

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ”
КАФЕДРА ИИТ

ОТЧЁТ
по лабораторной работе №4

Выполнил:
студент 3 курса
группы ПО-9
Мельничук В.М.
Проверил:
Крощенко А.А.

Брест 2024

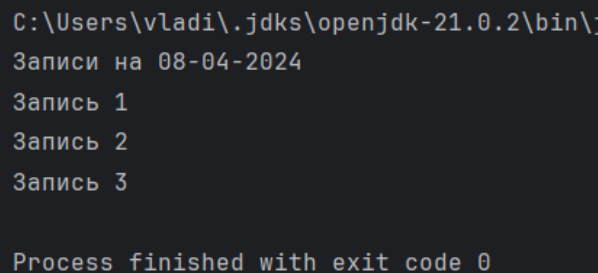
Цель работы: приобрести практические навыки в области объектно-ориентированного проектирования.

Вариант 1

Задание 1

Создать класс Notepad (записная книжка) с внутренним классом или классами, с помощью объектов которого могут храниться несколько записей на одну дату.

Выходные данные:



```
C:\Users\vladi\.jdk\openjdk-21.0.2\bin\
Записи на 08-04-2024
Запись 1
Запись 2
Запись 3

Process finished with exit code 0
```

Код программы:

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

class Notepad {
    private HashMap<String, DayRecord> records;

    public Notepad() {
        records = new HashMap<>();
    }

    public void addRecord(String date, String record) {
        if (!records.containsKey(date)) {
            records.put(date, new DayRecord());
        }
        records.get(date).add(record);
    }

    public List<String> getRecords(String date) {
        if (records.containsKey(date)) {
            return records.get(date).getRecords();
        }
        return new ArrayList<>();
    }
}

private class DayRecord {
    private ArrayList<String> records;

    public DayRecord() {
        records = new ArrayList<>();
    }

    public void add(String record) {
        records.add(record);
    }
}
```

```

        public List<String> getRecords() {
            return records;
        }
    }
}

public class task1 {
    public static void main(String[] args) {
        Notepad notepad = new Notepad();

        notepad.addRecord("08-04-2024", "Запись 1");
        notepad.addRecord("08-04-2024", "Запись 2");
        notepad.addRecord("08-04-2024", "Запись 3");
        notepad.addRecord("05-04-2024", "Запись 4");
        notepad.addRecord("05-04-2024", "Запись 5");

        List<String> records = notepad.getRecords("08-04-2024");
        System.out.println("Записи на 08-04-2024");
        for (String record : records) {
            System.out.println(record);
        }
    }
}

```

Задание 2

Создать класс Строка, используя классы Слово, Символ.

Выходные данные:

```

C:\Users\vladi\.jdk\openjdk-21.0.
ПРИВЕТ
Process finished with exit code 0

```

Код программы

```

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

class Line {
    private List<Word> words;

    public Line(){
        words = new ArrayList<>();
    }
    public void addWord(Word word){
        words.add(word);
    }
    public List<Word> getWords(){
        return words;
    }
}

class Word{
    private List<Symbol> symbols;
    public Word(){
        symbols = new ArrayList<>();
    }
    public void addSymbol(Symbol symbol){

```

```

        symbols.add(symbol);
    }
    public List<Symbol> getSymbols(){
        return symbols;
    }
}

class Symbol {
    private char symbol;

    public Symbol(char symbol){
        this.symbol = symbol;
    }

    public char getSymbol(){
        return symbol;
    }
}

class task2{
    public static void main(String[] args){
        Symbol symbol1 = new Symbol('П');
        Symbol symbol2 = new Symbol('Р');
        Symbol symbol3 = new Symbol('И');
        Symbol symbol4 = new Symbol('Б');
        Symbol symbol5 = new Symbol('Е');
        Symbol symbol6 = new Symbol('Т');

        Word word1 = new Word();
        word1.addSymbol(symbol1);
        word1.addSymbol(symbol2);

        Word word2 = new Word();
        word2.addSymbol(symbol3);
        word2.addSymbol(symbol4);
        word2.addSymbol(symbol5);
        word2.addSymbol(symbol6);

        Line line = new Line();
        line.addWord(word1);
        line.addWord(word2);

        List<Word> words = line.getWords();
        for(Word word : words){
            List<Symbol> symbols = word.getSymbols();
            for(Symbol symbol : symbols){
                System.out.print(symbol.getSymbol());
            }
        }
    }
}

```

Задание 3

Система Факультатив. Преподаватель объявляет запись на Курс. Студент записывается на Курс, обучается и по окончании Преподаватель выставляет Оценку, которая сохраняется в Архиве. Студентов, Преподавателей и Курсов при обучении может быть несколько.

Выходные данные:

```
C:\Users\vladi\.jdk\openjdk-21.0.2\bin\java.exe "-javaagent:E:\Soft
Запись на курс "Математика" объявлена преподавателем Иванченко А.В..
Запись на курс "Физика" объявлена преподавателем Василенко В.Д..
Студент MeLenkov записался на курс Математика
Студент MoLankova записался на курс Математика
Студент MeLenkov записался на курс Физика
Студент MoLankova записался на курс Физика
Обучение на факультативе Математика началось.
Обучение на факультативе Физика началось.
Обучение на факультативе Математика завершено.
Обучение на факультативе Физика завершено.
Оценка 4 выставлена MeLenkov
Оценка 2 выставлена MeLenkov
Оценка 5 выставлена MoLankova
Оценка 3 выставлена MoLankova
Студент: MeLenkov, Курс: Математика, Оценка: 4
Студент: MeLenkov, Курс: Физика, Оценка: 2
Студент: MoLankova, Курс: Математика, Оценка: 5
Студент: MoLankova, Курс: Физика, Оценка: 3

Process finished with exit code 0
```

Код программы:

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

class Elective{
    private String name;
    private Teacher teacher;
    private List<Student> students;

    public Elective(String name, Teacher teacher){
        this.name = name;
        this.teacher = teacher;
        students = new ArrayList<>();
    }

    public void registration(Student student){
        students.add(student);
    }

    public String getName(){
        return name;
    }

    public void learning(){
        System.out.println("Обучение на факультативе " + name + " началось.");
    }
}
```

```

    }

    public void finish(){
        System.out.println("Обучение на факультативе " + name + "
завершено.");
    }

}

class Teacher{
    private String name;

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

    public void advertReg(Elective elective){
        System.out.println("Запись на курс \"" + elective.getName() + "\"
объявлена преподавателем " + name + ".");
    }

    public void setMark(Student student, int mark, Elective elective){
        Archive.saveMark(student, mark, elective);
        System.out.println("Оценка " + mark + " выставлена " +
student.getName());
    }
}

class Student{
    private int id;
    private String name;

    public Student(Integer id, String name){
        this.id = id;
        this.name = name;
    }

    public void registration(Elective elective){
        elective.registration(this);
        System.out.println("Студент " + name + " записался на курс " +
elective.getName());
    }

    public String getName(){
        return name;
    }
}

class Archive{
    private static List<String> marks;

    public Archive() {

```

```

        marks = new ArrayList<>();
    }
    public static void saveMark(Student student, int mark, Elective elective){
        marks.add("Студент: " + student.getName() + ", Курс: " +
elective.getName() + ", Оценка: " + mark);
    }

    public List<String> getMarks(){
        return marks;
    }
}
public class task3{
    public static void main(String[] args){
        Teacher teacher = new Teacher("Иванченко А.В.");
        Teacher teacher1 = new Teacher("Василенко В.Д.");

        Elective elective = new Elective("Математика", teacher);
        Elective elective1 = new Elective("Физика", teacher1);
        Archive archive = new Archive();

        teacher.advertReg(elective);
        teacher1.advertReg(elective1);

        Student student1 = new Student(210663, "Melenkov");
        Student student2 = new Student(210664, "Molankova");

        student1.registration(elective);
        student2.registration(elective);

        student1.registration(elective1);
        student2.registration(elective1);

        elective.learning();
        elective1.learning();
        elective.finish();
        elective1.finish();

        teacher.setMark(student1, 4, elective);
        teacher1.setMark(student1, 2, elective1);

        teacher.setMark(student2, 5, elective);
        teacher1.setMark(student2, 3, elective1);

        List<String> marks = archive.getMarks();
        for(String mark : marks){
            System.out.println(mark);
        }
    }
}

```