**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине «Технологии разработки программного обеспечения»

на тему: «Развитие представлений о разработке программ. Объектно-ориентированное программирование в Java»

Выполнил: студент гр. ИТП-21

Трацевский И. С.

Принял: преподаватель

Малиновский И. Л.

Гомель 2023

**Вариант:** 19.

**Задания:**

1.1 Создать иерархию классов для учёта выплаты стипендии студентам по итогам экзаменационной сессии.

1.2 Только студенты, обучающиеся на бюджетной форме и сдавшие сессию в срок, получают стипендию.

1.3 Студент, имеющий средний бал меньше 5 стипендию не получает, от 6 до 7 получают минимальную стипендию, от 6 до 8 – увеличенную на 25%, от 8 до 10 – на 50%.

1.4 Ввести данные с информацией о 10 студентах, которые получили стипендию, увеличенную на 25%.

1.5 Вывести студентов, обучающихся на платной основе.

1.6 Вывести средний балл по итогам сессии по всем студентам.

**Ход работы:**

Перед разработкой самой программы согласно заданию: необходимо разработать *UML-*диаграмму иерархии классов будущей программы. Исходя из условия задания можно выделить три основных класса: абстрактный класс студент, который будет содержать общую информацию о студенте, класс студента на стипендии, который в дополнении к общей информации будет содержать данные об оценках и срок сдачи лабораторных работ и класс студента-платника, содержащего информацию о цене обучения. На рисунке 1 изображена требуемая *UML-*диаграмма.

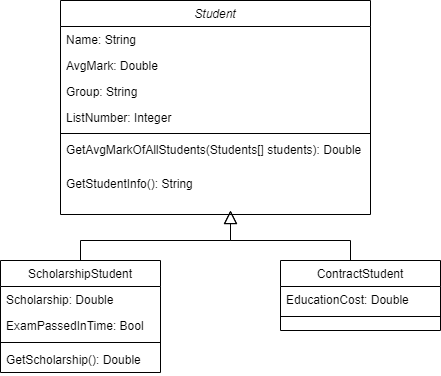


Рисунок 1 – *UML*-диаграмма классов программы

После разработки диаграммы можно приступить к написанию кода приложения. Каждый класс будет размещаться в отдельном файле.

Для реализации возможности вычисления размера стипендии в классе *ScholarshipStudent* был создан метод устанавливающий значение стипендии в зависимости от средней оценки и сдачи лабораторных работ. На рисунках 2 и 3 продемонстрирована работа программы по подсчету среднего балла всех студентов и получении информации о студентах-платниках.

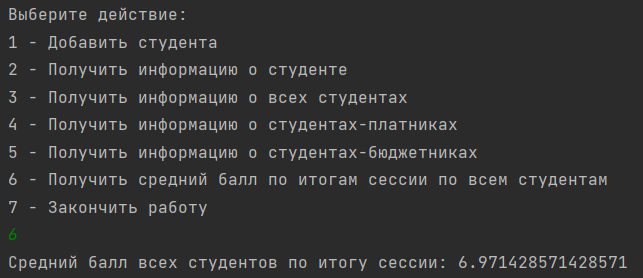


Рисунок 2 – Подсчет среднего балла студентов

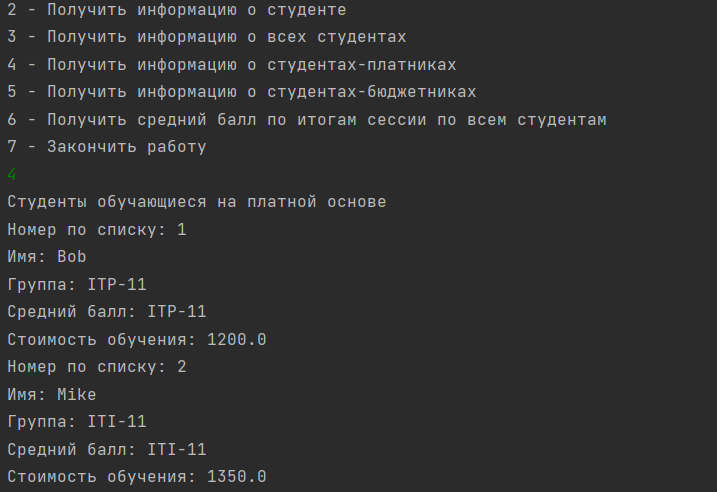


Рисунок 3 – Получение информации о студентах-платниках

Листинг разработанной программы представлен в приложении А.

**Вывод:** в результате выполнения лабораторной работы была разработана программа для управления данными о студентах в период сдачи сессии. Программа позволяет получать информацию о разных группах студентов и отслеживать наличие или отсутствие стипендии у студентов-бюджетников.

**ПРИЛОЖЕНИЕ А**

(обязательное)

Листинг классов программы

public abstract class Student {  
 protected String name;  
 protected String group;  
 protected double avgMark;  
 protected int listNumber;  
 public Student(String name, String group, Double avgMark, int listNumber){  
 this.name = name;  
 this.group = group;  
 this.avgMark = avgMark;  
 this.listNumber = listNumber;  
 }  
 public static Double getAvgMarkOfAllStudents(Student[] students){  
 double sumOfMarks = 0.0;  
 for(int i = 0; i< students.length;i++){  
 sumOfMarks += students[i].getAvgMark();  
 }  
 return sumOfMarks/students.length;  
 }  
 public Double getAvgMark() {return avgMark;}  
 public String getName(){return name;}  
 public String getGroup(){return group;}  
 public int getListNumber(){return listNumber;}  
 public abstract String getStudentInfo();  
}

public class ScholarshipStudent extends Student{  
 private double scholarShip = 100;  
 private boolean examPassedInTime;  
 public ScholarshipStudent(String name, String group, Double avgMark,int listNumber,Boolean examPassedInTime){  
 super(name,group,avgMark,listNumber);  
 this.examPassedInTime = examPassedInTime;  
 }  
 public Double getScholarShip() {  
 if (examPassedInTime) {  
 double avgMark = getAvgMark();  
 if (avgMark < 5) {  
 scholarShip = 0;  
 } else if (avgMark >= 5 && avgMark < 6) {  
 scholarShip = scholarShip \* 1;  
 } else if (avgMark >= 6 && avgMark < 8) {  
 scholarShip = scholarShip \* 1.25;  
 } else if (avgMark >= 8 && avgMark <= 10) {  
 scholarShip = scholarShip \* 1.5;  
 }  
 } else {  
 scholarShip = 0;  
 }  
 return scholarShip;  
 }t  
 public String getStudentInfo(){  
 return "Номер по списку: " + listNumber + "\nИмя: " + name +  
 "\nГруппа: " + group + "\nСредний балл: " + avgMark + "\n" +  
 (examPassedInTime ? "Сессия сдана вовремя" : "Сессия сдана не вовремя") +  
 "\nСтипендия: " + getScholarShip();  
 }  
}

public class ContractStudent extends Student{  
 private double educationCost;  
 public ContractStudent(String name, String group, Double avgMark,int listNumber,double educationCost){  
 super(name,group,avgMark,listNumber);  
 this.educationCost = educationCost;  
 }  
 public String getStudentInfo(){  
 return "Номер по списку: " + listNumber + "\nИмя: " + name + "\nГруппа: " + group + "\nСредний балл: " + group + "\nСтоимость обучения: " + educationCost;  
 }  
}

Листинг основной программы

import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Boolean isExit = false;  
 Student[] students = new Student[0];  
 Student[] studentsExample = {  
 new ContractStudent("Bob","ITP-11",6.1,1,1200),  
 new ContractStudent("Mike","ITI-11",9.1,2,1350),  
 new ScholarshipStudent("Jim","ITP-21",6.1,3,true),  
 new ScholarshipStudent("Bob","ITP-11",5.0,4,true),  
 new ScholarshipStudent("David","IPT-11",9.4,5,true),  
 new ScholarshipStudent("Jordan","IPT-11",3.7,6,true),  
 new ScholarshipStudent("Jim","IPT-11",9.4,5,false)  
 };  
 while(!isExit){  
 System.out.println("Выберите действие:");  
 System.out.println("1 - Добавить студента");  
 System.out.println("2 - Получить информацию о студенте");  
 System.out.println("3 - Получить информацию о всех студентах");  
 System.out.println("4 - Получить информацию о студентах-платниках");  
 System.out.println("5 - Получить информацию о студентах-бюджетниках");  
 System.out.println("6 - Получить средний балл по итогам сессии по всем студентам");  
 System.out.println("7 - Закончить работу");  
 Scanner userChoiseScaner = new Scanner(System.in);  
 if (userChoiseScaner.hasNextInt()) {  
 int userChoise = userChoiseScaner.nextInt();  
 switch(userChoise){  
 case (1) :  
 Scanner studentAddInput = new Scanner(System.in);  
 System.out.println("Введите имя студента");  
 String studentName = studentAddInput.nextLine();  
 System.out.println("Введите группу студента");  
 String studentGroup = studentAddInput.nextLine();  
 System.out.println("Введите средний балл студента");  
 Double studentAvgMark = studentAddInput.nextDouble();  
 System.out.println("Введите тип студента (1 - на бюджете/2 - платный)");  
 int studentType = studentAddInput.nextInt();  
 Student[] array = Arrays.copyOf(students, students.length + 1);  
 if(studentType == 1){  
 System.out.println("Студент сдал сессию в срок? (1 - да, 2 - нет)");  
 int examPassTimeNumber = studentAddInput.nextInt();  
 Boolean examPassConfirm;  
 if(examPassTimeNumber == 1){  
 examPassConfirm = true;  
 } else{  
 examPassConfirm = false;  
 }  
 array[students.length] = new ScholarshipStudent(studentName,studentGroup,studentAvgMark,array.length,examPassConfirm);  
 students = array;  
 }else if(studentType == 2){  
 System.out.println("Введите стоимость оплаты за обучение");  
 Double studentEduCost = studentAddInput.nextDouble();  
 array[students.length] = new ContractStudent(studentName,studentGroup,studentAvgMark,array.length,studentEduCost);  
 students = array;  
 }  
 break;  
 case 2:  
 Scanner studentSearchInput = new Scanner(System.in);  
 System.out.println("Введите номер студента по списку");  
 int studentListNumber = studentSearchInput.nextInt()-1;  
 System.out.println(students[studentListNumber].getStudentInfo());  
 break;  
 case 3:  
 for (Student student: studentsExample) {  
 System.out.println(student.getStudentInfo());  
 }  
 break;  
 case 4:  
 System.out.println("Студенты обучающиеся на платной основе");  
 for (Student student: studentsExample) {  
 if(student instanceof ContractStudent){  
 System.out.println(student.getStudentInfo());  
 }  
 }  
 break;  
 case 5:  
 System.out.println("Студенты обучающиеся на бюджете");  
 for (Student student: studentsExample) {  
 if(student instanceof ScholarshipStudent){  
 System.out.println(student.getStudentInfo());  
 }  
 }  
 break;  
 case 6:  
 System.out.println("Средний балл всех студентов по итогу сессии: " + Student.getAvgMarkOfAllStudents(studentsExample));  
 break;  
 case 7:  
 isExit = true;  
 break;  
 }  
 } else {  
 System.out.println("Извините, но это явно не число.");  
 }  
 }  
 }  
}