

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Програмування - 2
Об'єктно-орієнтоване програмування

МЕТОДИЧНІ ВКАЗІВКИ

до виконання розрахункових робіт для студентів спеціальності
„Автоматизація та комп'ютерно-інтегровані технології”

Рекомендовано кафедрою автоматизації хімічних виробництв
протокол №9 від 20.06.2017

Київ 2017

Програмування – 2. Об'єктно-орієнтоване програмування: Метод. вказівки до викон. розрахункових робіт для студ. спеціальності „Автоматизація та комп'ютерно-інтегровані технології” // Уклад.: Д.О. Ковалюк – К.: НТУУ «КПІ», 2017. – 19с.

Рекомендовано кафедрою
автоматизації хімічних виробництв
протокол №9 від 20.06.2017

Навчальне видання

Програмування - 2
Об'єктно-орієнтоване програмування

Методичні вказівки до виконання розрахункових робіт для студентів спеціальності „Автоматизація та комп'ютерно-інтегровані технології”

Укладачі: Ковалюк Дмитро Олександрович, доц.

Відповідальний

редактор А.І. Жученко, докт. техн. наук, проф.

Авторська редакція

ЗМІСТ

Мета і завдання розрахункової роботи	4
Структура розрахункової роботи	5
Вимоги до оформлення та подання звіту	7
Порядок захисту	7
Варіанти завдань	8
Приклад виконання	9

МЕТА І ЗАВДАННЯ РОЗРАХУНКОВОЇ РОБОТИ

Метою виконання розрахункової роботи є одержання необхідних знань та вмінь об'єктно-орієнтованого програмування на мові Java.

В результаті виконання роботи студенти закріплюють наступні знання:

- типи даних;
- функції;
- класи і об'єкти;
- принципи ООП – інкапсуляцію, наслідування, поліморфізм;
- обробку виняткових ситуацій;
- зчитування даних;
- використання колекцій;
- використання в проекті бібліотек сторонніх розробників.

Після виконання розрахункової роботи студенти повинні вміти:

- створювати програмне забезпечення для обробки даних;
- виконувати збірку, розгортання і відлагодження проекту.

.

СТРУКТУРА РОЗРАХУНКОВОЇ РОБОТИ

Розрахункова робота повинна складатися з наступних частин:

1. Титульний лист.
2. Технічне завдання.
3. Зміст.
4. Розділ 1 – Опис предметної області.
5. Розділ 2 – Структура програмного забезпечення.
6. Висновки.
7. Список використаних джерел.
8. Додатки (лістинг програм і скріншоти).

Технічне завдання

Виконується постановка задачі обробки даних, наводяться вимоги до функціональності програмного забезпечення. Наводяться початкові вхідні дані.

Розділ 1 - Опис предметної області

Навести структуру власних класів, які реалізують предметну область. Описати призначення їх полів і методів. Результат представити у вигляді таблиць. Вітається використання UML та CASE-засобів. Можна також використовувати згенеровану документацію JAVA.

Розділ 2 - Структура програмного забезпечення

Навести перелік пакетів, класів, інтерфейсів, типів даних, які реалізують бізнес-логіку роботи програми.

Програмне забезпечення має виконувати наступні функції

I. Робота в інтерактивному режимі з використанням меню.

II. Завантаження даних

- генерація даних на вимогу користувача
- завантаження з файлу
- перевірка даних на дублі, якщо потрібно – регенерувати дані, або вивести відповідне повідомлення
- Завантаження даних повинно бути реалізовано з використанням поліморфізму
- При завантаженні даних з файлу передбачити обробку виняткової ситуації

III Робота з масивами об'єктів (колекціями)

- Виконання операцій, вказаних у технічному завданні

IV Збірка проекту з використанням зовнішніх бібліотек

Висновки

Проаналізувати результати роботи. Зробити висновки про переваги або недоліки застосування JAVA для поставленої задачі.

Додатки

В додатках наводяться лістинги програм, та скріншоти їх виконання.

ВИМОГИ ДО ОФОРМЛЕННЯ ТА ПОДАННЯ ЗВІТУ

1. Пояснювальна записка – редактор MS Office.
2. Формули – редактор формул microsoft equation або mathtype – не допускається набір формул з використанням комбінацій різного формату шрифтів (*italic*, **bold**).
3. Скріншоти програмних засобів вставляються як рисунки MS Office.
4. Нумерація таблиць, формул, рисунків – в межах розділу.

ПОРЯДОК ЗАХИСТУ

1. Не пізніше, ніж за 2 тижні до кінця семестру надати електронний варіант пояснювальної записки та програмну реалізацію задачі на перевірку.
2. За наявності суттєвих зауважень, які не дозволяють позитивно оцінити роботу, - доопрацювати.
3. захист відбувається з використанням ЕОМ, на якому студент демонструє програмну реалізацію поставленої задачі та відповідає на питання викладача.

ВАРІАНТИ ЗАВДАНЬ

	Теоретична частина
1	Інформаційна система деканат. Функції: нарахування стипендії студентам, виведення списку студентів по рейтингу, виведення студентів, що проживають в певному гуртожитку
2	Інформаційна система магазин. Функції: облік товару, виведення списку товарів та їх кількості, вивести товари, термін дії яких виходить через певну кількість днів
3	Інформаційна система пошта. Функції: виведення списку рекомендованих листів, розрахунок вартості посилки залежно від її ваги, виведення кількості листів для відповідного міста
4	Інформаційна система Євробачення. Функції: виведення списку учасників, оцінка учасників, виведення трійки переможців
5	Інформаційна система засоби автоматизації. Функції: виведення списку технічних засобів, виведення групи технічних засобів (датчики, контролери, перетворювачі)
6	Інформаційна система автосалон. Функції: виведення списку авто, пошук авто за запитом (складним) користувача, сортування за параметрами
7	Інформаційна система вокзал. Функції: виведення списку рейсів між довільними містами, пошук та бронювання вільних місць на вказаний напрямок
8	Інформаційна система кінотеатр. Функції: виведення списку фільмів, виведення списку сеансів для конкретного фільму, виведення списку фільмів по категорії (дитячі, фантастика)
9	Інформаційна система Google Analytics. Функції: виведення відвідуваності сайту за останні три місяці, вивести джерела трафіку за зростанням
10	Інформаційна система Готель. Функції: виведення списку вільних номерів на певну дату, бронювання номеру, виведення номерів певної категорії (люкс, стандарт, двомісні, адномісні)

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інженерно-хімічний факультет

Кафедра «Автоматизація хімічних виробництв»

Розрахункова робота
Програмування-2. Об'єктно-орієнтованого програмування

Варіант 1
«Інформаційна система деканат»

Виконав:	Студент групи ЛА-61, С.О.Кубах
Перевірив:	Доцент каф. АХВ, Д.О. Ковалюк

Технічне завдання

Метою виконання розрахункової роботи є одержання необхідних знань та вмінь об'єктно-орієнтованого програмування на мові Java.

Інформаційна система деканат.

Функції:

- виведення списку студентів
- сортування за параметрами
- виведення найкращого студента
- виведення студента з найменшою кількістю балів
- виведення студентів жіночої статі
- виведення студентів-бюджетників

Початкові вхідні дані:

- Дані отримані при створенні Excel документа.
- Дані отриманні з масиву об'єктів.

Розділ 1 - Опис предметної області

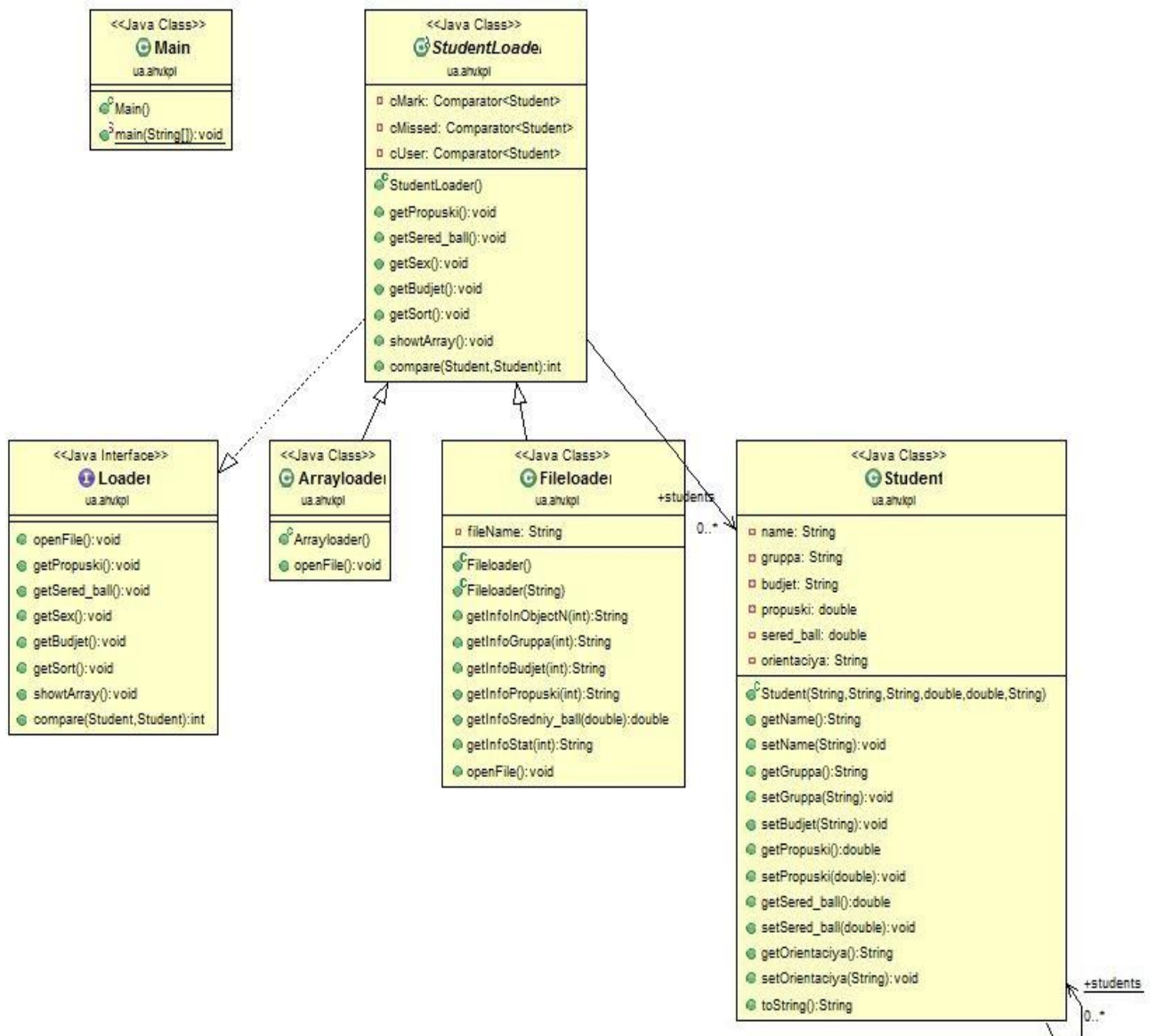


Табл 1.1 UML структура класів

Інформація про фреймворку Maven і використаних бібліотеках:

- Apache Maven 3.5.0 (2017-04-03T22:39:06+03:00)
- Maven home: C:\Users...
- Java version: 1.8.0_111, vendor: Oracle Corporation
- Apache POI 3.16

Розділ 2 - Структура програмного забезпечення

Абстрактний клас StudentLoader:

- Метод getPropuski () – визначає студента з найменшою кількістю пропусків.
- Метод getSered_bal () – визначає студента з кращим балом.
- Метод getSex () – визначає студента жіночої статі.
- Метод getBudjet () – визнає студентів-бюджетників.
- Метод getSort () – сортує студентів за параметрами.
- Метод showtArray () – виводить масив студентів.

Інтерфейс Loader:

- Визначає імена методів і списки аргументів для класів:
- ArrayLoader.
- FileLoader.
- Метод (getPropuski (), getSered_bal (), getSex (), getBudjet (), getSort (), showtArray ()) – наслідує абстрактний клас StudentLoader.

Клас ArrayLoader:

- Метод openFille() – містить масив об'єктів.

Клас FileLoader:

- Поле fileName - містить в собі отформатовану таблицю Книга1.xls
- Методи з префіксом get отримують з таблиці інформацію у відповідні поля та повертають їх.
- Метод openFile() створює потік на читання необхідного файлу Книга1.xls

Клас Student:

- Містить поля про студента та інкапсуляцію.

Клас Main:

- Містить сканер.
- Містить конструкцію Switch.
- Містить екземпляри інших класів.
- Виводить інформацію про студентів.

Висновки

Робота виконана із застосуванням зовнішньої бібліотеки для редагування і зчитування Excel файлів - Apache POI. Всі завдання в роботі виконуються і працюють справно без єдиної помилки.

Файл зчитується, інформація заноситься в масив об'єктів, і форматується як необхідно користувачу. У даній програмі мова програмування Java заснована на об'єктно-орієнтованому програмуванні, ідеально підходить для виконання всіх поставлених задач.

Його можливості разом із зовнішніми бібліотеками дозволяють без особливих зусиль працювати з такими форматами даних і реалізовувати безліч корисних функцій.

Додатки

The screenshot shows an IDE with a Java project. The main file, `Main.java`, contains the following code:

```

1 package ua.ahv.kpi;
2 import java.util.Scanner;
3
4
5 public class Main {
6     public static void main(String[] args) {
7         // new Student("Kripov Artem", "LA-54", "Budjet", 4, 4.5, "men");
8         // new Student("Pasichnik Jenya", "LA-63", "Kontrakt", 108, 3.2, "men");
9         // new Student("Sedaya Ira", "LA-63", "Budjet", 18, 4.8, "women");
10        // new Student("Ostrovskiy German", "LA-54", "Budjet", 28, 4.1, "men");
11    }
12 }

```

The console output shows the execution of the program:

```

<terminated> Main (10) [Java Application] C:\Program Files\Java\jdk1.8.0_121\bin\javaw.exe (17 мая 2017 г., 19:19:50)
3.Вывести Женского пола
4.Отсортировать (по имени)
5.Вывести бюджетников
Введите число(0 выход): 4
Сортированный массив:
[name=Kripov Artem, gruppa=LA-54, budjet=Budjet, propuski=4.0, sered_ball=4.5, orientaciya=men]
[name=Krivcov Egor, gruppa=LA-54, budjet=Kontrakt, propuski=40.0, sered_ball=5.0, orientaciya=men]
[name=Morozov Stas, gruppa=LA-63, budjet=Kontrakt, propuski=35.0, sered_ball=3.45, orientaciya=men]
[name=Ostrovskiy German, gruppa=LA-54, budjet=Budjet, propuski=28.0, sered_ball=4.1, orientaciya=men]
[name=Pasichnik Jenya, gruppa=LA-63, budjet=Kontrakt, propuski=108.0, sered_ball=3.2, orientaciya=men]
[name=Sedaya Ira, gruppa=LA-63, budjet=Budjet, propuski=18.0, sered_ball=4.8, orientaciya=women]

Выберите пункт:
1.Вывести с лучшим баллом
2.Вывести с меньшими пропусками
3.Вывести Женского пола
4.Отсортировать (по имени)
5.Вывести бюджетников
Введите число(0 выход): 5
Студенты бюджетники:
[name=Kripov Artem, gruppa=LA-54, budjet=Budjet, propuski=4.0, sered_ball=4.5, orientaciya=men]
[name=Ostrovskiy German, gruppa=LA-54, budjet=Budjet, propuski=28.0, sered_ball=4.1, orientaciya=men]
[name=Sedaya Ira, gruppa=LA-63, budjet=Budjet, propuski=18.0, sered_ball=4.8, orientaciya=women]

```

Main

```

package ua.ahv.kpi;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        System.out.println("Откуда загрузить данные?");
        System.out.println("1.Загрузить данные с массива");
        System.out.println("2.Загрузить данные с файла");
        System.out.print("Введите число (0 выход): ");

        Loader studentsData = new Arrayloader(); // загружаем все равно
        switch (scan.next()) {
            case "1":
                System.out.println("1");
                studentsData = new Arrayloader(); // загрузили список
                System.out.println("Данные загружены с массива");
                break;
            case "2":
                System.out.println("2");
                studentsData = new Fileloader(); // файл по умолчанию
                System.out.println("Данные загружены с файла");
                break;
            case "0":
                System.out.println("Выход");
                System.exit(0);
                break;
            default:
                System.out.println("Неправильно");
        }
        studentsData.showtArray();

        while (true) {
            System.out.println("\nВыберите пункт:");
            System.out.println("1.Вывести с лучшим баллом");
            System.out.println("2.Вывести с меньшими пропусками");
            System.out.println("3.Вывести Женского пола");
            System.out.println("4.Отсортировать (по фамилии)");
            System.out.println("5.Вывести бюджетников");
            System.out.print("Введите число (0 выход): ");
        }
    }
}

```

```
switch (scan.next()) {  
    case "1":  
        studentsData.getSered_ball();  
        break;  
    case "2":  
        studentsData.getPropuski();  
        break;  
    case "3":  
        studentsData.getSex();  
        break;  
    case "4":  
        studentsData.getSort();  
        break;  
    case "5":  
        studentsData.getBudjet();  
        break;  
    case "0":  
        System.out.println("Выход");  
        System.exit(0);  
        break;  
    default:  
        System.out.println("Неправильно");  
}  
  
}  
  
}
```

ArrayLoader

```
package ua.ahv.kpi;

public class Arrayloader extends StudentLoader {

    public Arrayloader() {
        openFile(); // загружаем во внутренний массив
        Student.students = students;
    }

    @Override
    public void openFile() { // вернуть массив студентов
        students.add(new Student("Kripov Artem", "LA-54", "Budjet", 4, 4.5,
"men"));
        students.add(new Student("Pasichnik Jenya", "LA-63", "Kontrakt",
108, 3.2, "men"));
        students.add(new Student("Sedaya Ira", "LA-63", "Budjet", 18, 4.8,
"women"));
        students.add(new Student("Ostrovskiy German", "LA-54", "Budjet",
28, 4.1, "men"));
        students.add(new Student("Krivcov Egor", "LA-54", "Kontrakt", 40,
5, "men"));
        students.add(new Student("Morozov Stas", "LA-63", "Kontrakt", 35,
3.45, "men"));
    }
}
```

Student

```
package ua.ahv.kpi;
import java.util.List;

public class Student {
    public static List<Student> students;

    private String name;
    private String gruppa;
    private String budget;
    private double propuski;
    private double sered_ball;
    private String orientaciya;

    public Student(String name, String gruppa, String budget, double
propuski, double sered_ball, String orientaciya) {
        this.name = name;
        this.gruppa = gruppa;
        this.budget = budget;
        this.propuski = propuski;
        this.sered_ball = sered_ball;
        this.orientaciya = orientaciya;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getGruppa() {
        return gruppa;
    }

    public void setGruppa(String gruppa) {
        this.gruppa = gruppa;
    }

    public String getBudget() {
        return budget;
    }

    public void setBudget(String budget) {
        this.budget = budget;
    }

    public double getPropuski() {
        return propuski;
    }

    public void setPropuski(double propuski) {
```

```

        this.propuski = propuski;
    }

    public double getSered_ball() {
        return sered_ball;
    }

    public void setSered_ball(double sered_ball) {
        this.sered_ball = sered_ball;
    }

    public String getOrientaciya() {
        return orientaciya;
    }

    public void setOrientaciya(String orientaciya) {
        this.orientaciya = orientaciya;
    }

    public String toString() {
        return "[name=" + name + ", gruppa=" + gruppa + ", budjet=" +
budjet + ", propuski=" + propuski
                + ", sered_ball=" + sered_ball + ", orientaciya=" +
orientaciya + "]";
    }
}

```