้นที่ เทียบกับความเข้มของรังสี จากดวงอาทิตย์ต่อหน่วยพื้นที่เดียวกัน และผลจากข้อมูลข้างต้น ได้แสดงให้เห็นว่าท้องฟ้าโปร่ง และ ท้องฟ้ามีเมฆปกคลุมทึบ จะมีผลเพียงเล็กน้อยต่อประสิทธิภาพ ของความส่องสว่าง ตามมุมของดวงอาทิตย์ (ในแต่ละช่วงเวลาของปี) แต่จะมีการลดลงอย่างมาก ของปริมาณรังสีตรงดวงอาทิตย์ (Sun's direct radiation) จะเกิดขึ้นเมื่อ มุมอัลติจูดของดวงอาทิตย์มีค่าน้อยกว่า 30 องศา อย่างไรก็ตามการเปรียบเทียบผลในยุคแรกนั้น อาจมีความคลาดเคลื่อนของผลจากอุปกรณ์ที่ใช้ตรวจสอบได้ แต่ในปัจจุบันเครื่องมือที่ใช้ได้มีการพัฒนาให้มีความแม่นยำสูง การวัดประสิทธิภาพของแสงสว่างจากดวงอาทิตย์ จากเครื่องมือที่ทันสมัย ทำให้ผลที่ได้จากการตรวจสอบมีความถูกต้อง และสามารถเชื่อถือได้ ซึ่งสามารถสรุปผลการศึกษาข้างต้น ได้ดังนี้

ความสามารถในการวัดความสัมพันธ์ ของรังสีกระจายทั้งหมด (Global radiation) จากท้องฟ้าโปร่ง, ท้องฟ้ามีเมฆปกคลุมทึบ (Clear sky , Overcast sky and Average sky) ซึ่งมีค่าเฉลี่ยของความสว่าง ที่พบอยู่ระหว่าง 119 +/- 2 , 120 +/- 5 และ 116 +/- 7 ลูเมนต่อวัตต์ (lm/W) ตามลำดับ และค่าเฉลี่ยของท้องฟ้า ที่เป็นแบบรวม (Global radiation) จะพบว่ามีความแตกต่างกันจากน้อยที่สุด 105 lm/W ในเดือนมีนาคมถึง 128 lm/W ในเดือนพฤศจิกายน แต่แท้จริงแล้ว จุดประสงค์ในการศึกษาเรื่องแสงสว่างนั้น ค่าของแสงสว่างที่นำมาใช้งานจะเป็นค่าของความสว่างจากดวงอาทิตย์ และจากท้องฟ้าในลักษณะต่างๆ เป็นตัวกำหนดด้วย นอกจากนี้ จากการแสดงข้อมูลด้วยกราฟการกระจายแสงของดวงอาทิตย์ จะแสดงให้เห็นถึงค่าที่เกือบคงที่ค่าหนึ่งของประสิทธิภาพของความสว่าง ซึ่งจะได้ประมาณ 117 lm/W สำหรับมุมอัลติจูดของดวงอาทิตย์ ที่มากกว่า 25 องศา และจะลดลงเหลือ 90 lm/W เมื่อมุมอัลติจูดของดวงอาทิตย์เท่ากับ 7½ องศา โดยทั่วไปเพื่อสะดวกในการคำนวณจะใช้ค่าความสว่าง 100 lm/w ซึ่งสามารถยอมรับได้ในความถูกต้อง และความสว่างของดวงอาทิตย์ ในสภาพท้องฟ้าที่ปราศจากเมฆปกคลุม (ท้องฟ้าโปร่ง) จะต่ำกว่าประมาณ 20%

จากการทดสอบทั่วไป ไม่ได้แสดงให้เห็นถึงความสว่าง ที่ได้จากรังสีดวงอาทิตย์ ของค่าเฉลี่ยจากท้องฟ้า ที่จะตรงข้ามกับค่ารวมของ ปริมาณรังสีจากดวงอาทิตย์ (Global Radiation) แต่การรวมกันของค่าเฉลี่ยของรังสีจากท้องฟ้าที่เกี่ยวกับค่าเฉลี่ยความสว่างของท้องฟ้า และจากรายงานที่เกี่ยวข้อง จะใช้ค่าเฉลี่ยที่เท่ากับ 125 ลูเมนต่อวัตต์ เป็นค่ามาตรฐาน ในการพิจารณาความสว่างของแสงจากท้องฟ้าตลอดทั้งปี และจากการพิจารณาถึง ค่าสูงสุด ของการกระจายสเปกตรัม ของพลังงาน จากรังสีดวงอาทิตย์ ในสภาพท้องฟ้าโปร่ง จากคลื่นสั้น

เปรียบเทียบกับท้องฟ้า แบบมีเมฆปกคลุมทั้งหมด ที่สามารถจะลดทอนความสัมพัทธ์ ระหว่างปริมาณ ของพลังงานในคลื่นแสงอินฟราเรด ที่เป็นส่วนหนึ่งของสเปกตรัมคลื่นแสงทั้งหมดนั้น จะมีผลทำให้ประสิทธิภาพ ของความสว่างของแสงในท้องฟ้าโปร่ง เพิ่มขึ้นเป็น 150 ลูเมนต่อวัตต์

การหาค่าเฉลี่ยความสว่าง (Average Illumination) ของแสงจากดวงอาทิตย์ หรือท้องฟ้า จากค่าของรังสีดวงอาทิตย์ (Solar Radiation) และ การแปลงค่าเฉลี่ยของรังสีจากดวงอาทิตย์ หรือท้องฟ้า สามารถเปลี่ยนแปลงได้ดังสมการต่อไปนี้

$$\text{Average sun/sky illumination (lm/ft}^2\text{)} = \text{Average sun/sky radiation (w/ft}^2\text{)} \\ \times \text{luminous efficiency of sun/sky radiation (lm/W)}$$

หากอัตราส่วนค่ารังสีดวงอาทิตย์ต่อค่ารังสีจากท้องฟ้า (Sun / Sky radiation) มีหน่วยเป็น มิลลิวัตต์ต่อตารางเซนติเมตร (milliwatts per sq.cm. , mW/cm²) ซึ่งเป็นหน่วยของที่ใช้ในกรมอุตุนิยมวิทยาของอังกฤษ จะมีการแปลงดังนี้

$$\text{Average sun/sky illumination (lm/ft}^2\text{)} = 0.929 \times \text{Average sun/sky radiation (mw/ft}^2\text{)} \\ \times \text{luminous efficiency of sun/sky radiation (lm/W)}$$

และจากสมการข้างต้น IES. (Illuminating Engineering Society) ได้ใช้ในการอ้างอิงในการคำนวณหา ค่าประสิทธิภาพความสว่าง (Luminous efficiency) ของท้องฟ้าเป็นรายชั่วโมง ตลอดทั้งปีของเมือง Eskdalemuir ในทางใต้ของประเทศสกอตแลนด์(Southern Scotland)และเมืองเลวิก (Lerwick) บนเกาะไอสแลนด์ (Shetland, Island) ซึ่งค่าความสว่างของรังสีจากดวงอาทิตย์ที่ใช้ในการคำนวณสามารถพิจารณาได้จากตารางข้างล่างนี้

Source	Luminous efficiency (lm/W)
Sun (solar altitude < 7 1/2°)	90
Sun (solar altitude > 25°)	117
Sun (suggested mean)	100
Sky (clear)	150
Sky (average)	125
Global (average)	115

ตารางแสดงค่าประสิทธิภาพความสว่าง (Luminous efficiency)

ของแสงสว่างธรรมชาติ หน่วยเป็น ลูเมน ต่อวัตต์

ที่มา : Daylighting by R.G.Hopkinson, PhD, page 51

ภาคผนวก ค. หน่วยความส่องสว่าง
(ที่มา : Daylight , page 570)

หน่วยพื้นฐาน

Lumen (lm) = หน่วยของฟลักซ์แสงสว่าง (flux)

Candela (cd) = หน่วยของความเข้มแสง (Flux per unit solid angle)

1 แคนเดลา (Candela) = 1 ลูเมนต่อสเตเรเดียน (Lumen per steradian)

ลูเมนต่อตารางฟุต (Lumen per square foot, lm/ft^2) เป็นหน่วยของ ฟุตแคนเดิล (fc.)

ลูเมนต่อตารางเมตร (Lumen per square metre, lm/m^2) เป็นหน่วยของ ลักซ์ (lux.)

ลูเมนต่อตารางเซนติเมตร (Lumen per square centimetre, lm/cm^2) เป็นหน่วยของ ฟลิอท (phot.)

1 milliphot = 10 lux

การแปลงหน่วย

1 lm/ft^2	=	10.76	lux
	=	0.001076	phot
	=	1.076	milliphot
1 Lux	=	0.0929	lm/ft^2
	=	0.0001	phot
	=	0.1	milliphot
1 lm/ft^2	~	10	lux
เมื่อ พื้นที่ 1 ตารางเมตร	=	พื้นที่ 10.76 ตารางฟุต	

แคนเดลา ต่อตารางเมตร (Candela per square metre, cd/m^2) เป็นหน่วย นิต (nit, nt)

แคนเดลา ต่อตารางเซนติเมตร (Candela per square metre, cd/m^2) เป็นหน่วย สติบ (Stib, sb)

1 cd/ft^2	=	10.76	cd/m^2
	=	10.76	nits
	=	0.001076	stibs
1 cd/m^2	=	0.0929	cd/ft^2
1 stib	=	1	cd/cm^2
	=	929	cd/ft^2

Luminance = Luminance x Reflectance

Foot-lambert (ft-L) , Foot- candle = $1/\pi$ 1 cd/ft^2

Lambert (L) (= 1,000 millilamberts) = $1/\pi$ 1 cd/ft^2 (1 foot-lambert ~ 1 millilambert)

**ภาคผนวก ง. รายละเอียดข้อมูลภาษาคอมพิวเตอร์ ที่ใช้ในการเขียนโปรแกรม
(Source Code and Algorithm)**

program Daylight;

uses

Forms,

Unit1 in 'Unit1.pas' {Form1},
 Unit2 in 'Unit2.pas' {newdata},
 Unit3 in 'Unit3.pas' {Lighttrans},
 Unit4 in 'Unit4.pas' {atmos},
 Unit5 in 'Unit5.pas' {Roomdata},
 Unit6 in 'Unit6.pas' {lumen},
 fstlogo in 'fstlogo.pas' {Firstlogo},
 Unit7 in 'Unit7.pas' {Lumengraph},
 skyfac in 'skyfac.pas' {skyfactor},
 intref in 'intref.pas' {Intref},
 toplumen in 'toplumen.pas' {lumentop},
 inputtop in 'inputtop.pas' {topinput},
 nproject in 'nproject.pas' {projectname},
 datawth in 'datawth.pas' {wthdata},
 blind in 'blind.pas' {blinds},
 contour in 'contour.pas' {skycontour},
 bcommand in 'bcommand.pas' {Commandbox},
 tdsky in 'tdsky.pas' {sky3d},
 multiwin in 'multiwin.pas' {Mulwindow},
 boxdata in 'boxdata.pas' {Databox},
 progmer in 'progmer.pas' {programer},
 tabledat in 'tabledat.pas' {datatable},
 skyhelp in 'skyhelp.pas' {helpbox},
 grharea in 'grharea.pas' {Grapharea},
 calarea in 'calarea.pas' {Areacal},
 diretwin in 'diretwin.pas' {windirect},
 daywith in 'daywith.pas' {Withdaylight};

{\$R *.RES}

begin

Application.Initialize;
 Application.CreateForm(TFirstlogo, Firstlogo);

Application.CreateForm(TForm1, Form1);
 Application.CreateForm(Tnewdata, newdata);
 Application.CreateForm(TLighttrans, Lighttrans);
 Application.CreateForm(Tatmos, atmos);
 Application.CreateForm(TRoomdata, Roomdata);
 Application.CreateForm(Tlumen, lumen);
 Application.CreateForm(TLumengraph, Lumengraph);
 Application.CreateForm(Tskyfactor, skyfactor);
 Application.CreateForm(TIntref, Intref);
 Application.CreateForm(Tlumentop, lumentop);
 Application.CreateForm(Ttopinput, topinput);
 Application.CreateForm(Tprojectname, projectname);
 Application.CreateForm(Twthdata, wthdata);
 Application.CreateForm(Tblinds, blinds);
 Application.CreateForm(Tskycontour, skycontour);
 Application.CreateForm(TCommandbox,
 Commandbox);
 Application.CreateForm(Tsky3d, sky3d);
 Application.CreateForm(TMulwindow, Mulwindow);
 Application.CreateForm(TDatabox, Databox);
 Application.CreateForm(Tprogramer, programer);
 Application.CreateForm(Tdatatable, datatable);
 Application.CreateForm(Thelpbox, helpbox);
 Application.CreateForm(TGrapharea, Grapharea);
 Application.CreateForm(TAreacal, Areacal);
 Application.CreateForm(Twindirect, windirect);
 Application.CreateForm(TWithdaylight, Withdaylight);
 Application.Run;
 end.

unit bcommand;

```

interface
uses
    Windows, Messages, SysUtils, Classes, Graphics,
    Controls, Forms, Dialogs, ExtCtrls, Buttons, StdCtrls,
    MPlayer;
type
    TCommandbox = class(TForm)
        SpeedButton1: TSpeedButton;
        Shape1: TShape;
        SpeedButton3: TSpeedButton;
        SpeedButton2: TSpeedButton;
        SpeedButton4: TSpeedButton;
        SpeedButton5: TSpeedButton;
        SpeedButton6: TSpeedButton;
        SpeedButton7: TSpeedButton;
        SpeedButton8: TSpeedButton;
        SpeedButton9: TSpeedButton;
        SpeedButton10: TSpeedButton;
        SpeedButton11: TSpeedButton;
        SpeedButton12: TSpeedButton;
        SpeedButton13: TSpeedButton;
        SpeedButton14: TSpeedButton;
        Shape2: TShape;
        SpeedButton15: TSpeedButton;
        Panel1: TPanel;
        Shape3: TShape;
        SpeedButton16: TSpeedButton;
        SpeedButton17: TSpeedButton;
        SpeedButton18: TSpeedButton;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Label4: TLabel;
        Label5: TLabel;
        Label6: TLabel;
        Label7: TLabel;
        Label8: TLabel;
        Label9: TLabel;
        Label10: TLabel;
        Label11: TLabel;
        SpeedButton19: TSpeedButton;
        SpeedButton20: TSpeedButton;
        Label12: TLabel;
        Label13: TLabel;
        MediaPlayer1: TMediaPlayer;
        procedure SpeedButton12Click(Sender: TObject);
        procedure SpeedButton16Click(Sender: TObject);
        procedure SpeedButton14Click(Sender: TObject);
        procedure SpeedButton18Click(Sender: TObject);
        procedure SpeedButton17Click(Sender: TObject);
        procedure SpeedButton1Click(Sender: TObject);
        procedure SpeedButton2Click(Sender: TObject);
        procedure SpeedButton3Click(Sender: TObject);
        procedure SpeedButton4Click(Sender: TObject);
        procedure SpeedButton5Click(Sender: TObject);
        procedure SpeedButton6Click(Sender: TObject);
        procedure SpeedButton7Click(Sender: TObject);
        procedure SpeedButton8Click(Sender: TObject);
        procedure SpeedButton9Click(Sender: TObject);
        procedure SpeedButton10Click(Sender: TObject);
        procedure SpeedButton11Click(Sender: TObject);
        procedure SpeedButton13Click(Sender: TObject);
        procedure SpeedButton15Click(Sender: TObject);
        procedure memoclear(panel:integer);
        procedure mapcal;
        procedure panelcal;
        procedure panelcheck(panel:integer);
        procedure atm_cal(var atm:real);
        procedure intreflect(var irc:real);
        procedure trans_cal(var trans:real);
        procedure CIEtable(W,H,D :real; var Ei :real);
    end;

```

```

procedure sf_cal(rfx,rfy,rfz : real; var sfv:real);
procedure sumoutput(sfv,irc,tr,atm :real; var Ell:real);
procedure comparecase(line,lneN:integer);
procedure rowcompare;
procedure Sumcompare;
procedure panelcase(panel:integer; ill:real);
procedure SpeedButton19Click(Sender: TObject);
procedure SpeedButton20Click(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Commandbox: TCommandbox;

implementation

uses Unit1, contour, skyfac, tdsky, multiwin, boxdata,
grharea;

{$R *.DFM}

procedure Tcommandbox.atm_cal(var atm:real);
var
  Ehk,Evk,Exr,atm_t,atm_m,Bua,Bka,Bga,Buu,Bku,Bgu:real;
  Ext,Edv,Edh,Ma,Ku,Ei :real;
begin
  with databox do begin
    Ehk := strtfloat(label41.caption);
    Evk := strtfloat(label42.caption);
    Exr := strtfloat(label51.caption);
    atm_t := strtfloat(label52.caption);
    atm_m := strtfloat(label53.caption);
    Bua := strtfloat(label63.caption);
    Bka := strtfloat(label64.caption);
    Bga := strtfloat(label65.caption);
    Buu := strtfloat(label66.caption);
    Bku := strtfloat(label67.caption);
    Bgu := strtfloat(label68.caption);
  end;
  Edv := 0; Edh := 0;
  Ext := ((Ehk+Edh)/2)*Exr;
  Ma := (Edv*Bua)+(Evk*Bka)+(Ext*Bga);
  Ku := (Edv*Buu)+(Evk*Bku)+(Ext*Bgu);

  if Edv = 0 then Ei := Ma else Ei := Ma+Ku;
  atm := Ei*atm_t*atm_m;
end;

procedure Tcommandbox.intreflect(var irc:real);
var c_re,w_re,f_re,wall,ceil,floor,evg,
  Hevg,Rfw,Rcw : real;
  rm_w,rm_d,rm_h,win_w,win_h,win_s : real;
  low_mid_win,up_mid_win,low_wall,up_wall,
  all_sur,win : real;
  c : integer;
begin
  C := 5;
  with databox do begin
    c_re := strtfloat(label54.caption);
    w_re := strtfloat(label55.caption);
    f_re := strtfloat(label56.caption);
    rm_w := strtfloat(label44.caption);
    rm_d := strtfloat(label45.caption);
    rm_h := strtfloat(label46.caption);
    win_w := strtfloat(label47.caption);
    win_h := strtfloat(label48.caption);
    win_s := strtfloat(label49.caption);
  end;
  low_mid_win := (win_h/2)+win_s;

```

```

up_mid_win := (rm_h - low_mid_win);

all_sur := 2*((rm_w*rm_h)+(rm_d*rm_h)+
(rm_w*rm_d));
wall := (2*(rm_w+rm_d)*rm_h)-(win_h*win_w);
low_wall := (2*(rm_w+rm_d)*low_mid_win)-
(rm_w*low_mid_win);
up_wall := (2*(rm_w+rm_d)* up_mid_win)
-(rm_w* up_mid_win);
ceiling := (rm_w*rm_d);
floor := (rm_w*rm_d);
win := (win_w*win_h);
Evg := ((win*0.15)+(wall*w_re)+(ceiling*c_re)
+(floor*f_re))/(all_sur);
Rfw :=((low_wall*w_re)+(floor*f_re))/(low_wall+floor);
Rcw :=(( up_wall*w_re)+(ceiling*c_re) /(up_wall+ceiling);

```

```

if evg > 0.98 then evg := 0.98;
Hevg := (all_sur*(1-0.019-evg));
if Hevg <= 0 then Hevg := all_sur*0.01;
IRC := ((0.85*win)*(39*Rfw+C*Rcw))/abs(Hevg);

```

```

with skyfactor do begin
label36.caption := format('%5.2n',[Evg]);
label27.caption := format('%5.2n',[Rfw]);
label28.caption := format('%5.2n',[Rcw]);
label62.caption := format('%5.2n',[all_sur]);
label63.caption := format('%5.2n',[win]);
end;

```

```
end;
```

```
procedure Tcommandbox.trans_cal(var trans:real);
```

```
var mat,vis,net : real;
```

```
begin
```

```

with databox do begin
mat := strtfloat(label57.caption);
vis := strtfloat(label58.caption);

```

```
net := strtfloat(label59.caption);
```

```
end;
```

```
Trans := mat*vis*net;
```

```
end;
```

```
procedure Tcommandbox.CIEtable
```

```
(W,H,D:real; var Ei:real);
```

```
var SF:real;
```

```
begin
```

```
if D=0 then D:=0.001;
```

```
SF:=(50/Pi)*
```

```
(ArcTan(W/D)-(D/Sqrt(Sqr(D)+Sqr(H)))
```

```
*Arctan(W/Sqrt(sqr(D)+Sqr(H))) );
```

```
if SF<0.01 then SF:=0;
```

```
Ei:=SF;
```

```
end;
```

```
procedure Tcommandbox.sf_cal
```

```
(rfx,rfy,rfz:real; var sfv:real);
```

```
var win_w,win_h,rm_w,rm_h,rm_d,win_s,win_d :real;
```

```
w0,w1,w2,h0,h1,h2,E1,E2,E3,E4 : real;
```

```
begin
```

```
(*Input Data*)
```

```
with databox do begin
```

```
rm_w := strtfloat(label44.caption);
```

```
rm_d := strtfloat(label45.caption);
```

```
rm_h := strtfloat(label46.caption);
```

```
win_w := strtfloat(label47.caption);
```

```
win_h := strtfloat(label48.caption);
```

```
win_s := strtfloat(label49.caption);
```

```
win_d := strtfloat(label50.caption);
```

```
end;
```

```
(*-----*)
```

```
if rfx > win_w+win_d then begin
```

```
w1 := rfx-(win_w+win_d);
```

```
w2 := win_w;
```

```
end;
```

```

if rfx = win_w+win_d then begin
    w1 := 0; w2 := win_w;
    end;
if (rfx < win_d+win_w)and(rfx > win_d) then begin
    w1 := (win_w+win_d)-rfx;
    w2 := win_w-w1;
    end;
if rfx = win_d then begin w1 := 0;
    w2 := win_w;
    end;
if rfx < win_d then begin w1 := (win_d-rfx);
    w2 := win_w;
    end;
w0 := w1+w2;
(*-----*)
h0 := (win_h+win_s)-rfz;
if h0 <=0 then h1:=win_h+win_s;
if h0 > win_h then begin h1 := h0-win_h;
    h2 := win_h;
    end;
if h0 <= win_h then begin h1 := 0;
    h2 := win_h;
    end;

(*-----*)
if (rfx < win_d+win_w)and(rfx > win_d) then begin
    cietable(w1,h0,rfy,E1);
    cietable(w2,h0,rfy,E2);
    cietable(w1,h1,rfy,E3);
    cietable(w2,h1,rfy,E4);
    sfv := (E1+E2)-(E3+E4);
    end else begin
    cietable(w0,h0,rfy,E1);
    cietable(w1,h0,rfy,E2);
    cietable(w0,h1,rfy,E3);
    cietable(w1,h1,rfy,E4);
    sfv := (E1-E2-E3)+E4;
    end;
end;
(*-----*)
end;

procedure TCommandbox.SpeedButton12Click
    (Sender: TObject);
begin
    mediaplayer1.filename:='Data\SDend.wav';
    mediaplayer1.open; mediaplayer1.play;

    if speedbutton12.caption = 'Next panels' then begin
        label12.visible := false;
        label13.visible := false;
        skycontour.visible := false;
        sky3d.visible := false;
        mulwindow.visible := true
    end;

    if speedbutton12.caption = 'Return Command' then
    begin
        skycontour.visible := false;
        sky3d.visible := false;
        form1.enabled := true;
        end;
    close;
end;

procedure TCommandbox.SpeedButton16Click
    (Sender: TObject);
begin
    mediaplayer1.filename:='Data\SDenter.wav';
    mediaplayer1.open; mediaplayer1.play;

    label5.caption := '0'; label6.caption := '0';
    label11.caption := 'Input 2 digits value :';
    label11.visible := true;
    speedbutton12.enabled := false;
    speedbutton13.enabled := false;

```

```

speedbutton15.enabled := false;
speedbutton17.enabled := false;
speedbutton18.enabled := false;
end;

procedure TCommandbox.SpeedButton14Click
  (Sender: TObject);
var txt : string; A,B,row:integer;
begin
  mediaplayer1.filename:='Data\SDenter.wav';
  mediaplayer1.open; mediaplayer1.play;

  A := strtoint(label5.caption);
  B := strtoint(label6.caption);
  if a = 0 then txt := label6.caption else
  txt := label5.caption + label6.caption;
  if (a=0)and(b=0) then txt := '1';

  if label11.caption='Input 2 digits value :' then
  begin
    if speedbutton17.enabled = false then
      label7.caption := txt;
    if speedbutton16.enabled = false then
      label8.caption := txt;
    end;

  if label11.caption='Select analysis row :' then
  begin
    grapharea.visible := true;
    row := strtoint(txt);
    Grapharea.insertdata(row);
    Grapharea.maskedit1.text:=inttostr(row);
    end;

  label5.visible := false;
  label6.visible := false;
  label11.visible := false;

```

```

speedbutton12.enabled := true;
speedbutton13.enabled := true;
speedbutton15.enabled := true;
speedbutton16.enabled := true;
speedbutton17.enabled := true;
speedbutton18.enabled := true;
end;

procedure TCommandbox.SpeedButton18Click
  (Sender: TObject);
var rm_d,rm_w : real; xpoint,ypoint:integer;
begin
  mediaplayer1.filename:='Data\SDenter.wav';
  mediaplayer1.open; mediaplayer1.play;

  rm_w := strtofloat(skyfactor.label38.caption);
  rm_d := strtofloat(skyfactor.label39.caption);
  xpoint := strtoint(label7.caption);
  ypoint := strtoint(label8.caption);
  if xpoint = 0 then xpoint := 1;
  if ypoint = 0 then ypoint := 1;
  label9.caption := format('%8.2n',[rm_w/(xpoint+1)]);
  label10.caption := format('%8.2n',[rm_d/(ypoint+1)]);
  label3.visible := true; label4.visible := true;
  label9.visible := true; label10.visible := true;
end;

procedure TCommandbox.SpeedButton17Click(Sender:
TObject);
begin
  mediaplayer1.filename:='Data\SDenter.wav';
  mediaplayer1.open; mediaplayer1.play;
  label5.caption := '0'; label6.caption := '0';
  label11.caption := 'Input 2 digits value :';
  label11.visible := true;
  speedbutton12.enabled := false;

```

```

speedbutton13.enabled := false;
speedbutton15.enabled := false;
speedbutton16.enabled := false;
speedbutton18.enabled := false;
end;

procedure TCommandbox.SpeedButton1Click(Sender:
TObject);
begin
mediaplayer1.filename:='Data\SD1.wav';
mediaplayer1.open; mediaplayer1.play;
if label6.visible = false then begin
label6.Caption := '1'; label6.visible := true;
end else begin
label5.caption := label6.caption;
label5.visible := true;
label6.caption := '1';
end;
end;

procedure TCommandbox.SpeedButton2Click(Sender:
TObject);
begin
mediaplayer1.filename:='Data\SD2.wav';
mediaplayer1.open; mediaplayer1.play;
if label6.visible = false then begin
label6.Caption := '2'; label6.visible := true;
end else begin
label5.caption := label6.caption;
label5.visible := true; label6.Caption := '2';
end;
end;

procedure TCommandbox.SpeedButton3Click(Sender:
TObject);
begin
mediaplayer1.filename:='Data\SD3.wav';

```

```

mediaplayer1.open; mediaplayer1.play;
if label6.visible = false then begin
label6.Caption := '3'; label6.visible := true;
end else begin
label5.caption := label6.caption;
label5.visible := true;label6.Caption := '3';
end;
end;

procedure TCommandbox.SpeedButton4Click(Sender:
TObject);
begin
mediaplayer1.filename:='Data\SD4.wav';
mediaplayer1.open; mediaplayer1.play;
if label6.visible = false then begin
label6.Caption := '4'; label6.visible := true;
end else begin
label5.caption := label6.caption;
label5.visible := true; label6.Caption := '4';
end;
end;

procedure TCommandbox.SpeedButton5Click(Sender:
TObject);
begin
mediaplayer1.filename:='Data\SD5.wav';
mediaplayer1.open; mediaplayer1.play;
if label6.visible = false then begin
label6.Caption := '5'; label6.visible := true;
end else begin
label5.caption := label6.caption;
label5.visible := true; label6.Caption := '5';
end;
end;

procedure TCommandbox.SpeedButton6Click(Sender:
TObject);

```



```

begin
mediaplayer1.filename:='Data\SD6.wav';
mediaplayer1.open; mediaplayer1.play;
if label6.visible = false then begin
label6.Caption := '6'; label6.visible := true;
end else begin
label5.caption := label6.caption;
label5.visible := true; label6.Caption := '6';
end;
end;

procedure TCommandbox.SpeedButton7Click(Sender:
TObject);
begin
mediaplayer1.filename:='Data\SD7.wav';
mediaplayer1.open; mediaplayer1.play;
if label6.visible = false then begin
label6.Caption := '7'; label6.visible := true;
end else begin
label5.caption := label6.caption;
label5.visible := true; label6.Caption := '7';
end;
end;

procedure TCommandbox.SpeedButton8Click(Sender:
TObject);
begin
mediaplayer1.filename:='Data\SD8.wav';
mediaplayer1.open; mediaplayer1.play;
if label6.visible = false then begin
label6.Caption := '8'; label6.visible := true;
end else begin
label5.caption := label6.caption;
label5.visible := true; label6.Caption := '8';
end;
end;

```

```

procedure TCommandbox.SpeedButton9Click(Sender:
TObject);
begin
mediaplayer1.filename:='Data\SD9.wav';
mediaplayer1.open; mediaplayer1.play;
if label6.visible = false then begin
label6.Caption := '9'; label6.visible := true;
end else begin
label5.caption := label6.caption;
label5.visible := true; label6.Caption := '9';
end;
end;

procedure TCommandbox.SpeedButton10Click(Sender:
TObject);
begin
mediaplayer1.filename:='Data\SD0.wav';
mediaplayer1.open; mediaplayer1.play;
if label6.visible = false then begin
label6.Caption := '0'; label6.visible := true;
end else begin
label5.caption := label6.caption;
label5.visible := true; label6.Caption := '0';
end;
end;

procedure TCommandbox.SpeedButton11Click(Sender:
TObject);
begin
mediaplayer1.filename:='Data\SDenter.wav';
mediaplayer1.open; mediaplayer1.play;

label5.caption := '0'; label5.visible := false;
label6.caption := '0'; label6.visible := false;
label3.visible := false; label4.visible := false;
label9.visible := false; label10.visible := false;
end;

```

```

procedure Tcommandbox.memoclear(panel:integer);
begin
  case panel of
    0: begin skycontour.memo1.lines.clear;
        skycontour.memo2.lines.clear;
        skycontour.memo1.lines[0]:='Points';
        sky3d.memo1.lines.clear;
        sky3d.memo1.lines[0]:='Panel1';
        end;
    1: begin skycontour.memo1.lines.clear;
        skycontour.memo2.lines.clear;
        skycontour.memo1.lines[0]:='Points';
        sky3d.memo1.lines.clear;
        sky3d.memo1.lines[0]:='Panel1';
        end;
    2: begin skycontour.memo1.lines.clear;
        skycontour.memo2.lines.clear;
        skycontour.memo1.lines[0]:='Points';
        sky3d.memo2.lines.clear;
        sky3d.memo2.lines[0]:='Panel2';
        end;
    3: begin skycontour.memo1.lines.clear;
        skycontour.memo2.lines.clear;
        skycontour.memo1.lines[0]:='Points';
        sky3d.memo3.lines.clear;
        sky3d.memo3.lines[0]:='Panel3';
        end;
    4: begin skycontour.memo1.lines.clear;
        skycontour.memo2.lines.clear;
        skycontour.memo1.lines[0]:='Points';
        sky3d.memo4.lines.clear;
        sky3d.memo4.lines[0]:='Panel4';
        end;
    5: begin skycontour.memo1.lines.clear;
        skycontour.memo2.lines.clear;
        skycontour.memo1.lines[0]:='Points';
        sky3d.memo5.lines.clear;
        sky3d.memo5.lines[0]:='Panel5';
        end;
  end;
end;

procedure Tcommandbox.Sumoutput(sfv,irc,tr,atm:real;
var Ell:real);
begin
  Ell := ((sfv+(0.6*irc))/100)*(Tr*atm);
end;

procedure
Tcommandbox.comparecase(line,IneN:integer);
var panel:integer;
begin
  panel:=strtoint(commandbox.label13.caption);
  case panel of
    1: sky3d.memo1.lines[line]:=inttostr(IneN);
    2: sky3d.memo2.lines[line]:=inttostr(IneN);
    3: sky3d.memo3.lines[line]:=inttostr(IneN);
    4: sky3d.memo4.lines[line]:=inttostr(IneN);
    5: sky3d.memo5.lines[line]:=inttostr(IneN);
  end;
end;

procedure Tcommandbox.rowcompare;
var panel,max_x,max_y,C,D,E,F,N,M:integer;
check:boolean;
    Ine1,Ine2,Ine3,IneN:integer;
begin
  panel := strtoint(commandbox.label13.caption);
  max_x := strtoint(commandbox.label7.caption);
  max_y := strtoint(commandbox.label8.caption);

  C:=1; D:=1; M:=1;
  for C:=1 to max_x do begin

```

```

E:=0; F:=0; N:=1; check:=true;
D:=M;
for C:=1 to max_x - 2 do begin
  lne1 := strtoint(skycontour.memo1.lines[D]);
  if lne1 <> 0 then begin
    if check = true then begin
      E:=D+max_y; F:=D+(max_y*2);
      lne2:=strtoint(skycontour.memo1.lines[E]);
      lne3:=strtoint(skycontour.memo1.lines[F]);

      if (lne3>=lne1)and(lne2>=lne3) then begin
        lneN:=trunc(((lne3-lne1)/2)+lne1);
        comparecase(E,lneN);
        skycontour.memo1.lines[E]:= inttostr(lneN);
        check:=false;
      end;
      if (lne3>=lne1)and(lne2<=lne1) then begin
        lneN:=trunc(((lne3-lne1)/2)+lne1);
        comparecase(E,lneN);
        skycontour.memo1.lines[E]:= inttostr(lneN);
        check:=false;
      end;
      if (lne3<=lne1)and(lne2<=lne1) then begin
        lneN:=trunc(((lne3-lne1)/2)+lne1);
        comparecase(E,lneN);
        skycontour.memo1.lines[E]:= inttostr(lneN);
        check:=false;
      end;
      if (lne3<=lne1)and(lne2>=lne1) then begin
        lneN:=trunc(((lne3-lne1)/2)+lne1);
        comparecase(E,lneN);
        skycontour.memo1.lines[E]:= inttostr(lneN);
        check:=false;
      end;
    end else check:=true;(*if check ture*)
  end;(* if lne<>0*)
  D:=D+(max_y);
  {N:=N+1;}
end;(*end of for max_x loop*)
M:=M+1;
end;(*end of for max_y loop*)

end;

procedure Tcommandbox.Sumcompare;
var
  max_x,max_y,C,D,E,F,G,H,lne1,lne2,lne3,lneN,panel:inte
ger;
  max,max1:integer; check:boolean;
  {lne1s,lne2s,lne3s:string;}
begin
  panel := strtoint(commandbox.label13.caption);
  max_x := strtoint(commandbox.label7.caption);
  max_y := strtoint(commandbox.label8.caption);

  C:=1; G:=1; D:=1; max1:=1;
  for c:=1 to max_x do begin

    for C:=1 to max_y do begin
      max := strtoint(skycontour.memo1.lines[G]);
      if max1 > max then max1:=max1 else
max1:=max;
      G:=G+1;
    end;

    E:=0; F:=0; check:=true;
    for c:=1 to max_y-2 do begin
      lne1 := strtoint(skycontour.memo1.lines[D]);
      if lne1=0 then begin
        E:=D+1; F:=D+2;
        lne2:=strtoint(skycontour.memo1.lines[E]);
        lne3:=strtoint(skycontour.Memo1.lines[F]);
        if (lne2>0) and (lne2>lne3) then begin
          lneN := trunc((lne3-lne1)*5/7);

```



```

    ill[D] := E||;
    skycontour.memo1.lines.add(inttostr(trunc(ill[D])));
    sky3d.memo1.lines.add(inttostr(trunc(ill[D])));
    D := D + 1;
    N := N + 1;
    end;
    M := M+1; N := 1;
end;
skycontour.visible := true;
end;

procedure Tcommandbox.panelcheck(panel:integer);
begin
    case panel of
    0: begin sky3d.memo1.visible:=true;
        sky3d.memo2.visible:=false;
        sky3d.memo3.visible:=false;
        sky3d.memo4.visible:=false;
        sky3d.memo5.visible:=false;
        sky3d.totalmemo.visible:=false;
    end;
    1: begin sky3d.memo1.visible:=true;
        sky3d.memo2.visible:=false;
        sky3d.memo3.visible:=false;
        sky3d.memo4.visible:=false;
        sky3d.memo5.visible:=false;
        sky3d.totalmemo.visible:=false;
    end;
    2: begin sky3d.memo1.visible:=false;
        sky3d.memo2.visible:=true;
        sky3d.memo3.visible:=false;
        sky3d.memo4.visible:=false;
        sky3d.memo5.visible:=false;
        sky3d.totalmemo.visible:=false;
    end;
    3: begin sky3d.memo1.visible:=false;
        sky3d.memo2.visible:=false;
        sky3d.memo3.visible:=true;
        sky3d.memo4.visible:=false;
        sky3d.memo5.visible:=false;
        sky3d.totalmemo.visible:=false;
    end;
    4: begin sky3d.memo1.visible:=false;
        sky3d.memo2.visible:=false;
        sky3d.memo3.visible:=false;
        sky3d.memo4.visible:=true;
        sky3d.memo5.visible:=false;
        sky3d.totalmemo.visible:=false;
    end;
    5: begin sky3d.memo1.visible:=false;
        sky3d.memo2.visible:=false;
        sky3d.memo3.visible:=false;
        sky3d.memo4.visible:=false;
        sky3d.memo5.visible:=true;
        sky3d.totalmemo.visible:=false;
    end;
end;

procedure Tcommandbox.panelcase(panel:integer;
ill:real);
begin
    case panel of
    1 : begin
        skycontour.memo1.lines.add(inttostr(trunc(ill)));
        sky3d.memo1.lines.add(inttostr(trunc(ill)));
    end;
    2 : begin
        skycontour.memo1.lines.add(inttostr(trunc(ill)));
        sky3d.memo2.lines.add(inttostr(trunc(ill)));
    end;
    3 : begin
        skycontour.memo1.lines.add(inttostr(trunc(ill)));
        sky3d.memo3.lines.add(inttostr(trunc(ill)));
    end;

```

```

end;
4 : begin
    skycontour.memo1.lines.add(inttostr(trunc(ill)));
    sky3d.memo4.lines.add(inttostr(trunc(ill)));
end;
5 : begin
    skycontour.memo1.lines.add(inttostr(trunc(ill)));
    sky3d.memo5.lines.add(inttostr(trunc(ill)));
end;
end;
end;

procedure Tcommandbox.panelcal;
var ill : array[1..2000] of real;
    panel,L,M,N,D,W,l_max,w_max : integer;
    tr,atm,irc,rfx,rfy,rfz,sfv,rm_w,rm_d,Ell : real;
    win_w,win_h,win_s,win_d,win_l : real;
    Tpoint,Rm_dx,Rm_dx1,rm_r,lng:real;
    Tcount,Rcount:integer;
begin
    panel := strtoint (commandbox.label13.caption);
    rm_w := strtoint(databox.label44.caption);
    rm_d := strtoint(databox.label45.caption);
    rm_dx := strtoint(databox.label40.caption);
    rfz := strtoint(databox.label62.caption);
    w_max := strtoint (commandbox.label7.caption);
    l_max := strtoint (commandbox.label8.caption);

    memoclear(panel);

    L:=1; W:=1; M := 1; N := 1; D := 1;
    intreflect(irc);
    trans_cal(tr);
    atm_cal(atm);

    (*Calculate without window depth*)
    if databox.label40.caption='0' then begin
        for W := 1 to W_max do begin
            rfx := rm_w*M/(W_max+1);
            for L := 1 to L_max do begin
                rfy := rm_d*N/(L_max+1);
                sf_cal(rfx,rfy,rfz,sfv);
                sumoutput(sfv,irc,tr,atm,ell);
                ill[D] := Ell;
                panelcase(panel,ill[D]);
                D := D + 1;
                N := N + 1;
            end;
            M := M+1; N := 1;
        end;
    end;(*end of if databox.label40.caption='0'*)

    (*Calculate with window depth*)
    if databox.label40.caption<>'0' then begin
        rm_dx1 := rm_d - rm_dx;
        lng := rm_d/(l_max+1);
        Tpoint := lng;
        Tcount := 0;
        while Tpoint < rm_dx do begin
            Tpoint := Tpoint + lng;
            Tcount := Tcount + 1;
        end;

        Rcount := l_max - Tcount;
        Rm_r := rm_dx1 - (Rcount*lng);

        L:=1; W:=1; M := 1; N := 1; D := 1;
        for W := 1 to W_max do begin
            rfx := rm_w*M/(W_max+1);
            (*Before windows*)
            for L := 1 to Tcount do begin

```

```

    ill[D] := 0;
    panelcase(panel,ill[D]);
    D := D + 1;
end;
(*After Windows*)
L:=1; N := 0; (*D := 1;*)
for L := 1 to Rcount do begin
    if N = 0 then begin
        if rm_r <> 0 then begin
            rfy := rm_r;
            sf_cal(rfx,rfy,rfz,sfv);
            sumoutput(sfv,irc,tr,atm,ell);
            ill[D] := Ell;
            panelcase(panel,ill[D]);
        end else begin
            ill[D] := 0;
            panelcase(panel,ill[D]);
        end;
    end;
    if N <> 0 then begin
        rfy := rm_r+(lng*N);
        sf_cal(rfx,rfy,rfz,sfv);
        sumoutput(sfv,irc,tr,atm,ell);
        ill[D] := Ell;
        panelcase(panel,ill[D]);
    end;
    (*sf_cal(rfx,rfy,rfz,sfv);
    sumoutput(sfv,irc,tr,atm,ell);
    ill[D] := Ell;
    panelcase(panel,ill[D]);*)
    D := D + 1;
    N := N + 1;
end; (*end of for*)
M := M+1; N := 1;
end;(*end of for loop of window depth*)
end;(*end of if databox.label40.caption<>'0'*)

    skycontour.visible := true;
end;

procedure TCommandbox.SpeedButton13Click(Sender:
TObject);
var panel : integer; max_x,max_y:integer;
    win_h,win_s,rfz,h1:real;
begin
    {win_h := strtfloat(databox.label48.caption);
    win_s := strtfloat(databox.label49.caption);
    rfz := strtfloat(databox.label62.caption);}
    max_y := strtoint(commandbox.label8.caption);
    max_x := strtoint(commandbox.label7.caption);

    {h1 := (win_h+win_s)-rfz;
    if h1<=0 then h1:=win_s;}

    if speedbutton12.caption='Return Command' then
begin
    mapcal;
    win_h := strtfloat(databox.label48.caption);
    win_s := strtfloat(databox.label49.caption);
    rfz := strtfloat(databox.label62.caption);
    h1 := (win_h+win_s)-rfz;
    if h1<=0 then h1:=win_s;

    if h1>(2.5*win_h) then begin
        case max_y of
            31..40 : begin sumcompare; end;
            41..50 : begin sumcompare;
                sumcompare;
                end;
        end;
    end;
    (*case max_x of
    41..50 : begin sumcompare;
        end;
    end;*)
end;

```

```

        end>(*end of if h1>2*winh*)
end; (*end of if Return command*)

if speedbutton12.caption='Next panels' then begin
panel := strtoint(commandbox.label13.caption);
case panel of
1 : begin
databox.label47.caption:= mulwindow.maskedit1.text;
databox.label48.caption:= mulwindow.maskedit2.text;
databox.label50.caption:= mulwindow.maskedit3.text;
databox.label49.caption:= mulwindow.maskedit4.text;
databox.label40.caption:= mulwindow.maskedit5.text;
        end;
2 : begin
databox.label47.caption := mulwindow.maskedit6.text;
databox.label48.caption := mulwindow.maskedit7.text;
databox.label50.caption := mulwindow.maskedit8.text;
databox.label49.caption := mulwindow.maskedit9.text;
databox.label40.caption:=mulwindow.maskedit10.text;
        end;
3 : begin with databox do begin
label47.caption := mulwindow.maskedit11.text;
label48.caption := mulwindow.maskedit12.text;
label50.caption := mulwindow.maskedit13.text;
label49.caption := mulwindow.maskedit14.text;
label40.caption := mulwindow.maskedit15.text;
        end;
        end;
4 : begin with databox do begin
label47.caption := mulwindow.maskedit16.text;
label48.caption := mulwindow.maskedit17.text;
label50.caption := mulwindow.maskedit18.text;
label49.caption := mulwindow.maskedit19.text;
label40.caption := mulwindow.maskedit20.text;
        end;
        end;
end;
end;

```

```

panelcal;
win_h := strtfloat(databox.label48.caption);
win_s := strtfloat(databox.label49.caption);
rfz := strtfloat(databox.label62.caption);
h1 := (win_h+win_s)-rfz;
if h1<=0 then h1:=win_s;

        if h1>(2.5*win_h) then begin
                case max_y of
31..40 : begin sumcompare; end;
41..50 : begin sumcompare;
sumcompare;
                end;
                end;
        end;
(*case max_x of
29..40 : begin sumcompare;
                sumcompare; end;
41..50 : begin sumcompare;
sumcompare;
                sumcompare;
                end;
        end;*)
end>(*end of if h1>2*winh*)
end>(*end of if begin*)

commandbox.SpeedButton15.enabled:=true;
commandbox.SpeedButton20.enabled:=true;
end;

procedure TCommandbox.SpeedButton15Click(Sender:
TObject);
begin
        sky3d.visible:=true;
end;

procedure TCommandbox.SpeedButton19Click(Sender:
TObject);

```



```

var pz1,pz2,pz3,pz4,pz5,tpz,D:integer;
begin
    skycontour.memo1.lines.clear;
    sky3d.totalmemo.lines.clear;
    with sky3d do begin
        memo1.visible:=false;
        memo2.visible:=false;
        memo3.visible:=false;
        memo4.visible:=false;
        memo5.visible:=false;
        Totalmemo.visible:=true;

        D:=1; totalmemo.lines[0]:='Integrate';
            skycontour.memo1.lines[0]:='Integrate';
        repeat
            if memo1.lines[1]<>" then pz1:=strtoint(memo1.lines[D])
                else pz1:=0;
            if memo2.lines[1]<>" then pz2:=strtoint(memo2.lines[D])
                else pz2:=0;
            if memo3.lines[1]<>" then pz3:=strtoint(memo3.lines[D])
                else pz3:=0;
            if memo4.lines[1]<>" then pz4:=strtoint(memo4.lines[D])
                else pz4:=0;
            if memo5.lines[1]<>" then pz5:=strtoint(memo5.lines[D])
                else pz5:=0;

            tpz := pz1 + pz2 + pz3 + pz4 + pz5;
            sky3d.totalmemo.lines.add(inttostr(tpz));
            skycontour.memo1.lines.add(inttostr(tpz));
            D:= D+1;
        until memo1.lines[D]="";
        end; (*end of with*)
    end;

    label11.caption := 'Select analysis row :';
    label11.visible := true;

    speedbutton12.enabled := false;
    speedbutton13.enabled := false;
    speedbutton15.enabled := false;
    speedbutton16.enabled := false;
    speedbutton17.enabled := false;
    speedbutton18.enabled := false;
end;

end.

procedure TCommandbox.SpeedButton20Click(Sender:
TObject);
begin
    label5.caption := '0'; label6.caption := '0';

```

unit blinds;

```

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Buttons, ComCtrls,
  ExtCtrls;
type
  Tblinds = class(TForm)
    GroupBox1: TGroupBox;
    Panel1: TPanel;
    Panel2: TPanel;
    Image1: TImage;
    Image2: TImage;
    Label1: TLabel;
    Label2: TLabel;
    RadioButton1: TRadioButton;
    Shape1: TShape;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    RadioButton4: TRadioButton;
    RadioButton5: TRadioButton;
    GroupBox2: TGroupBox;
    Panel3: TPanel;
    Image3: TImage;
    Shape2: TShape;
    TrackBar1: TTrackBar;
    Label3: TLabel;
    Label4: TLabel;
    Shape3: TShape;
    Shape4: TShape;
    SpeedButton1: TSpeedButton;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    CheckBox1: TCheckBox;
    Label16: TLabel;
    Label17: TLabel;
    Label18: TLabel;
    Label19: TLabel;
    Label20: TLabel;
    Label21: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    SpeedButton2: TSpeedButton;
    procedure SpeedButton1Click(Sender: TObject);
    procedure RadioButton1Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
    procedure RadioButton3Click(Sender: TObject);
    procedure RadioButton4Click(Sender: TObject);
    procedure RadioButton5Click(Sender: TObject);
    procedure TrackBar1Change(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  blinds: Tblinds;
implementation
uses Unit1, skyfac;
{$R *.DFM}

```

```

procedure bndtable(Bref,Bang,Sang:integer;var
Bua,Bka,Bga,Buu,Bku,Bgu:string);
var fri:textfile; B_ref1,B_ref2 : string;
C,B_ang,S_ang:integer;
B_ua,B_ka,B_ga,B_uu,B_ku,B_gu:string;
checked : boolean;
begin
  if Bref = 10 then B_ref1 := 'Ten Percent Blinds
Reflectance';
  if Bref = 30 then B_ref1 := 'Thirty Percent Blinds
Reflectance';
  if Bref = 50 then B_ref1 := 'Fifty Percent Blinds
Reflectance';
  if Bref = 70 then B_ref1 := 'Seventy Percent Blinds
Reflectance';
  if Bref = 90 then B_ref1 := 'Ninety Percent Blinds
Reflectance';
  if Bang = 0 then B_ang := 11;
  if Bang = 15 then B_ang := 21;
  if Bang = 30 then B_ang := 31;
  if Bang = 45 then B_ang := 41;
  if Bang = 60 then B_ang := 51;
  case Sang of
0..7 : S_ang := 4;
8..22 : S_ang := 5;
23..37 : S_ang := 6;
38..52 : S_ang := 7;
53..67 : S_ang := 8;
68..82 : S_ang := 9;
83..90 : S_ang := 10;
end;
if Sang > 90 then S_ang := 10;

Assignfile(fri,'data\blinds.dat');
reset(fri);
checked := false;
while NOT Eof(fri) and (checked = false) do

```

```

begin
  Readln (fri,B_ref2);
  if B_ref2 = B_ref1 then begin
    for C := 1 to S_ang do begin
      Readln(fri,B_ua);
      end>(*begin of for loop*)
    for C := 1 to 18 do begin
      Readln(fri,B_uu);
      end;
      checked := true;
    end;
  end>(*While*)
closefile(fri);

Assignfile(fri,'data\blinds.dat');
reset(fri);
checked := false;
while NOT Eof(fri) and (checked = false) do
begin
  Readln (fri,B_ref2);
  if B_ref2 = B_ref1 then begin
    for C := 1 to 14 do begin
      Readln(fri,B_ka);
      end>(*begin of for loop*)
    for C := 1 to 4 do begin
      Readln(fri,B_ga);
      end>(*begin of for loop*)
    for C := 1 to 14 do begin
      Readln(fri,B_ku);
      end>(*begin of for loop*)
    for C := 1 to 4 do begin
      Readln(fri,B_gu);
      end>(*begin of for loop*)
      checked := true;
    end;
  end>(*While*)
closefile(fri);

```

```

Bua := copy(B_ua,B_ang,5);
Bka := copy(B_ka,B_ang,5);
Bga := copy(B_ga,B_ang,5);
Buu := copy(B_uu,B_ang,5);
Bku := copy(B_ku,B_ang,5);
Bgu := copy(B_gu,B_ang,5);
end;

procedure Tblinds.SpeedButton1Click(Sender: TObject);
var Bang,Sang,Bref:integer;
Bua,Bka,Bga,Buu,Bku,Bgu:string;
begin
    if checkbox1.checked = false then begin
    if radiobutton1.Checked = true then Bref := 10;
    if radiobutton2.Checked = true then Bref := 30;
    if radiobutton3.Checked = true then Bref := 50;
    if radiobutton4.Checked = true then Bref := 70;
    if radiobutton5.Checked = true then Bref := 90;
    Bang := strtoint(label8.caption);
    Sang := strtoint(label9.caption);
    bndtable(Bref,Bang,Sang,Bua,Bka,Bga,Buu,Bku,Bgu);
    label10.caption := Bua;
    label11.caption := Bka;
    label12.caption := Bga;
    label13.caption := Buu;
    label14.caption := Bku;
    label15.caption := Bgu;
    end else begin
    label10.caption := '1.00';
    label11.caption := '1.00';
    label12.caption := '1.00';
    label13.caption := '1.00';
    label14.caption := '1.00';
    label15.caption := '1.00';
    end;
end;

```

```

skyfactor.label70.caption := label10.caption;
skyfactor.label71.caption := label11.caption;
skyfactor.label72.caption := label12.caption;
skyfactor.label73.caption := label13.caption;
skyfactor.label74.caption := label14.caption;
skyfactor.label75.caption := label15.caption;
form1.enabled := true;
close;
end;

procedure Tblinds.RadioButton1Click(Sender: TObject);
begin
    image2.picture.loadfromfile('image\shade10.bmp');
end;

procedure Tblinds.RadioButton2Click(Sender: TObject);
begin
    image2.picture.loadfromfile('image\shade30.bmp');
end;

procedure Tblinds.RadioButton3Click(Sender: TObject);
begin
    image2.picture.loadfromfile('image\shade50.bmp');
end;

procedure Tblinds.RadioButton4Click(Sender: TObject);
begin
    image2.picture.loadfromfile('image\shade70.bmp');
end;

procedure Tblinds.RadioButton5Click(Sender: TObject);
begin
    image2.picture.loadfromfile('image\shade90.bmp');
end;

procedure Tblinds.TrackBar1Change(Sender: TObject);
var ang : integer;

```

```

begin
  ang := trackbar1.position;
  case ang of
  1 : begin
    image3.picture.loadfromfile('image\bindang10.bmp');
    label8.caption := '0';
    end;
  2 : begin
    image3.picture.loadfromfile('image\bindang15.bmp');
    label8.caption := '15';
    end;
  3 : begin
    image3.picture.loadfromfile('image\bindang30.bmp');
    label8.caption := '30';
    end;
  4 : begin
    image3.picture.loadfromfile('image\bindang45.bmp');
    label8.caption := '45';
    end;
  5 : begin
    image3.picture.loadfromfile('image\bindang60.bmp');
    label8.caption := '60';
    end;
  end;
end;

```

```

procedure Tblinds.CheckBox1Click(Sender: TObject);

```

```

begin
  if checkbox1.checked = true then begin
    radiobutton1.enabled := false;
    radiobutton2.enabled := false;
    radiobutton3.enabled := false;
    radiobutton4.enabled := false;
    radiobutton5.enabled := false;
    trackbar1.enabled := false;
  end else begin
    radiobutton1.enabled := true;

```

```

    radiobutton2.enabled := true;
    radiobutton3.enabled := true;
    radiobutton4.enabled := true;
    radiobutton5.enabled := true;
    trackbar1.enabled := true;

```

```

  end;

```

```

end;

```

```

procedure Tblinds.SpeedButton2Click(Sender: TObject);

```

```

begin

```

```

  if label10.visible = false then begin

```

```

    label10.visible := true; label11.visible := true;
    label12.visible := true; label13.visible := true;
    label14.visible := true; label15.visible := true;
    label16.visible := true; label17.visible := true;
    label18.visible := true; label19.visible := true;
    label20.visible := true; label21.visible := true;

```

```

  end else begin

```

```

    label10.visible := false; label11.visible := false;
    label12.visible := false; label13.visible := false;
    label14.visible := false; label15.visible := false;
    label16.visible := false; label17.visible := false;
    label18.visible := false; label19.visible := false;
    label20.visible := false; label21.visible := false;

```

```

  end;

```

```

end;

```

```

end.

```

unit boxdata;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, Buttons, StdCtrls, ExtCtrls;

type

TDataBox = class(TForm)

Shape1: TShape;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

Label24: TLabel;

Label25: TLabel;

Label26: TLabel;

Label27: TLabel;

Label28: TLabel;

Label29: TLabel;

Label30: TLabel;

Label31: TLabel;

Label32: TLabel;

Label33: TLabel;

Label34: TLabel;

Label35: TLabel;

Label36: TLabel;

Label37: TLabel;

Shape2: TShape;

SpeedButton1: TSpeedButton;

Label41: TLabel;

Label42: TLabel;

Label43: TLabel;

Label44: TLabel;

Label45: TLabel;

Label46: TLabel;

Label47: TLabel;

Label48: TLabel;

Label49: TLabel;

Label50: TLabel;

Label51: TLabel;

Label52: TLabel;

Label53: TLabel;

Label54: TLabel;

Label55: TLabel;

Label56: TLabel;

Label57: TLabel;

Label58: TLabel;

Label59: TLabel;

Label60: TLabel;

Label61: TLabel;

Label62: TLabel;

Label63: TLabel;

Label64: TLabel;

Label65: TLabel;

```

Label66: TLabel;
Label67: TLabel;
Label68: TLabel;
Label38: TLabel;
Label40: TLabel;
procedure SpeedButton1Click(Sender: TObject);
procedure readdata;
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Databox: TDatabox;
implementation

uses Unit2, Unit1, Unit5, Unit4, intref, Unit3, blind;
{$R *.DFM}

procedure TDatabox.readdata;
begin
  with databox do begin
    if newdata.radiobutton1.checked then
label27.caption := 'Clear sky';
    if newdata.radiobutton2.checked then
label27.caption := 'Partly cloudy sky';
    if newdata.radiobutton3.checked then
label27.caption := 'Overcast sky';
    label41.caption := newdata.edit1.text;
    label42.Caption := newdata.edit2.text;
    label43.Caption := newdata.edit3.text;
    label44.caption := roomdata.edit1.text;
    label45.caption := roomdata.edit2.text;
    label46.caption := roomdata.edit3.text;
    label47.caption := roomdata.edit4.text;
    label48.caption := roomdata.edit5.text;
    label49.caption := roomdata.edit6.text;
    label50.caption := roomdata.edit7.text;
    label51.caption := atmos.label4.caption;
    label52.caption := atmos.label8.caption;
    label53.caption := atmos.label7.caption;
    label54.caption := intref.label4.caption;
    label55.caption := intref.label5.caption;
    label56.caption := intref.label6.caption;
    label57.caption := lighttrans.label4.caption;
    label58.caption := lighttrans.edit1.text;
    label59.caption := lighttrans.edit2.text;
    label60.caption := roomdata.edit8.text;
    label61.caption := roomdata.edit9.text;
    label62.caption := roomdata.edit10.text;
    label63.caption := blinds.label10.caption;
    label64.caption := blinds.label11.caption;
    label65.caption := blinds.label12.caption;
    label66.caption := blinds.label13.caption;
    label67.caption := blinds.label14.caption;
    label68.caption := blinds.label15.caption;
  end;
end;

procedure TDatabox.SpeedButton1Click(Sender:
TObject);
begin
  form1.enabled := true;
  close;
end;

end.

```

unit calarea;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, Grids, Buttons,
ExtCtrls, Mask;

type

TAreacal = class(TForm)

Shape1: TShape;

Shape2: TShape;

SpeedButton1: TSpeedButton;

StringGrid1: TStringGrid;

Label1: TLabel;

SpeedButton2: TSpeedButton;

SpeedButton3: TSpeedButton;

Shape3: TShape;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Shape4: TShape;

Label11: TLabel;

MaskEdit1: TMaskEdit;

MaskEdit2: TMaskEdit;

MaskEdit3: TMaskEdit;

MaskEdit4: TMaskEdit;

MaskEdit5: TMaskEdit;

Label12: TLabel;

Shape5: TShape;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

SpeedButton4: TSpeedButton;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

Label24: TLabel;

Label25: TLabel;

Label26: TLabel;

Label27: TLabel;

Label28: TLabel;

Label29: TLabel;

Label30: TLabel;

Label31: TLabel;

Label32: TLabel;

MaskEdit6: TMaskEdit;

Label33: TLabel;

Label34: TLabel;

MaskEdit7: TMaskEdit;

Label35: TLabel;

MaskEdit8: TMaskEdit;

SpeedButton5: TSpeedButton;

MaskEdit9: TMaskEdit;

Shape7: TShape;

Label36: TLabel;

Label37: TLabel;

Label38: TLabel;

MaskEdit10: TMaskEdit;

Label39: TLabel;

Label40: TLabel;

Shape6: TShape;

SpeedButton6: TSpeedButton;


```

Label41: TLabel;
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure area_cal;
procedure areasum(var points:integer);
procedure requiredarea(points:integer; var area:real);
procedure rrcal(var rcr:real);
procedure copcal(var cop:real);
procedure clearcell;
procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton4Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure wattmax(Ei:integer; var max:integer);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Areacal: TAreacal;
implementation
uses bcommand, boxdata, grharea, daywith;

{$R *.DFM}

procedure TAreacal.SpeedButton1Click
  (Sender: TObject);
begin
  label5.visible :=false; label6.visible:=false;
  label7.visible :=false; label8.visible:=false;
  label10.visible:=false;
  label28.visible:=false; label29.visible:=false;
  label30.visible:=false; label31.visible:=false;
  label32.visible:=false;
close;
end;

procedure Tareacal.clearcell;
var C,D,E,F : integer;
begin
  C:=0; D:=1; E:=1; F:=1;
  for E:=1 to 6 do begin
    for F:=1 to 50 do begin
      stringgrid1.cells[C,D] := "";
      D:=D+1;
    end;
    C:=C+1; D:=1;
  end;
end;

procedure TAreacal.area_cal;
const Colhead : array[0..5] of string =
  ('Dp.Points',' Curve','Required',' Surplus','
Under','Summary');
  rowhead : array[0..50] of string =
  (' 0',' 1',' 2',' 3',' 4',' 5',' 6',
' 7',' 8',' 9',' 10',' 11',' 12',' 13',
' 14',' 15',' 16',' 17',' 18',' 19',' 20',
' 21',' 22',' 23',' 24',' 25',' 26',' 27',
' 28',' 29',' 30',' 31',' 32',' 33',' 34',
' 35',' 36',' 37',' 38',' 39',' 40',' 41',
' 42',' 43',' 44',' 45',' 46',' 47',' 48',
' 49',' 50');
var C,D,rm_d,max_y,mline,req,cv:integer;
  base:real; cv1:string;
begin
  rm_d := strtoint(databox.label45.caption);
  mline := strtoint(grapharea.maskedit2.text);
  max_y := strtoint(commandbox.label8.caption);
  if max_y = 0 then max_y := 1;
  base := rm_d / max_y;

```

```

clearcell;
(*Draw col and row head*)
for c:=0 to 5 do begin
    stringgrid1.Cells[c,0] := colhead[c];
end;
for c:=0 to max_y do begin
    stringgrid1.cells[0,c+1] := rowhead[c];
end;
(*Draw illuminance*)
D:=1;
for c:=1 to max_y do begin
stringgrid1.cells[1,D+1] := grapharea.memo1.lines[D];
    D:=D+1;
end;
(*Draw required illuminance area*)
req := trunc(mline*base);
D:=1;
for c:=1 to max_y+1 do begin
    stringgrid1.cells[2,D] := inttostr(mline);
    D:=D+1;
end;
(*Draw surplus illuminance*)
D:=1;
for c:=1 to max_y+1 do begin
    cv1 := stringgrid1.cells[1,D];
    if cv1="" then cv := 0;
    if cv1<>" then cv := strtoint(cv1);
    (*Cv := strtoint(stringgrid1.cells[1,D]);*)
    if cv > mline then
        stringgrid1.cells[3,D] := inttostr(cv - mline);
    if (mline > cv) and (cv<>0) then
        stringgrid1.cells[4,D] := inttostr(mline - cv);
    D:=D+1;
end;
D:=1;
for c:=1 to max_y+1 do begin
    Cv1 := stringgrid1.cells[4,D];
    if cv1<>" then
        stringgrid1.cells[5,D]:=inttostr(mline*strtoint
(cv1))
    else stringgrid1.cells[5,D]:='0';
    D:=D+1;
end;
end;
procedure TAreacal.SpeedButton2Click(Sender:
TObject);
begin
    area_cal;
end;
procedure Tareacal.areasum(var points:integer);
var C,D,CV,sum,mline,max_y(*points1*):integer;
req,req1:real;
begin
    label5.visible:=true; label6.visible:=true;
    label7.visible:=true; label8.visible:=true;
    label10.visible:=true;
    max_y := strtoint(commandbox.label8.caption);
    mline := strtoint(grapharea.maskedit2.text);
    points:=0; (*points1:=0;*) sum:=0;
    D:=1;
    for c:=1 to max_y+1 do begin
        cv := strtoint(stringgrid1.cells[5,D]);
        if cv<>0 then begin
            sum := sum+cv;
            points := points+1;
        end else begin
            sum := sum+cv;
            points := points+1;
            (*points1:=points1+1;*)
        end;
    D:=D+1;

```

```
end;
```

```
(*if points = 0 then points:=points1;*)
req := sum/(points*mline);
req1 := req*100/mline;
label5.caption:= grapharea.maskedit2.text;
label6.caption:= inttostr(sum);
label7.caption:= inttostr(points);
label8.caption:= format('%8.2n',[req]);
label10.caption:= format('%8.2n',[req1]);
```

```
end;
```

```
procedure TAreacal.requiredarea(points:integer; var
area:real);
```

```
var max_y:integer;
```

```
rm_w,rm_d,rm_r:real;
```

```
begin
```

```
max_y := strtoint(commandbox.label8.caption);
```

```
if Max_y = 0 then max_y := 1;
```

```
rm_w := strtfloat(databox.label44.caption);
```

```
rm_d := strtfloat(databox.label45.caption);
```

```
rm_r := (points+1)*(rm_d/max_y);
```

```
area := rm_r*rm_w;
```

```
maskedit1.text := format('%3.2n',[area]);
```

```
end;
```

```
procedure TAreacal.rcrcal(var rcr:real);
```

```
var rm_w,rm_d,rm_h: real;
```

```
begin
```

```
rm_w := strtfloat(Databox.label44.caption);
```

```
rm_d := strtfloat(Databox.label45.caption);
```

```
rm_h := strtfloat(Databox.label46.caption);
```

```
rcr := 5*rm_h*(rm_w+rm_d)/(rm_w*rm_d);
```

```
maskedit6.text := format('%0.2n',[rcr]);
```

```
end;
```

```
procedure TAreacal.SpeedButton3Click(Sender:
TObject);
```

```
var points:integer;
```

```
area,rcr,cop:real;
```

```
begin
```

```
areasum(points);
```

```
requiredarea(points,area);
```

```
rcrcal(rcr);
```

```
copcal(cop);
```

```
end;
```

```
procedure TAreacal.SpeedButton4Click(Sender:
TObject);
```

```
var req,req1,Ecost:real;
```

```
mline,sum,points,WoL,WoB:integer;
```

```
cu,llf,TWoL,TWoB,Heat,Cool,Teng,cop,area,effi,ill:real;
```

```
begin
```

```
label28.visible:=true; label29.visible:=true;
```

```
label30.visible:=true; label31.visible:=true;
```

```
label32.visible:=true;
```

```
mline := strtoint(label5.caption);
```

```
sum := strtoint(label6.caption);
```

```
points := strtoint(label7.caption);
```

```
req := sum/(points*mline);
```

```
req1 := req*100/mline;
```

```
WoL := strtoint(maskedit2.text);
```

```
WoB := strtoint(maskedit3.text);
```

```
cu := strtfloat(maskedit4.text);
```

```
llf := strtfloat(maskedit5.text);
```

```
cop := strtfloat(maskedit9.text);
```

```
ill := strtfloat(maskedit10.text);
```

```
requiredarea(points,area);
```

```
effi := ill/WoL;
```

```

TWoL := (req*area)/(effi*cu*lf);
TWoB := TWoL*(1+(WoB/WoL));
Heat := TWoL*1*1.2;
Cool := Heat/cop;

TEng := TWoB + Cool;

label28.Caption := format('%8.2n',[TWoL]);
label29.Caption := format('%8.2n',[TWoB]);
label30.Caption := format('%8.2n',[Heat]);
label31.Caption := format('%8.2n',[Cool]);
label32.Caption := format('%8.2n',[TEng]);
end;

procedure Tareacal.copcal(var cop:real);
var btu,w:real;
begin
    Btu := trunc(strtfloat(maskedit7.Text));
    w := strtfloat(maskedit8.text);
    cop := Btu/(w*3.412);
    maskedit9.text := format('%4.2n',[cop]);
end;

procedure TAreacal.SpeedButton5Click(Sender: TObject);
var cop:real;
begin
    copcal(cop);
end;

procedure TAreacal.wattmax(Ei:integer; var max:integer);
begin
    with Withdaylight do begin
        case Ei of
            0..40 : begin max := 40;
                    label22.caption:=' 40'; label23.caption:=' 30';
                    label24.caption:=' 20'; label25.caption:=' 10';
                    end;
            41..50 : begin max := 50;
                    label22.caption:=' 50'; label23.caption:=' 38';
                    label24.caption:=' 25'; label25.caption:=' 13';
                    end;
            51..80 : begin max := 80;
                    label22.caption:=' 80'; label23.caption:=' 60';
                    label24.caption:=' 40'; label25.caption:=' 20';
                    end;
            81..100 : begin max := 100;
                    label22.caption:='100'; label23.caption:=' 75';
                    label24.caption:=' 50'; label25.caption:=' 25';
                    end;
            101..150 : begin max := 150;
                    label22.caption:='150'; label23.caption:='113';
                    label24.caption:=' 75'; label25.caption:=' 38';
                    end;
            151..200 : begin max := 200;
                    label22.caption:='200'; label23.caption:='150';
                    label24.caption:='100'; label25.caption:=' 50';
                    end;
            201..400 : begin max := 400;
                    label22.caption:='400'; label23.caption:='300';
                    label24.caption:='200'; label25.caption:='100';
                    end;
            401..500 : begin max := 500;
                    label22.caption:='500'; label23.caption:='375';
                    label24.caption:='250'; label25.caption:='125';
                    end;
            501..800 : begin max := 800;
                    label22.caption:='800'; label23.caption:='600';
                    label24.caption:='400'; label25.caption:='200';
                    end;
            801..1000 : begin max := 1000;
                    label22.caption:='1000'; label23.caption:='750';
                    label24.caption:='500'; label25.caption:='250';
                    end;
            1001..1500 : begin max := 1500;

```

```

label22.caption:='1500'; label23.caption:='1125';
label24.caption:='750'; label25.caption:='375';
    end;
1501..2000 : begin max := 2000;
    label22.caption:='2000'; label23.caption:='1500';
    label24.caption:='1000'; label25.caption:='750';
    end;
2001..4000 : begin max := 4000;
    label22.caption:='4000'; label23.caption:='3000';
    label24.caption:='2000'; label25.caption:='1000';
    end;
4001..6000 : begin max := 6000;
    label22.caption:='6000'; label23.caption:='4500';
    label24.caption:='3000'; label25.caption:='1500';
    end;
6001..10000 : begin max := 10000;
    label22.caption:='10000'; label23.caption:='7500';
    label24.caption:='5000'; label25.caption:='2500';
    end;
10001..15000 : begin max := 15000;
    label22.caption:='15000'; label23.caption:='12500';
    label24.caption:='7500'; label25.caption:='3750';
    end;
15001..20000 : begin max := 20000;
    label22.caption:='20000'; label23.caption:='15000';
    label24.caption:='10000'; label25.caption:='5000';
    end;
20001..40000 : begin max := 40000;
    label22.caption:='40000'; label23.caption:='35000';
    label24.caption:='20000'; label25.caption:='10000';
    end;
40001..50000 : begin max := 50000;
    label22.caption:='50000'; label23.caption:='37500';
    label24.caption:='25000'; label25.caption:='12500';
    end;
50001..100000 : begin max := 100000;
    label22.caption:='100000'; label23.caption:='75000';
    label24.caption:='50000'; label25.caption:='25000';
    end;
100001..400000 : begin max := 400000;
    label22.caption:='400000'; label23.caption:='300000';
    label24.caption:='200000'; label25.caption:='100000';
    end;
end;
end>(*End of with*)
end;

procedure TAreacal.SpeedButton6Click(Sender:
TObject);
var Ecost,Ecost1,mline,req,TEng1,TEng2:real;
    Wmax,WoL,WoB,points,sum,GraphH,GraphT:integer;
    cu,llf,TWol,TWoB,Heat,Cool,TEng,cop,area,effi,ill:real;
begin
    Ecost:=
    strtofloat(Withdaylight.Maskedit1.text)/1000;
    mline := strtfloat(label5.caption);
    sum := strtoint(label6.caption);
    points := strtoint(label7.caption);
    req := sum/(points*mline);

    WoL := strtoint(maskedit2.text);
    WoB := strtoint(maskedit3.text);
    cu := strtfloat(maskedit4.text);
    llf := strtfloat(maskedit5.text);
    cop := strtfloat(maskedit9.text);
    ill := strtfloat(maskedit10.text);

    requiredarea(points,area);
    effi := ill/WoL;

    GraphH:=withdaylight.shape11.Height;
    GraphT:=withdaylight.shape11.Top;
    (*Without DAYLIGHT*)
    with withdaylight do begin

```

```

TWoL := (mline*Area)/(effi*cu*lf);
TWoB := TWoL*(1+(WoB/WoL));
Heat := TWoL*1*1.2;
Cool := Heat/cop;
TEng := TWoB + Cool;
label28.caption := format('%8.2n',[mline]);
label29.caption := format('%8.2n',[TWoL]);
label30.caption := format('%8.2n',[TWoB]);
label38.caption := format('%8.2n',[Heat]);
label39.caption := format('%8.2n',[Cool]);
label40.caption := format('%8.2n',[TEng]);
label3.Caption := format('%8.2n',[TEng*Ecost]);
Teng1 := Teng;
wattmax(trunc(TEng),Wmax);
shape14.top:= GraphT+trunc(GraphH-
(TEng*GraphH/Wmax));
shape14.height:= trunc(TEng*GraphH/Wmax);
end;

(*Using DAYLIGHT*)
with withdaylight do begin
TWoL := (req*Area)/(effi*cu*lf);
TWoB := TWoL*(1+(WoB/WoL));
Heat := TWoL*1*1.2;
Cool := Heat/cop;
TEng := TWoB + Cool;
label31.caption := format('%8.2n',[req]);
label32.caption := format('%8.2n',[TWoL]);
label33.caption := format('%8.2n',[TWoB]);
label34.caption := format('%8.2n',[Heat]);
label35.caption := format('%8.2n',[Cool]);
label36.caption := format('%8.2n',[TEng]);
label37.Caption := format('%8.2n',[TEng*Ecost]);
Teng2 := Teng;
shape13.top:= GraphT+trunc(GraphH-
(TEng*GraphH/Wmax));
shape13.height:= trunc(TEng*GraphH/Wmax);
end;
Withdaylight.label44.caption :=
format('%8.2n',[TEng1-TEng2]);
withdaylight.visible := true;
end;
end.

```

unit contour;

```

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Buttons,
  Mask, printers;
type
  Tskycontour = class(TForm)
    SpeedButton1: TSpeedButton;
    Shape1: TShape;
    SpeedButton2: TSpeedButton;
    Shape3: TShape;
    Label1: TLabel;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    Shape2: TShape;
    Label2: TLabel;
    Label3: TLabel;
    Memo1: TMemo;
    Shape4: TShape;
    Memo2: TMemo;
    PrintDialog1: TPrintDialog;
    Label4: TLabel;
    MaskEdit1: TMaskEdit;
    CheckBox1: TCheckBox;
    Label5: TLabel;
    MaskEdit2: TMaskEdit;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
  end;

```

```

private
  { Private declarations }
public
  { Public declarations }
end;

var
  skycontour: Tskycontour;

implementation

uses skyfac, Unit1, bcommand;
{$R *.DFM}

procedure Tskycontour.SpeedButton1Click(Sender:
TObject);
begin
  close;
end;

procedure Tskycontour.SpeedButton3Click(Sender:
TObject);
var x_max,y_max,D,l,lng,scale:integer;
    mat,ln,ln1,line:string;
begin
  skycontour.memo2.Lines.clear;
  x_max := strtoint(commandbox.label7.caption);
  y_max := strtoint(commandbox.label8.caption);
  scale := trunc(strtofloat(maskedit2.Text));

  mat := 'Matrix width x depth = '+
        inttostr(x_max) + ' x ' + inttostr(y_max);
  skycontour.memo2.lines[0] := mat;
  skycontour.memo2.font.size := 1*scale;

  D := 1; L := 1; line:="";

```

```

for l:=1 to x_max do begin
  for l:=1 to y_max do begin
    lne := memo1.lines[D];
    lng := length(lne);
    case lng of
      1 : lne1 := ' ' +lne;
      2 : lne1 := ' ' +lne;
      3 : lne1 := ' ' +lne;
      4 : lne1 := ' ' +lne;
      5 : lne1 := ' ' +lne;
      6 : lne1 := ' ' +lne;
    end;
    line := line + lne1;
    D := D+1;
  end;
  memo2.lines.add(line);
  line := "";
end;

end;

procedure Tskycontour.SpeedButton2Click(Sender:
TObject);
var Poutput:textfile; N:longint;
begin
  if label4.visible = false then begin
    label4.visible:=true; maskedit1.visible:=true;
    checkbox1.visible:=true;
  end else begin
    printdialog1.execute;
    assignpm(poutput);
    rewrite(poutput);
    if checkbox1.checked=true then
      printer.canvas.font.style := [fsbold];
    printer.canvas.font.size := strtoint(maskedit1.text);
    writeln(poutput,"");
    writeln(poutput,"");
    writeln(poutput,"");
  end;
end;

procedure Tskycontour.SpeedButton4Click(Sender:
TObject);
begin
  with commandbox do begin
    label5.caption := '0'; label6.caption := '0';
    label11.caption := 'Select analysis row :';
    label11.visible := true;
  end;
end;

procedure Tskycontour.SpeedButton5Click(Sender:
TObject);
begin
  commandbox.Sumcompare;
end;

procedure Tskycontour.SpeedButton6Click(Sender:
TObject);
begin
  commandbox.rowcompare;
end;
end.

```


unit datawth;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, ExtCtrls, ComCtrls,
Grids, Calendar, Buttons, Math, Mask;

type

Twthdata = class(TForm)

GroupBox1: TGroupBox;

Calendar1: TCalendar;

UpDown1: TUpDown;

UpDown2: TUpDown;

SpeedButton1: TSpeedButton;

Panel1: TPanel;

Shape1: TShape;

Label1: TLabel;

Panel2: TPanel;

Shape2: TShape;

Label2: TLabel;

ComboBox1: TComboBox;

ComboBox2: TComboBox;

Label3: TLabel;

Shape3: TShape;

Shape4: TShape;

Shape5: TShape;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Shape6: TShape;

SpeedButton2: TSpeedButton;

Shape7: TShape;

Image1: TImage;

Shape8: TShape;

Label8: TLabel;

SpeedButton3: TSpeedButton;

MaskEdit1: TMaskEdit;

ComboBox3: TComboBox;

MaskEdit2: TMaskEdit;

MaskEdit3: TMaskEdit;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

ComboBox4: TComboBox;

Image2: TImage;

SpeedButton4: TSpeedButton;

procedure UpDown1Click

(Sender: TObject; Button: TUDBtnType);

procedure SpeedButton1Click(Sender: TObject);

procedure UpDown2Click

(Sender: TObject; Button: TUDBtnType);

procedure SpeedButton2Click(Sender: TObject);

procedure SpeedButton3Click(Sender: TObject);

procedure another_az

(mth,dat,time:integer; var at,az:integer);

procedure sunat_az

(mth,dat,tme:integer; var at,az:integer);

procedure SpeedButton4Click(Sender: TObject);

procedure ComboBox2Change(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

wthdata: Twthdata;

implementation

uses Unit1, Unit2, diretwin;

```
{SR *.DFM}
```

```
procedure Twthdata.UpDown1Click(Sender: TObject;
Button: TUDBtnType);
```

```
var mth,dat : integer;
```

```
begin
```

```
    mth := updown1.position;
```

```
    dat := strtoint(label2.caption);
```

```
    case mth of
```

```
    1 : begin calendar1.month := mth;
```

```
        label1.caption := 'Jan';
```

```
    end;
```

```
    2 : begin label1.caption := 'Feb';
```

```
        if(label2.caption='30')or(label2.caption='31')then
```

```
            begin calendar1.day := 28;
```

```
            label2.caption := '28';
```

```
        end;
```

```
        calendar1.month := mth;
```

```
    end;
```

```
    3 : begin calendar1.month := mth;
```

```
        label1.caption := 'Mar';
```

```
    end;
```

```
    4 : begin label1.caption := 'Apr';
```

```
        if label2.caption='31'then
```

```
            begin calendar1.day := 30;
```

```
            label2.caption:='30';
```

```
        end;
```

```
        calendar1.month := mth;
```

```
    end;
```

```
    5 : begin calendar1.month := mth;
```

```
        label1.caption := 'May';
```

```
    end;
```

```
    6 : begin label1.caption := 'Jun';
```

```
        if label2.caption='31'then
```

```
            begin calendar1.day := 30;
```

```
            label2.caption:='30';
```

```
        end;
```

```
        calendar1.month := mth;
```

```
    end;
```

```
    7 : begin calendar1.month := mth;
```

```
        label1.caption := 'Jul';
```

```
    end;
```

```
    8 : begin calendar1.month := mth;
```

```
        label1.caption := 'Aug';
```

```
    end;
```

```
    9 : begin label1.caption := 'Sep';
```

```
        if label2.caption='31'then
```

```
            begin calendar1.day := 30;
```

```
            label2.caption:='30';
```

```
        end;
```

```
        calendar1.month := mth;
```

```
    end;
```

```
    10: begin calendar1.month := mth;
```

```
        label1.caption := 'Oct';
```

```
    end;
```

```
    11: begin label1.caption := 'Nov';
```

```
        if label2.caption='31'then
```

```
            begin calendar1.day := 30;
```

```
            label2.caption:='30';
```

```
        end;
```

```
        calendar1.month := mth;
```

```
    end;
```

```
    12: begin calendar1.month := mth;
```

```
        label1.caption := 'Dec';
```

```
    end;
```

```
end;
```

```
end;
```

```
procedure Twthdata.SpeedButton1Click(Sender:
TObject);
```

```
begin
```

```
    close;
```

```
    form1.enabled := true;
```

```
end;
```

```

procedure Twthdata.UpDown2Click(Sender: TObject;
Button: TUDBtnType);
var dat : integer; mth : string;
begin
    mth := label1.caption;
    if (mth='Jan')or(mth='Mar')or(mth='May')or
(mth='Jul')or(mth='Aug')or(mth='Oct')or
(mth='Dec') then begin
        updown2.max := 31;
    end;
    if (mth='Apr')or(mth='Jun')or(mth='Sep')or(mth='Nov')
        then begin updown2.max := 30;
    end;
    if mth='Feb'then updown2.max := 29;

    dat := updown2.position;
    label2.caption := inttostr(dat);
    calendar1.day := dat;
end;

procedure Twthdata.sunat_az(mth,dat,tme:integer; var
at,az:integer);
var month,time : integer;
begin
    if mth=12 then month:=1;
    if (mth=1)or(mth=11) then month:=2;
    if (mth=2)or(mth=10) then month:=3;
    if (mth=3)or(mth=9) then month:=4;
    if (mth=4)or(mth=8) then month:=5;
    if (mth=5)or(mth=7) then month:=6;
    if mth=6 then month:=7;
    if (tme=7)or(tme=17) then time:=7;
    if (tme=8)or(tme=16) then time:=8;
    if (tme=9)or(tme=15) then time:=9;
    if (tme=10)or(tme=14)then time:=10;
    if (tme=11)or(tme=13)then time:=11;

```

```

if tme=12 then time:=12;
case month of
1 : begin case time of
        7 : begin at:=8; az:=63; end;
        8 : begin at:=20; az:=58; end;
        9 : begin at:=32; az:=50; end;
        10: begin at:=43; az:=38; end;
        11: begin at:=50; az:=22; end;
        12: begin at:=53; az:=0; end;
    end;
end;
2 : begin case time of
        7 : begin at:=9; az:=67; end;
        8 : begin at:=22; az:=62; end;
        9 : begin at:=34; az:=53; end;
        10: begin at:=45; az:=42; end;
        11: begin at:=53; az:=24; end;
        12: begin at:=56; az:=0; end;
    end;
end;
3 : begin case time of
        7 : begin at:=12; az:=76; end;
        8 : begin at:=26; az:=71; end;
        9 : begin at:=39; az:=64; end;
        10: begin at:=51; az:=52; end;
        11: begin at:=61; az:=32; end;
        12: begin at:=65; az:=0; end;
    end;
end;
4 : begin case time of
        7 : begin at:=14; az:=87; end;
        8 : begin at:=29; az:=82; end;
        9 : begin at:=43; az:=77; end;
        10: begin at:=57; az:=67; end;
        11: begin at:=69; az:=49; end;
        12: begin at:=76; az:=0; end;
    end;
end;

```

```

end;
5 : begin case time of
    7 : begin at:=17; az:=98; end;
    8 : begin at:=32; az:=95; end;
    9 : begin at:=46; az:=92; end;
    10: begin at:=61; az:=89; end;
    11: begin at:=75; az:=83; end;
    12: begin at:=86; az:=0; end;
end;
end;
6 : begin case time of
    7 : begin at:=19; az:=107; end;
    8 : begin at:=33; az:=105; end;
    9 : begin at:=47; az:=104; end;
    10: begin at:=61; az:=106; end;
    11: begin at:=74; az:=114; end;
    12: begin at:=84; az:=180; end;
end;
end;
7 : begin case time of
    7 : begin at:=19; az:=111; end;
    8 : begin at:=33; az:=109; end;
    9 : begin at:=46; az:=109; end;
    10: begin at:=60; az:=113; end;
    11: begin at:=73; az:=125; end;
    12: begin at:=81; az:=180; end;
end;
end;
end; (*end of case*)
end;

procedure Twthdata.another_az(mth,dat,time:integer; var
at,az:integer);
var ast,et,dcl,lsm,lan,lon,H,A1,A2,a0:real; min:integer;
begin
    lsm:= strtfloat(maskedit1.text);
    lan:= strtfloat(maskedit2.text);

```

```

lon:= strtfloat(maskedit3.text);
case mth of
    1: begin et:= -11.2; dcl:= -20.0; end;
    2: begin et:= -13.9; dcl:= -10.8; end;
    3: begin et:= -7.5; dcl:= 0.0; end;
    4: begin et:= 1.1; dcl:= 11.6; end;
    5: begin et:= 3.3; dcl:= 20.0; end;
    6: begin et:= -1.4; dcl:= 23.45; end;
    7: begin et:= -6.2; dcl:= 20.6; end;
    8: begin et:= -2.4; dcl:= 12.3; end;
    9: begin et:= 7.5; dcl:= 0.0; end;
    10: begin et:= 15.4; dcl:= -10.5; end;
    11: begin et:= 13.8; dcl:= -19.8; end;
    12: begin et:= 1.6; dcl:= -23.45;end;
end;
case time of
    7 : min:= 300;
    8 : min:= 240;
    9 : min:= 180;
    10: min:= 120;
    11: min:= 60;
    12: min:= 0;
    13: min:= 60;
    14: min:= 120;
    15: min:= 180;
    16: min:= 240;
    17: min:= 300;
end;

H := 0.25 * ( min - ( et + 4*(lsm - lon) ) );

A1:= cos(lan*pi/180)*cos(dcl*pi/180)*cos(H*pi/180)
    + sin(lan*pi/180)*sin(dcl*pi/180);
at:= round((arcsin(a1))*(180/pi));
a0:= ((arcsin(a1))*(180/pi));
A2:= ((sin(a0*pi/180)*sin(lan*pi/180))-
    sin(dcl*pi/180))

```

```

        
$$/(\cos(a0*\pi/180)*\cos(lan*\pi/180));$$

    az:= round((arccos(a2))*(180/pi));
end;

procedure Twthdata.SpeedButton2Click(Sender: TObject);
var mth,dat,tme,time,at,az : integer;
begin
    mth := calendar1.month;
    dat := calendar1.day;
    tme := combobox2.itemindex;
    case tme of
    0 : time := 7;
    1 : time := 8;
    2 : time := 9;
    3 : time := 10;
    4 : time := 11;
    5 : time := 12;
    6 : time := 13;
    7 : time := 14;
    8 : time := 15;
    9 : time := 16;
    10: time := 17;
    end;
    if maskedit1.enabled=false then
        sunat_az(mth,dat,time,at,az) else
        another_az(mth,dat,time,at,az);
    label4.caption := inttostr(at);
    label5.caption := inttostr(az);
    windirect.label14.caption:= inttostr(at);
    windirect.label15.caption:= inttostr(az);
end;

procedure Twthdata.SpeedButton3Click(Sender: TObject);
begin
    if Maskedit1.enabled = false then begin
        label8.caption:='Another Locations';

```

```

        Maskedit1.enabled:=true; Maskedit2.enabled:=true;
        Maskedit3.enabled:=true;
        combobox3.enabled:=true;
    combobox4.enabled:=true;
    end else begin
        label8.caption:='Lacation 14 North (Thailand,
        Bangkok)';
        Maskedit1.enabled:=false;
    Maskedit2.enabled:=false;
        Maskedit3.enabled:=false;
        combobox3.enabled:=false;
        combobox4.enabled:=false;
    end;
end;

procedure Twthdata.SpeedButton4Click(Sender:
TObject);
var Fz:integer;
begin
    with windirect do begin
        Fz := trackbar1.position;
        case Fz of
        -18 : image1.picture.LoadFromFile
        ('image\wind_n.bmp');
        -17..-15 : image1.picture.LoadFromFile
        ('image\wind_nnw.bmp');
        -14..-13 : image1.picture.LoadFromFile
        ('image\wind_nw.bmp');
        -12..-10 : image1.picture.LoadFromFile
        ('image\wind_wnw.bmp');
        -9 : image1.picture.LoadFromFile
        ('image\wind_w.bmp');
        -8..-6 : image1.picture.LoadFromFile
        ('image\wind_wws.bmp');
        -5.. 4 : image1.picture.LoadFromFile
        ('image\wind_sw.bmp');
        -3..-1 : image1.picture.LoadFromFile

```

```

('image\wind_sws.bmp');
0 :
image1.picture.LoadFromFile('image\wind_s.bmp');
1..3 : image1.picture.LoadFromFile
('image\wind_ese.bmp');
4..5 :
image1.picture.LoadFromFile('image\wind_se.bmp');
6..8 : image1.picture.LoadFromFile
('image\wind_ees.bmp');
9 :
image1.picture.LoadFromFile('image\wind_e.bmp');
10..12 : image1.picture.LoadFromFile
('image\wind_een.bmp');
13..14 : image1.picture.LoadFromFile
('image\wind_ne.bmp');
15..17 : image1.picture.LoadFromFile
('image\wind_nne.bmp');
18 :
image1.picture.LoadFromFile('image\wind_n.bmp');
end;
end>(* END OF WITH*)
if newdata.radiobutton1.checked = true then begin
windirect.label26.caption := 'Clear sky'end;
if newdata.radiobutton2.checked = true then begin
windirect.label26.caption := 'Partly and Cloudy sky'
end;
if newdata.radiobutton3.checked = true then begin
windirect.label26.caption := 'Overcast sky' end;
windirect.label14.caption:=wthdata.label4.caption;
windirect.label15.caption:=wthdata.label5.caption;
form1.enabled:=false;
windirect.visible:=true;
end;

```

```

procedure Twthdata.ComboBox2Change(Sender:
TObject);
var mth,dat,tme,time,at,az : integer;

```

```

begin
mth := calendar1.month;
dat := calendar1.day;
tme := combobox2.itemindex;
case tme of
0 : time := 7;
1 : time := 8;
2 : time := 9;
3 : time := 10;
4 : time := 11;
5 : time := 12;
6 : time := 13;
7 : time := 14;
8 : time := 15;
9 : time := 16;
10: time := 17;
end;
if maskedit1.enabled=false then
sunat_az(mth,dat,time,at,az) else
another_az(mth,dat,time,at,az);
label4.caption := inttostr(at);
label5.caption := inttostr(az);
windirect.label14.caption:= inttostr(at);
windirect.label15.caption:= inttostr(az);
end;
end.

```

unit daywith;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics,

Controls, Forms, Dialogs, ExtCtrls, StdCtrls, Mask, Buttons;

type

TWithdaylight = class(TForm)

Shape1: TShape; Shape2: TShape;

Shape3: TShape; Shape4: TShape;

SpeedButton1: TSpeedButton;

Label1: TLabel; Label2: TLabel;

MaskEdit1: TMaskEdit;

Shape5: TShape; Shape6: TShape;

Shape7: TShape; Shape8: TShape;

Shape9: TShape; Shape10: TShape;

GroupBox1: TGroupBox;

Shape11: TShape; Shape12: TShape;

Shape13: TShape; Shape14: TShape;

Shape15: TShape; Shape16: TShape;

Shape17: TShape; Shape18: TShape;

Label4: TLabel; Label5: TLabel;

Label6: TLabel; Label7: TLabel;

Label8: TLabel; Label9: TLabel;

Label10: TLabel; Label11: TLabel;

Shape19: TShape; Shape20: TShape;

Shape21: TShape; Shape22: TShape;

Shape23: TShape; Shape24: TShape;

Label12: TLabel; Label13: TLabel;

Label14: TLabel; Label15: TLabel;

Label16: TLabel; Label17: TLabel;

Label18: TLabel; Label19: TLabel;

Label20: TLabel; Label21: TLabel;

Label22: TLabel; Label23: TLabel;

Label24: TLabel;

Label25: TLabel; Label26: TLabel;

Label27: TLabel; Label28: TLabel;

Label29: TLabel; Label30: TLabel;

Label31: TLabel; Label32: TLabel;

Label33: TLabel; Label34: TLabel;

Label35: TLabel; Label36: TLabel;

Label37: TLabel; Label3: TLabel;

Label38: TLabel; Label39: TLabel;

Label40: TLabel; Label41: TLabel;

Label42: TLabel; Label43: TLabel;

Label44: TLabel; Label45: TLabel;

procedure SpeedButton1Click(Sender: TObject);

procedure SpeedButton2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Withdaylight: TWithdaylight;

implementation

{\$R *.DFM}

procedure TWithdaylight.SpeedButton1Click(Sender:
TObject);

begin

close;

end;

procedure TWithdaylight.SpeedButton2Click(Sender:
TObject);

var W1,W2:integer;Ecost:real;

begin

W1:= trunc(strtfloat(label36.caption));

W2:= trunc(strtfloat(label40.caption));

Ecost := strtfloat(Maskedit1.text);

label37.caption:=format('%8.2n',[W1*Ecost]);

label3.caption :=format('%8.2n',[W2*Ecost]);

end;

end.

unit fstlogo;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, ExtCtrls, StdCtrls, MPlayer;

type

TFirstlogo = class(TForm)

Timer1: TTimer;

Timer2: TTimer;

Image1: TImage;

procedure Timer1Timer(Sender: TObject);

procedure Timer2Timer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Firstlogo: TFirstlogo;

implementation

uses Unit1, bcommand;

{\$R *.DFM}

procedure TFirstlogo.Timer1Timer(Sender: TObject);

begin

image1.picture.LoadFromFile('image\stlogo.bmp');

Timer2.enabled:=true;

Timer1.enabled:=false;

Timer1.destroy;

with commandbox do begin

mediaplayer1.filename:='Data\SDtitle.wav';

mediaplayer1.open; mediaplayer1.play;

end;

(*with commandbox do begin

mediaplayer1.filename:='Data\SDtitle.wav';

mediaplayer1.open; mediaplayer1.play;

end;

form1.visible := true;

firstlogo.visible := false;

firstlogo.free;*)

end;

procedure TFirstlogo.Timer2Timer(Sender: TObject);

begin

(*Timer1.free;*)

(*with commandbox do begin

mediaplayer1.filename:='Data\SDtitle.wav';

mediaplayer1.open; mediaplayer1.play;

end;*)

form1.visible := true;

firstlogo.visible := false;

firstlogo.free;

end;

end.

unit grharea;

```

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, ExtCtrls, Buttons, StdCtrls,
  Mask, Printers, Grids;
type
  TGrapharea = class(TForm)
    Shape1: TShape;
    Shape2: TShape;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Memo1: TMemo;
    SpeedButton4: TSpeedButton;
    MaskEdit1: TMaskEdit;
    SpeedButton5: TSpeedButton;
    MaskEdit2: TMaskEdit;
    SpeedButton6: TSpeedButton;
    PrintDialog1: TPrintDialog;
    MaskEdit3: TMaskEdit;
    MaskEdit4: TMaskEdit;
    Edit1: TEdit;
    Edit2: TEdit;
    StringGrid1: TStringGrid;
    SpeedButton7: TSpeedButton;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;

    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure insertdata(row:integer);
    procedure cleardraw;
    procedure illvalue(var maxp:integer);
    procedure
      depthvalue(Gw1,Gh1,orw,orh,ori,sz:integer);
    procedure
      griddraw(Gw1,Gh1,Gw2,Gh2,orw,orh,ori,sz:integer);
    procedure
      graphdraw(Gw1,Gh1,orw,orh,ori,sz,maxp:integer);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton7Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Grapharea: TGrapharea;
implementation
uses Unit1, bcommand, contour, calarea, boxdata;
{$R *.DFM}

procedure TGrapharea.SpeedButton1Click(Sender:
TObject);
begin
  label1.visible:=false; label2.visible:=false;
  label3.visible:=false; label4.visible:=false;
  label5.visible:=false; label6.visible:=false;
  label7.visible:=false; label8.visible:=false;
  label9.visible:=false;
  stringgrid1.visible:=false;

```

```

form1.enabled:=true;
close;
end;

procedure TGrapharea.cleardraw;
begin
    refresh;
end;

procedure TGrapharea.insertdata(row:integer);
var xp,yp,xy,D,C:integer;
begin
    memo1.lines.clear;
    xp := strtoint(commandbox.label7.caption);
    yp := strtoint(commandbox.label8.caption);
    xy := (row-1)*yp;
    D := xy+1; memo1.lines[0] := 'Points';
    for c:=1 to yp do begin
        memo1.lines.add(skycontour.memo1.lines[D]);
        D:=D+1;
    end;
end;

procedure TGrapharea.illvalue(var maxp:integer);
var Ei1,max,max1,D:integer;
begin
    label1.visible:=true; label2.visible:=true;
    label3.visible:=true; label4.visible:=true;
    label5.visible:=true;
    max1:=0; D:=1;

    repeat
        max := strtoint(memo1.lines[D]);
        if max1 > max then max1:=max1 else
max1:=max;
        D:=D+1;
    until memo1.lines[D]="";

    Ei1 := max1;
    case Ei1 of
        0..500 : begin maxp := 500;
            label1.caption:=' 500'; label2.caption:=' 425';
            label3.caption:=' 250'; label4.caption:=' 125';
            end;
        501..1000 : begin maxp := 1000;
            label1.caption:='1000'; label2.caption:=' 750';
            label3.caption:=' 500'; label4.caption:=' 250';
            end;
        1001..1500 : begin maxp := 1500;
            label1.caption:='1500'; label2.caption:='1150';
            label3.caption:=' 750'; label4.caption:=' 350';
            end;
        1501..2000 : begin maxp := 2000;
            label1.caption:='2000'; label2.caption:='1500';
            label3.caption:='1000'; label4.caption:=' 500';
            end;
        2001..3000 : begin maxp := 3000;
            label1.caption:='3000'; label2.caption:='2250';
            label3.caption:='1500'; label4.caption:=' 750';
            end;
        3001..4000 : begin maxp := 4000;
            label1.caption:='4000'; label2.caption:='3000';
            label3.caption:='2000'; label4.caption:='1000';
            end;
        4001..5000 : begin maxp := 5000;
            label1.caption:='5000'; label2.caption:='3750';
            label3.caption:='2500'; label4.caption:='1250';
            end;
        5001..6000 : begin maxp := 6000;
            label1.caption:='6000'; label2.caption:='4500';
            label3.caption:='3000'; label4.caption:='1500';
            end;
        6001..7000 : begin maxp := 7000;
            label1.caption:='7000'; label2.caption:='3520';

```

```

label3.caption:='3500'; label4.caption:='1750';
    end;
7001..8000 : begin maxp := 8000;
    label1.caption:='8000'; label2.caption:='6000';
    label3.caption:='4000'; label4.caption:='2000';
    end;
8001..9000 : begin maxp := 9000;
    label1.caption:='9000'; label2.caption:='6750';
    label3.caption:='4500'; label4.caption:='2250';
    end;
9001..10000 : begin maxp := 10000;
    label1.caption:='10000'; label2.caption:=' 7500';
    label3.caption:=' 5000'; label4.caption:=' 2500';
    end;
10001..12000 : begin maxp := 12000;
    label1.caption:='12000'; label2.caption:=' 9000';
    label3.caption:=' 7000'; label4.caption:=' 3000';
    end;
12001..14000 : begin maxp := 14000;
    label1.caption:='14000'; label2.caption:='10500';
    label3.caption:=' 7000'; label4.caption:=' 3500';
    end;
14001..15000 : begin maxp := 15000;
    label1.caption:='15000'; label2.caption:='11250';
    label3.caption:=' 7500'; label4.caption:=' 3750';
    end;
15001..18000 : begin maxp := 18000;
    label1.caption:='18000'; label2.caption:='13500';
    label3.caption:=' 9000'; label4.caption:=' 4500';
    end;
18001..20000 : begin maxp := 20000;
    label1.caption:='20000'; label2.caption:='15000';
    label3.caption:='10000'; label4.caption:=' 5000';
    end;
20001..40000 : begin maxp := 40000;
    label1.caption:='40000'; label2.caption:='30000';
    label3.caption:='20000'; label4.caption:='10000';
    end;
40001..100000 : begin maxp := 100000;
    label1.caption:='100000';
    label2.caption:='75000';
    label3.caption:='50000';
    label4.caption:='25000';
    end;
100001..200000 : begin maxp := 200000;
    label1.caption:='200000';
    label2.caption:='150000';
    label3.caption:='100000';
    label4.caption:='50000';
    end;
end>(*end of case*)
end;

procedure
TGrapharea.depthvalue(Gw1,Gh1,orw,orh,orl,sz:integer);
var rm_d:real; max_y,C,D,E:integer;
begin
    rm_d := strtofloat(databox.label45.caption);
    max_y := strtoint(commandbox.label8.caption);
    label6.left := trunc(Gw1+orw+((sz*max_y)*2/5)-8);
    label6.Top := trunc(Gh1+orh+(orl)+40);
    label7.left := trunc(Gw1+orw+((sz*max_y)*3/10));
    stringgrid1.defaultcolwidth := sz;
    stringgrid1.colcount := max_y;
    stringgrid1.width:= max_y*sz+20;
    stringgrid1.top := trunc(Gh1+orh+(orl)+10);
    stringgrid1.left:= trunc(Gw1+orw+sz-8);
    D:=0; E:=1;
    for c:=1 to max_y do begin
        stringgrid1.cells[D,0] := format('%1.1n',[rm_d*E/
(max_y+1)]);
        D:=D+1; E:=E+1;
    end;
end;

```

```

procedure TGrapharea.griddraw
(Gw1,Gh1,Gw2,Gh2,orw,orh,orl,sz:integer);
var max_y,D,C:integer;
begin
  max_y:=strtoint(commandbox.label8.caption);
  (*if max_y>20 then max_y:=20;*)
  with canvas do begin {draw grid}
    {horizontal line}
    pen.color := clblue; pen.width := 1;
    moveto(Gw1+orw,Gh1+orh+orl);
    lineto(Gw1+orw+sz*(max_y+1),Gh1+orh+orl);
    moveto(Gw1+orw,Gh1+orh+trunc(orl*3/4));
    lineto(Gw1+orw+sz*(max_y+1),Gh1+orh+trunc(orl*3/4));
    moveto(Gw1+orw,Gh1+orh+trunc(orl*1/2));
    lineto(Gw1+orw+sz*(max_y+1),Gh1+orh+trunc(orl*1/2));
    moveto(Gw1+orw,Gh1+orh+trunc(orl*1/4));
    lineto(Gw1+orw+sz*(max_y+1),Gh1+orh+trunc(orl*1/4));
    moveto(Gw1+orw,Gh1+orh);
    lineto(Gw1+orw+sz*(max_y+1),Gh1+orh);
  } {vertical line}
  D:=0;
  for c:=1 to max_y+2 do begin
    moveto(Gw1+orw+sz*D,Gh1+orh+orl) ;
    lineto(Gw1+orw+sz*D,Gh1+orh);
    D:=D+1;
  end;
  (*label1.top := trunc(Gh1+orh-8);*)
  label2.top := trunc(Gh1+orh+(orl/4)-8);
  label3.top := trunc(Gh1+orh+(orl/2)-8);
  label4.top := trunc(Gh1+orh+(orl*3/4)-8);
  label5.top := trunc(Gh1+orh+(orl)-8);
  end; {end with of canvas draw grid}
end;

```

```

procedure TGrapharea.SpeedButton2Click(Sender:
TObject);

```

```

var Gw1,Gh1,Gw2,Gh2,orw,orh,orl,sz:integer;
  maxp:integer; scale:real;
begin
  scale := strtoint(maskedit4.text);
  gw1:=0; gh1:=0; gw2:=0; gh2:=0;
  sz:=trunc(3.5*scale);
  orw:=70; orh:=80; orl:=trunc(24*scale);
  .
  if memo1.lines[1]<>" then begin
    cleardraw;
    ilvalue(maxp);
    griddraw(Gw1,Gh1,Gw2,Gh2,orw,orh,orl,sz);
    depthvalue(Gw1,Gh1,orw,orh,orl,sz);
    graphdraw(Gw1,Gh1,orw,orh,orl,sz,maxp);
    grapharea.label8.visible := false;
  end else grapharea.label8.visible := true;
end;

procedure TGrapharea.SpeedButton4Click(Sender:
TObject);
var row:integer;
begin
  if memo1.Lines[1]<>" then begin
    row := strtoint(maskedit1.text);
    Grapharea.insertdata(row);
    label8.visible:=false;
  end else label8.visible:=true;
end;

procedure
TGrapharea.graphdraw(Gw1,Gh1,orw,orh,orl,sz,maxp:int
eger);
var Ei : array[1..50] of integer;
  Ell: array[1..50] of integer;
  C,D,max_y:integer;
begin
  max_y := strtoint(commandbox.label8.caption);

```

```

C:=1; D:=1;
for c:=1 to max_y do begin
    Ei[D] := strtoint(memo1.lines[D]);
    D:= D+1;
end;
C:=1; D:=1;
for c:=1 to max_y do begin
    Ell[D] := trunc(Gh1+orh+orl-(Ei[D]*orl/maxp));
    D := D+1;
end;
with canvas do begin (*Draw cruve*)
    pen.color := cllime; pen.width := 2;
    C:=1; D:=1;
    for c:=1 to max_y-1 do begin
        moveto(Gw1+orw+(D*sz),Ell[D]);
        D:=D+1;
        lineto(Gw1+orw+(D*sz),Ell[D]);
    end;(*end of for loop*)
end;
with canvas do begin (*Draw circle*)
    pen.color := cllime; pen.width := 2;
    brush.color := clblack;
    D:=1; C:=1;
    for c:=1 to max_y do begin
        ellipse(Gw1+orw+(D*sz)-3, Ell[D]-3, Gw1+orw+
(D*sz)+5 ,Ell[D]+5);
        D:=D+1;
    end; (*end of for loop*)
end;

procedure TGrapharea.SpeedButton5Click(Sender:
TObject);
var mline,maxp,max_y:integer; scale:real;
    Gw1,Gh1,Gw2,Gh2,orw,orh,orl,sz:integer;
begin
    scale := strtofloat(maskedit4.text);
    gw1:=0; gh1:=0; gw2:=0; gh2:=0; sz:=trunc
(3.5*scale);
    orw:=70; orh:=80; orl:=trunc(24*scale);
    mline := strtoint(maskedit2.text);
    max_y := strtoint(commandbox.label8.caption);
    if memo1.Lines[1]<>" then begin
        illvalue(maxp);
        with canvas do begin
            pen.color := cired; pen.width := 2;
            brush.color := clblack;
            moveto(Gw1+orw,trunc(Gh1+orh+orl-
(mline*orl/maxp)));
            lineto(Gw1+orw+((max_y+1)*sz),trunc
(Gh1+orh+orl-(mline*orl/maxp)));
            ellipse(Gw1+orw-3,trunc(Gh1+orh+orl-
(mline*orl/maxp))-3,
                Gw1+orw+5,trunc(Gh1+orh+orl-
(mline*orl/maxp))+5);
            ellipse(Gw1+orw+((max_y+1)*sz)-3, trunc
(Gh1+orh+orl-(mline*orl/maxp))-3,
                Gw1+orw+((max_y+1)*sz)+5, trunc
(Gh1+orh+orl-(mline*orl/maxp))+5);
            label8.visible:=false;
        end;
    end else label8.visible:=true;
end;

procedure TGrapharea.SpeedButton6Click(Sender:
TObject);
var Gw1,Gh1,Gw2,Gh2,orw,orh,orl,sz:integer;
    mline,maxp,max_y,C,D:integer;
    Grect:Trect;
    Ei : array[1..50] of integer;
    Ell: array[1..50] of integer;
    scale:real;
begin

```

```

mline := strtoint(maskedit2.text);
maxp  := strtoint(label1.caption);
max_y := strtoint(commandbox.label8.caption);
scale := strtfloat(maskedit3.text);
gw1:=100; gh1:=500; gw2:=2750; gh2:=2200;
sz:=trunc(14*scale);
orw:=200; orh:=400; orl:=trunc(90*scale);

if printdialog1.execute then begin
  Grect:=rect(Gw1,Gh1,Gw2,Gh2);
  with printer do begin
    beginDoc;
    with canvas do begin {draw grid}
      pen.color := clblue; pen.width := 6;
      Rectangle(Grect.left,Grect.top,Grect.right,Grect.bottom);
      {horizontal line}
      pen.color := clblue; pen.width := 2;
      moveto(Gw1+orw,Gh1+orh+orl);
      lineto(Gw1+orw+sz*(max_y+1),Gh1+orh+orl);
      moveto(Gw1+orw,Gh1+orh+trunc(orl*3/4));
      lineto(Gw1+orw+sz*(max_y+1),Gh1+orh+trunc(orl*3/4));
      moveto(Gw1+orw,Gh1+orh+trunc(orl*1/2));
      lineto(Gw1+orw+sz*(max_y+1),Gh1+orh+trunc(orl*1/2));
      moveto(Gw1+orw,Gh1+orh+trunc(orl*1/4));
      lineto(Gw1+orw+sz*(max_y+1),Gh1+orh+trunc(orl*1/4));
      moveto(Gw1+orw,Gh1+orh);
      lineto(Gw1+orw+sz*(max_y+1),Gh1+orh);
      {vertical line}
      D:=0;
      for c:=1 to max_y+2 do begin
        moveto(Gw1+orw+sz*D,Gh1+orh+orl);
        lineto(Gw1+orw+sz*D,Gh1+orh);
        D:=D+1;
      end;
    end; {end with of canvas draw grid}
  (*Draw daylight curve*)
  C:=1; D:=1;
  for c:=1 to max_y do begin
    Eil[D] := strtoint(memo1.lines[D]);
    D:= D+1;
  end;
  C:=1; D:=1;
  for c:=1 to max_y do begin
    Eil[D] := trunc(Gh1+orh+orl-(Ei
[D]*orl/maxp));
    D := D+1;
  end;
  with canvas do begin (*Draw cruve*)
    pen.color := clNavy; pen.width := 8;
    C:=1; D:=1;
    for c:=1 to max_y-1 do begin
      moveto(Gw1+orw+(D*sz),Eil[D]);
      D:=D+1;
      lineto(Gw1+orw+(D*sz),Eil[D]);
    end;(*end of for loop*)
  end;
  with canvas do begin (*Draw circle*)
    pen.color := clNavy; pen.width := 8;
    brush.color := clwhite;
    D:=1; C:=1;
    for c:=1 to max_y do begin
      ellipse(Gw1+orw+(D*sz)-14,Eil[D]-
14,Gw1+orw+(D*sz)+14,Eil[D]+14);
      D:=D+1;
    end; (*end of for loop*)
  end;
  (*Draw mark line*)
  if maskedit2.text<>'0' then begin
    with canvas do begin
      pen.color := clblue; pen.width := 8;
      brush.color := clwhite;
      moveto(Gw1+orw,trunc(Gh1+orh+orl-
(mline*orl/maxp)));

```

```

        lineto(Gw1+orw+((max_y+1)*sz),trunc
(Gh1+orh+orl-(mline*orl/maxp)));
        ellipse(Gw1+orw-14,trunc(Gh1+orh+orl-
(mline*orl/maxp))-14, Gw1+orw+14,trunc(Gh1+orh+
orl-(mline*orl/maxp))+14);
        ellipse(Gw1+orw+((max_y+1)*sz)-14,
trunc(Gh1+orh+orl-(mline*orl/maxp))-14, Gw1+orw+
((max_y+1)*sz)+14, trunc(Gh1+orh+orl-
(mline*orl/maxp))+14);
        end;
        end; (*End of if*)
    enddoc;
    end>(*end of with*)

end;
end;

```

```

procedure TGrapharea.SpeedButton3Click(Sender:
TObject);

```

```

var points:integer;
    area,rcr,cop:real;
begin
    if memo1.lines[1]<>" then begin
        Areacal.visible := true; label8.visible:=false; end
    else label8.visible:=true;
        areacal.area_cal;
        with areacal do begin
            areasum(points);
            requiredarea(points,area);
            rrcal(rcr);
            copcal(cop);
        end;
    end;
end;

```

```

procedure TGrapharea.SpeedButton7Click(Sender:
TObject);
begin

```

```

if memo1.lines[1]<>" then begin
    if stringgrid1.visible = false then begin
        stringgrid1.visible:= true;
        label6.visible:=true; label7.visible:=true;
        label9.visible:=true;    end else begin
            stringgrid1.visible:= false;
            label6.visible:=true; label7.visible:=true;
            label9.visible:=true;    end;
            label8.visible:=false;
        end else label8.visible:=true;
    end;

end.

```

unit inputtop;

```

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Buttons,
  Mask;
type
  Ttopinput = class(TForm)
    GroupBox1: TGroupBox;
    Panel1: TPanel;
    SpeedButton1: TSpeedButton;
    Shape1: TShape; Edit1: TEdit;
    Label1: TLabel; Label2: TLabel;
    Edit2: TEdit; Label3: TLabel;
    Edit3: TEdit;
    Label4: TLabel; Label5: TLabel;
    Label6: TLabel;
    RadioGroup1: TRadioGroup;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    GroupBox2: TGroupBox;
    RadioButton4: TRadioButton;
    RadioButton5: TRadioButton;
    RadioButton6: TRadioButton;
    GroupBox3: TGroupBox;
    RadioButton7: TRadioButton;
    RadioButton8: TRadioButton;
    RadioButton9: TRadioButton;
    Shape2: TShape; Shape4: TShape;
    SpeedButton2: TSpeedButton;
    Label7: TLabel; Edit4: TEdit;
    Label8: TLabel;
    Label9: TLabel; Label10: TLabel;
    SpeedButton3: TSpeedButton;
    Image1: TImage; Shape5: TShape;
    GroupBox4: TGroupBox;
    Panel2: TPanel; Shape3: TShape;
    MaskEdit1: TMaskEdit;
    MaskEdit2: TMaskEdit;
    Label11: TLabel; Label12: TLabel;
    SpeedButton4: TSpeedButton;
    Image2: TImage;
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  topinput: Ttopinput;
implementation
uses Unit1, toplumen, Unit5;
{$R *.DFM}

procedure Ttopinput.SpeedButton2Click(Sender:
TObject);
begin
  form1.enabled := true;
  close;
end;

procedure Ttopinput.SpeedButton1Click(Sender:
TObject);
begin
  if speedbutton1.caption = 'Well Depth' then begin
    image1.picture.loadfromfile('image\skylht2.bmp');
    edit3.Enabled := true;
    groupbox3.enabled := true;
  end;
end;

```



```

speedbutton1.caption := 'No Well Depth';
end else begin
image1.picture.loadfromfile('imageskylght1.bmp');
edit3.text := '0';
edit3.Enabled := false;
groupbox3.enabled := false;
speedbutton1.caption := 'Well Depth';
lumentop.label35.caption := '100';
end;
end;

procedure Ttopinput.SpeedButton4Click(Sender: TObject);
begin
    if speedbutton4.Caption = 'Reference enable' then
        begin
            maskedit1.Enabled := true;
            maskedit2.Enabled := true;
            speedbutton4.Caption := 'Reference disable';
        end else begin
            maskedit1.Enabled := false;
            maskedit2.Enabled := false;
            speedbutton4.Caption := 'Reference enable';
        end;
end;

procedure Ttopinput.SpeedButton3Click(Sender: TObject);
var rm_w,rm_d,rm_h,sk_w,sk_d,sk_h,Nsky,sk_x,sk_y:real;
begin
    label10.caption := ' Data has been checked ';
    label12.caption := ' Pass. ';
    rm_w := strtofloat(roomdata.edit1.text);
    rm_d := strtofloat(roomdata.edit2.text);
    rm_h := strtofloat(roomdata.edit3.text);
    sk_w := strtofloat(edit1.text);
    sk_d := strtofloat(edit2.text);
    sk_h := strtofloat(edit3.text);
    Nsky := strtofloat(edit2.text);
    sk_x := strtofloat(maskedit1.text);
    sk_y := strtofloat(maskedit2.text);
    if Maskedit1.enabled = true then begin
        label10.caption := 'Number of skylight must be';
        label12.caption := 'equal 1';
        edit4.text := '1';
    end;
    if sk_w > rm_w then begin
        label10.caption := 'Skylight width than';
        label12.caption := 'room width.';
    end;
    if sk_d > rm_d then begin
        label10.caption := 'Skylight length than';
        label12.caption := 'room length.';
    end;
    if sk_h > rm_h then begin
        label10.caption := 'Skylight height than';
        label12.caption := 'room height.';
    end;
    if (sk_h*sk_d)*Nsky > rm_w*rm_d then begin
        label10.caption := 'Skylight more area than';
        label12.caption := 'room area.';
    end;
    if (sk_x+sk_w)>rm_w then begin
        label10.caption := 'Skylight over room width';
        label12.caption := 'Check x value.';
    end;
    if (sk_y+sk_d)>rm_d then begin
        label10.caption := 'Skylight over room length';
        label12.caption := 'Check y value.';
    end;
    label10.visible := true;
    label12.visible := true;
end;
end.

```

unit intref;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, Buttons, StdCtrls, ComCtrls,
ExtCtrls;

type

TIntref = class(TForm)

GroupBox1: TGroupBox;

Panel1: TPanel;

SpeedButton1: TSpeedButton;

Image1: TImage;

Image2: TImage;

Image3: TImage;

GroupBox2: TGroupBox;

Label4: TLabel;

TrackBar1: TTrackBar;

GroupBox3: TGroupBox;

TrackBar2: TTrackBar;

GroupBox4: TGroupBox;

TrackBar3: TTrackBar;

Label5: TLabel;

Label6: TLabel;

procedure SpeedButton1Click(Sender: TObject);

procedure TrackBar1Change(Sender: TObject);

procedure TrackBar2Change(Sender: TObject);

procedure TrackBar3Change(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Intref: TIntref;

implementation

uses Unit5;

{\$R *.DFM}

procedure TIntref.SpeedButton1Click(Sender: TObject);

begin

Roomdata.enabled := true;

close;

end;

procedure TIntref.TrackBar1Change(Sender: TObject);

var tp1 : integer; tpa : real;

begin

tp1 := trackbar1.Position;

tpa := tp1/100;

label4.caption := format('%1.2n',[tpa]);

case tp1 of

 96..100: Image1.picture.loadfromfile
 ('image\shade100.bmp'); 86..95 : Image1.picture.loadfromfile
 ('image\shade90.bmp'); 76..85 : Image1.picture.loadfromfile
 ('image\shade80.bmp'); 66..75 : Image1.picture.loadfromfile
 ('image\shade70.bmp'); 56..65 : Image1.picture.loadfromfile
 ('image\shade60.bmp'); 46..55 : Image1.picture.loadfromfile
 ('image\shade50.bmp'); 36..45 : Image1.picture.loadfromfile
 ('image\shade40.bmp'); 26..35 : Image1.picture.loadfromfile
 ('image\shade30.bmp'); 16..25 : Image1.picture.loadfromfile
 ('image\shade20.bmp');

```

        6..15 : Image1.picture.loadfromfile
('image\shade10.bmp');
        0..5  : Image1.picture.loadfromfile
('image\shade0.bmp');
    end;
end;

procedure TIntrefT.TrackBar2Change(Sender: TObject);
var tp2 : integer; tpb : real;
begin
    tp2 := trackbar2.Position;
    tpb := tp2/100;
    label5.caption := format('%1.2n',[tpb]);
    case tp2 of
        96..100: Image2.picture.loadfromfile
('image\shade100.bmp');
        86..95  : Image2.picture.loadfromfile
('image\shade90.bmp');
        76..85  : Image2.picture.loadfromfile
('image\shade80.bmp');
        66..75  : Image2.picture.loadfromfile
('image\shade70.bmp');
        56..65  : Image2.picture.loadfromfile
('image\shade60.bmp');
        46..55  : Image2.picture.loadfromfile
('image\shade50.bmp');
        36..45  : Image2.picture.loadfromfile
('image\shade40.bmp');
        26..35  : Image2.picture.loadfromfile
('image\shade30.bmp');
        16..25  : Image2.picture.loadfromfile
('image\shade20.bmp');
        6..15   : Image2.picture.loadfromfile
('image\shade10.bmp');
        0..5    : Image2.picture.loadfromfile
('image\shade0.bmp');
    end;
end;

```

```

end;

procedure TIntrefT.TrackBar3Change(Sender: TObject);
var tp3 : integer; tpc : real;
begin
    tp3 := trackbar3.Position;
    tpc := tp3/100;
    label6.caption := format('%1.2n',[tpc]);
    case tp3 of
        96..100: Image3.picture.loadfromfile
('image\shade100.bmp');
        86..95  : Image3.picture.loadfromfile
('image\shade90.bmp');
        76..85  : Image3.picture.loadfromfile
('image\shade80.bmp');
        66..75  : Image3.picture.loadfromfile
('image\shade70.bmp');
        56..65  : Image3.picture.loadfromfile
('image\shade60.bmp');
        46..55  : Image3.picture.loadfromfile
('image\shade50.bmp');
        36..45  : Image3.picture.loadfromfile
('image\shade40.bmp');
        26..35  : Image3.picture.loadfromfile
('image\shade30.bmp');
        16..25  : Image3.picture.loadfromfile
('image\shade20.bmp');
        6..15   : Image3.picture.loadfromfile
('image\shade10.bmp');
        0..5    : Image3.picture.loadfromfile
('image\shade0.bmp');
    end;
end;

end.

```

unit multiwin;**interface****uses**

Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, ExtCtrls, StdCtrls, Mask,
ComCtrls, Buttons;

type

TMulwindow = class(TForm)

Panel1: TPanel;

Shape1: TShape;

SpeedButton1: TSpeedButton;

PageControl1: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

TabSheet3: TTabSheet;

GroupBox1: TGroupBox;

MaskEdit1: TMaskEdit;

MaskEdit2: TMaskEdit;

Label1: TLabel;

Label2: TLabel;

GroupBox2: TGroupBox;

Panel2: TPanel;

Label3: TLabel;

Label4: TLabel;

MaskEdit3: TMaskEdit;

MaskEdit4: TMaskEdit;

Panel3: TPanel;

Label5: TLabel;

MaskEdit5: TMaskEdit;

Shape2: TShape;

SpeedButton2: TSpeedButton;

Image1: TImage;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

Label24: TLabel;

Label25: TLabel;

Label26: TLabel;

Label27: TLabel;

Label28: TLabel;

Label29: TLabel;

Label30: TLabel;

Label31: TLabel;

Label32: TLabel;

Label33: TLabel;

Label34: TLabel;

Label35: TLabel;

Label36: TLabel;

Label37: TLabel;

Label38: TLabel;

Label39: TLabel;

Label40: TLabel;

Shape3: TShape;

TabSheet4: TTabSheet;

SpeedButton3: TSpeedButton;

SpeedButton4: TSpeedButton;

Shape4: TShape;

Label49: TLabel;
Label50: TLabel;
Label51: TLabel;
Label52: TLabel;
Image2: TImage;
Image3: TImage;
GroupBox3: TGroupBox;
GroupBox4: TGroupBox;
Shape5: TShape;
Panel4: TPanel;
Label54: TLabel;
Label55: TLabel;
MaskEdit6: TMaskEdit;
MaskEdit7: TMaskEdit;
Label56: TLabel;
Label57: TLabel;
Label58: TLabel;
MaskEdit8: TMaskEdit;
MaskEdit9: TMaskEdit;
SpeedButton5: TSpeedButton;
MaskEdit10: TMaskEdit;
Label59: TLabel;
Label60: TLabel;
Label61: TLabel;
GroupBox5: TGroupBox;
GroupBox6: TGroupBox;
Label62: TLabel;
Label63: TLabel;
MaskEdit11: TMaskEdit;
MaskEdit12: TMaskEdit;
Shape6: TShape;
Panel6: TPanel;
Panel7: TPanel;
Label64: TLabel;
Label65: TLabel;
Label66: TLabel;
MaskEdit13: TMaskEdit;
MaskEdit14: TMaskEdit;
MaskEdit15: TMaskEdit;
SpeedButton6: TSpeedButton;
Label67: TLabel;
Label68: TLabel;
Label69: TLabel;
Image4: TImage;
Panel5: TPanel;
Image5: TImage;
Image6: TImage;
Image7: TImage;
GroupBox7: TGroupBox;
GroupBox8: TGroupBox;
Label70: TLabel;
Label71: TLabel;
MaskEdit16: TMaskEdit;
MaskEdit17: TMaskEdit;
Shape7: TShape;
Panel8: TPanel;
Panel9: TPanel;
Image8: TImage;
Image9: TImage;
Label72: TLabel;
Label73: TLabel;
Label74: TLabel;
MaskEdit18: TMaskEdit;
MaskEdit19: TMaskEdit;
MaskEdit20: TMaskEdit;
Label75: TLabel;
Label76: TLabel;
Label77: TLabel;
SpeedButton7: TSpeedButton;
Label41: TLabel;
Label42: TLabel;
TabSheet5: TTabSheet;
GroupBox9: TGroupBox;
GroupBox10: TGroupBox;

```

Shape8: TShape;
Panel10: TPanel;
Panel11: TPanel;
Label43: TLabel;
Label44: TLabel;
Label45: TLabel;
Label46: TLabel;
Label47: TLabel;
MaskEdit21: TMaskEdit;
MaskEdit22: TMaskEdit;
SpeedButton8: TSpeedButton;
MaskEdit23: TMaskEdit;
MaskEdit24: TMaskEdit;
MaskEdit25: TMaskEdit;
Label48: TLabel;
Label78: TLabel;
Label79: TLabel;
Image10: TImage;
Image11: TImage;
Label80: TLabel;
Label81: TLabel;
Label82: TLabel;
Label83: TLabel;
Label84: TLabel;
procedure SpeedButton1Click(Sender: TObject);
procedure TabSheet2DragDrop(Sender, Source:
TObject; X, Y: Integer);
procedure TabSheet3DragDrop(Sender, Source:
TObject; X, Y: Integer);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure SpeedButton7Click(Sender: TObject);
procedure SpeedButton8Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
private
{ Private declarations }

public
{ Public declarations }
end;

var
Mulwindow: TMulwindow;

implementation

uses Unit1, bcommand, boxdata;

{$R *.DFM}

procedure TMulwindow.SpeedButton1Click(Sender:
TObject);
begin
commandbox.speedbutton16.enabled:=true;
commandbox.speedbutton17.enabled:=true;
form1.enabled := true;
close;
end;

procedure TMulwindow.TabSheet2DragDrop(Sender,
Source: TObject; X,
Y: Integer);
begin
image4.picture.loadfromfile('image\mul-a.bmp');
image5.picture.loadfromfile('image\mul-c.bmp');
end;

procedure TMulwindow.TabSheet3DragDrop(Sender,
Source: TObject; X,
Y: Integer);
begin
image6.picture.loadfromfile('image\mul-a.bmp');
image7.picture.loadfromfile('image\mul-c.bmp');
end;

```

```

procedure TMultiwindow.SpeedButton2Click(Sender:
TObject);
begin
    label52.visible:=true;
    label80.visible:=true;
    commandbox.label13.caption := '1';
    commandbox.label13.visible := true;
    commandbox.label12.visible := true;
    commandbox.SpeedButton16.enabled:=true;
    commandbox.SpeedButton17.enabled:=true;
    (*commandbox.SpeedButton15.enabled:=false;
    commandbox.SpeedButton20.enabled:=false;*)
    commandbox.panelcheck(1);
    commandbox.visible := true;
commandbox.SpeedButton12.caption:='Next panels';
    close;
end;

```

```

procedure TMultiwindow.SpeedButton5Click(Sender:
TObject);
begin
    label81.visible:=true;
    commandbox.label13.caption := '2';
    commandbox.label13.visible := true;
    commandbox.label12.visible := true;
    commandbox.SpeedButton16.enabled:=false;
    commandbox.SpeedButton17.enabled:=false;
    (*commandbox.SpeedButton15.enabled:=false;
    commandbox.SpeedButton20.enabled:=false;*)
    commandbox.panelcheck(2);
    commandbox.visible := true;
    commandbox.SpeedButton12.caption:='Next panels';
    close;
end;

```

```

procedure TMultiwindow.SpeedButton6Click(Sender:
TObject);
begin
    label82.visible:=true;
    commandbox.label13.caption := '3';
    commandbox.label13.visible := true;
    commandbox.label12.visible := true;
    commandbox.SpeedButton16.enabled:=false;
    commandbox.SpeedButton17.enabled:=false;
    (*commandbox.SpeedButton15.enabled:=false;
    commandbox.SpeedButton20.enabled:=false;*)
    commandbox.panelcheck(3);
    commandbox.visible := true;
    commandbox.SpeedButton12.caption:='Next panels';
    close;
end;

```

```

procedure TMultiwindow.SpeedButton7Click(Sender:
TObject);
begin
    label83.visible:=true;
    commandbox.label13.caption := '4';
    commandbox.label13.visible := true;
    commandbox.label12.visible := true;
    commandbox.SpeedButton16.enabled:=false;
    commandbox.SpeedButton17.enabled:=false;
    commandbox.panelcheck(4);
    commandbox.visible := true;
    commandbox.SpeedButton12.caption:='Next panels';
    close;
end;

```

```

procedure TMultiwindow.SpeedButton8Click(Sender:
TObject);
begin
    label84.visible:=true;
    commandbox.label13.caption := '5';

```

```
commandbox.label13.visible := true;
commandbox.label12.visible := true;
commandbox.SpeedButton16.enabled:=false;
commandbox.SpeedButton17.enabled:=false;
(*commandbox.SpeedButton15.enabled:=false;
commandbox.SpeedButton20.enabled:=false;*)
commandbox.panelcheck(5);
commandbox.visible := true;
commandbox.SpeedButton12.caption:='Next panels';
close;
```

```
end;
```

```
procedure TMultiwindow.SpeedButton3Click(Sender:
```

```
TObject);
```

```
begin
```

```
    databox.visible:=true;
```

```
end;
```

```
end.
```


unit nproject;

```

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls;

type
  Tprojectname = class(TForm)
    Image1: TImage;
    SpeedButton1: TSpeedButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    procedure SpeedButton1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  projectname: Tprojectname;

implementation

uses Unit1, bcommand;

{$R *.DFM}

```

```

procedure Tprojectname.SpeedButton1Click(Sender:
TObject);
begin
  with commandbox do begin
    mediaPlayer1.filename:='Data\SDnew.wav';
    mediaPlayer1.open; mediaPlayer1.play;
  end;
  form1.enabled := true;
  close;
end;

end.

```

unit progmer;

```
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, ExtCtrls, StdCtrls, Buttons,
  Mask, printers;
```

type

```
Tprogramer = class(TForm)
  Shape1: TShape;
  Memo1: TMemo;
  Shape2: TShape;
  SpeedButton1: TSpeedButton;
  Label1: TLabel;
  Panel1: TPanel;
  Image1: TImage;
  SpeedButton2: TSpeedButton;
  SpeedButton3: TSpeedButton;
  SpeedButton4: TSpeedButton;
  Label2: TLabel;
  SpeedButton5: TSpeedButton;
  Label3: TLabel;
  PrintDialog1: TPrintDialog;
  SpeedButton6: TSpeedButton;
  Label4: TLabel;
  MaskEdit1: TMaskEdit;
  CheckBox1: TCheckBox;
  Label5: TLabel;
  Label6: TLabel;
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton2Click(Sender: TObject);
  procedure SpeedButton5Click(Sender: TObject);
  procedure SpeedButton6Click(Sender: TObject);
  procedure SpeedButton3Click(Sender: TObject);
  procedure SpeedButton4Click(Sender: TObject);
```

```
private
  { Private declarations }
public
  { Public declarations }
end;
var
  programer: Tprogramer;
implementation
uses Unit1;
{$R *.DFM}

procedure Tprogramer.SpeedButton1Click(Sender:
TObject);
begin
  label4.visible :=false; maskedit1.visible:=false;
  checkbox1.visible:=false;
  form1.enabled := true;
  close;
end;
procedure Tprogramer.SpeedButton2Click(Sender:
TObject);
var pic_w,pic_h:integer;
begin
  label2.caption:='Resume';
  memo1.Font.style:=[];
  memo1.font.size:=8;
  memo1.font.color:=cllime;
  memo1.lines.LoadFromFile('dataresume.txt');
  if memo1.lines[30]<>' then label3.visible:=true;
  image1.picture.loadfromfile('image\mol1.bmp');
  pic_h := image1.picture.height;
  pic_w := image1.picture.width;
  if pic_h > 280 then pic_h:=280;
  if pic_w > 280 then pic_w:=280;
  panel1.height := pic_h+4;
  panel1.width := pic_w+4;
```

```

end;

procedure Tprogramer.SpeedButton5Click(Sender:
TObject);
begin
    label2.caption:='Thanks';
    memo1.Font.style:=[fsBold];
    memo1.font.size:=8;
    memo1.font.color:=clyellow;
    memo1.lines.LoadFromFile('data\thanks.txt');
    if memo1.lines[30]<>" then label3.visible:=true;
end;

procedure Tprogramer.SpeedButton6Click(Sender:
TObject);
var poutput:textfile; n:longint; fs:string;
begin
    if label4.visible = false then begin
        label4.visible:=true; maskedit1.visible:=true;
        checkbox1.visible:=true;
    end else begin
        printdialog1.execute;
        assignpm(poutput);
        rewrite(poutput);
        if checkbox1.checked=true then
            printer.canvas.font.style := [fsbold];
            printer.canvas.font.size := strtoint(maskedit1.text);
            writeln(poutput,"");
            writeln(poutput,"");
            writeln(poutput,"");
            for n:=0 to memo1.lines.count - 1 do
                writeln(poutput, memo1.lines[n]);
            system.closefile(poutput);
        end;
    end;
end;

```

```

procedure Tprogramer.SpeedButton3Click(Sender:
TObject);
var pic,pic_w,pic_h:integer; fri:textfile; text:string;
    line1,line2,C,D:integer;
begin
    pic := strtoint(label5.caption);
    label2.caption:='My historical photo';
    memo1.Font.style:=[fsbold];
    memo1.font.size:=9;
    memo1.font.color:=clwhite;
    case pic of
        1 : begin image1.picture.loadfromfile
('image\mol2.bmp');
            label5.caption:='2'; line1:=1;line2:=10;end;
        2 : begin image1.picture.loadfromfile
('image\mol3.bmp');
            label5.caption:='3'; line1:=11;line2:=20;end;
        3 : begin image1.picture.loadfromfile
('image\mol4.bmp');
            label5.caption:='4'; line1:=21;line2:=30;end;
        4 : begin image1.picture.loadfromfile
('image\mol5.bmp');
            label5.caption:='5'; line1:=31;line2:=40;end;
        5 : begin image1.picture.loadfromfile
('image\mol6.bmp');
            label5.caption:='6'; line1:=41;line2:=50;end;
        6 : begin image1.picture.loadfromfile
('image\mol7.bmp');
            label5.caption:='7'; line1:=51;line2:=60;end;
        7 : begin image1.picture.loadfromfile
('image\mol8.bmp');
            label5.caption:='8'; line1:=61;line2:=70;end;
        8 : begin image1.picture.loadfromfile
('image\mol9.bmp');
            label5.caption:='1'; line1:=71;line2:=80;end;
    end;
    memo1.lines.clear; D:=1;

```

```

memo1.lines[0] := "";
memo1.lines[3] := "";
Assignfile(fri,'data\myphoto.dat');
reset(fri);
  for c:=1 to line2 do begin
    if (D>=line1) and (D<=line2) then begin
      readln(fri,text);
      memo1.lines.add(text);
    end else readln(fri,text);
    D:=D+1;
  end;
closefile(fri);
pic_w := image1.picture.width;
pic_h := image1.picture.height;
if pic_h > 280 then pic_h:=280;
if pic_w > 230 then pic_w:=230;
panel1.height := pic_h+4;
panel1.Width := pic_w+4;
label3.visible:=false;
end;

procedure Tprogramer.SpeedButton4Click(Sender:
TObject);
var pic,pic_w,pic_h:integer; fri:textfile; text:string;
    line1,line2,C,D:integer;
begin
  pic := strtoint(label6.caption);
  label2.caption:='Bachelor degree thesis';
  memo1.Font.style:=[fsbold];
  memo1.font.size:=9;
  memo1.font.color:=clwhite;
  case pic of
    1 : begin
image1.picture.loadfromfile('image\mol_t1.bmp');
      label6.caption:='2'; line1:=1;line2:=20; end;
    2 : begin
image1.picture.loadfromfile('image\mol_t2.bmp');
      label6.caption:='3'; line1:=21;line2:=50; end;
    3 : begin
image1.picture.loadfromfile('image\mol_t3.bmp');
      label6.caption:='4'; line1:=51;line2:=70; end;
    4 : begin
image1.picture.loadfromfile('image\mol_t4.bmp');
      label6.caption:='5'; line1:=71;line2:=80;end;
    5 : begin
image1.picture.loadfromfile('image\mol_t5.bmp');
      label6.caption:='6'; line1:=81;line2:=90;end;
    6 : begin
image1.picture.loadfromfile('image\mol_t6.bmp');
      label6.caption:='1'; line1:=91;line2:=100;end;
  end;
  memo1.lines.clear; D:=1;
  memo1.lines[0] := "";
  memo1.lines[3] := "";
  Assignfile(fri,'data\mythesis.dat');
  reset(fri);
    for c:=1 to line2 do begin
      if (D>=line1) and (D<=line2) then begin
        readln(fri,text);
        memo1.lines.add(text);
      end else readln(fri,text);
      D:=D+1;
    end;
  closefile(fri);
  pic_w := image1.picture.width;
  pic_h := image1.picture.height;
  if pic_h > 280 then pic_h:=280;
  if pic_w > 280 then pic_w:=280;
  panel1.height := pic_h+4;
  panel1.width := pic_w+4;
  label3.visible:=false;
end;
end.

```

unit skyfac;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, ExtCtrls, Buttons, StdCtrls,
ComCtrls;

type

Tskyfactor = class(TForm)

StatusBar1: TStatusBar;

GroupBox1: TGroupBox;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

SpeedButton1: TSpeedButton;

SpeedButton2: TSpeedButton;

SpeedButton3: TSpeedButton;

SpeedButton6: TSpeedButton;

Shape1: TShape;

GroupBox2: TGroupBox;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

Label24: TLabel;

Label37: TLabel;

Label38: TLabel;

Label39: TLabel;

Label40: TLabel;

Label41: TLabel;

Label42: TLabel;

Label43: TLabel;

Label44: TLabel;

Label45: TLabel;

Label46: TLabel;

Label47: TLabel;

Label48: TLabel;

Label49: TLabel;

Label50: TLabel;

Label51: TLabel;

Label52: TLabel;

Label53: TLabel;

Label54: TLabel;

Label55: TLabel;

Label56: TLabel;

Timer1: TTimer;

Label36: TLabel;

Shape4: TShape;

Label57: TLabel;

Label58: TLabel;

Label59: TLabel;

Label60: TLabel;

Shape2: TShape;

Label25: TLabel;

Shape5: TShape;

Shape6: TShape;

```

Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label31: TLabel;
Label34: TLabel;
Label64: TLabel;
Label32: TLabel;
Label33: TLabel;
Shape7: TShape;
Label35: TLabel;
Label61: TLabel;
Label62: TLabel;
Label63: TLabel;
Shape3: TShape;
SpeedButton4: TSpeedButton;
Label26: TLabel;
Label65: TLabel;
Label66: TLabel;
Label67: TLabel;
Label68: TLabel;
Label69: TLabel;
Label70: TLabel;
Label71: TLabel;
Label72: TLabel;
Label73: TLabel;
Label74: TLabel;
Label75: TLabel;
Label76: TLabel;
Panel1: TPanel;
Image1: TImage;
procedure SpeedButton6Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
(*procedure intreflect;
procedure sf_cal;
procedure Trans_cal;
procedure atm_cal;*)

procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton4Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  skyfactor: Tskyfactor;
implementation
uses Unit1, Unit2, Unit5, Unit3, Unit4, intref, Unit7,
  contour, bcommand, tdsky;
{$R *.DFM}

procedure Tskyfactor.SpeedButton6Click(Sender:
TObject);
begin
  form1.enabled := true;
  close;
end;

procedure Tskyfactor.Timer1Timer(Sender: TObject);
begin
  with skyfactor do begin
    if newdata.radiobutton1.checked then
      label1.caption := 'Clear sky';
    if newdata.radiobutton2.checked then
      label1.caption := 'Partly cloudy sky';
    if newdata.radiobutton3.checked then
      label1.caption := 'Overcast sky';
    label32.caption := newdata.edit2.text;
    label37.Caption := newdata.edit1.text;
    label38.caption := roomdata.edit1.text;
    label39.caption := roomdata.edit2.text;
  end;
end;

```

```

label40.caption := roomdata.edit3.text;
label41.caption := roomdata.edit4.text;
label42.caption := roomdata.edit5.text;
label43.caption := lightrans.label4.caption;
label44.caption := lightrans.edit1.text;
label45.caption := lightrans.edit2.text;
label46.caption := atmos.label4.caption;
label47.caption := atmos.label8.caption;
label48.caption := atmos.label7.caption;
label49.caption := intrefl.label4.caption;
label50.caption := intrefl.label5.caption;
label51.caption := intrefl.label6.caption;
label52.caption := roomdata.edit8.text;
label53.caption := roomdata.edit9.text;
label54.caption := roomdata.edit10.text;
label55.caption := roomdata.edit6.text;
label56.caption := roomdata.edit7.text;
end;

end;

procedure Tskyfactor.SpeedButton1Click(Sender:
TObject);
var rfx,rfy,rfz,sfv,tr,irc,atm,Ei : real;
begin
    rfx := strtofloat(skyfactor.label52.caption);
    rfy := strtofloat(skyfactor.label53.caption);
    rfz := strtofloat(skyfactor.label54.caption);

    commandbox.intreflect(irc);
    commandbox.trans_cal(tr);
    commandbox.sf_cal(rfx,rfy,rfz,sfv);
    commandbox.atm_cal(atm);

    Ei := ((sfv+irc)/100)*(Tr*atm);

    label29.caption := format('%5.2n',[irc]);
    label34.caption := format('%5.2n',[tr] );

    label30.caption := format('%5.2n',[sfv]);
    label64.caption := format('%5.2n',[ei] );
    StatusBar1.Panels[1].text := 'Analysis Complete';
end;

procedure Tskyfactor.SpeedButton4Click(Sender:
TObject);
begin
    if label70.visible = false then begin
        label26.visible := true; label65.visible := true;
        label66.visible := true; label67.visible := true;
        label68.visible := true; label69.visible := true;
        label70.visible := true; label71.visible := true;
        label72.visible := true; label73.visible := true;
        label74.visible := true; label75.visible := true;
    end else begin
        label26.visible := false; label65.visible := false;
        label66.visible := false; label67.visible := false;
        label68.visible := false; label69.visible := false;
        label70.visible := false; label71.visible := false;
        label72.visible := false; label73.visible := false;
        label74.visible := false; label75.visible := false;
    end;
end;

end;

procedure Tskyfactor.SpeedButton2Click(Sender:
TObject);
var rfx,rfy,rfz,rm_d,Ei,irc,tr,atm,sfv,Ell : real;
    ill : array[1..10] of real; C,D : integer;
begin
    rm_d := strtofloat(label39.caption);
    rfx := strtofloat(skyfactor.label52.caption);
    rfz := strtofloat(skyfactor.label54.caption);
    C := 1; D:= 1;

    commandbox.intreflect(irc);
    commandbox.trans_cal(tr);

```

```

commandbox.atm_cal(atm);

for c := 1 to 9 do begin
  rfy := m_d*D/10;
  commandbox.sf_cal(rfx,rfy,rfz,sfv);
  commandbox.sumoutput(sfv,irc,tr,atm,Ell);
  ill[D] := Ell;
  D := D + 1
end;

with lumengraph do begin
(*label26.caption := inttostr(trunc(ill[1]+ill[1]));*)
label27.caption := inttostr(trunc(ill[1]));
label28.caption := inttostr(trunc(ill[2]));
label29.caption := inttostr(trunc(ill[3]));
label30.caption := inttostr(trunc(ill[4]));
label31.caption := inttostr(trunc(ill[5]));
label32.caption := inttostr(trunc(ill[6]));
label33.caption := inttostr(trunc(ill[7]));
label34.caption := inttostr(trunc(ill[8]));
label35.caption := inttostr(trunc(ill[9]));
label36.caption := inttostr(trunc(ill[9]));

visible := true;
image1.picture.loadfromfile('image\ugrapg.bmp');
caption := 'Skyfactor Side Lighting Analysis';
end;

close;
end;

procedure Tskyfactor.SpeedButton3Click(Sender:
TObject);
begin
  commandbox.visible := true;
  commandbox.speedbutton12.caption:='Return
Command';
skycontour.visible := true;
commandbox.label13.caption := '1';
close;
end;
end.

```


unit skyhelp;

```

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, Buttons, StdCtrls, Mask,
  printers;

type
  Thelpbox = class(TForm)
    Memo1: TMemo;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    ComboBox1: TComboBox;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    Label2: TLabel;
    MaskEdit1: TMaskEdit;
    CheckBox1: TCheckBox;
    PrintDialog1: TPrintDialog;
    Label3: TLabel;
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  helpbox: Thelpbox;

implementation
uses Unit1;
{$R *.DFM}

procedure Thelpbox.SpeedButton2Click(Sender: TObject);
begin

```

```

  label2.visible :=false; maskedit1.visible:=false;
  checkbox1.visible:=false;
  form1.enabled:=true;
  memo1.lines.clear;
  close;
end;

procedure Thelpbox.SpeedButton3Click(Sender:
TObject);
var poutput:textfile; n:longint; fs:string;
begin
  if label2.visible = false then begin
    label2.visible:=true; maskedit1.visible:=true;
    checkbox1.visible:=true;
  end else begin
    printdialog1.execute;
    assignpm(poutput);
    rewrite(poutput);
    if checkbox1.checked=true then
      printer.canvas.font.style := [fsbold];
    printer.canvas.font.size := strtoint(maskedit1.text);
    writeln(poutput,"");
    writeln(poutput,"");
    writeln(poutput,"");
    for n:=0 to memo1.lines.count - 1 do
      writeln(poutput, memo1.lines[n]);
    system.closefile(poutput);
  end;
end;

procedure Thelpbox.ComboBox1Change(Sender:
TObject);
var page:integer;
begin
  Page := Combobox1.itemindex;
  case Page of
    0 : begin

```

```
Label3.caption:='Daylighting 1.0 Design Concept';
```

```
memo1.font.size:=10;
```

```
memo1.font.color:=clNavy;
```

```
memo1.lines.LoadFromFile('data\concept.txt');
```

```
end;
```

```
1 : begin
```

```
Label3.caption:='Daylighting 1.0 Manual in Thai';
```

```
memo1.font.size:=10;
```

```
memo1.font.color:=clNavy;
```

```
memo1.lines.LoadFromFile('data\TManual.txt');
```

```
end;
```

```
end;
```

```
end;
```

```
end.
```

unit tabledat;

```

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Buttons, ExtCtrls,
  printers, Mask;

type
  Tdatatable = class(TForm)
    Shape1: TShape;
    SpeedButton1: TSpeedButton;
    Memo1: TMemo;
    SpeedButton2: TSpeedButton;
    Label1: TLabel; Label2: TLabel;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    SpeedButton7: TSpeedButton;
    SpeedButton8: TSpeedButton;
    PrintDialog1: TPrintDialog;
    Label3: TLabel; Label4: TLabel;
    MaskEdit1: TMaskEdit;
    CheckBox1: TCheckBox;
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
    procedure SpeedButton7Click(Sender: TObject);
    procedure SpeedButton8Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }

    end;

var
  datatable: Tdatatable;

implementation
uses Unit1;
{$R *.DFM}
procedure Tdatatable.SpeedButton2Click(Sender:
TObject);
begin
  label1.caption:='Blinds Table';
  label1.visible:=true;
  memo1.Font.style:=[fsbold];
  memo1.font.size:=15;
  memo1.font.color:=clWhite;
  memo1.lines.LoadFromFile('data\blinds.dat');
  if memo1.lines[30]<>" then label2.visible:=true
  else label2.visible:=false;
end;

procedure Tdatatable.SpeedButton3Click(Sender:
TObject);
begin
  label1.caption:='C.I.E. Standard Overcast Sky Table';
  label1.visible:=true;
  memo1.Font.style:=[fsbold];
  memo1.font.size:=10;
  memo1.font.color:=clWhite;
  memo1.lines.LoadFromFile('data\ciesky1.dat');
  if memo1.lines[30]<>" then label2.visible:=true
  else label2.visible:=false;
end;

procedure Tdatatable.SpeedButton4Click(Sender:
TObject);
begin
  label1.caption:=
    'Extraterrestrial Solar Radiation Intensity';

```

```

label1.visible:=true;
memo1.Font.style:=[fsbold];
memo1.font.size:=13;
memo1.font.color:=clwhite;
memo1.lines.LoadFromFile('data\solarET.dat');
if memo1.lines[30]<>" then label2.visible:=true
else label2.visible:=false;
end;

procedure Tdatatable.SpeedButton1Click(Sender:
TObject);
begin
    label3.visible:=false; label4.visible:=false;
    maskedit1.visible:=false; checkbox1.visible:=false;
    memo1.Lines.clear;
    form1.enabled:=true; label2.visible:=false;
    close;
end;

procedure Tdatatable.SpeedButton5Click(Sender:
TObject);
begin
    Memo1.CopyToClipboard;
end;

procedure Tdatatable.SpeedButton6Click(Sender:
TObject);
begin
    label1.caption:="Table of Recommended illuminances";
    label1.visible:=true;
    memo1.Font.style:=[fsbold];
    memo1.font.size:=12; memo1.font.color:=clwhite;
    memo1.lines.LoadFromFile('data\workedp.dat');
    if memo1.lines[15]<>" then label2.visible:=true
    else label2.visible:=false;
end;

```

```

procedure Tdatatable.SpeedButton7Click(Sender:
TObject);
begin
    label1.caption:='CIE Visual tasks recommended
illuminances';
    label1.visible:=true;
    memo1.Font.style:=[fsbold];
    memo1.font.size:=14; memo1.font.color:=cllime;
    memo1.lines.LoadFromFile('data\taskarea.dat');
    if memo1.lines[15]<>" then label2.visible:=true
    else label2.visible:=false;
end;

procedure Tdatatable.SpeedButton8Click(Sender:
TObject);
var poutput:textfile; n:longint; fs:string;
begin
    if label3.visible = false then begin
        label3.visible:=true; maskedit1.visible:=true;
        label4.visible:=true; checkbox1.visible:=true;
    end else begin
        printdialog1.execute;
        assignpm(poutput);
        rewrite(poutput);
        if checkbox1.checked=true then
            printer.canvas.font.style := [fsbold];
            printer.canvas.font.size := strtoint(maskedit1.text);
            writeln(poutput,""); writeln(poutput,"");
            writeln(poutput,"");
            for n:=0 to memo1.lines.count - 1 do
                writeln(poutput, memo1.lines[n]);
            system.closefile(poutput);
        end;
    end;
end.

```

unit tdsky;

```

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, Buttons, ExtCtrls, StdCtrls,
  Mask;
type
  Tsky3d = class(TForm)
    Shape1: TShape;
    Memo1: TMemo;
    Label2: TLabel;
    Label3: TLabel;
    Shape3: TShape;
    MaskEdit1: TMaskEdit;
    MaskEdit2: TMaskEdit;
    MaskEdit3: TMaskEdit;
    Edit1: TEdit;
    Shape4: TShape;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    MaskEdit4: TMaskEdit;
    Edit5: TEdit;
    Panel1: TPanel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    Memo2: TMemo;
    Memo3: TMemo;
    Memo4: TMemo;
    Memo5: TMemo;
    TotalMemo: TMemo;
    SpeedButton4: TSpeedButton;
    procedure per(l:char,x,y,z:double);
    procedure clearwindow;
    procedure ddraw;
    procedure flrbrush;
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  sky3d: Tsky3d;
implementation
uses Unit1, bcommand;
{$R *.DFM}

procedure Tsky3d.flrbrush;
var x_max,y_max,zh:integer;
begin
  x_max := strtoint(commandbox.label7.caption);
  y_max := strtoint(commandbox.label8.caption);
  zh := trunc(strtoint(maskedit4.text));
  with sky3d.Canvas do begin
    pen.color:=clred;
    per('m',x_max,0,0); per('l',0,0,0);per('l',0,y_max,0);
    per('m',x_max-0.4,0,0.4);per('l',x_max,0,0);per
('l',x_max-0.4,0,-0.4);
    per('m',0,y_max-0.4,0.4);per('l',0,y_max,0);per
('l',0,y_max-0.4,-0.4);
    per('m',0,0,0);per('l',0,0,300/zh);
    per('m',0.4,-0.4,trunc(300/zh)-0.4);per('l',0,0,300/zh);
    per('l',-0.4,0.4,trunc(300/zh)-0.4);
  end;
end;

procedure Tsky3d.SpeedButton1Click(Sender: TObject);

```

```

begin
  close;
end;

procedure Tsky3d.clearwindow;
begin
  (*with sky3d.Canvas do begin
  pen.color:=clblack; brush.color:=clblack;
  Rectangle(81,0,642,433);
  end;*)
  refresh;
end;

procedure Tsky3d.ddraw;
var
  x_max,y_max,mg,D,D1,D2,l,px,py,pz,py1.px1,pz1:integer;
  x:boolean;
  panel:integer;
begin
  panel := strtoint(commandbox.label13.caption);
  x_max := strtoint(commandbox.label7.caption);
  y_max := strtoint(commandbox.label8.caption);
  mg := strtoint(maskedit4.text)*10;
  D := 1; L := 1; D1:=1;
  px:=0; py:=0; x:=true;

  if totalmemo.visible=false then begin
    for l:=1 to x_max do begin
      repeat
        case panel of
          1: pz := strtoint(memo1.lines[D]);
          2: pz := strtoint(memo2.lines[D]);
          3: pz := strtoint(memo3.lines[D]);
          4: pz := strtoint(memo4.lines[D]);
          5: pz := strtoint(memo5.lines[D]);
        end;
        if x = true then begin
          per('m',px,py,trunc(pz/mg));
          x := false; end
        else per('l',px,py,trunc(pz/mg));
          D := D+1; py := py+1;
          until py = y_max;
          px:= px+1; x:=true;
          py:= 0;
        end;
        x := true; px1:=0; py1:=0; D1:=1; D2:=0;
        for l:=1 to y_max do begin
          for l:=1 to x_max do begin
            case panel of
              1: pz1 := strtoint(memo1.lines[D1]);
              2: pz1 := strtoint(memo2.lines[D1]);
              3: pz1 := strtoint(memo3.lines[D1]);
              4: pz1 := strtoint(memo4.lines[D1]);
              5: pz1 := strtoint(memo5.lines[D1]);
            end;
            if x = true then begin
              per('m',px1,py1,trunc(pz1/mg));
              x := false; end
            else per('l',px1,py1,trunc(pz1/mg));
              D1 := D1 + y_max;
              px1 := px1 + 1;
            end;
            py1 := py1+1;x:=true;px1:=0;
            D2:=D2+1; D1:=D2+1;
          end;
        end>(*end of if 'totalmemo.visible=false*');

        if totalmemo.visible=true then begin
          for l:=1 to x_max do begin
            repeat
              pz := strtoint(totalmemo.lines[D]);
              if x = true then begin
                per('m',px,py,trunc(pz/mg));
                x := false; end

```

```

        else per('l',px,py,trunc(pz/mg));
    D := D+1;
    py := py+1;
    until py = y_max;
    px:= px+1; x:=true;
    py:= 0;
end;
x := true; px1:=0; py1:=0; D1:=1; D2:=0;
for l:=1 to y_max do begin
    for l:=1 to x_max do begin
        pz1 := strtoint(totalmemo.lines[D1]);
        if x = true then begin
            per('m',px1,py1,trunc(pz1/mg));
            x := false; end
        else per('l',px1,py1,trunc(pz1/mg));
            D1 := D1 + y_max;
            px1 := px1 + 1;
        end;
        py1 := py1+1;x:=true;px1:=0;
        D2:=D2+1; D1:=D2+1;
    end;
end;(*end of if 'totalmemo.visible=true'*)
end;

procedure Tsky3d.per(l:char;x,y,z:double);
var xe,ye,ze,s1,s2,c1,c2:double; sx,sy,d,a1,a2:integer;
t,p:real;
begin
    d :=strtoint(sky3d.maskedit1.text)*100;
    a1:=strtoint(sky3d.maskedit2.text);
    a2:=strtoint(sky3d.maskedit3.text);
    t :=3.14159/180*(a1);
    p :=3.14159/180*(a2);
    s1:=sin(t); c1:=cos(t); s2:=sin(p); c2:=cos(p);
    xe:= -(x*s1+y*c1);
    ye:= -x*c1*c2 - y*s1*c2 + z*s2;
    ze:= -(x*s2*c1 - y*s1*s2 - z*c2 + 100);

```

```

    sx:= trunc((sky3d.width*3/7) +d*xe/ze);
    sy:= trunc((sky3d.height*4/7)+d*ye/ze);
    with sky3d.canvas do begin
        Pen.width:=1;
        Pen.color:=cllime;
        end;
    if upcase(l)='L' then canvas.lineto(sx,sy);
    if upcase(l)='M' then canvas.moveto(sx,sy);
end;

procedure Tsky3d.SpeedButton2Click(Sender: TObject);
begin
    clearwindow;
    if speedbutton3.caption='Boundary' then
        sky3d.flrbrush;
    ddraw;
end;

procedure Tsky3d.SpeedButton3Click(Sender: TObject);
begin
    if speedbutton3.caption ='Boundary' then
        speedbutton3.caption:='Without' else
        speedbutton3.caption:='Boundary';
end;

procedure Tsky3d.SpeedButton4Click(Sender: TObject);
begin
    maskedit3.text:='65';
    maskedit2.text:='55';
    clearwindow;
    if speedbutton3.caption='Boundary' then
        sky3d.flrbrush;
    ddraw;
end;
end.

```

unit toplumen;**interface****uses**

Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, ExtCtrls, StdCtrls, Buttons,
ComCtrls;

type

Tlumentop = class(TForm)

 GroupBox1: TGroupBox;

 Label1: TLabel;

 Label2: TLabel;

 Label3: TLabel;

 Label4: TLabel;

 Label5: TLabel;

 Label6: TLabel;

 Label7: TLabel;

 Label8: TLabel;

 Label9: TLabel;

 Label10: TLabel;

 Label11: TLabel;

 SpeedButton2: TSpeedButton;

 SpeedButton3: TSpeedButton;

 Shape2: TShape;

 Shape3: TShape;

 SpeedButton1: TSpeedButton;

 Label12: TLabel;

 Label17: TLabel;

 Label18: TLabel;

 Label19: TLabel;

 Label20: TLabel;

 Label21: TLabel;

 Label22: TLabel;

 Label23: TLabel;

 Label24: TLabel;

 Label25: TLabel;

 Label26: TLabel;

 Label27: TLabel;

 Label28: TLabel;

 Label29: TLabel;

 Timer1: TTimer;

 Label31: TLabel;

 Label32: TLabel;

 GroupBox2: TGroupBox;

 Label13: TLabel;

 Label14: TLabel;

 Label15: TLabel;

 Label30: TLabel;

 Label33: TLabel;

 Label34: TLabel;

 Label35: TLabel;

 Label36: TLabel;

 Panel1: TPanel;

 Label37: TLabel;

 Label38: TLabel;

 Label39: TLabel;

 TrackBar1: TTrackBar;

 Label40: TLabel;

 Label41: TLabel;

 Label42: TLabel;

 Label43: TLabel;

 Label44: TLabel;

 Label45: TLabel;

 Label46: TLabel;

 Shape1: TShape;

 Image1: TImage;

 Label47: TLabel;

 Label48: TLabel;

 Label49: TLabel;

 Label50: TLabel;

 Label51: TLabel;

 Shape4: TShape;

 Label52: TLabel;


```

Label53: TLabel;
Label54: TLabel;
Panel2: TPanel;
Label16: TLabel;
Label56: TLabel;
Label57: TLabel;
Label55: TLabel;
Label58: TLabel;
Label59: TLabel;
Label60: TLabel;
Label61: TLabel;
Label62: TLabel;
Label63: TLabel;
Image2: TImage;
Label64: TLabel;
Label65: TLabel;
Label66: TLabel;
Label67: TLabel;
Lb56: TLabel;
lb57: TLabel;
lb65: TLabel;
lb58: TLabel;
lb67: TLabel;
lb59: TLabel;
lb66: TLabel;
lb60: TLabel;
lb61: TLabel;
lb64: TLabel;
lb62: TLabel;
lb63: TLabel;
procedure SpeedButton1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);

(*procedure topcal;*)

private
  { Private declarations }
public
  { Public declarations }
end;
var
  lumentop: Tlumentop;
implementation
uses Unit1, Unit2, Unit5, inputtop, Unit3, Unit4;
{$R *.DFM}

procedure Tlumentop.Timer1Timer(Sender: TObject);
begin
  with lumentop do begin
    if newdata.radiobutton1.checked then label1.caption
:= 'Clear sky';
    if newdata.radiobutton2.checked then label1.caption
:= 'Partly cloudy sky';
    if newdata.radiobutton3.checked then label1.caption
:= 'Overcast sky';
    label17.caption := newdata.edit3.text;
    label18.Caption := newdata.edit1.text;
    label19.caption := roomdata.edit1.text;
    label20.caption := roomdata.edit2.text;
    label21.caption := roomdata.edit3.text;
    label22.caption := topinput.edit1.text;
    label23.caption := topinput.edit2.text;
    label24.caption := topinput.edit3.text;
    label25.caption := lighttrans.edit1.text;
    label26.caption := lighttrans.edit2.text;
    label27.caption := lighttrans.label4.caption;
    label28.caption := atmos.label8.caption;
    label29.caption := atmos.label7.caption;
    label30.caption := topinput.edit4.text;
  end;
  with lumentop do begin
    if topinput.radiobutton1.checked then

```

```

        label33.caption := '80';
if topinput.radiobutton2.checked then
    label33.caption := '50';
if topinput.radiobutton3.checked then
    label33.caption := '30';
    end;
with lumentop do begin
    if topinput.radiobutton4.checked then
        label34.caption := '50';
    if topinput.radiobutton5.checked then
        label34.caption := '30';
    if topinput.radiobutton6.checked then
        label34.caption := '20';
    end;
with lumentop do begin
    if topinput.speedbutton1.caption='Well Depth'
then label35.caption:='100'
    else begin
    if topinput.radiobutton7.checked then
        label35.caption := '80';
    if topinput.radiobutton8.checked then
        label35.caption := '60';
    if topinput.radiobutton9.checked then
        label35.caption := '40';
    end;
    end;
end;

procedure Tlumentop.TrackBar1Change(Sender: TObject);
begin
    label47.caption := inttostr(Trackbar1.position);
end;

procedure rcrtable (ceil_r,wall_r:integer; rcr:real; var cu
:real);
begin
    case ceil_r of

```

```

80 : begin case wall_r of
50 : begin
    if (rcr<1) then begin
        cu := 1.19-((rcr-0)*(1.19-1.05));
        end;
    if (rcr>=1) and (rcr<2) then begin
        cu := 1.05-((rcr-1)*(1.05-0.93));
        end;
    if (rcr>=2) and (rcr<3) then begin
        cu := 0.93-((rcr-2)*(0.93-0.83));
        end;
    if (rcr>=3) and (rcr<4) then begin
        cu := 0.83-((rcr-3)*(0.83-0.75));
        end;
    if (rcr>=4) and (rcr<5) then begin
        cu := 0.75-((rcr-4)*(0.75-0.67));
        end;
    if (rcr>=5) and (rcr<6) then begin
        cu := 0.67-((rcr-5)*(0.67-0.62));
        end;
    if (rcr>=6) and (rcr<7) then begin
        cu := 0.62-((rcr-6)*(0.62-0.57));
        end;
    if (rcr>=7) and (rcr<8) then begin
        cu := 0.57-((rcr-7)*(0.57-0.54));
        end;
    if (rcr>=8) and (rcr<9) then begin
        cu := 0.54-((rcr-8)*(0.54-0.53));
        end;
    if (rcr>=9) and (rcr<10) then begin
        cu := 0.53-((rcr-9)*(0.53-0.52));
        end;
    if rcr>=10 then cu := 0.52;
    end;
30 : begin
    if (rcr<1) then begin
        cu := 1.19-((rcr-0)*(1.19-1.00));

```



```

    cu := 0.87-((rcr-2)*(0.87-0.79));
        end;
if (rcr>=3) and (rcr<4) then begin
    cu := 0.79-((rcr-3)*(0.79-0.71));
        end;
if (rcr>=4) and (rcr<5) then begin
    cu := 0.71-((rcr-4)*(0.71-0.64));
        end;
if (rcr>=5) and (rcr<6) then begin
    cu := 0.64-((rcr-5)*(0.64-0.59));
        end;
if (rcr>=6) and (rcr<7) then begin
    cu := 0.59-((rcr-6)*(0.59-0.55));
        end;
if (rcr>=7) and (rcr<8) then begin
    cu := 0.55-((rcr-7)*(0.55-0.52));
        end;
if (rcr>=8) and (rcr<9) then begin
    cu := 0.52-((rcr-8)*(0.52-0.51));
        end;
if (rcr>=9) and (rcr<10) then begin
    cu := 0.51-((rcr-9)*(0.51-0.50));
        end;
if rcr>=10 then cu := 0.50;
end;
30 : begin
if (rcr<1) then begin
    cu := 1.11-((rcr-0)*(1.11-0.95));
        end;
if (rcr>=1) and (rcr<2) then begin
    cu := 0.95-((rcr-1)*(0.95-0.83));
        end;
if (rcr>=2) and (rcr<3) then begin
    cu := 0.83-((rcr-2)*(0.83-0.73));
        end;
if (rcr>=3) and (rcr<4) then begin
    cu := 0.73-((rcr-3)*(0.73-0.64));
        end;
if (rcr>=4) and (rcr<5) then begin
    cu := 0.64-((rcr-4)*(0.64-0.57));
        end;
if (rcr>=5) and (rcr<6) then begin
    cu := 0.57-((rcr-5)*(0.57-0.52));
        end;
if (rcr>=6) and (rcr<7) then begin
    cu := 0.52-((rcr-6)*(0.52-0.48));
        end;
if (rcr>=7) and (rcr<8) then begin
    cu := 0.48-((rcr-7)*(0.48-0.46));
        end;
if (rcr>=8) and (rcr<9) then begin
    cu := 0.46-((rcr-8)*(0.46-0.45));
        end;
if (rcr>=9) and (rcr<10) then begin
    cu := 0.45-((rcr-9)*(0.45-0.44));
        end;
if rcr>=10 then cu := 0.44;
end;
10 : begin
if (rcr<1) then begin
    cu := 1.11-((rcr-0)*(1.11-0.92));
        end;
if (rcr>=1) and (rcr<2) then begin
    cu := 0.92-((rcr-1)*(0.92-0.78));
        end;
if (rcr>=2) and (rcr<3) then begin
    cu := 0.78-((rcr-2)*(0.78-0.68));
        end;
if (rcr>=3) and (rcr<4) then begin
    cu := 0.68-((rcr-3)*(0.68-0.59));
        end;
if (rcr>=4) and (rcr<5) then begin
    cu := 0.59-((rcr-4)*(0.59-0.52));
        end;

```

```

if (rcr>=5) and (rcr<6) then begin
  cu := 0.52-((rcr-5)*(0.52-0.47));
  end;
if (rcr>=6) and (rcr<7) then begin
  cu := 0.47-((rcr-6)*(0.47-0.43));
  end;
if (rcr>=7) and (rcr<8) then begin
  cu := 0.43-((rcr-7)*(0.43-0.41));
  end;
if (rcr>=8) and (rcr<9) then begin
  cu := 0.41-((rcr-8)*(0.41-0.40));
  end;
if (rcr>=9) and (rcr<10) then begin
  cu := 0.40;
  end;
if rcr>=10 then cu := 0.40;
end;
end; (*end of wall reflectivity*)
end; (*end of 50% ceiling reflectivity*)
20 : begin case wall_r of
  50 : begin
    if (rcr<1) then begin
      cu := 1.04-((rcr-0)*(1.04-0.92));
      end;
    if (rcr>=1) and (rcr<2) then begin
      cu := 0.92-((rcr-1)*(0.92-0.83));
      end;
    if (rcr>=2) and (rcr<3) then begin
      cu := 0.83-((rcr-2)*(0.83-0.75));
      end;
    if (rcr>=3) and (rcr<4) then begin
      cu := 0.75-((rcr-3)*(0.75-0.68));
      end;
    if (rcr>=4) and (rcr<5) then begin
      cu := 0.68-((rcr-4)*(0.68-0.61));
      end;
    if (rcr>=5) and (rcr<6) then begin
      cu := 0.61-((rcr-5)*(0.61-0.57));
      end;
    if (rcr>=6) and (rcr<7) then begin
      cu := 0.57-((rcr-6)*(0.57-0.53));
      end;
    if (rcr>=7) and (rcr<8) then begin
      cu := 0.53-((rcr-7)*(0.53-0.51));
      end;
    if (rcr>=8) and (rcr<9) then begin
      cu := 0.51-((rcr-8)*(0.51-0.50));
      end;
    if (rcr>=9) and (rcr<10) then begin
      cu := 0.50-((rcr-9)*(0.50-0.49));
      end;
    if rcr>=10 then cu := 0.49;
    end;
  30 : begin
    if (rcr<1) then begin
      cu := 1.04-((rcr-0)*(1.04-0.90));
      end;
    if (rcr>=1) and (rcr<2) then begin
      cu := 0.90-((rcr-1)*(0.90-0.79));
      end;
    if (rcr>=2) and (rcr<3) then begin
      cu := 0.79-((rcr-2)*(0.79-0.70));
      end;
    if (rcr>=3) and (rcr<4) then begin
      cu := 0.70-((rcr-3)*(0.70-0.62));
      end;
    if (rcr>=4) and (rcr<5) then begin
      cu := 0.62-((rcr-4)*(0.62-0.56));
      end;
    if (rcr>=5) and (rcr<6) then begin
      cu := 0.56-((rcr-5)*(0.56-0.51));
      end;
    if (rcr>=6) and (rcr<7) then begin
      cu := 0.51-((rcr-6)*(0.51-0.47));

```

```

        end;
if (rcr>=7) and (rcr<8) then begin
    cu := 0.47-((rcr-7)*(0.47-0.45));
    end;
if (rcr>=8) and (rcr<9) then begin
    cu := 0.45-((rcr-8)*(0.45-0.44));
    end;
if (rcr>=9) and (rcr<10) then begin
    cu := 0.44-((rcr-9)*(0.44-0.43));
    end;
if rcr>=10 then cu := 0.43;
end;
10 : begin
if (rcr<1) then begin
    cu := 1.04-((rcr-0)*(1.04-0.88));
    end;
if (rcr>=1) and (rcr<2) then begin
    cu := 0.88-((rcr-1)*(0.88-0.76));
    end;
if (rcr>=2) and (rcr<3) then begin
    cu := 0.76-((rcr-2)*(0.76-0.66));
    end;
if (rcr>=3) and (rcr<4) then begin
    cu := 0.66-((rcr-3)*(0.66-0.58));
    end;
if (rcr>=4) and (rcr<5) then begin
    cu := 0.58-((rcr-4)*(0.58-0.51));
    end;
if (rcr>=5) and (rcr<6) then begin
    cu := 0.51-((rcr-5)*(0.51-0.46));
    end;
if (rcr>=6) and (rcr<7) then begin
    cu := 0.46-((rcr-6)*(0.46-0.43));
    end;
if (rcr>=7) and (rcr<8) then begin
    cu := 0.43-((rcr-7)*(0.43-0.41));
    end;
if (rcr>=8) and (rcr<9) then begin
    cu := 0.41-((rcr-8)*(0.41-0.40));
    end;
if (rcr>=9) and (rcr<10) then begin
    cu := 0.40-((rcr-9)*(0.41-0.39));
    end;
if rcr>=10 then cu := 0.39;
end;
end; (*End of wall reflectivity*)
end; (*End of 20% Ceiling Reflectivity*)
end; (* End of case*)

end;

procedure topcal(mm_h : real; var Ei : real);
var sky_w,sky_l,sky_d,rm_w,rm_d,Edn,Ehk : real;
    mat,vis,net,atm_t,atm_m,num : real;
    ceil_re,wall_re,well_re : integer;
    rcrc,tr,sky_area,wk_area,cu : real;
begin
    Edn := strtfloat(lumentop.label17.caption);
    Ehk := strtfloat(lumentop.label18.caption);
    rm_w := strtfloat(lumentop.label19.caption);
    rm_d := strtfloat(lumentop.label20.caption);
    (*rm_h := strtfloat(lumentop.label21.caption);*)
    sky_w := strtfloat(lumentop.label22.caption);
    sky_l := strtfloat(lumentop.label23.caption);
    sky_d := strtfloat(lumentop.label24.caption);
    mat := strtfloat(lumentop.label25.caption);
    vis := strtfloat(lumentop.label26.caption);
    net := strtfloat(lumentop.label27.caption);
    atm_t := strtfloat(lumentop.label28.caption);
    atm_m := strtfloat(lumentop.label29.caption);
    num := strtfloat(lumentop.label30.caption);
    ceil_re := strtoint(lumentop.label33.caption);
    wall_re := strtoint(lumentop.label34.caption);
    well_re := strtoint(lumentop.label35.caption);

```

```

rcrc := 5*rm_h*(rm_w+rm_d)/(rm_w*rm_d);
lumentop.label48.caption := format('%5.2n',[rcrc]);

tr := atm_t*(mat*vis*net)*(well_re/100);
lumentop.label49.caption := format('%5.2n',[tr]);

sky_area := (sky_l*sky_w*num);
lumentop.label50.caption := format('%5.2n',[sky_area]);

wk_area := (rm_w*rm_d);
lumentop.label51.caption := format('%5.2n',[wk_area]);

rcrtable(ceil_re,wall_re,rcrc,cu);

Ei := (Edn*Tr+Ehk*Tr)*Cu*(sky_area/wk_area);
(*lumentop.label54.caption := format('%5.2n',[Ei]);*)

```

```
end;
```

```

procedure Tlumentop.SpeedButton2Click(Sender:
TObject);

```

```
var rm_h,Ei : real;
```

```
begin
```

```

rm_h := strtofloat(lumentop.label21.caption);
topcal((rm_h-0.75),Ei);
lumentop.label54.caption := format('%5.2n',[Ei]);
label53.visible := true;

```

```
end;
```

```

procedure Tlumentop.SpeedButton1Click(Sender:
TObject);

```

```
begin
```

```

label53.visible := false;
form1.enabled := true;
label56.visible := false; label57.visible := false;
label58.visible := false;

```

```

label59.visible := false; label60.visible := false;
label61.visible := false; label62.visible := false;
label63.visible := false; label64.visible := false;
label65.visible := false; label66.visible := false;
label67.visible := false;
lb56.visible := false; lb57.visible := false;
lb58.visible := false; lb59.visible := false;
lb60.visible := false; lb61.visible := false;
lb62.visible := false; lb63.visible := false;
lb64.visible := false; lb65.visible := false;
lb66.visible := false; lb67.visible := false;
close;

```

```
end;
```

```

procedure Tlumentop.SpeedButton3Click(Sender:
TObject);

```

```
var dive :integer; rm_h,Ei : real;
```

```
begin
```

```

label56.caption := ""; label57.caption := "";
label58.caption := ""; label59.caption := "";
label60.caption := ""; label61.caption := "";
label62.caption := ""; label62.caption := "";
label63.caption := ""; label64.caption := "";
label65.caption := ""; label66.caption := "";
label67.caption := "";
lb56.caption := ""; lb57.caption := "";
lb58.caption := ""; lb59.caption := "";
lb60.caption := ""; lb61.caption := "";
lb62.caption := ""; lb63.caption := "";
lb64.caption := ""; lb65.caption := "";
lb66.caption := ""; lb67.caption := "";

```

```
rm_h := strtofloat(lumentop.label21.caption);
```

```
dive := trackbar1.position;
```

```
case dive of
```

```
1 : begin topcal(rm_h,Ei);
```

```
label63.Caption := format('%5.2n',[Ei]);
```

```

    label63.visible := true;
    lb63.Caption := format('%5.2n',[rm_h]);
    lb63.visible := true;
end;
2 : begin topcal((rm_h/2),Ei);
    label59.Caption := format('%5.2n',[Ei]);
    label59.visible := true;
    lb59.Caption := format('%5.2n',[(rm_h/2)]);
    lb59.visible := true;
    topcal(rm_h,Ei);
    label63.Caption := format('%5.2n',[Ei]);
    label63.visible := true;
    lb63.Caption := format('%5.2n',[rm_h]);
    lb63.visible := true;
end;
3 : begin topcal((rm_h/3),Ei);
    label65.Caption := format('%5.2n',[Ei]);
    label65.visible := true;
    lb65.Caption := format('%5.2n',[(rm_h/3)]);
    lb65.visible := true;
    topcal((rm_h*2/3),Ei);
    label60.Caption := format('%5.2n',[Ei]);
    label60.visible := true;
    lb60.Caption := format('%5.2n',[(rm_h*2/3)]);
    lb60.visible := true;
    topcal(rm_h,Ei);
    label63.Caption := format('%5.2n',[Ei]);
    label63.visible := true;
    lb63.Caption := format('%5.2n',[rm_h]);
    lb63.visible := true;
end;
4 : begin topcal((rm_h*1/4),Ei);
    label65.Caption := format('%5.2n',[Ei]);
    label65.visible := true;
    lb65.Caption := format('%5.2n',[(rm_h/4)]);
    lb65.visible := true;
    topcal((rm_h*2/4),Ei);
    label66.Caption := format('%5.2n',[Ei]);
    label66.visible := true;
    lb66.Caption := format('%5.2n',[(rm_h/2)]);
    lb66.visible := true;
    topcal((rm_h*3/4),Ei);
    label61.Caption := format('%5.2n',[Ei]);
    label61.visible := true;
    lb61.Caption:=format('%5.2n',[(rm_h*3/4)]);
    lb61.visible:=true;
    topcal(rm_h,Ei);
    label63.Caption := format('%5.2n',[Ei]);
    label63.visible := true;
    lb63.Caption := format('%5.2n',[rm_h]);
    lb63.visible := true;
end;
5 : begin
    topcal((rm_h*1/5),Ei);
    label57.Caption := format('%5.2n',[Ei]);
    label57.visible := true;
    lb57.Caption := format('%5.2n',[(rm_h/5)]);
    lb57.visible := true;
    topcal((rm_h*2/5),Ei);
    label67.Caption := format('%5.2n',[Ei]);
    label67.visible := true;
    lb67.Caption := format('%5.2n',[(rm_h*2/5)]);
    lb67.visible := true;
    topcal((rm_h*3/5),Ei);
    label60.Caption := format('%5.2n',[Ei]);
    label60.visible := true;
    lb60.Caption:=format('%5.2n',[(rm_h*3/5)]);
    lb60.visible:=true;
    topcal((rm_h*4/5),Ei);
    label64.Caption := format('%5.2n',[Ei]);
    label64.visible := true;
    lb64.Caption:=format('%5.2n',[(rm_h*4/5)]);
    lb64.visible:=true;
    topcal(rm_h,Ei);

```



```

label63.Caption := format('%5.2n',[Ei]);
label63.visible := true;
lb63.Caption := format('%5.2n',[rm_h]);
lb63.visible := true;
end;
6 : begin
topcal((rm_h*1/6),Ei);
label57.Caption := format('%5.2n',[Ei]);
label57.visible := true;
lb57.Caption := format('%5.2n',[(rm_h/6)]);
lb57.visible := true;
topcal((rm_h*2/6),Ei);
label58.Caption := format('%5.2n',[Ei]);
label58.visible := true;
lb58.Caption := format('%5.2n',[(rm_h*2/6)]);
lb58.visible := true;
topcal((rm_h*3/6),Ei);
label59.Caption := format('%5.2n',[Ei]);
label59.visible := true;
lb59.Caption:=format('%5.2n',[(rm_h*3/6)]);
lb59.visible:=true;
topcal((rm_h*4/6),Ei);
label60.Caption := format('%5.2n',[Ei]);
label60.visible := true;
lb60.Caption:=format('%5.2n',[(rm_h*4/6)]);
lb60.visible:=true;
topcal((rm_h*5/6),Ei);
label61.Caption := format('%5.2n',[Ei]);
label61.visible := true;
lb61.Caption:=format('%5.2n',[(rm_h*5/6)]);
lb61.visible := true;
topcal(rm_h,Ei);
label62.Caption := format('%5.2n',[Ei]);
label62.visible := true;
lb62.Caption := format('%5.2n',[rm_h]);
lb62.visible := true;
end;

```

```

7 : begin
topcal((rm_h*1/7),Ei);
label57.Caption := format('%5.2n',[Ei]);
label57.visible := true;
lb57.Caption := format('%5.2n',[(rm_h/7)]);
lb57.visible := true;
topcal((rm_h*2/7),Ei);
label58.Caption := format('%5.2n',[Ei]);
label58.visible := true;
lb58.Caption := format('%5.2n',[(rm_h*2/7)]);
lb58.visible := true;
topcal((rm_h*3/7),Ei);
label59.Caption := format('%5.2n',[Ei]);
label59.visible := true;
lb59.Caption := format('%5.2n',[rm_h*3/7]);
lb59.visible := true;
topcal((rm_h*4/7),Ei);
label60.Caption := format('%5.2n',[Ei]);
label60.visible := true;
lb60.Caption := format('%5.2n',[rm_h*4/7]);
lb60.visible := true;
topcal((rm_h*5/7),Ei);
label61.Caption := format('%5.2n',[Ei]);
label61.visible := true;
lb61.Caption := format('%5.2n',[rm_h*5/7]);
lb61.visible := true;
topcal((rm_h*6/7),Ei);
label62.Caption := format('%5.2n',[Ei]);
label62.visible := true;
lb62.Caption := format('%5.2n',[rm_h*6/7]);
lb62.visible := true;
topcal(rm_h,Ei);
label63.Caption := format('%5.2n',[Ei]);
label63.visible := true;
lb63.Caption := format('%5.2n',[rm_h]);
lb63.visible := true;
end;

```

```

8 : begin
    topcal((rm_h*1/8),Ei);
    label56.Caption := format('%5.2n',[Ei]);
    label56.visible := true;
    lb56.Caption := format('%5.2n',[rm_h/8]);
    lb56.visible := true;
    topcal((rm_h*2/8),Ei);
    label57.Caption := format('%5.2n',[Ei]);
    label57.visible := true;
    lb57.Caption := format('%5.2n',[rm_h*2/8]);
    lb57.visible := true;
    topcal((rm_h*3/8),Ei);
    label58.Caption := format('%5.2n',[Ei]);
    label58.visible := true;
    lb58.Caption := format('%5.2n',[rm_h*3/8]);
    lb58.visible := true;
    topcal((rm_h*4/8),Ei);
    label59.Caption := format('%5.2n',[Ei]);
    label59.visible := true;
    lb59.Caption := format('%5.2n',[rm_h*4/8]);
    lb59.visible := true;
    topcal((rm_h*5/8),Ei);
    label60.Caption := format('%5.2n',[Ei]);
    label60.visible := true;
    lb60.Caption := format('%5.2n',[rm_h*5/8]);
    lb60.visible := true;
    topcal((rm_h*6/8),Ei);
    label61.Caption := format('%5.2n',[Ei]);
    label61.visible := true;
    lb61.Caption := format('%5.2n',[rm_h*6/8]);
    lb61.visible := true;
    topcal((rm_h*7/8),Ei);
    label62.Caption := format('%5.2n',[Ei]);
    label62.visible := true;
    lb62.Caption := format('%5.2n',[rm_h*7/8]);
    lb62.visible := true;
    topcal(rm_h,Ei);
    label63.Caption := format('%5.2n',[Ei]);
    label63.visible := true;
    lb63.Caption := format('%5.2n',[rm_h]);
    lb63.visible := true;
end;
end;
end;
end.

```

ประวัติผู้เขียน

นายกมล เกียรติเรืองภมลา เกิดเมื่อวันที่ 12 พฤษภาคม 2516 โรงพยาบาลหัวเฉียว กรุงเทพมหานคร เข้าศึกษาในระดับชั้นมัธยมศึกษาตอนต้นและมัธยมศึกษาตอนปลายที่โรงเรียนเทพศิรินทร์ สอบเทียบมัธยมศึกษาตอนปลายเพื่อเข้าศึกษาในระดับอุดมศึกษา ณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง สำเร็จการศึกษาระดับปริญญาตรี สาขาวิศวกรรมศาสตร์ บัณฑิต จากสถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง เมื่อปีการศึกษา 2539 (คะแนนเฉลี่ยสะสม 3.21) และเข้าศึกษาต่อหลักสูตรสถาปัตยกรรมศาสตร์มหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย ในภาควิชาสถาปัตยกรรมศาสตร์ สาขาวิชาเทคโนโลยีทางอาคาร ในปีการศึกษา 2540 เคยได้รับรางวัลนักศึกษาวิชาทหารดีเด่น กรมการรักษาดินแดน ในปี 2534 , รางวัลที่ 3 จากการประกวดออกแบบผลงานนักศึกษาของบริษัท เจริญซีเมนต์ไทย จำกัด ในปี 2535 และผลงานทางวิชาการ การพัฒนาโปรแกรมคอมพิวเตอร์ เพื่อช่วยวิเคราะห์การออกแบบสถาปัตยกรรม สำหรับอาคารสูงในประเทศไทย ได้ลงตีพิมพ์เผยแพร่ในนิตยสาร Arch and Idea ฉบับที่ 14 เดือนพฤศจิกายน พ.ศ. 2537

ประสบการณ์การทำงาน สถาปนิกฝึกงานบริษัท เออร์เบิน อาร์คิเทค จำกัด และสถาปนิกฝ่ายออกแบบบริษัท นันทวัน จำกัด

