Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №3

З предмету: «Інженерія програмного забезпечення»

номер залікової книжки 2230

Виконав: студент групи ІО-22

Щербина М.В.

Перевірив: Абу Усбах А.Н.

Київ 2013р.

Варіант 10

10. Визначити специфікації класів та реалізацію методів для маніпулювання зображеннями з можливістю їх “пізнього завантаження”. Реалізувати бізнес-метод для визначення кольору точки за його координатами.

**Завдання**

1. Ознайомитись з призначенням та видами шаблонів проектування ПЗ. Вивчити класифікацію шаблонів проектування ПЗ. Знати назви шаблонів, що відносяться до певного класу.

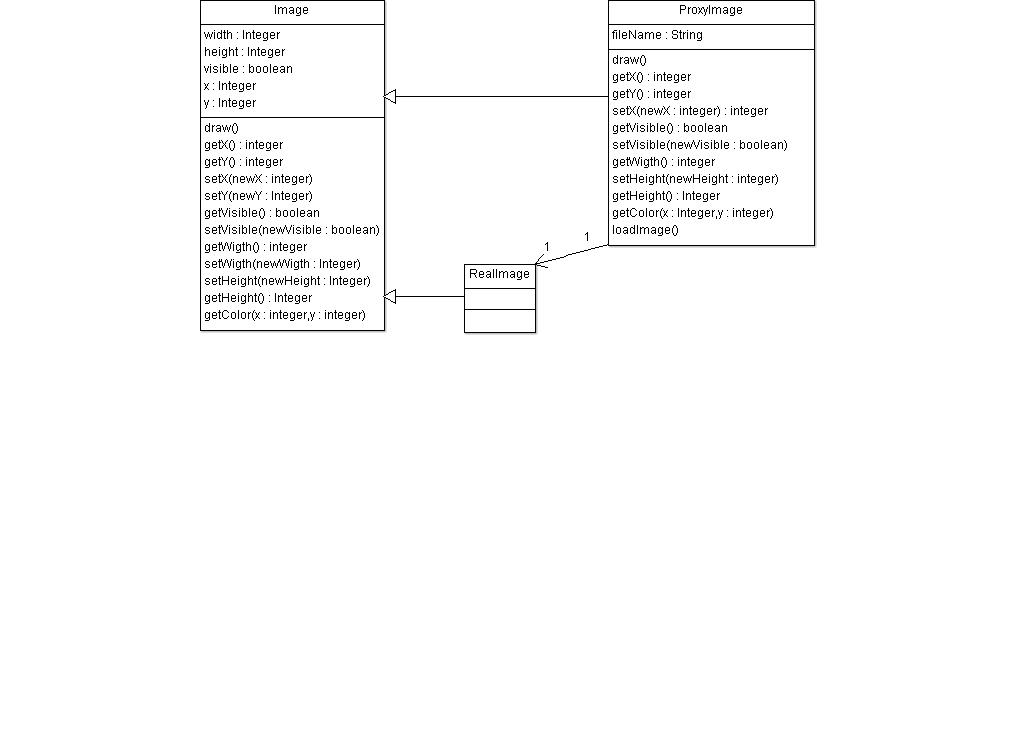
2. Вивчити структурні шаблонів проектування ПЗ. Знати загальну характеристику структурних шаблонів та призначення кожного з них.

3. Детально вивчити структурні шаблони проектування Composite, Decorator та Proxy. Для кожного з них:

* вивчити Шаблон, його призначення, альтернативні назви, мотивацію, випадки коли його застосування є доцільним та результати такого застосування;
* знати особливості реалізації Шаблону, споріднені шаблони, відомі випадки його застосування в програмних додатках;
* вільно володіти структурою Шаблону, призначенням його класів та відносинами між ними;
* вміти розпізнавати Шаблон в UML діаграмі класів та будувати сирцеві коди Java-класів, що реалізують шаблон.

4. В підготованому проекті (ЛР1) створити програмний пакет com.lab111.labwork3. В пакеті розробити інтерфейси і класи, що реалізують завдання (згідно варіанту) з застосуванням одного чи декількох шаблонів (п.3). В розроблюваних класах повністю реалізувати методи, пов'язані з функціюванням Шаблону. Методи, що реалізують бізнес-логіку закрити заглушками з виводом на консоль інформації про викликаний метод та його аргументи.

5. За допомогою автоматизованих засобів виконати повне документування розроблених класів (також методів і полів), при цьому документація має в достатній мірі висвітлювати роль певного класу в загальній структурі Шаблону та особливості конкретної реалізації.



**package** com.lab111.labwork3;

/\*\*

\* реальный обьект

\*

\*/

**public** **class** RealImage **extends** Image {

**public** RealImage(**int** width, **int** height, **boolean** visible, **int** x, **int** y) {

**super**(width, height, visible, x, y);

}

}**package** com.lab111.labwork3;

/\*\*

\* Интерфейс с которым будет работать пользователь

\*

\*/

**public** **class** Image {

**protected** **int** width;

**protected** **int** height;

**protected** **boolean** visible;

**protected** **int** x;

**protected** **int** y;

/\*\*

\* конструктор

\*

\* **@param** width

\* - ширина

\* **@param** height

\* -высота

\* **@param** visible

\* - видимость

\* **@param** x

\* - кооридината

\* **@param** y

\* -координата

\*/

**public** Image(**int** width, **int** height, **boolean** visible, **int** x, **int** y) {

**super**();

**this**.width = width;

**this**.height = height;

**this**.visible = visible;

**this**.x = x;

**this**.y = y;

}

/\*\*

\* нарисовать катринку

\*/

**public** **void** draw() {

System.*out*.println("draw()");

}

**public** **int** getX() {

**return** x;

}

**public** **int** getY() {

**return** y;

}

**public** **void** setX(**int** newX) {

x = newX;

}

**public** **void** setY(**int** newY) {

y = newY;

}

**public** **boolean** getVisible() {

**return** visible;

}

**public** **void** setVisible(**boolean** newVisible) {

visible = newVisible;

}

**public** **int** getWigth() {

**return** width;

}

**public** **void** setWigth(**int** newWigth) {

width = newWigth;

}

**public** **void** setHeight(**int** newHeight) {

height = newHeight;

}

**public** **int** getHeight() {

**return** height;

}

/\*\*

\* возращает цвет пикселя по координате

\*

\* **@param** x

\* **@param** y

\* **@return**

\*/

**public** **int** getColor(**int** x, **int** y) {

System.*out*.println("getColor x = " + x + ", y = " + y);

**return** 0;

}

}**package** com.lab111.labwork3;

/\*\*

\* Класс котрый реализует паттерн Proxy

\*

\*/

**public** **class** ProxyImage **extends** Image {

**public** ProxyImage(**int** width, **int** height, **boolean** visible, **int** x, **int** y,

String fileName) {

**super**(width, height, visible, x, y);

**this**.fileName = fileName;

}

**private** String fileName;

**private** RealImage realImage;//ссылка на настоящий обьект

/\*

\* нарисовать обьект

\*/

**public** **void** draw() {

//если обьект еще незагружен, то загружается

**if** (realImage == **null**)

loadImage();

realImage.draw();

}

**public** **int** getX() {

**if** (realImage == **null**)

**return** x;

**else**

**return** realImage.getX();

}

**public** **int** getY() {

**if** (realImage == **null**)

**return** y;

**else**

**return** realImage.getY();

}

**public** **void** setX(**int** newX) {

**if** (realImage == **null**)

x = newX;

**else**

realImage.setX(newX);

}

**public** **boolean** getVisible() {

**if** (realImage == **null**)

**return** visible;

**else**

**return** realImage.getVisible();

}

**public** **void** setVisible(**boolean** newVisible) {

**if** (realImage == **null**)

visible = newVisible;

**else**

realImage.setVisible(newVisible);

}

**public** **int** getWigth() {

**if** (realImage == **null**)

**return** width;

**else**

**return** realImage.getWigth();

}

**public** **void** setHeight(**int** newHeight) {

**if** (realImage == **null**)

height = newHeight;

**else**

realImage.setHeight(newHeight);

}

**public** **int** getHeight() {

**if** (realImage == **null**)

**return** height;

**else**

**return** realImage.getHeight();

}

**public** **int** getColor(**int** x, **int** y) {

**if** (realImage == **null**)

loadImage();

**return** realImage.getColor(x, y);

}

/\*\*

\* загружает и создает реальный обьект

\*/

**private** **void** loadImage() {

realImage = **new** RealImage(width, height, visible, x, y);

}

}

**package** com.lab111;

**import** com.lab111.labwork3.\*;

/\*\*

\* Template first class.

\*

\*/

**public** **final** **class** TestMain {

/\*\*

\* Constructor.

\*

\*/

**private** TestMain() {

**super**();

}

/\*\*

\* Invokes at application startup.

\* **@param** args Parameters from command line

\*/

**public** **static** **void** main(**final** String[] args) {

Image image=**new** ProxyImage(1, 2, **true**, 0, 0, "image.jpg");

**int** x=image.getX();

**int** y=image.getY();

image.draw();

x=image.getX();

y=image.getY();

}

}