

Національний технічний університет України «Київський політехнічний  
інститут»  
Кафедра обчислювальної техніки

*Лабораторна робота №9*

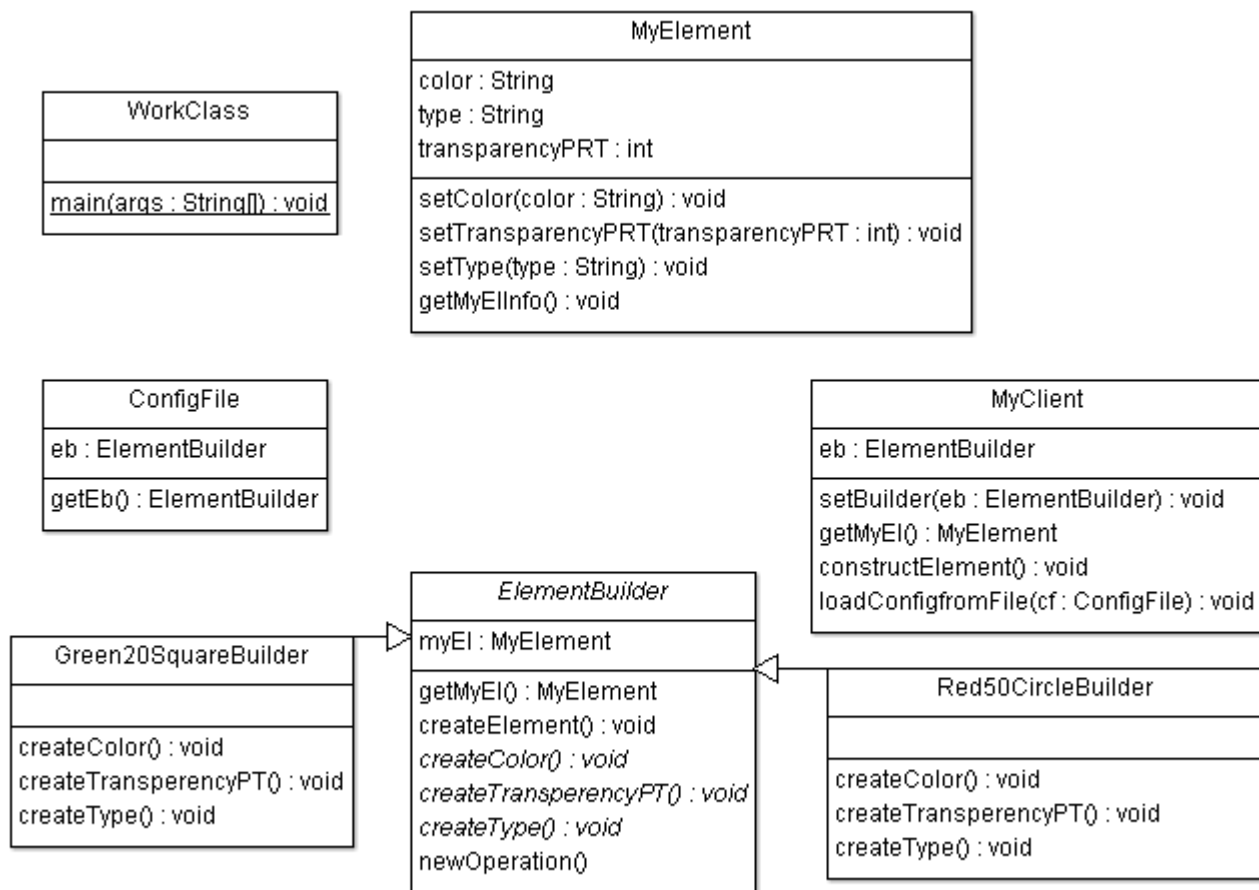
*з дисципліни «Інженерія програмного забезпечення»*

*Виконав:  
студент 2 курсу  
ФІОТ гр. ІО-32  
Довгаль Д.С.  
Залікова книжка №3211*

Варіанти (3211 mod 11)

10. Визначити специфікації класів для подання елементів векторного графічного редактору (примітив і композит). Реалізувати можливість побудови композитного зображення на основі завантаженого файлу-специфікації.

Київ 2014 р.



```

package lab111.labwork9;

/**
 * @author Dima
 * @version 21.10.2014
 * Realize Builder design pattern. This is the Client.
 */
public class MyClient {

    /**
     * Builder, wich we will use
     */
    ElementBuilder eb;

    /**
     * setter for our builder
     * @param eb builder to set.
     */
    public void setBuilder(ElementBuilder eb) {
        this.eb = eb;
    }

    /**
     * Getter for our element
     * @return our element
     */
    public MyElement getMyEl() {
        return eb.getMyEl();
    }

    /**
     * Do all operations to finally construct our element.
     */
    public void constructElement() {

```

```

        eb.createElement();
        eb.createColor();
        eb.createTransparencyPT();
        eb.createType();
    }

    /**
     * Loads config from our ConfigFile.
     * @param cf our ConfigFile.
     */
    public void loadConfigfromFile(ConfigFile cf) {
        setBuilder(cf.getEb());
    }
}

```

```
package lab111.labwork9;
```

```

/**
 * @author Dima
 * @version 21.10.2014
 * Realize Builder design pattern. This is the product of our builder.
 */
public class MyElement {

    /**
     * Parameters, simply to all elements.
     */
    private String color= "";
    private String type= "";
    private int transparencyPRT= 100;

    /**
     * Setter for color.
     * @param color color wich will use.
     */
    public void setColor(String color) {
        this.color = color;
    }

    /**
     * Setter for transparencyPRT
     * @param transparencyPRT transparencyPRT wich will use.
     */
    public void setTransparencyPRT(int transparencyPRT) {
        this.transparencyPRT = transparencyPRT;
    }

    /**
     * Setter for type.
     * @param type type wich will use.
     */
    public void setType(String type) {
        this.type = type;
    }

    /**
     * To get some info about element.
     */
    public void getMyElInfo() {
        System.out.println("This is "+ type+ " whith "+ transparencyPRT+ "%, and its color
is "+ color);
    }
}

```

```

package lab111.labwork9;

/**
 * @author Dima
 * @version 21.10.2014
 * Realize Builder design pattern. This is the a concrete Builder.
 * Its creates red circle with 50% transparency.
 */
public class Red50CircleBuilder extends ElementBuilder {

    @Override
    public void createColor() {
        myEl.setColor("RED");
    }

    @Override
    public void createTransperencyPT() {
        myEl.setTransparencyPRT(50);
    }

    @Override
    public void createType() {
        myEl.setType("Circle");
    }
}

```

```

package lab111.labwork9;

/**
 * @author Dima
 * @version 21.10.2014
 * Only workclass.
 */
public class WorkClass {
    public static void main(String[] args) {

        //создаю файл и клиента, для пождальших опреаций с ними
        ConfigFile myFile= new ConfigFile();
        MyClient client = new MyClient();

        //создаю строителя с 1 вариантом
        ElementBuilder red50CircleBuilder = new Red50CircleBuilder();

        //говорю клиенту, что именно строить
        client.setBuilder(red50CircleBuilder);
        client.constructElement();

        //узнаю инфу
        MyElement myEl1= client.getMyEl();
        myEl1.getMyElInfo();
        System.out.println();

        //загружаю конфигурацию с файла
        client.loadConfigfromFile(myFile);
        client.constructElement();

        //узнаю инфу
        MyElement myEl2= client.getMyEl();
        myEl2.getMyElInfo();
    }
}

```

```

package lab111.labwork9;

```

```

/**

```

```

* @author Dima
* @version 21.10.2014
* Only file.
*/
public class ConfigFile {

    /**
     * Info that our file have.
     */
    private ElementBuilder eb= new Green20SquareBuilder();

    /**
     * Getter for get info from our file.
     * @return info, that file have.
     */
    public ElementBuilder getEb() {
        return eb;
    }
}

package lab111.labwork9;

/**
 * @author Dima
 * @version 21.10.2014
 * Realize Builder design pattern. This is the Abstract Builder.
 */
abstract public class ElementBuilder {

    /**
     * Element, which we will construct.
     */
    protected MyElement myEl;

    /**
     * Getter for element
     * @return element.
     */
    public MyElement getMyEl() {
        return myEl;
    }

    /**
     * Method to firstly create our element.
     */
    public void createElement() {
        myEl= new MyElement();
    }

    //other abstract methods

    /**
     * Set color for element.
     */
    public abstract void createColor();

    /**
     * Set pts of Transperency for element.
     */
    public abstract void createTransperencyPT();

    /**
     * Set type for element.
     */
    public abstract void createType();
}

```

```
package lab111.labwork9;

/**
 * @author Dima
 * @version 21.10.2014
 * Realize Builder design pattern. This is the a concrete Builder.
 * Its creates green square with 20% transparency.
 */
public class Green20SquareBuilder extends ElementBuilder {

    @Override
    public void createColor() {
        myEl.setColor("GREEN");
    }

    @Override
    public void createTransperencyPT() {
        myEl.setTransparencyPRT(20);
    }

    @Override
    public void createType() {
        myEl.setType("Square");
    }
}
```