МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №4

Специальность ПО-5(о)

Выполнил

Д. С. Бриштен,

студент группы ПО-5

Проверил

А. А. Крощенко

ст. преп. каф. ИИТ

Брест 2021

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

Вариант 4

Задание 1. Создать класс Зачетная Книжка с внутренним классом, с помощью объектов которого можно хранить информацию о сессиях, зачетах, экзаменах.

Выполнение:

Код программы

import java.util.Scanner;

public class \_StudentBook {

    public static Scanner ob = new Scanner(System.in);

    public info info;

    public String name;

    public \_StudentBook(){

        System.out.print("Enter the name of student: ");

        setNameStudent(ob.next());

        info = new info();

    }

    public String setNameStudent(){

        return name;

    }

    public void setNameStudent(String name){

        this.name = name;

    }

    public class info {

        int pr,st,sq;

        public info(){

            System.out.print("Session num: ");

            setPr(ob.nextInt());

            System.out.print("Remaining credits: ");

            setSt(ob.nextInt());

            System.out.print("Remaining exams: ");

            setSq(ob.nextInt());

        }

        public int getPr(){

            return pr;

        }

        public void setPr(int pr){

            this.pr=pr;

        }

        public int getSt(){

            return st;

        }

        public void setSt(int st){

            this.st=st;

        }

        public int getSq(){

            return sq;

        }

        public void setSq(int sq){

            this.sq = sq;

        }

        public void print(){

            System.out.print("The student " + setNameStudent() + " passed " +

            + getPr() + " session " + getSt() + " credits and " + getSq() + " exams");

        }

    }

    static public void main(String[]args){

        \_StudentBook ct = new \_StudentBook();

        ct.info.print();

    }

}

Спецификация ввода:

name = <имя студента>

pr = <номер сессии>

st = <оставшиеся зачёты>

sq = <оставшиеся экзамены>

Спецификация вывода:

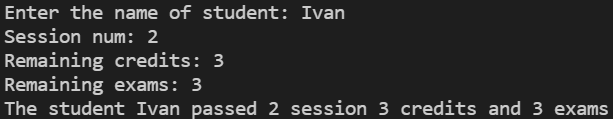
Используя метод print()

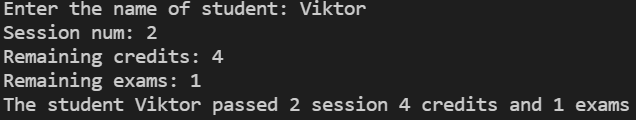
<name> + <pr> + <st> + <sq> and text

Пример:



Рисунки с результатами работы программы





Задание 2. Создать класс Текст, используя классы Страница, Слово.

Выполнение:

Код программы

public class TextRunner {

    public static void main(String[] args) {

        Word word = new Word("Java");

        Text text = new Text(word);

        Word word1 = new Word("is");

        Word word2 = new Word("the best");

        Word word3 = new Word("end");

        Page oun = new Page();

        oun.addWord(word1);

        oun.addWord(word2);

        System.out.println("Page " + oun.getNumPage() + ": " + oun.getValue());

        Page two = new Page();

        two.addWord(word3);

        System.out.println("Page " + two.getNumPage() + ": " + two.getValue());

        text.addText(oun);

        text.addText(two);

        System.out.println("Title: " + text.getTitle());

        System.out.println("Text: " + text.getText());

    }

}

class Text {

    private String title, value;

    private String text = "";

    Text(Word word) {

        title = word.getValue();

    }

    Text(Page numPage) {

        value = numPage.getValue();

    }

    String getTitle() {

        return title;

    }

    void addText(Word word) {

        text += word.getValue();

    }

    void addText(Page numPage) {

        text += numPage.getValue();

    }

    String getText() {

        return text;

    }

}

class Page {

    private String value = "";

    static int numPage = 0;

    Page() {

        numPage++;

    }

    void addWord(Word word) {

        value += " " + word.getValue();

    }

    int getNumPage(){

        return numPage;

    }

    String getValue() {

        return value;

    }

}

class Word {

    private String value;

    Word(String value) {

        this.value = value;

    }

    String getValue() {

        return value;

    }

}

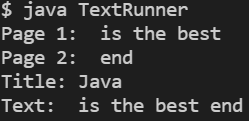
Спецификация вывода:

value – текст на данной странице

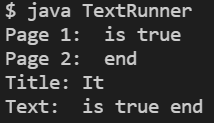
title – заголовок текста

text – полный текст

Пример:



Рисунки с результатами работы программы



Задание 3. Система Вступительные экзамены. Абитуриент регистрируется на Факультет, сдает Экзамены. Преподаватель выставляет Оценку. Система подсчитывает средний балл и определяет Абитуриентов, зачисленных в учебное заведение.

Выполнение:

Код программы

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Map;

class Abiturient implements Printable {

    private String Name, Surname;

    public Integer assessment;

    private Faculty facultet;

    private ArrayList<Exams> exams;

    private Exams exam = new Exams("Show");

    Abiturient(String name, String surname, String Facultet, ArrayList<Exams> exams) {

        this.Name = name;

        this.Surname = surname;

        this.facultet = new Faculty(name, surname, Facultet);

        this.exams = exams;

    }

    Abiturient(String name, String surname) {

        this.Name = name;

        this.Surname = surname;

    }

    String getName() { return this.Name; }

    String getSurname() { return this.Surname; }

    public void show() {

        System.out.println("\nAbiturient: " + this.Surname + " " + this.Name);

    }

    public void showResult() {

        System.out.println("Abiturient " + this.Surname + " " + this.Name +

        " of faculty " + this.facultet.getFaculty() + " passed exams " + exam.show());

    }

}

class Faculty extends Abiturient {

    private String Namefacult, name, surname;

    Faculty(String name, String surname, String namefacult) {

        super(name, surname);

        this.name = name;

        this.surname = surname;

        this.Namefacult = namefacult;

    }

    String getFaculty() { return this.Namefacult; }

    void showFacult() {

        System.out.println("Abiturient " + this.name + " "

        + this.surname + " registered with the faculty: " + this.Namefacult);

    }

}

class Exams {

    private String exam;

    static private ArrayList<Exams> exams = new ArrayList<Exams>();

    Exams(String Exam) {

        this.exam = Exam;

    }

    void setExams(Exams oun, Exams two) {

        exams.add(oun);

        exams.add(two);

    }

    ArrayList<Exams> getExams() { return exams; }

    String show() {

        return exams.get(0).exam + " and " + exams.get(1).exam;

    }

}

class Teacher extends Abiturient {

    private String name, surname;

    public int Assessment;

    Teacher(String name, String surname, Integer assessment) {

        super(name, surname);

        this.name = name;

        this.surname = surname;

        this.Assessment = assessment;

    }

    @Override

    public void show() {

        System.out.println("The teacher gave the grade " + this.Assessment

        + " for student " + this.name + " " + this.surname);

    }

}

class Assessment {

    private int max = 10;

    private int min = 0;

    Integer getAssessment() { return (int)(Math.random()\*((max-min)+1))+min; }

}

interface Printable {

    void show();

}

public class EntranceExam {

    public static void main(String[] args) {

        HashMap<String, Integer> studentsResult = new HashMap<String, Integer>();

        Abiturient Putin = new Abiturient("Vladimir", "Putin");

        Putin.show();

        String Facultet = "FEIS";

        Exams math = new Exams("math");

        Exams history = new Exams("history");

        Assessment assessment = new Assessment();

        math.setExams(math, history);

        ArrayList<Exams> exams = history.getExams();

        Putin = new Abiturient(Putin.getName(), Putin.getSurname(), Facultet, exams);

        Faculty facultet = new Faculty(Putin.getName(), Putin.getSurname(), Facultet);

        facultet.showFacult();

        Putin.showResult();

        Teacher Petya = new Teacher(Putin.getName(), Putin.getSurname(), assessment.getAssessment());

        Putin.assessment = Petya.Assessment;

        Petya.show();

        studentsResult.put(Putin.getSurname(), Putin.assessment);

        Abiturient Kim = new Abiturient("Kim", "Than In");

        Kim.show();

        Kim = new Abiturient(Kim.getName(), Kim.getSurname(), Facultet, exams);

        Faculty facultet2 = new Faculty(Kim.getName(), Kim.getSurname(), Facultet);

        facultet2.showFacult();

        Kim.showResult();

        Teacher Petya2 = new Teacher(Kim.getName(), Kim.getSurname(), assessment.getAssessment());

        Kim.assessment = Petya2.Assessment;

        Petya2.show();

        studentsResult.put(Kim.getSurname(), Kim.assessment);

        System.out.println("\nStudents who passed the exams");

        for (Map.Entry<String, Integer> entry : studentsResult.entrySet())

            if (entry.getValue() >= 4)

                System.out.println(entry.getKey());

    }

}

Спецификация вывода:

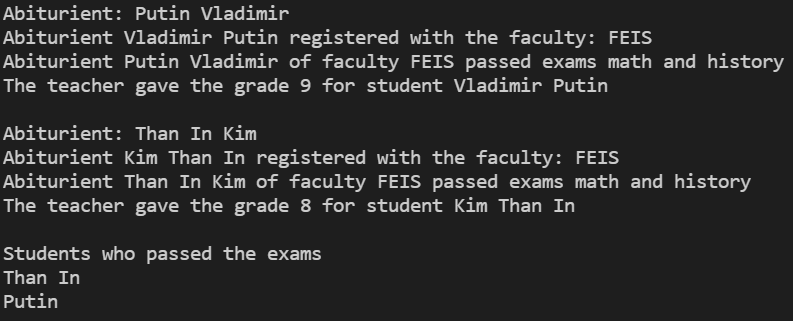
Метод Abiturient::show(): <фамилия> <имя>

Метод Abiturient::showResult(): <фамилия> <имя> <факультет> <экзамены>

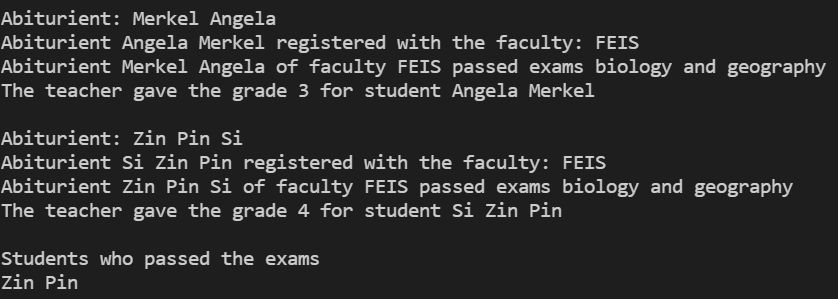
Метод Faculty::showFacult(): <имя> <факультет>

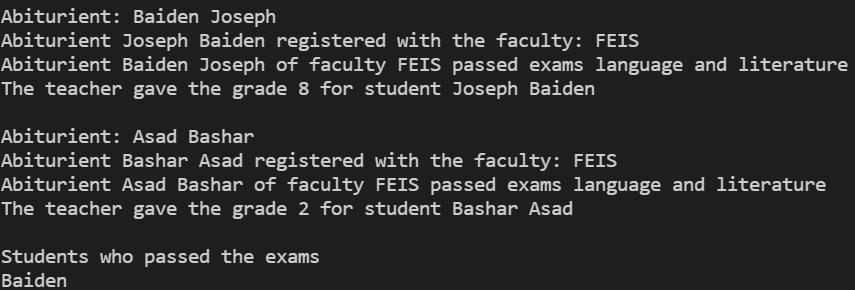
Метод Teacher::show(): <оценка> <имя> < фамилия>

Пример:



Рисунки с результатами работы программы





Вывод: Получил начальные навыки в работе с классами и объектами в языке программирования Java.