static const int NUMROWS = 4;

static const int NUMCOLS = 5;

Graphics^ graphics;

Brush^ redBrush;

Brush^ yellowBrush;

Brush^ greenBrush;

Pen^ blackPen;

private: System::Void MyForm\_Load(System::Object^ sender, System::EventArgs^ e) {

graphics = panel1->CreateGraphics();

redBrush = gcnew SolidBrush(Color::Red);

yellowBrush = gcnew SolidBrush(Color::Yellow);

greenBrush = gcnew SolidBrush(Color::Green);

blackPen = gcnew Pen(Color::Black);

}

private: System::Void btnShow\_Click(System::Object^ sender, System::EventArgs^ e) {

panel1->Refresh();

for (int row = 0; row < NUMROWS; row++)

{

for (int col = 0; col < NUMCOLS; col++)

{

Rectangle seat = Rectangle(75 + col \* 75, 40 + row \* 40, 25, 25);

graphics->DrawRectangle(blackPen, seat);

}

}

}

private: double CalculateMean(double (&scores)[NUMROWS][NUMCOLS])

{

double sum;

int students = NUMROWS \* NUMCOLS;

for (int row = 0; row < NUMROWS; row++)

{

for (int col = 0; col < NUMCOLS; col++)

{

sum += scores[row][col];

}

}

return sum / students;

}

private: System::Void btnGroup\_Click(System::Object^ sender, System::EventArgs^ e) {

double score[NUMROWS][NUMCOLS] =

{

{ 45, 65, 11, 98, 66 },

{ 56, 77, 78, 56, 56 },

{ 87, 71, 78, 90, 78 },

{ 76, 75, 72, 79, 83 }

};

double mean = CalculateMean(score);

txtMean->Text = mean.ToString();

for (int row = 0; row < NUMROWS; row++)

{

for (int col = 0; col < NUMCOLS; col++)

{

Rectangle seat = Rectangle(75 + col \* 75, 40 + (row \* 40), 25, 25);

if (score[row][col] >= 80)

graphics->FillRectangle(greenBrush, seat);

else if (score[row][col] >= mean)

graphics->FillRectangle(yellowBrush, seat);

else

graphics->FillRectangle(redBrush, seat);

graphics->DrawRectangle(blackPen, seat);

}

}

}

private: System::Void btnHighLow\_Click(System::Object^ sender, System::EventArgs^ e) {

double score[NUMROWS][NUMCOLS] =

{

{ 45, 65, 11, 98, 66 },

{ 56, 77, 78, 56, 56 },

{ 87, 71, 78, 90, 78 },

{ 76, 75, 72, 79, 83 }

};

HighLow(score);

}

private: System::Void HighLow(double(&scores)[NUMROWS][NUMCOLS]){

double Highest;

double Lowest = 100;

for (int row = 0; row < NUMROWS; row++)

{

for (int col = 0; col < NUMCOLS; col++)

{

if (scores[row][col] > Highest) Highest = scores[row][col];

else if (scores[row][col] < Lowest) Lowest = scores[row][col];

}

}

lblHighest0->Visible = true;

lblHighest1->Visible = true;

lblHighest1->Text = Highest.ToString();

lblLowest0->Visible = true;

lblLowest1->Visible = true;

lblLowest1->Text = Lowest.ToString();

}