과목: 자료구조

교수: 최재현 교수님

자료구조 <과제 2>

- Rectangle 클래스 작성-

홍지훈

이름: 홍지훈

학과: 소프트웨어학부

분반: 나

학번 : 20201777

0. 과제

[과제2] Rectangle 클래스 작성

1. 시작점의 좌표(x,y)와 길이(width) 와 높이(height)로 표현되는 Rectangle 클래스를 작성.

2. 기본생성자 및 소멸자(파괴자) 작성 ( 추가 생성 및 소멸자를 구현해도 됨 )

3. Rectangle 클래스의 데이터를 입력받는 read 함수 작성

4. Rectangle 클래스의 데이터를 출력하는 print 함수 작성

5. +, /, ==, << 4개의 연산자 오버로딩을 구현

  5.1 + 기호는 합집합 --> 2개의 사각형을 내부에 포함하는 가장작은 사각형의 시작좌표 및 길이와, 높이를 계산 ( 강의내용 참고 )

  5.2 / 기호는 교집합 --> 2개의 사각형이 교차되는 부분의 사각형의 시작좌표 및 길이와 높이를 계산 ( 강의내용 참고 )

  5.3 == 기호는 동일성 판단 --> 2개의 사각형이 동일한 좌표 및 길이와 높이를 가지고 있는지를 판단 ( 교재 참고 )

  5.4 << 기호는 출력 --> cout 를 통해 출력할수 있도록 연산자를 구현 ( 강의내용 및 교재 참고 )

\*\* 5.5 넓이를 구하는 getArea()함수 작성 ( 강의내용 참고 )

6. 위에서 구현된 기능의 동작을 확인할수 있는 main() 함수를 작성

강의내용 및 교재에 거의 모든 힌트가 존재하지만,

위와 과제와 관련된 문의사항은 질의응답 게시판 및 메일을 이용해 언제든지 질문 가능.

제출방법은 이전과제의 제출방법과 동일

\*\*\* 좌표는 모두 양수만 처리하는 것으로 가정합니다.

1. 소스코드

1-1. Rectangle.h

#pragma once

#include <iostream>

using namespace std;

class Rectangle

{

private:

    int x;

    int y;

    int width;

    int height;

public:

    Rectangle();

    Rectangle(int input\_x, int input\_y, int w, int h);

    ~Rectangle();

    //합집합

    Rectangle operator+(Rectangle& r);

    //교집합

    Rectangle operator/(Rectangle& r);

    //동일성판단

    bool operator==(Rectangle& r);

    int getX();

    int getY();

    int getWidth();

    int getHeight();

    void read();

    void print();

    void getArea();

};

1-2. Rectangle.cpp

#include "Rectangle.h"

Rectangle::Rectangle()

{

    x = 0;

    y = 0;

    width = 0;

    height = 0;

}

Rectangle::Rectangle(int input\_x, int input\_y, int w, int h)

{

    x = input\_x;

    y = input\_y;

    width = w;

    height = h;

}

Rectangle::~Rectangle()

{

    //객체 사라질때

}

Rectangle Rectangle::operator+(Rectangle& r)

{

    Rectangle rect;

    if (r.x < x)

        rect.x = r.x;

    else

        rect.x = x;

    if (r.y < y)

        rect.y = r.y;

    else

        rect.y = y;

    if (r.x + r.width > x + width)

        rect.width = (r.x + r.width) - rect.x;

    else

        rect.width = (x + width) - rect.x;

    if (r.y + r.height > y + height)

        rect.height = (r.y + r.height) - rect.y;

    else

        rect.height = (y + height) - rect.y;

    return rect;

}

Rectangle Rectangle::operator/(Rectangle& r)

{

    Rectangle rect;

    if (r.x < x)

        rect.x = x;

    else

        rect.x = r.x;

    if (r.y < y)

        rect.y = y;

    else

        rect.y = r.y;

    if (r.x + r.width > x + width)

        rect.width = (x + width) - rect.x;

    else

        rect.width = (r.x + r.width) - rect.x;

    if (r.y + r.height > y + height)

        rect.height = (y + height) - rect.y;

    else

        rect.height = (r.y + r.height) - rect.y;

    if (rect.width <= 0 || rect.height <= 0) {

        rect.x = 0;

        rect.y = 0;

        rect.width = 0;

        rect.height = 0;

    }

    return rect;

}

bool Rectangle::operator==(Rectangle& r)

{

    if (x == r.x && y == r.y && width == r.width && height == r.height)

        return true;

    else return false;

}

void Rectangle::read()

{

    cout << "INPUT X : ";

    cin >> x;

    cout << "INPUT Y : ";

    cin >> y;

    cout << "INPUT WIDTH : ";

    cin >> width;

    cout << "INPUT HEIGHT : ";

    cin >> height;

}

void Rectangle::print()

{

    cout << "X : " << x << endl << "Y : " << y << endl;

    cout << "WIDTH : " << width << endl << "HEIGHT : " << height << endl;

}

void Rectangle::getArea()

{

    cout << "AREA : " << width \* height << endl;

}

int Rectangle::getX()

{

    return x;

}

int Rectangle::getY()

{

    return y;

}

int Rectangle::getWidth()

{

    return width;

}

int Rectangle::getHeight()

{

    return height;

}

1-3. main.cpp

#include <iostream>

#include "Rectangle.h"

using namespace std;

ostream& operator<<(ostream& os, Rectangle& r)

{

    os << "X: " << r.getX() << " Y: " << r.getY() << endl;

    os << "WIDTH: " << r.getWidth() << " HEIGHT: " << r.getHeight() << endl;

    return os;

}

int main()

{

    Rectangle rect;

    Rectangle rect1(11, 7, 4, 3);

    Rectangle rect2(5,2,4,6);

    Rectangle rect3(3,6,3,4);

    //input (3,3,5,3)

    rect.read();

    //result: X:3, Y:3, W:5, H:3

    rect.print();

    //result: X:11, Y:7, W:4, H:3

    rect1.print();

    //result: X:3, Y:3, W:12, H:7

    //AREA: 12\*7 = 84

    Rectangle u = rect + rect1;

    u.print();

    u.getArea();

    //result: X:5, Y:6, W:1, H:2

    Rectangle i = rect2 / rect3;

    i.print();

    //result: X:O, Y:0, W:0, H:0

    Rectangle T1 = rect / rect1;

    cout << T1;

    bool e1 = rect == rect1;

    if (e1)

        cout << "rect == rect1: True" << endl;

    else

        cout << "rect == rect1: False" << endl;

    bool e2 = rect == rect;

    if (e2)

        cout << "rect == rect: True" << endl;

    else

        cout << "rect == rect: False" << endl;

    //area: 15

    rect.getArea();

    cout << rect << rect1;

    return 0;

}

2. 실행 화면

