ФГАОУВО «Южно-Уральский государственный университет (НИУ)»

Институт естественных и точных наук

Кафедра «Прикладная математика и программирование»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

по дисциплине «Объектно-ориентированное программирование»

Автор работы

студент группы ЕТ-212

_____ Гайнуллина Э.Д.

_____2022 г.

Работа зачтена с оценкой

_____

_____А.К.Демидов

_____2022 г.

Челябинск, 2017

1 Постановка задачи

I. Базовый класс для всех вариантов:

```
class Figure
{
  int c; // цвет
  bool visible;
protected:
  int x,y; // базовая точка
  virtual void draw();
public:
  Figure(int c, int x, int y);
  ~Figure();
  void move(int x, int y); // сместить фигуру в точку (x,y)
               // видимая фигура гасится, затем рисуется в другом
месте
               // у невидимой просто меняются поля x,y
  void setcolor(int c); // установить цвет фигуры
                        // видимая фигура рисуется новым цветом
                        // у невидимой просто меняется поле c
  int getcolor() const; // получить цвет
  void hide();    // спрятать: нарисовать черный прямоугольник
                  //           по размерам area()
  void show();    // показать
  bool isvisible() const; // видима?
  virtual void area(int &x1,int &y1,int &x2,int &y2) const;
                 // получить размеры прямоугольной области, содержащей
фигуру
};
```

Определить реализацию методов класса Figure.
Методы area и draw нужно определить как чисто виртуальные.
Как нужно определить деструктор Figure и производных классов, чтобы видимый объект исчезал с экрана при уничтожении?
Определить производный класс

5. Прямоугольник со скругленными углами
Rectangle(цвет линий, x и y левого нижнего угла, длина, высота, радиус)

Определить дополнительный метод в производном классе для изменения размеров:

```
  void setsizes(длина, высота);
```

или `void setsizes(длина, высота, радиус);`
или `void setsizes(радиус, угол1, угол2);`
и т.д., т.е. изменение значений, указываемых в аргументах конструтора, начиная с четвертого.

От написанного класса произвести новый дочерний класс - закрашенная фигура.
Например, закрашенный ромб (FillRomb ← Romb ← Figure).
Добавить к параметрам конструктора цвет заполнения.
Определить дополнительный метод для изменения цвета заполнения:

```
void setfillcolor(int c);
```

II. Реализовать main с тестами

2 Описание интерфейса класса

```
class Figure
{
   private:
      int c; // цвет
      bool visible; // видимость
   protected:
      int x,y; // базовая точка
      virtual void draw(); // нарисовать
   public:
      Figure(int c, int x, int y): c(c), x(x), y(y){visible =
false;} // Конструктор
      virtual ~Figure() {} // Деструктор
      void move(int x, int y); // переместить фигуру в точку
      void setcolor(int c); // установить цвет фигуры
      int getcolor() const { return c; } // получить цвет
      void hide(); // спрятать фигуру
      void show(); // показать фигуру
      bool isvisible() const { return visible; } // видима?
      virtual void area(int &x1, int &y1, int &x2, int &y2)
const = 0;
   // получить размеры прямоугольной области, содержащей фигуру
};


class Rectangl: public Figure
{
   protected:
      int w, h, r; // длинна, высота и радиус прямоугольника
      void draw(); // нарисовать
   public:
      Rectangl(int c, int x, int y, int w, int h, int r):
Figure(c, x, y), w(w), h(h), r(r){} // конструктор
      ~Rectangl(){hide();} // деструктор
```

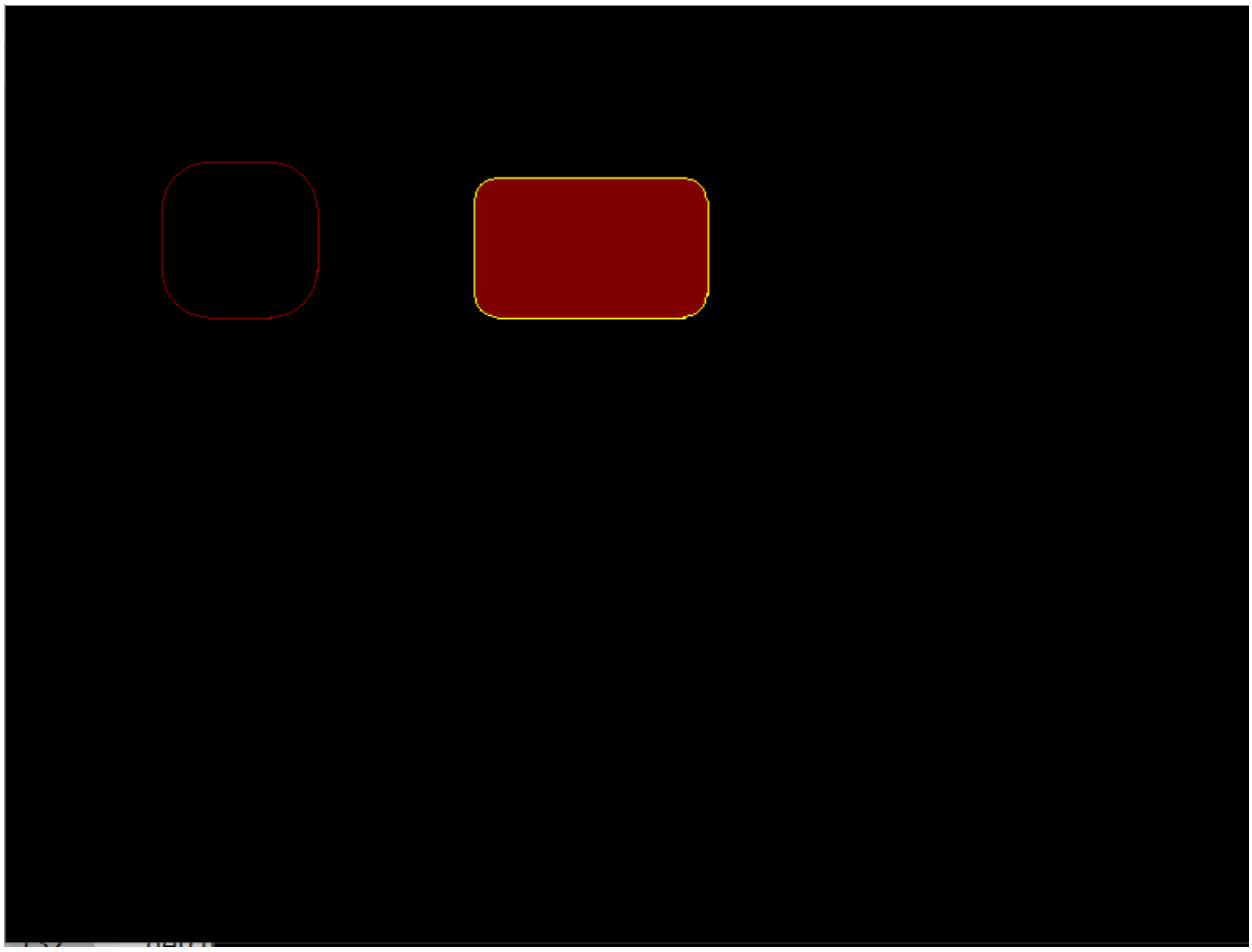
```
      void setsizes(int w, int h, int r);// изменение размера
      void area(int &x1,int &y1,int &x2,int &y2) const; //
получить размеры прямоугольной области, содержащей фигуру
};

class FillRectangle: public Rectangl
{
   private:
      int fillColor; // цвет закраски
      void draw(); // нарисовать
   public:
      FillRectangle(int c, int x, int y, int w, int h, int r,
int fillColor): Rectangl(c, x, y, w, h, r),
fillColor(fillColor){} //конструктор
      void setfillcolor(int c); // изменить цвет закраски
};
```

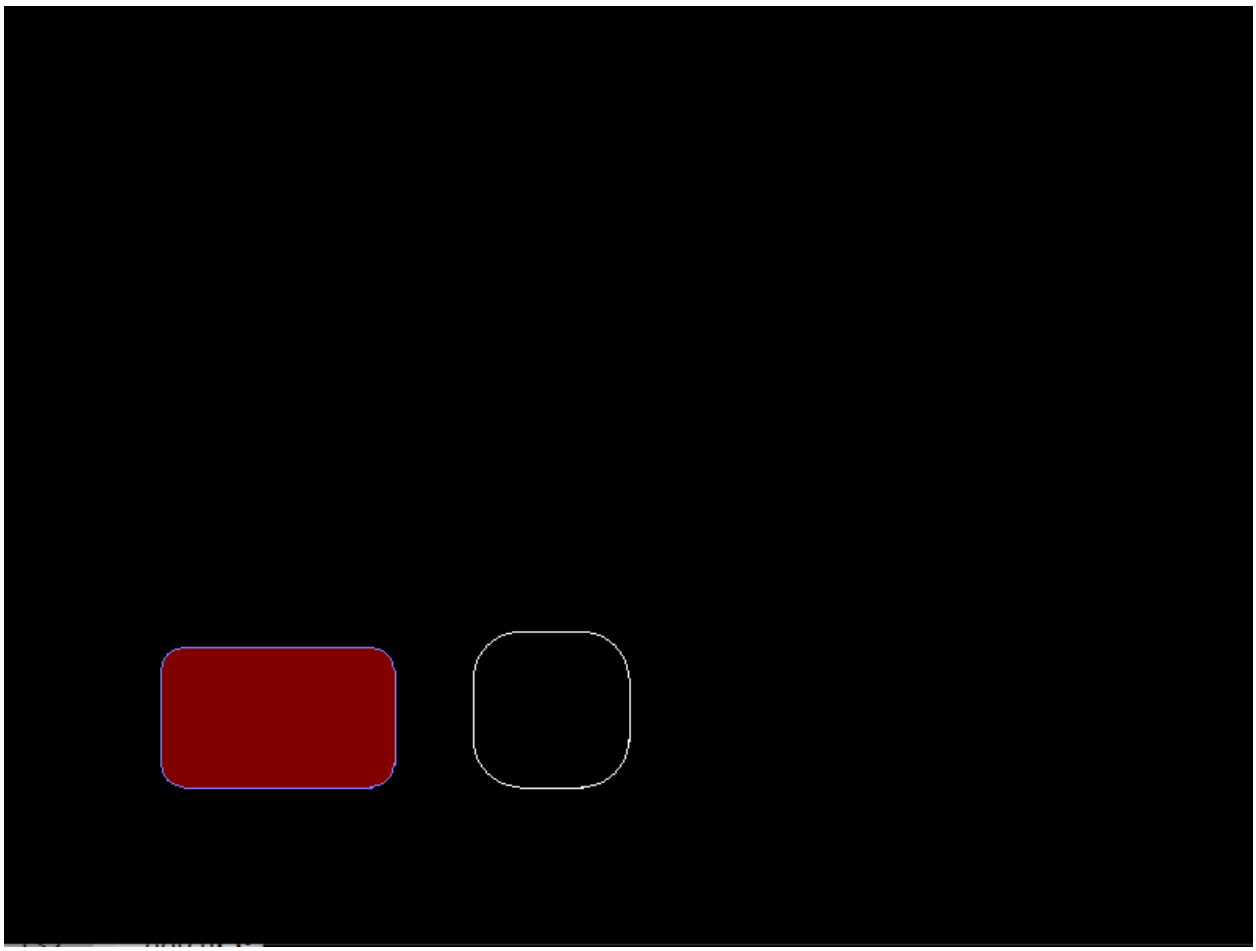
3 Описание тестов для проверки классов

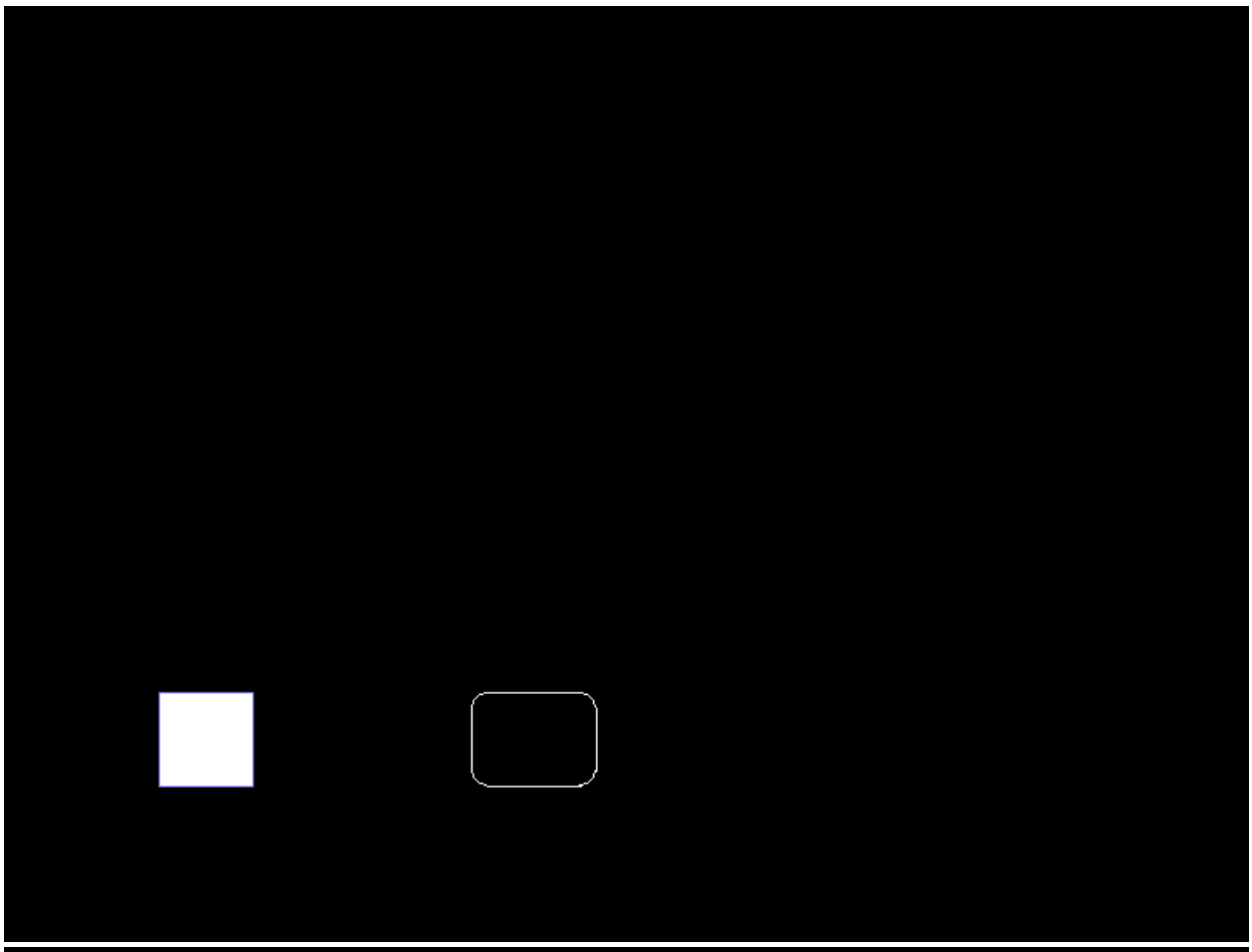
```
int main() {

   initwindow(800, 600);

   Figure *a = new Rectangl(RED, 100, 200, 100, 100, 30);

   Figure *b = new FillRectangle(YELLOW, 300, 200, 150, 90, 15,
RED);



   a->show();

   b->show();

   getch();

   a->hide();

   b->hide();

   getch();

   a->move(300, 500);

   b->move(100, 500);

   a->show();

   b->show();

   getch();

   a->setcolor(WHITE);

   b->setcolor(LIGHTBLUE);

   getch();

// проверяем изменение размеров, обе фигуры меняются

   if (Rectangl *r=dynamic_cast<Rectangl *>(a)) r->setsizes(80,
60, 10);

   if (Rectangl *r=dynamic_cast<Rectangl *>(b)) r->setsizes(60,
60, 0);

   getch();

// проверяем перекраску, фигура a не должна измениться

   if (FillRectangle *r=dynamic_cast<FillRectangle *>(a)) r-
>setfillcolor(WHITE);
```

```
   if (FillRectangle *r=dynamic_cast<FillRectangle *>(b)) r-
>setfillcolor(WHITE);

   getch();

// проверяем исчезновение с экрана при удалении

   delete a;

   delete b;

   getch();

   return 0;

}
```

Полученные результаты







4 Листинг реализации класса

```
void Figure::setcolor(int c) {
   this->c = c;
   if (visible) draw();
}

void Figure::move(int x, int y) {
   bool s = visible;
   if (s) hide();
   this->x = x;
   this->y = y;
   if (s) show();
}

void Figure::hide() {
   if (visible == 0) return;
   int x1, y1, x2, y2;
   area(x1, y1, x2, y2);
   setfillstyle(SOLID_FILL, BLACK);
   bar(x1, y1, x2, y2);
   visible = 0;
}

void Figure::show() {
   if (visible) return;
   draw();
   visible = 1;
}

void Rectangl::area(int &x1,int &y1,int &x2,int &y2) const {
   x1 = x;
   y1 = y-h;
   x2 = x+w;
   y2 = y;
}

void Rectangl::setsizes(int w, int h, int r) {
   bool s = isvisible();
   if (s) hide();
   this->w = w;
   this->h = h;
   this->r = r;
   if (s) show();
}

void Rectangl::draw() {
   ::setcolor(getcolor());
   arc(x+w-r, y-h+r, 0, 90, r);
   arc(x+r, y-h+r, 90, 180, r);
   arc(x+r, y-r, 180, 270, r);
   arc(x+w-r, y-r, 270, 360, r);
   line(x+w, y-r, x+w, y-h+r);
   line(x+w-r, y-h, x+r, y-h);
```

```
   line(x, y-h+r, x, y-r);
   line(x+r, y, x+w-r, y);
}

void FillRectangle::draw() {
   setfillstyle(SOLID_FILL, fillColor);
   Rectangl::draw();
   floodfill(x+w/2, y-h/2, getcolor());
}

void FillRectangle::setfillcolor(int c) {
   fillColor = c;
   if (isvisible()) draw();
}
```