

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
Кафедра ИИТ

Отчет по лабораторной работе №4

Выполнила:  
Студентка группы  
АС-50  
Дряпко А. В.  
Проверил:  
Крощенко А.А.

Брест 2020

#### Вариант 4

**Задание 1.** Реализовать указанный класс, включив в него вспомогательный внутренний класс или классы. Реализовать 2-3 метода (на выбор). Продемонстрировать использование реализованных классов.

4) Создать класс Зачетная Книжка с внутренним классом, с помощью объектов которого можно

**хранить информацию о сессиях, зачетах, экзаменах.**

**Код программы:**

**Class RecordBook**

```
import java.util.ArrayList;
```

```
public class RecordBook {

    public ArrayList<Session> sessionsList = new ArrayList<Session>();

    public void showRecordBook() {
        for(Session item : sessionsList) {
            System.out.println("\nСессия " + item.sessionId);
            for(Session.SessionField item2 : item.sessionFieldsList) {
                System.out.print(item2.fieldType+"\n");
                System.out.print("Номер предмета: "+item2.fieldId+"\t");
                System.out.print("Предмета: "+item2.fieldSubject+"\t");
                System.out.print("Оценка: "+item2.fieldMark+"\n");
                System.out.print("Преподаватель: "+item2.lecturer+"\n");
            }
        }
    }

    public class Session {

        public int sessionId;
        public ArrayList<SessionField> sessionFieldsList = new
ArrayList<SessionField>();

        public Session(int id, ArrayList<SessionField> testList) {
            this.sessionId = id;
            this.sessionFieldsList = testList;
        }

        public Session() {
        }

        public class SessionField {

            public int fieldId;
            public String fieldSubject;
            public int fieldMark;
            public String fieldType;
            public String lecturer;
        }
    }
}
```

```

        public SessionField(int fieldId, String fieldSubject, int
fieldMark, String fieldType, String lecturer) {
            this.fieldId = fieldId;
            this.fieldSubject = fieldSubject;
            this.fieldMark = fieldMark;
            this.fieldType = fieldType;
            this.lecturer=lecturer;
        }
    }
}

```

### **Class Main**

```

import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {
        RecordBook recordBook = new RecordBook();
        ArrayList<RecordBook.Session.SessionField> data = new
ArrayList<RecordBook.Session.SessionField>();
        data.add(new RecordBook().new Session().new
SessionField(1,"БиБД",9,"Экзамен"
, "Буров А.А"));
        data.add(new RecordBook().new Session().new
SessionField(2,"ГимС",7,"Зачет",
"Студнева Г.И.));
        recordBook.sessionsList.add(new RecordBook().new Session(1,data));
        data.clear();
        data.add(new RecordBook().new Session().new
SessionField(1,"МС",10,"Экзамен",
"Маргулис И.Н.));
        data.add(new RecordBook().new Session().new
SessionField(3,"СПО",9,"Зачет",
"Климова Л.С.));
        recordBook.sessionsList.add(new RecordBook().new Session(2,data));
        recordBook.showRecordBook();
    }
}

```

```
Сессия 1
Экзамен
Номер предмета: 1    Предмета: МС    Оценка: 10
Преподаватель: Маргулис И.Н.
Зачет
Номер предмета: 3    Предмета: СПО    Оценка: 9
Преподаватель: Климова Л.С.
```

```
Сессия 2
Экзамен
Номер предмета: 1    Предмета: МС    Оценка: 10
Преподаватель: Маргулис И.Н.
Зачет
Номер предмета: 3    Предмета: СПО    Оценка: 9
Преподаватель: Климова Л.С.
```

**Задание 2.** Реализовать агрегирование. При создании класса агрегируемый класс объявляется как атрибут (локальная переменная, параметр метода). Включить в каждый класс 2-3 метода на выбор. Продемонстрировать использование разработанных классов.

4) Создать класс Текст, используя классы Страница, Слово.

**Код программы:**

```
import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {
        Text text = new Text();
        text.addPage(1, "Живи", "и", "радуйся");
        text.addPage(2, "Не", "надо", "бояться", "густого", "тумана");
        text.showText();
    }

    public static class Text {
        //Поля и методы класса Текст
        public ArrayList<Page> text = new ArrayList<Page>();

        public void addPage(int id, String...words) {
            Page temp = new Page(id);
            temp.addWords(words);
            text.add(temp);
        }

        public void showText() {
            for(Page item : text) {
                System.out.println("\nСтраница №"+item.pageNumber);
                for(Page.Word item2 : item.page) {
                    System.out.print(item2.word+" ");
                }
            }
        }
    }
}
```

```

    }

    public class Page {
        public int pageNumber;
        public ArrayList<Word> page = new ArrayList<Word>();

        public Page(int pageNumber) {
            this.pageNumber = pageNumber;
        }

        public void addWords(String...words){
            for (int i = 0; i < words.length; i++) {
                page.add(new Word(words[i]));
            }
        }

        public class Word {
            public String word;

            public Word(String word) {
                this.word = word;
            }
        }
    }
}

```

```

Страница №1
Живи и радуйся
Страница №2
Не надо бояться густого тумана
Process finished with exit code 0

```

**Задание 3.** Построить модель программной системы с применением отношений (обобщения, агрегации, ассоциации, реализации) между классами. Задать атрибуты и методы классов. Реализовать (если необходимо) дополнительные классы.

Продемонстрировать работу разработанной системы.

4) Система Вступительные экзамены. Абитуриент регистрируется на Факультет, сдает Экзамены. Преподаватель выставляет Оценку. Система подсчитывает средний балл и определяет Абитуриентов, зачисленных в учебное заведение.

**Код программы:**

**Class Faculty**

```
package com.company.Members;
```

```
import com.company.Exams.Exam;
```

```
import com.company.Faculty;
```

```

public class Abiturient extends Man {
    private Faculty faculty;

    public void setFaculty(Faculty f) {
        faculty = f;
    }
    public Faculty getFaculty() {
        return faculty;
    }
    public void passExam(Exam exam) {
        exam.addPassedAbiturent(this);
    }

    public Abiturient(int aId, String aName, String aSurname ) {
        name = aName;
        surname = aSurname;
        id = aId;
    }
}

```

### **Class Main**

```

package com.company;
import com.company.Exams.Exam;
import com.company.Members.Abiturient;
import com.company.Members.Lecturer;

import java.util.*;
import java.util.stream.Collectors;

public class Main {

    public static void main(String[] args) {
        ArrayList<Faculty> listOfFaculties = new ArrayList<Faculty>();
        ArrayList<Exam> listOfExams = new ArrayList<Exam>();
        ArrayList<Lecturer> listOfLecturers = new ArrayList<Lecturer>();
        ArrayList<Abiturient> listOfAbiturients = new
ArrayList<Abiturient>();

        int ids[] = {0,0,0,0}; //ID: Абитуриент, Преподаватель, Факультет,
Экзамен
        Abiturient ab1 = new Abiturient(++ids[0], "Никита", "Лещёв");
        Abiturient ab2 = new Abiturient(++ids[0], "Олег", "Бочонский");
        Abiturient ab3 = new Abiturient(++ids[0], "Екатерина", "Богрова");
        Abiturient ab4 = new Abiturient(++ids[0], "Елисей", "Щавельский");
        Abiturient ab5 = new Abiturient(++ids[0], "Мария", "Кушкина");
        Abiturient ab6 = new Abiturient(++ids[0], "Яков", "Крупин");
        Abiturient ab7 = new Abiturient(++ids[0], "Владлен", "Дешевых");
        Abiturient ab8 = new Abiturient(++ids[0], "Харитон", "Вятт");
        Abiturient ab9 = new Abiturient(++ids[0], "Ника", "Мухоморова");
        Lecturer Kroschenko = new
Lecturer(++ids[1], "Александр", "Крощенко", "Д1");
    }
}

```

```

    Lecturer Golovko = new
Lecturer(++ids[1], "Владимир", "Головко", "Д2");
    Lecturer Bezobrazov = new
Lecturer(++ids[1], "Сергей", "Безобразов", "Д1");
    Lecturer Hveschuk = new
Lecturer(++ids[1], "Владимир", "Хвещук", "Д2");

    Faculty feis = new Faculty(++ids[2], "ФЭИС");
    Faculty ef = new Faculty(++ids[2], "ЭФ");

    Exam Math = new Exam(++ids[3], "Математика");
    Exam Phys = new Exam(++ids[3], "Физика");
    Exam Eng = new Exam(++ids[3], "Английский язык");
    Exam Rus = new Exam(++ids[3], "Русский язык");

    Math.setExaminator(Kroschenko);
    Phys.setExaminator(Golovko);
    Eng.setExaminator(Bezobrazov);
    Rus.setExaminator(Hveschuk);

    feis.setFacultyExam(Math);
    feis.setFacultyExam(Phys);
    feis.showFacultyExams();

    feis.regAbiturient(ab1);
    feis.regAbiturient(ab2);
    feis.regAbiturient(ab3);
    feis.regAbiturient(ab4);
    feis.regAbiturient(ab5);
    feis.showAbiturents();

    ab1.passExam(Math);
    ab2.passExam(Math);
    ab3.passExam(Math);
    ab4.passExam(Math);
    ab5.passExam(Math);
    ab1.passExam(Phys);
    ab2.passExam(Phys);
    ab3.passExam(Phys);
    ab4.passExam(Phys);
    ab5.passExam(Phys);

    Kroschenko.setMark(Math, ab1, 7);
    Kroschenko.setMark(Math, ab2, 3);
    Kroschenko.setMark(Math, ab3, 2);
    Kroschenko.setMark(Math, ab4, 10);
    Kroschenko.setMark(Math, ab5, 5);
    Golovko.setMark(Phys, ab1, 7);
    Golovko.setMark(Phys, ab2, 10);
    Golovko.setMark(Phys, ab3, 6);
    Golovko.setMark(Phys, ab4, 3);
    Golovko.setMark(Phys, ab5, 6);

```

```

        ef.setFacultyExam(Math);
        ef.setFacultyExam(Eng);
        ef.showFacultyExams();

        ef.regAbiturient(ab6);
        ef.regAbiturient(ab7);
        ef.regAbiturient(ab8);
        ef.regAbiturient(ab9);
        ef.showAbiturents();

        ab6.passExam(Math);
        ab7.passExam(Math);
        ab8.passExam(Math);
        ab9.passExam(Math);
        ab6.passExam(Eng);
        ab7.passExam(Eng);
        ab8.passExam(Eng);
        ab9.passExam(Eng);

        Kroschenko.setMark(Math, ab6, 7);
        Kroschenko.setMark(Math, ab7, 3);
        Kroschenko.setMark(Math, ab8, 2);
        Kroschenko.setMark(Math, ab9, 10);
        Bezobrazov.setMark(Eng, ab6, 7);
        Bezobrazov.setMark(Eng, ab7, 10);
        Bezobrazov.setMark(Eng, ab8, 6);
        Bezobrazov.setMark(Eng, ab9, 3);

        Math.showPassedAbiturients();
        Phys.showPassedAbiturients();
        Eng.showPassedAbiturients();
        getResults(feis);
        getResults(ef);
    }

    public static void getResults(Faculty f) {
        System.out.println("\033[34;1;24m"+"#\t Результаты вступительных  

        экзаменов по факультету : "+f.facultyName+"\033[0m");
        ArrayList<Abiturient> tempAbiturientLiST = f.getAbiturients();
        ArrayList<Exam> tempExamsList = f.getExamsList();
        HashMap<Abiturient, Double> newStudents = new HashMap<>();
        for(Abiturient item : tempAbiturientLiST) {
            double tempMark = 0;
            for(Exam item2 : tempExamsList) {
                tempMark +=
                Integer.parseInt(item2.getPassedAbiturientMark(item));
            }
            tempMark = tempMark/tempExamsList.size();
            newStudents.put(item, tempMark);
        }
    }

```



```

        for(Map.Entry<Abiturient,Double> item : newStudents.entrySet()) {
            System.out.println("\033[35;1;24m"+" \t "+item.getKey().name+"
"+item.getValue()+"\033[0m");
        }
    }
}

```

### **Class Abiturient**

```
package com.company.Members;
```

```
import com.company.Exams.Exam;
```

```
import com.company.Faculty;
```

```
public class Abiturient extends Man {
    private Faculty faculty;
```

```

    public void setFaculty(Faculty f) {
        faculty = f;
    }

```

```

    public Faculty getFaculty() {
        return faculty;
    }

```

```

    public void passExam(Exam exam) {
        exam.addPassedAbiturent(this);
    }

```

```

    public Abiturient(int aId, String aName, String aSurname ) {
        name = aName;
        surname = aSurname;
        id = aId;
    }
}

```

```
}
```

### **Class Lecturer**

```
package com.company.Members;
```

```
import com.company.Exams.Exam;
```

```
public class Lecturer extends Man {
    private String position;
```

```

    public void setMark(Exam exam, Abiturient abiturient, int mark) {
        exam.setMark(this,abiturient,mark);
    }

```

```

    public Lecturer(int lId, String lName, String lSurname,String lPosition
) {
        name = lName;
        surname = lSurname;
        id = lId;
        position = lPosition;
    }
}

```

```
}
```

### **Class Man**

```

package com.company.Members;

public class Man {
    public String name;
    public String surname;
    public int id;
}

Class Exam
package com.company.Exams;

import com.company.Members.Abiturient;
import com.company.Members.Lecturer;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class Exam {
    private int id;
    public String examName;
    private Lecturer Examiner;
    private HashMap<Abiturient,String> PassedAbiturients = new HashMap<>();

    public String getPassedAbiturientMark(Abiturient abiturient) {
        if(PassedAbiturients.containsKey(abiturient)) {
            return PassedAbiturients.get(abiturient);
        } else {
            System.out.println("#\t Данный абитуриент не сдавал экзамен");
            return "null";
        }
    }

    public void showPassedAbiturients() {
        System.out.println("\033[31;1;24m"+"#\t Список сдавших экзамен по
предмету : "+this.examName+"\033[0m");
        System.out.println("#\t Всего сдало :
"+this.PassedAbiturients.size());
        for(Map.Entry<Abiturient,String> item :
PassedAbiturients.entrySet()) {
            System.out.println("\t "+item.getKey().name+"
"+item.getKey().surname
            +"\n\t\tОценка : "+item.getValue());
        }
    }

    public void setMark(Lecturer lecturer, Abiturient abiturient, int mark)
    {
        if(this.Examiner == lecturer) {
            if(this.PassedAbiturients.containsKey(abiturient)) {
this.PassedAbiturients.put(abiturient,String.valueOf(mark));
            } else {
                System.out.println("#\t ! Этот абитуриент не сдавал
экзамен");
            }
        } else {
    
```

```

        System.out.println("#\t ! Вы не являетесь экзаменатором этого
предмета");
    }
}
public void addPassedAbiturent(Abiturient abiturient) {
    PassedAbiturients.put(abiturient,"Не выставлено");
}
public void setExaminator(Lecturer lecturer) {
    Examinator = lecturer;
}
public Lecturer getExaminator() {
    return Examinator;
}

//Конструктор класса Экзамен
public Exam(int id, String examName) {
    this.id = id;
    this.examName = examName;
}
}

```

```

# Список экзаменов для поступления на факультет : ФЗИС
Математика Экзаменатор : Александр Крощенко
Физика Экзаменатор : Владимир Головкин
# Список абитуриентов факультета ФЗИС
Николай Серый
Олег Сутулый
Екатерина Щербакова
Марат Вялый
Мария Красивая
# Список экзаменов для поступления на факультет : ЭФ
Математика Экзаменатор : Александр Крощенко
Английский язык Экзаменатор : Сергей Безобразов
# Список абитуриентов факультета ЭФ
Александр Смелый
Евгений Душный
Стелла Томная
Вероника Лесная

```

```

# Список сдавших экзамен по предмету : Английский язык
# Всего сдало : 4
Стелла Томная
Оценка : 6
Евгений Душный
Оценка : 10
Александр Смелый
Оценка : 7
Вероника Лесная
Оценка : 3
# Результаты вступительных экзаменов по факультету : ФЗИС
Марат 6.5
Мария 5.5
Олег 6.5
Николай 7.0
Екатерина 4.0
# Результаты вступительных экзаменов по факультету : ЭФ
Стелла 4.0
Евгений 6.5
Александр 7.0
Вероника 6.5

```

```

# Список сдавших экзамен по предмету : Математика
# Всего сдало : 9
Стелла Томная
Оценка : 2
Евгений Душный
Оценка : 3
Марат Вялый
Оценка : 10
Александр Смелый
Оценка : 7
Мария Красивая
Оценка : 5
Олег Сутулый
Оценка : 3
Николай Серый
Оценка : 7
Екатерина Щербакова
Оценка : 2
Вероника Лесная
Оценка : 10
# Список сдавших экзамен по предмету : Физика
# Всего сдало : 5
Марат Вялый
Оценка : 3
Мария Красивая
Оценка : 6
Олег Сутулый
Оценка : 10
Николай Серый
Оценка : 7
Екатерина Щербакова
Оценка : 6

```