|  |
| --- |
| #include <iostream> |
|  | using namespace std; |
|  |  |
|  | class Point { |
|  | public: |
|  | Point(); |
|  |  |
|  | Point (int xval, int yval); |
|  |  |
|  | void Move (int dx, int dy); |
|  |  |
|  | int Get\_X() const; |
|  | int Get\_Y() const; |
|  |  |
|  | void Set\_X(int xval); |
|  | void Set\_Y(int yval); |
|  |  |
|  | void print() const; |
|  |  |
|  | private: |
|  | int X; |
|  | int Y; |
|  | }; |
|  |  |
|  | Point::Point() { |
|  | X = 0; |
|  | Y = 0; |
|  | } |
|  |  |
|  | Point::Point(int xval, int yval) { |
|  | X = xval; |
|  | Y = yval; |
|  | } |
|  |  |
|  | void Point::Move(int dx, int dy) { |
|  | X += dx; |
|  | Y += dy; |
|  | } |
|  |  |
|  | int Point::Get\_X() const { |
|  | return X; |
|  | } |
|  |  |
|  | int Point::Get\_Y() const { |
|  | return Y; |
|  | } |
|  |  |
|  | void Point::Set\_X(int xval) { |
|  | X = xval; |
|  | } |
|  |  |
|  | void Point::Set\_Y(int yval) { |
|  | Y = yval; |
|  | } |
|  |  |
|  | void Point::print() const { |
|  | cout << "(" << X << ", " << Y << ")" << endl; |
|  | } |
|  |  |
|  | class Rectangle { |
|  | Point bottomleft, bottomright, topleft, topright; |
|  |  |
|  | int side1(); |
|  | int side2(); |
|  |  |
|  | public: |
|  | Rectangle(Point bottomleft, Point topright); |
|  | Rectangle(); |
|  |  |
|  | Point getBottomLeft(); |
|  | Point getBottomRight(); |
|  | Point getTopLeft(); |
|  | Point getTopRight(); |
|  |  |
|  | void print(); |
|  |  |
|  | int area(); |
|  | }; |
|  |  |
|  | Rectangle::Rectangle(Point bottomleft, Point topright){ |
|  | int trX = topright.Get\_X(); |
|  | int blY = bottomleft.Get\_Y(); |
|  | bottomright.Set\_X(trX); |
|  | bottomright.Set\_Y(blY); |
|  |  |
|  | int trY = topright.Get\_Y(); |
|  | int blX = bottomleft.Get\_X(); |
|  | topleft.Set\_X(blX); |
|  | topleft.Set\_Y(trY); |
|  | } |
|  |  |
|  | Rectangle::Rectangle() { |
|  | bottomleft.Set\_X(0); |
|  | bottomleft.Set\_Y(0); |
|  |  |
|  | bottomright.Set\_X(1); |
|  | bottomright.Set\_Y(0); |
|  |  |
|  | topleft.Set\_X(0); |
|  | topleft.Set\_Y(1); |
|  |  |
|  | topright.Set\_X(1); |
|  | topright.Set\_Y(1); |
|  | } |
|  |  |
|  | Point Rectangle::getBottomLeft() { |
|  | return bottomleft; |
|  | } |
|  |  |
|  | Point Rectangle::getBottomRight() { |
|  | return bottomright; |
|  | } |
|  |  |
|  | Point Rectangle::getTopLeft() { |
|  | return topleft; |
|  | } |
|  |  |
|  | Point Rectangle::getTopRight() { |
|  | return topright; |
|  | } |
|  |  |
|  | int Rectangle::side1() { |
|  | return topright.Get\_X() - bottomleft.Get\_X(); |
|  | } |
|  |  |
|  | int Rectangle::side2() { |
|  | return topright.Get\_Y() - bottomleft.Get\_Y(); |
|  | } |
|  |  |
|  | void Rectangle::print() { |
|  | bottomleft.print(); |
|  | bottomright.print(); |
|  | topleft.print(); |
|  | topright.print(); |
|  | } |
|  |  |
|  | int Rectangle::area() { |
|  | return (\*this).side1() \* (\*this).side2(); |
|  | } |
|  |  |
|  | int main() { |
|  |  |
|  | return 0; |
|  | } |