

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ОТЧЁТ

по лабораторной работе №4

Выполнил:
студент группы ПО-9
Дарашкевич Д.И.

Проверил:
Крощенко А.А.

Брест 2024

Цель работы: приобрести практические навыки в области объектно-ориентированного проектирования.

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:

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

public class ZauchetnayaKnizhka {
    private String studentName;
    private int studentId;
    private List<Session> sessions;

    public ZauchetnayaKnizhka(String studentName, int studentId) {
        this.studentName = studentName;
        this.studentId = studentId;
        this.sessions = new ArrayList<>();
    }

    private class Session {
        private String semester;
        private String subjects;
        private boolean passed;

        public Session(String semester, String subjects, boolean passed) {
            this.semester = semester;
            this.subjects = subjects;
            this.passed = passed;
        }

        public String getSessionInfo() {
            return "Семестр: " + semester + ", Предметы: " + subjects + ", Сдано: " + (passed
? "Да" : "Нет");
        }
    }

    public void addSession(String semester, String subjects, boolean passed) {
        Session session = new Session(semester, subjects, passed);
        sessions.add(session);
    }

    public void displaySessions() {
        System.out.println("Информация о сессиях студента " + studentName + ":");
        for (int i = 0; i < sessions.size(); i++) {
            System.out.println("Сессия " + (i + 1) + ": " + sessions.get(i).getSessionInfo());
        }
    }

    public int getSessionCount() {
        return sessions.size();
    }

    public static void main(String[] args) {
        ZauchetnayaKnizhka studentRecord = new ZauchetnayaKnizhka("Иванов Иван", 123456);
        studentRecord.addSession("Весенний 2023", "Математика, Физика, Информатика", true);
        studentRecord.addSession("Осенний 2023", "Физика, Химия, Биология", false);
    }
}
```

```

        studentRecord.addSession("Весенний 2024", "История, Литература, Обществознание",
true);

        studentRecord.displaySessions();
        System.out.println("Количество сессий: " + studentRecord.getSessionCount());
    }
}

```

Входные данные:

```

ZauchetnayaKnizhka studentRecord = new ZauchetnayaKnizhka("Иванов Иван", 123456);
studentRecord.addSession("Весенний 2023", "Математика, Физика, Информатика", true);
studentRecord.addSession("Осенний 2023", "Физика, Химия, Биология", false);
studentRecord.addSession("Весенний 2024", "История, Литература, Обществознание",
true);

```

Результат работы программы:

```

C:\Users\Legion\jdk\openjdk-22.0.1\bin\java.exe "-javaagent:D:\IntelliJ IDEA 2024.1.1\lib\idea_rt.j
-classpath "C:\Users\Legion\Desktop\6 семестр\СПП\lab4\1\out\production\1" ZauchetnayaKnizhka
Информация о сессиях студента Иванов Иван:
Сессия 1: Семестр: Весенний 2023, Предметы: Математика, Физика, Информатика, Сдано: Да
Сессия 2: Семестр: Осенний 2023, Предметы: Физика, Химия, Биология, Сдано: Нет
Сессия 3: Семестр: Весенний 2024, Предметы: История, Литература, Обществознание, Сдано: Да
Количество сессий: 3

Process finished with exit code 0

```

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:

Word

```

public class Word {
    private String word;

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

    public String getWord() {
        return word;
    }
}

```

Page

```

import java.util.ArrayList;
import java.util.List;

public class Page {
    private List<Word> words;

    public Page() {
        this.words = new ArrayList<>();
    }

    public void addWord(Word word) {
        words.add(word);
    }

    public List<Word> getWords() {
        return words;
    }
}

```

```
}
```

Text

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

```
public class Text {
    private List<Page> pages;

    public Text() {
        this.pages = new ArrayList<>();
    }

    public void addPage(Page page) {
        pages.add(page);
    }

    public void addWordToLastPage(Word word) {
        if (!pages.isEmpty()) {
            Page lastPage = pages.get(pages.size() - 1);
            lastPage.addWord(word);
        } else {
            Page newPage = new Page();
            newPage.addWord(word);
            pages.add(newPage);
        }
    }

    public String getText() {
        StringBuilder textBuilder = new StringBuilder();
        for (Page page : pages) {
            List<Word> words = page.getWords();
            for (Word word : words) {
                textBuilder.append(word.getWord()).append(" ");
            }
            textBuilder.append("\n");
        }
        return textBuilder.toString();
    }
}
```

```
}
```

Main

```
public class Main {
    public static void main(String[] args) {
        Text text = new Text();

        Word hello = new Word("Hello");
        Word world = new Word("world");
        Word java = new Word("Java");

        Page page1 = new Page();
        page1.addWord(hello);
        page1.addWord(world);
        text.addPage(page1);

        Page page2 = new Page();
        page2.addWord(java);
        text.addPage(page2);

        System.out.println(text.getText());
    }
}
```

```
}
```

Входные данные:

```
Word hello = new Word("Hello");
```

```
Word world = new Word("world");
Word java = new Word("Java");
```

Результат работы программы:

```
C:\Users\Legion\.jdk\openjdk-22.0.1\bin\java.exe "-javaagent:D:\IntelliJ IDEA 2024
-classpath "C:\Users\Legion\Desktop\6 семестр\СПП\lab4\2\out\production\2" Main
Hello world
Java

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 заведении.

Код программы:

Applicant

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

public class Applicant {
    private String name;
    private Faculty faculty;
    private List<Exam> exams;
    private Map<Exam, Integer> grades;

    public Applicant(String name, Faculty faculty) {
        this.name = name;
        this.faculty = faculty;
        this.exams = new ArrayList<>();
        this.grades = new HashMap<>();
    }

    public void registerForExam(Exam exam) {
        exams.add(exam);
    }

    public void takeExam(Exam exam, int grade) {
        if (exams.contains(exam)) {
            grades.put(exam, grade);
        } else {
            System.out.println("Error: This applicant is not registered for the exam.");
        }
    }

    public double getAverageGrade() {
        if (grades.isEmpty()) {
            return 0;
        }
        int sum = 0;
        for (int grade : grades.values()) {
            sum += grade;
        }
    }
}
```

```

        return (double) sum / grades.size();
    }

    public String getName() {
        return name;
    }
}

```

Exam

```

import java.time.LocalDate;

public class Exam {
    private String subject;
    private LocalDate date;

    public Exam(String subject, LocalDate date) {
        this.subject = subject;
        this.date = date;
    }
}

```

Faculty

```

import java.util.ArrayList;
import java.util.List;

public class Faculty {
    private String name;
    private List<Applicant> applicants;

    public Faculty(String name) {
        this.name = name;
        this.applicants = new ArrayList<>();
    }

    public void enroll(Applicant applicant) {
        applicants.add(applicant);
    }

    public List<Applicant> getApplicants() {
        return applicants;
    }

    public String getName() {
        return name;
    }
}

```

Teacher

```

public class Teacher {
    private String name;

    public Teacher(String name) {
        this.name = name;
    }

    public void grade(Applicant applicant, Exam exam, int grade) {
        applicant.takeExam(exam, grade);
    }
}

```

EnrollmentSystem

```

import java.util.ArrayList;
import java.util.List;

```

```

public class EnrollmentSystem {
    public void enrollStudents(Faculty faculty) {
        List<Applicant> admittedStudents = new ArrayList<>();
        for (Applicant applicant : faculty.getApplicants()) {
            double averageGrade = applicant.getAverageGrade();
            if (averageGrade >= 60) {
                admittedStudents.add(applicant);
            }
        }
        System.out.println("Admitted students for " + faculty.getName() + ":");
        for (Applicant admittedStudent : admittedStudents) {
            System.out.println(admittedStudent.getName());
        }
    }
}

```

Main

```

import java.time.LocalDate;

public class Main {
    public static void main(String[] args) {
        Faculty faculty = new Faculty("Computer Science");

        Applicant applicant1 = new Applicant("John", faculty);
        Applicant applicant2 = new Applicant("Alice", faculty);

        Exam mathExam = new Exam("Math", LocalDate.of(2024, 6, 1));
        Exam englishExam = new Exam("English", LocalDate.of(2024, 6, 2));
        applicant1.registerForExam(mathExam);
        applicant1.registerForExam(englishExam);
        applicant2.registerForExam(mathExam);

        Teacher teacher = new Teacher("Dr. Smith");

        teacher.grade(applicant1, mathExam, 70);
        teacher.grade(applicant1, englishExam, 80);
        teacher.grade(applicant2, mathExam, 55);

        faculty.enroll(applicant1);
        faculty.enroll(applicant2);

        EnrollmentSystem enrollmentSystem = new EnrollmentSystem();
        enrollmentSystem.enrollStudents(faculty);
    }
}

```

Входные данные:

```

Faculty faculty = new Faculty("Computer Science");

Applicant applicant1 = new Applicant("John", faculty);
Applicant applicant2 = new Applicant("Alice", faculty);

Exam mathExam = new Exam("Math", LocalDate.of(2024, 6, 1));
Exam englishExam = new Exam("English", LocalDate.of(2024, 6, 2));
applicant1.registerForExam(mathExam);
applicant1.registerForExam(englishExam);
applicant2.registerForExam(mathExam);

Teacher teacher = new Teacher("Dr. Smith");

teacher.grade(applicant1, mathExam, 70);
teacher.grade(applicant1, englishExam, 80);
teacher.grade(applicant2, mathExam, 55);

```

Результат работы программы:

```
C:\Users\Legion\.jdk\openjdk-22.0.1\bin\java.exe "-javaagent:D:\IntelliJ IDEA 2023.3\lib\idea_rt.jar" -classpath "C:\Users\Legion\Desktop\6 семестр\СПП\lab4\3\out\production\3" Main
Admitted students for Computer Science:
John

Process finished with exit code 0
```

Вывод: в ходе выполнения данной лабораторной работы я приобрел практические навыки в области объектно-ориентированного проектирования.