Міністерство освіти і науки України

Черкаський державний технологічний університет

Кафедра програмного забезпечення автоматизованих систем

Звіт

З лабораторної роботи №3

З предмету «Об’єктно-орієнтоване програмування»

Перевірив

Викладач кафедри

Програмного забезпечення автоматизованих систем

Крайовий В. М.

Виконав

Студент І курсу

Група ПЗ-154

Кравченко Артем Олегович

Виконав

Студент ІІ курсу

Група ПЗ-154

Мороз Б.В.

Черкаси 2016

**Тема:** Конструктори. Перевантаження конструкторів.

**Мета:** Навчитись створювати конструктори у власних класах та використовувати їх при створенні об’єктів. Навчитись створювати перевантажені конструктори.

**Постановка завдання:** В роботі використовується клас, створений в попередній лабораторній роботі.

1. Створити в своєму класі конструктор, в якому ініціалізуються значення полів. Початкові значення полів повинні бути максимально логічними. Випробувати роботу описаного конструктора та оцінити її результат.

2. Додати в свій клас ще два перевантажені конструктори. Випробувати роботу кожного з них та оцінити їх результат.

3. Зробити так, щоб як мінімум в одному випадку був виклик одного конструктора з іншого конструктора. Випробувати роботу зміненого(-них) конструктора та оцінити її результат.

4. Проаналізувати програму, написану в лабораторній роботі №1. Вказати, де в ній використовуються конструктори.

5. Додати в свій клас поле типу Color, якщо його там немає.

6. Додати до вікна об'єкт класу JColorChooser, який дозволяє вибирати колір. Забезпечити введення вибраного за допомогою JColorChooser кольору у відповідне поле об'єкту розробленого класу.

**Індивідуальне завдання:** Олівець.

**Хід роботи:**

**Лістинг програми:**

package com.company;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

class Pencil {

float sizeOfEraser;

float sizeOfGraphite;

float sizeOfPencil;

float angle;

float flexibility;

float power;

int sec;

boolean broken;

public Pencil(){

sizeOfEraser = 25;

sizeOfGraphite = 50;

sizeOfPencil = 1000;

angle = 20;

flexibility = 40;

broken = false;

}

@Override

public String toString() {

return "Pencil{" +

"sizeOfEraser=" + sizeOfEraser +

", sizeOfGraphite=" + sizeOfGraphite +

", sizeOfPencil=" + sizeOfPencil +

", angle=" + angle +

", flexibility=" + flexibility +

", sec=" + sec +

", broken=" + broken +

'}';

}

void write(int s){

sec = s;

if (broken == false) {

for (int i = 1; i <= sec; i++) {

if (sizeOfGraphite <= 0) {

System.out.println("No graphite");

break;

} else {

sizeOfGraphite = sizeOfGraphite - 1;

sizeOfPencil = sizeOfPencil - 1;

angle = angle + 5;

}

}

}

else{

System.out.println("Pencil is broken");

}

};

void erase(int s){

sec = s;

if (broken == false) {

for (int i = 1; i < sec; i++) {

if (sizeOfEraser < 0) {

System.out.println("No eraser");

break;

} else {

sizeOfEraser = sizeOfEraser - 1;

sizeOfPencil = sizeOfPencil - 1;

}

}

}

else {

System.out.println("Pencil is broken");

}

}

void bend(int s, float p){

sec = s;

power = p;

for (int i = 1 ; i < sec ; i++) {

if (broken == true){

System.out.println("Pencil is broken");

break;

}

if (sec \* power >= flexibility)

broken = true;

}

}

void shape(int s){

sec = s;

if (broken == false) {

for (int i = 1; i <= sec; i++) {

if (angle >= 20) {

sizeOfPencil = sizeOfPencil - 1;

sizeOfGraphite = sizeOfGraphite - 1;

angle = angle - 5;

} else {

System.out.println("Shaped");

break;

}

}

}

else{

System.out.println("Pencil is broken");

}

}

}

public class Main {

public static void main(String[] args) {

JFrame frame = new JFrame("Pencil");

JButton btn1 = new JButton("Write");

JButton btn2 = new JButton("Erase");

JButton btn3 = new JButton("Bend");

JButton btn4 = new JButton("Shape");

JPanel panel = new JPanel();

Pencil pencil = new Pencil();

System.out.println(pencil);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(200,100);

frame.setVisible(true);

frame.add(panel);

panel.setLayout(new GridLayout(2,2));

panel.add(btn1);

panel.add(btn2);

panel.add(btn3);

panel.add(btn4);

btn1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

pencil.write(5);

System.out.println(pencil);

}

});

btn2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

pencil.erase(5);

System.out.println(pencil);

}

});

btn3.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

pencil.bend(5, 10);

System.out.println(pencil);

}

});

btn4.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

pencil.shape(5);

System.out.println(pencil);

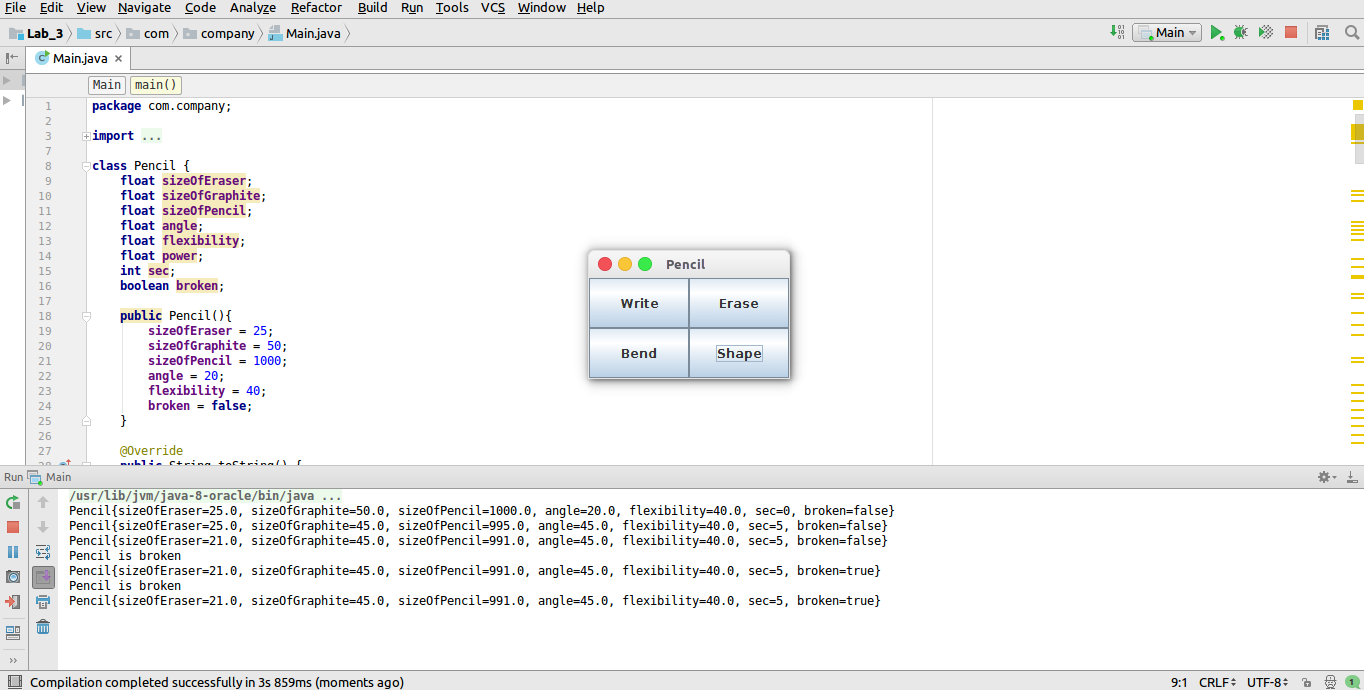
}

});

}

}

**Результат роботи програми:**



**Висновок:** У ході виконання лабораторної роботи, я навчився створювати конструктори у власних класах та використовувати їх при створенні об’єктів. Навчився створювати перевантажені конструктори.