**Computer Language 2023**

**Assignment #4**

**Due: 21/Apr 23:59:59**

**Student ID:\_\_21102054\_\_ Name: \_\_Lim Saeyeon (임새연)\_\_**

**For each question, 1) write your solution codes, 2) present a screenshot of your result, and 3) describe a short explanation about your solution. Without these components, you will be given some penalties.**

**1. Implement Grade class to make the following program work. Grade class’s main method is as follows:**

public static void main(String [] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Input the score of Math, Science, and English subject >> ");  
 int math = scanner.nextInt();  
 int science = scanner.nextInt();  
 int english = scanner.nextInt();  
 Grade me = new Grade(math, science, english);  
 System.*out*.println("Average is " + me.average());   
 scanner.close();  
}

**Output)**

텍스트이(가) 표시된 사진

자동 생성된 설명

**1) Your code**

import java.util.Scanner;

public class Grade {  
  
 int math;  
 int science;  
 int english;  
  
 public int average () {  
 return (math + science + english) / 3;  
 }  
   
 public Grade(int math, int science, int english) {  
 this.math = math;  
 this.science = science;  
 this.english = english;  
 };  
  
 public static void main(String [] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Input the score of Math, Science, and English subject >> ");  
 int math = scanner.nextInt();  
 int science = scanner.nextInt();  
 int english = scanner.nextInt();  
 Grade me = new Grade(math, science, english);  
 System.*out*.println("Average is " + me.average());  
 scanner.close();  
 }

**2) Your result (screenshot)**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**3) Your explanation**

The program defines a Grade class with three instance variables (math, science, english) representing the student's scores in each subject. The average() method calculates the average score by adding the scores in the three subjects and dividing the total by three.

In the main method, the program prompts the user to input the scores for the three subjects using the Scanner class. It then creates a new Grade object with the input values and calls the average() method to calculate and print the average score. Finally, the program closes the scanner.

**2. Implement Rectangle class to provide the following features:**

* Member fields: x, y, width, height (int type)
* Constructor(…): initializes x, y, width, and height fields using the given arguments
* getArea(): returns the area of the rectangle
* show(): prints the coordinate and the area of the rectangle
* contains(Rectangle rect): returns *true* if the current object contains the given Rectangle object *rect*
* The main method of Rectangle class and the expected output is as follows:

텍스트이(가) 표시된 사진

자동 생성된 설명

**Conceptual diagram)**

어두운이(가) 표시된 사진

자동 생성된 설명

**Output)**

텍스트이(가) 표시된 사진

자동 생성된 설명

**1) Your code**

public class Rectangle {  
 int x;  
 int y;  
 int width;  
 int height;  
  
 public int getArea() {  
 return (width \* height);  
 }

public boolean contains(Rectangle rectangle) {  
 if (x <= rectangle.x && y <= rectangle.y)  
 if (((x + width) >= (rectangle.x+rectangle.width))  
 && ((y + width) >= (rectangle.y + rectangle.width))) return true;  
 return false;  
 }

public void show() {  
 System.*out*.print("coordinate: (" + x + "," + y+ ") ");  
 System.*out*.println("area: " + width + "x" + height);  
 }

Rectangle(int x, int y, int width, int height){  
 this.x = x;  
 this.y = y;  
 this.width = width;  
 this.height = height;  
 }  
  
 public static void main(String[] args) {  
 Rectangle r = new Rectangle(2,2,9,7);  
 Rectangle s = new Rectangle(5,5,7,7);  
 Rectangle t = new Rectangle(1,1,10,10);  
  
 r.show();  
 System.*out*.println("s's area is " + s.getArea());  
 if (t.contains(r)) System.*out*.println("t contains r");  
 if (t.contains(s)) System.*out*.println("t contains s");  
 }  
}

**2) Your result (screenshot)**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**3) Your explanation**

The Rectangle class has instance variables x and y to represent the coordinates of the top-left corner of the rectangle, and width and height to represent the dimensions of the rectangle.

The getArea() method calculates and returns the area of the rectangle as the product of its width and height.

The contains() method checks whether another Rectangle object is completely contained within this Rectangle object. It does so by checking whether the top-left corner of the other Rectangle is within this Rectangle, and whether the bottom-right corner of the other Rectangle is also within this Rectangle.

The show() method prints the coordinates and area of the Rectangle object.

In the main() method, the program creates three Rectangle objects and prints their details. It then calculates and prints the area of the s rectangle, and checks if r and s are contained within t by calling the contains() method. If they are, it prints a message to indicate that.

1. **Write a program that works as follows. Your program must have Phone class (private name, private tel fields and their getters, and a constructor method which takes two input parameters should be implemented), and PhoneBook class. PhoneBook class must take the number of people to store from the user and creates Phone array. Information of a single person is stored in a single Phone instance. If a user types "stop", the program ends.**

텍스트이(가) 표시된 사진

자동 생성된 설명

**1) Your code**

import java.util.Scanner;

class Phone {  
 private String name;  
 private String tel;  
 public String getName() {  
 return name;  
 }  
 public String getTel() {  
 return tel;  
 }  
 public Phone(String name, String tel) {  
 this.name = name;  
 this.tel = tel;  
 }  
}  
  
class PhoneBook {  
 Scanner scanner = new Scanner(System.*in*);  
 Phone[] phoneBook;

void register() {  
 System.*out*.print("Number of person to store? >> ");  
 int num = scanner.nextInt();  
 phoneBook = new Phone[num];  
 for (int i = 0; i < num; i++) {  
 System.*out*.print("Name and Tel. No. >> ");  
 String name = scanner.next();  
 String tel = scanner.next();  
 Phone input = new Phone(name, tel);  
 phoneBook[i] = input;  
 }  
 }

String search(String name) {  
 for (int i = 0; i < phoneBook.length; i++)  
 if (name.equals(phoneBook[i].getName())) return phoneBook[i].getTel();  
 return null;  
 }

void execute() {  
 register();  
 while (true) {  
 System.*out*.print("Who do you wanna search for? >> ");  
 String input = scanner.next();  
 if (input.equals("stop")) break;  
 if (search(input) != null)  
 System.*out*.println(input + "'s telephone no. is " + search(input));  
 else System.*out*.println(input + " does not exist.");  
 }

scanner.close();  
 }

public PhoneBook() {  
 }  
  
 public static void main(String[] args) {  
 new PhoneBook().execute();  
 }  
}

**2) Your result (screenshot)**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**3) Your explanation:**

The "Phone" class has two private variables, name and tel, which represent the name and telephone number of a person. It also has two public methods, getName() and getTel(), which return the values of the two variables. It has a constructor that initializes the two variables with values passed as arguments.

The "PhoneBook" class has a Scanner object, an array of "Phone” objects, and three methods. The first method, register(), asks the user for the number of people to be stored in the phone book, creates an array of "Phone" objects with the specified size, and then prompts the user to enter the name and telephone number of each person to be stored in the phone book. It creates a new "Phone" object for each person and stores it in the array.

The second method, search(), takes a name as an argument and searches the phone book for a matching name. If a match is found, it returns the corresponding telephone number. If no match is found, it returns null.

The third method, execute(), calls register method to create the phone book, and then repeatedly prompts the user to enter a name to search for. If the user enters stop, the loop ends. Otherwise, the program calls the "search()" method to find the telephone number associated with the name entered by the user. If a match is found, the program displays the name and telephone number. Otherwise, it displays a message indicating that the name does not exist in the phone book.

The main() method creates an instance of the "PhoneBook" class and calls the "execute()" method to start the program.