МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ
КАФЕДРА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

ЛАБОРОТОРНА РОБОТА №6

з дисципліни «ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

на тему "Абстрактні класи та інтерфейси. Механізм зворотного визову"

Студента 2 курсу групи АД-181
Гежа Н. І.
Перевіряв
доцент Рудніченко Н. Д.

Одеса - 2019

ЗМІСТ

ВСТУП

Ціллю роботи є ознайомлення з абстрактними класами, інтерфейсами та зворотніми викликами, їх практичне використання.

ТЕОРЕТИЧНА ЧАСТИНА

Інтерфейси і абстрактні класи покращують структуру коду і сприяють відділенню інтерфейсу від реалізації. В першу чергу необхідно розглянути поняття абстрактного класу, який є проміжним ступенем між звичайним класом і інтерфейсом.

У прикладі з класами фігур, методи базового класу Shape завжди були «фіктивними». Пристрій може спробувати здійснити методу з класу Shape привела б до помилки в програмі. Це було пов'язано з тим, що клас Shape був потрібен лише для того, щоб визначити загальний інтерфейс всіх класів, похідних від нього, а вже похідні класи переобумовленої ці методи і реалізовували їх по-своєму.

Клас Shape визначав базову форму, спільність всіх похідних класів. Такі класи як Shape називають абстрактними базовими класами або просто абстрактними класами.

Якщо ви оголосите клас, похідний від абстрактного класу, але хочете мати можливість створення нових об'єктів нового типу, то ви повинні перевизначити всі абстрактні методи базового класу. Якщо це не буде зроблено, то похідний клас теж залишиться абстрактним, і компілятор змусить позначити новий клас ключовим словом abstract.

Можна позначити клас ключовим словом abstract навіть тоді, коли в ньому немає жодного абстрактного методу. Це буває корисно, коли необхідно просто заборонити створення екземплярів цього класу.

Завдання 1

1. Переробіть попередню лабораторну роботу з використанням механізму інтерфейсів;
2. Переробіть попередню лабораторну роботу з використанням механізму абстрактних класів;

Дані подано код до першої частини завдання:

Код абстрактного класу Person:

```
package AD181.Gezha;

public abstract class Person {
  private String surname = "";
  private String name = "";
  private int age = 0;

  Person(String surname, String name, int age) {
    this.setSurname(surname);
    this.setName(name);
    this.setAge(age);
  }

  public void setSurname(String surname) {
    this.surname = surname;
  }

  public void setName(String name) {
    this.name = name;
  }

  public void setAge(int age) {
    this.age = age;
  }

  public String getSurname() {
    return this.surname;
  }

  public String getName() {
    return this.name;
  }

  public int getAge() {
    return this.age;
  }

  public abstract String printInfo();
}
```

Код класу Lecturer:

```
package AD181.Gezha;
```

```
public class Lecturer extends Person {
  private String department = "";
  private double salary = 0.0;

  public Lecturer(String surname, String name, int age, String department,
double salary) {
    super(surname, name, age);
    this.department = department;
    this.salary = salary;
  }

  @Override
  public String printInfo() {
    return "Lecturer of " + this.getDepartment() + " department " +
this.getSurname() + " " +
           this.getName() + ", age: " + this.getAge() +
           ". Salary: " + this.getSalary();
  }

  public String getDepartment() {
    return department;
  }

  public void setDepartment(String department) {
    this.department = department;
  }

  public double getSalary() {
    return salary;
  }

  public void setSalary(double salary) {
    this.salary = salary;
  }
}
```

Код класу Student:

```
package AD181.Gezha;

public class Student extends Person {
  private String group = "";
  private String studentID = "";

  public Student(String surname, String name, int age, String group, String
studentID) {
    super(surname, name, age);
    this.group = group;
    this.studentID = studentID;
  }

  @Override
  public String printInfo() {
    return "Student of group " + this.getGroup() + " " + this.getSurname() + " "
+
           this.getName() + ", age: " + this.getAge() +
           ". Student ID: " + this.getStudentID();
  }

  public String getGroup() {
    return group;
```

```
  }

  public void setGroup(String group) {
    this.group = group;
  }

  public String getStudentID() {
    return studentID;
  }

  public void setStudentID(String studentID) {
    this.studentID = studentID;
  }
}
```

Далі подано код до другої частини завдання.

Код інтерфейсу Person:

```
package AD181.Gezha;

public interface Person {
  public String printInfo();

  void setSurname(String surname);

  void setName(String name);

  void setAge(int age);

  String getSurname();

  String getName();

  int getAge();
}
```

Код класу Student:

```
package AD181.Gezha;

public class Student implements Person {
  private String group = "";
  private String studentID = "";
  private String surname = "";
  private String name = "";
  private int age = 0;

  public Student(String surname, String name, int age, String group, String
studentID) {
    this.surname = surname;
    this.name = name;
    this.age = age;
    this.group = group;
    this.studentID = studentID;
  }

  @Override
  public String printInfo() {
    return "Student of group " + this.getGroup() + " " + this.getSurname() + " "
```

```
+
            this.getName() + ", age: " + this.getAge() +
            ". Student ID: " + this.getStudentID();
  }

  @Override
  public String getSurname() {
    return surname;
  }

  @Override
  public void setSurname(String surname) {
    this.surname = surname;
  }

  @Override
  public String getName() {
    return name;
  }

  @Override
  public void setName(String name) {
    this.name = name;
  }

  @Override
  public int getAge() {
    return age;
  }

  @Override
  public void setAge(int age) {
    this.age = age;
  }

  public String getGroup() {
    return group;
  }

  public void setGroup(String group) {
    this.group = group;
  }

  public String getStudentID() {
    return studentID;
  }

  public void setStudentID(String studentID) {
    this.studentID = studentID;
  }
}
```

Код класу Lecturer:

```
package AD181.Gezha;

public class Lecturer implements Person {
  private String department = "";
  private double salary = 0.0;
  private String surname = "";
  private String name = "";
```

```
  private int age = 0;

  public Lecturer(String surname, String name, int age, String department,
double salary) {
    this.age = age;
    this.surname = surname;
    this.name = name;
    this.department = department;
    this.salary = salary;
  }

  @Override
  public String printInfo() {
    return "Lecturer of " + this.getDepartment() + " department " +
this.getSurname() + " " +
           this.getName() + ", age: " + this.getAge() +
           ". Salary: " + this.getSalary();
  }

  @Override
  public void setSurname(String surname) {
    this.surname = surname;
  }

  @Override
  public void setName(String name) {
    this.name = name;
  }

  @Override
  public void setAge(int age) {
    this.age = age;
  }

  @Override
  public String getSurname() {
    return this.surname;
  }

  @Override
  public String getName() {
    return this.name;
  }

  @Override
  public int getAge() {
    return this.age;
  }

  public String getDepartment() {
    return department;
  }

  public void setDepartment(String department) {
    this.department = department;
  }

  public double getSalary() {
    return salary;
  }
```

```
  public void setSalary(double salary) {
    this.salary = salary;
  }
}
```

Код класу Main у обох випадках не було змінено:

```
package AD181.Gezha;

public class Main {
  public static void main(String[] args) {
    Student student1 = new Student("Ross", "Bob", 18, "BS-181", "11037");
    Student student2 = new Student("Cheese", "William", 20, "AA-113", "1498");
    Lecturer lecturer = new Lecturer("Smith", "John", 42, "physics", 2500.0);

    Person[] humans = {student1, student2, lecturer};
    for (Person human : humans) {
      System.out.println(human.printInfo());
    }
  }
}
```

Обидві версії програми були запущені та обидві вивела правильний результат (Рис. 1).

```
Student of group BS-181 Ross Bob, age: 18. Student ID: 11037
Student of group AA-113 Cheese William, age: 20. Student ID: 1498
Lecturer of physics department Smith John, age: 42. Salary: 2500.0
```

Рис. 1: результат роботи програми

Завдання 2

Завдання було подано у наступній формі (Рис. 2):

5. Необходимо описать интерфейс, содержащий одну функцию:

   ```
   int fold(int[] arr);
   ```

   Данный интерфейс является аккумулятором и позволяет собирать агрегирующие сведения о последовательности чисел. Необходимо реализовать два класса для этого интерфейса. Первый класс должен возвращать количество четных чисел в массиве. Второй класс должен возвращать количество нечетных чисел в массиве. Необходимо, чтобы приложение запросило у пользователя размер массива, элементы массива и выдало результаты вычисления для обоих классов.

Рис. 2: текст завдання

Код у класі CounterInterface:

```
package AD161.Gezha;

public interface CounterInterface {
    int fold(int[] arr);
}
```

Код у класі EvenNumbersCounter:

```
package AD161.Gezha;

public class EvenNumbersCounter implements CounterInterface {

    @Override
    public int fold(int[] arr) {
        int evenAmount = 0;
        for (int number: arr) {
            if (number % 2 == 0) {
                ++evenAmount;
            }
        }
        return evenAmount;
    }
}
```

Код у класі OddNumbersCounter:

```
package AD161.Gezha;

public class OddNumbersCounter implements CounterInterface {

    @Override
    public int fold(int[] arr) {
        int oddAmount = 0;
        for (int number: arr) {
            if (number % 2 == 1) {
                ++oddAmount;
```

```
            }
        }
        return oddAmount;
    }
}
```

Код у класі Main:

```
package AD161.Gezha;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int arraySize = input.nextInt();

        int[] array = new int[arraySize];
        for (int i = 0; i < arraySize; ++i) {
            System.out.print("Input array element #" + (i + 1) + ": ");
            array[i] = input.nextInt();
        }

        CounterInterface evenCounter = new EvenNumbersCounter();
        CounterInterface oddCounter = new OddNumbersCounter();

        System.out.println("Even numbers amount: " + evenCounter.fold(array));
        System.out.println("Odd numbers amount: " + oddCounter.fold(array));
    }
}
```

Програма була запущена 3 рази із різним вводом від користувача, вона вивела правильний результат у усіх 3х запусках (Рис. 3, Рис. 4, Рис. 5).

```
Enter array size: 5
Input array element #1: 123
Input array element #2: 321
Input array element #3: 789
Input array element #4: 987
Input array element #5: 101
Even numbers amount: 0
Odd numbers amount: 5
```

Рис. 3: запуск программы 1

```
Enter array size: 3
Input array element #1: 42
Input array element #2: 11037
Input array element #3: 2345
Even numbers amount: 1
Odd numbers amount: 2
```

Рис. 4: запуск программы 2

```
Enter array size: 1
Input array element #1: 4
Even numbers amount: 1
Odd numbers amount: 0
```

Рис. 5: запуск программы 3

ВИВІД

Було проведено ознайомлення та практичне використання абстрактних класів та інтерфейсів, ознайомлення з зворотніми викликами.

Посилання на github репозиторій: https://github.com/onpu-ad181ng/gezha-oop-lab6

СПИСОК ЛІТЕРАТУРИ

1. Форум StackOverflow
2. Інтернет-ресурс GeeksForGeeks
3. Інструкція щодо лабораторної роботи.