Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 8

З дисципліни «Інженерія програмного забезпечення»

На тему « ПОРОДЖУВАЛЬНІ ШАБЛОНИ. ШАБЛОНИ PROTOTYPE, SINGLETON, FACTORY METHOD»

ВИКОНАЛА:

студентка 2 курсу ФІОТ

групи ІВ-71

Молчанова Варвара

Залікова - 7110

Варіант – 1

ПЕРЕВІРИВ:

Старший викладач

к.т.н., с.н.с.

Антонюк А.І.

Київ – 2018

ЗАВДАННЯ

Варіант 1.

Визначити специфікації класів та реалізацію методів для механізму клонування графічних елементів у редакторі векторної графіки. Забезпечити можливість як глибокого так і поверхневого клонування.

ПРОГРАМНИЙ КОД

**package** com.lab111.labwork8;  
  
*/\*\*  
 \* Interface, that cloned objects  
 \** ***@author*** *Molchanova Varvara  
 \*/***public interface** Prototype {  
 **public** Prototype surfСlone();  
 **public** Prototype deepClone();  
  
}

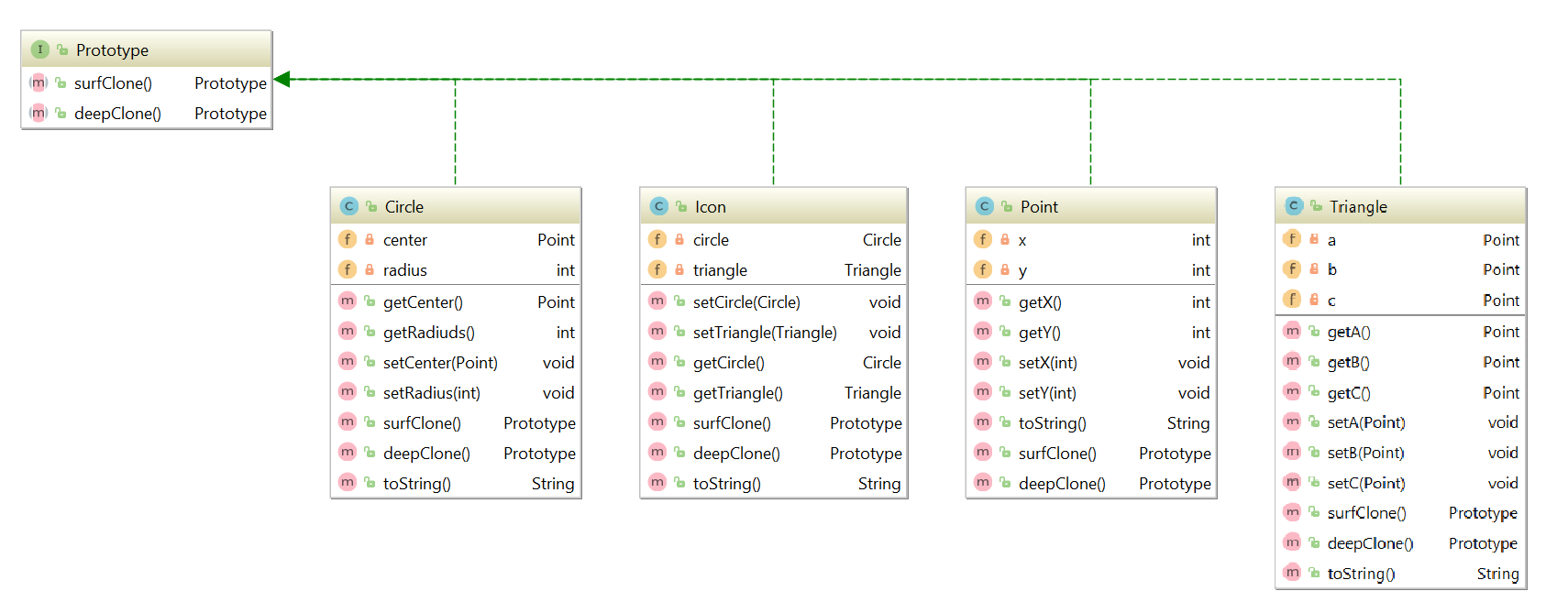
**package** com.lab111.labwork8;  
  
*/\*\*  
 \* Class, which implements interface Prototype.  
 \* Class, which describe point element of graphics  
 \** ***@author*** *Molchanova Varvara  
 \*/***public class** Point **implements** Prototype{  
 **private int x**;  
 **private int y**;  
 */\*\*  
 \*  
 \** ***@param x*** *x-coordinate on field  
 \** ***@param y*** *y-coordinate on field  
 \*/* **public** Point(**int** x, **int** y){  
 **this**.**x**=x;  
 **this**.**y**=y;  
 }  
 */\*  
 \* Getters and Setters  
 \*/* **public int** getX(){  
 **return this**.**x**;  
 }  
  
 **public int** getY(){  
 **return this**.**y**;  
 }  
 **public void** setX(**int** x){  
 **this**.**x**=x;  
 }  
 **public void** setY(**int** y){  
 **this**.**y**=y;  
 }  
 **public** String toString(){  
 **return "Point coordinates: "**+**"[ "**+**this**.**x**+**", "**+**this**.**y**+**" ]"**;  
 }  
 */\*\*  
 \* Surf-clone method  
 \*/* @Override  
 **public** Prototype surfСlone() {  
 *//* ***TODO Auto-generated method stub* return new** Point(**this**.**x**,**this**.**y**);  
 }  
 */\*\*  
 \* Deep-clone method  
 \*/* @Override  
 **public** Prototype deepClone() {  
 *//* ***TODO Auto-generated method stub* return this**.surfСlone();  
 }  
  
}

**package** com.lab111.labwork8;  
  
*/\*\*  
 \* Class, which implements interface Prototype.  
 \* Class, which describe triangle element of graphics  
 \** ***@author*** *Molchanova Varvara  
 \*/***public class** Triangle **implements** Prototype {  
 **private** Point **a**;  
 **private** Point **b**;  
 **private** Point **c**;  
 */\*\*  
 \*  
 \** ***@param a*** *anchor point  
 \** ***@param b*** *anchor point  
 \** ***@param c*** *anchor point  
 \*/* **public** Triangle(Point a, Point b, Point c){  
 **this**.**a** = a;  
 **this**.**b** = b;  
 **this**.**c** = c;  
  
 }  
 */\*  
 \* Getters and Setters  
 \*/* **public** Point getA(){  
 **return this**.**a**;  
 }  
 **public** Point getB(){  
 **return this**.**b**;  
 }  
 **public** Point getC(){  
 **return this**.**c**;  
 }  
 **public void** setA(Point a){  
 **this**.**a**=a;  
 }  
 **public void** setB(Point b){  
 **this**.**b**=b;  
 }  
 **public void** setC(Point c){  
 **this**.**c**=c;  
 }  
 */\*\*  
 \* Surf-clone method  
 \*/* @Override  
 **public** Prototype surfСlone() {  
 *//* ***TODO Auto-generated method stub* return new** Triangle(**this**.**a**,**this**.**b**,**this**.**c**);  
 }  
 */\*\*  
 \* Deep-clone method  
 \*/* @Override  
 **public** Prototype deepClone() {  
 *//* ***TODO Auto-generated method stub* return this**.surfСlone();  
 }  
 **public** String toString(){  
 **return "Triangle coordinates: "**+**"a="**+**this**.**a**.toString()+**", "**+**"b="**+**this**.**b**.toString()+**", "**+**"c="**+**this**.**c**.toString()+**"."**;  
 }  
  
}

**package** com.lab111.labwork8;  
  
*/\*\*  
 \* Class, which implements interface Prototype.  
 \* Class, which describe circle element of graphics  
 \** ***@author*** *Molchanova Varvara  
 \*/***public class** Circle **implements** Prototype{  
 **private** Point **center**;  
 **private int radius**;  
 */\*\*  
 \*  
 \** ***@param center*** *center of circle  
 \** ***@param radius*** *radius of circle  
 \*/* **public** Circle (Point center, **int** radius){  
 **this**.**center**=center;  
 **this**.**radius**=radius;  
 }  
 */\*  
 \* Getters and Setters  
 \*/* **public** Point getCenter(){  
 **return this**.**center**;  
 }  
 **public int** getRadiuds(){  
 **return this**.**radius**;  
 }  
 **public void** setCenter(Point center){  
 **this**.**center**=center;  
 }  
 **public void** setRadius(**int** radius){  
 **this**.**radius**=radius;  
 }  
 */\*\*  
 \* Surf-clone method  
 \*/* @Override  
 **public** Prototype surfСlone() {  
 *//* ***TODO Auto-generated method stub* return new** Circle(**this**.**center**,**this**.**radius**);  
 }  
  
 */\*\*  
 \* Deep clone method  
 \*/* @Override  
 **public** Prototype deepClone() {  
 *//* ***TODO Auto-generated method stub* return this**.surfСlone();  
 }  
 **public** String toString(){  
 **return "Circle coordinates: "**+**"center="**+**this**.**center**.toString()+**", "**+**"radius="**+**this**.**radius**+**"."**;  
 }  
  
  
}

**package** com.lab111.labwork8;  
  
*/\*\*  
 \* Class, which implements interface Prototype.  
 \* Class, which describe icon element of graphics  
 \** ***@author*** *Molchanova Varvara  
 \*/***public class** Icon **implements** Prototype {  
 **private** Circle **circle**;  
 **private** Triangle **triangle**;  
 */\*\*  
 \*  
 \** ***@param circle*** *circle element of icon  
 \** ***@param triangle*** *triangle element of icon  
 \*/* **public** Icon (Circle circle, Triangle triangle){  
 **this**.**circle**=circle;  
 **this**.**triangle**=triangle;  
 }  
 */\*  
 \* Getters and Setters  
 \*/* **public void** setCircle(Circle circle){  
 **this**.**circle**=circle;  
 }  
 **public void** setTriangle(Triangle triangle){  
 **this**.**triangle**=triangle;  
 }  
  
 **public** Circle getCircle (){  
 **return this**.**circle**;  
 }  
  
 **public** Triangle getTriangle(){  
 **return this**.**triangle**;  
 }  
 */\*\*  
 \* Surf-clone method  
 \*/* @Override  
 **public** Prototype surfСlone() {  
 *//* ***TODO Auto-generated method stub* return new** Icon(**this**.**circle**,**this**.**triangle**);  
 }  
 */\*\*  
 \* Deep-clone method  
 \*/* @Override  
 **public** Prototype deepClone() {  
 *//* ***TODO Auto-generated method stub* return this**.surfСlone();  
 }  
 **public** String toString(){  
 **return "Icon coordinates: "**+**circle**.toString()+**" "**+**triangle**.toString();  
 }  
  
}

ДІАГРАМА КЛАСІВ



РЕЗУЛЬТАТИ ТЕСТУВАННЯ ПРОГРАМИ

Properties of graphic elements

Point coordinates: [ 5, 7 ]

Triangle coordinates: a=Point coordinates: [ 5, 7 ], b=Point coordinates: [ 6, 8 ], c=Point coordinates: [ 9, 7 ].

Circle coordinates: center=Point coordinates: [ 5, 7 ], radius=6.

Icon coordinates: Circle coordinates: center=Point coordinates: [ 5, 7 ], radius=6. Triangle coordinates: a=Point coordinates: [ 5, 7 ], b=Point coordinates: [ 6, 8 ], c=Point coordinates: [ 9, 7 ].

Properties of clones

Point coordinates: [ 5, 7 ]

Triangle coordinates: a=Point coordinates: [ 5, 7 ], b=Point coordinates: [ 6, 8 ], c=Point coordinates: [ 9, 7 ].

Circle coordinates: center=Point coordinates: [ 5, 7 ], radius=6.

Icon coordinates: Circle coordinates: center=Point coordinates: [ 5, 7 ], radius=6. Triangle coordinates: a=Point coordinates: [ 5, 7 ], b=Point coordinates: [ 6, 8 ], c=Point coordinates: [ 9, 7 ].

ВИСНОВКИ

Ознайомились із породжувальними шаблонами Prototype, Singleton, Factory Method.

Отримали навички із застосування шаблонів.

Розроблена відповідна тестова програма.

Результати успішної роботи тестової програми наведені вище підтверджують правильність обраних рішень.