



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційні систем та технологій

**Лабораторна робота №10**  
із дисципліни «*Основи програмування*»  
**Тема:** «*Колекції. Списки*»

Виконали:  
Студенти групи ІА-24

Момоток Любов

Пархоменко Іван

Боднар Антон

Перевірив:

Колеснік Валерій Миколайович

Київ-2022

**Хід роботи:**

**1. Ознайомитись з javadoc для наступних інтерфейсів та класів:**

- Collection
- List
- ArrayList
- LinkedList
- Iterator
- RandomAccess

**2. Виконати завдання з таблиці 2 відповідно до свого варіанту у таблиці**

**1. Для цього:**

- проаналізувати завдання;
- створити зазначенні класи;
- для створення списків слід використовувати класи та інтерфейси з Collection Framework (заборонено використовувати масиви);
- усі списки мають бути типізованими (наприклад, ArrayList<Student>, а не просто ArrayList);
- при реалізації задач «1)», «2)», «3)» слід застосувати наступні методи перегляду колекцій у відповідності до свого варіанту (табл. 1):
  - а) нетипізований ітератор;
  - б) типізований ітератор;
  - с) типізований цикл «for-each»

**Код програми:**

```

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

public class Department {
    private final String nameOfDepartment;
    private final String chief;
    private List<Employee> employees = new ArrayList();

    public Department(String nameOfDepartment, String chief) {
        this.nameOfDepartment = nameOfDepartment;
        this.chief = chief;
    }

    private static final List<Employee> allEmployees = new ArrayList();
    public static void addEmployee(Employee employee) {
        allEmployees.add(employee);
    }

    public static List<Employee> getAllEmployees(){
        return allEmployees;
    }

    public String getChief() {
        return this.chief;
    }

    @Override
    public String toString() {
        return "Department(" + "Name of department=" + nameOfDepartment + ", Chief=" + chief + " )";
    }
}

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

public class Employee {
    private String firstname;
    private String surname;
    private int salary;

    public Employee(String firstname, String surname, int salary) {
        if (salary < 0) {
            throw new IllegalArgumentException();
        }
        this.firstname = firstname;
        this.surname = surname;
        this.salary = salary;
    }

    public String getName() {
        return this.firstname;
    }

    public String getSurname() {
        return this.surname;
    }

    public int getsSalary() {
        return this.salary;
    }

    @Override
    public String toString() {
        return firstname + " " + surname;
    }
}

```

```

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

public class Firm {
    private String nameOfFirm;
    private String head;

    public Firm(String nameOfFirm, String head) {
        this.nameOfFirm = nameOfFirm;
        this.head = head;
    }

    private static final List<Firm> allFirms= new ArrayList();

    public static void addFirm(Firm firm) {
        allFirms.add(firm);
    }
    public static List<Firm> getAllFirms() {
        return allFirms;
    }

    public String getHead() {
        return this.head;
    }

    public String getName() {
        return this.nameOfFirm;
    }

    private static final List<Department> allDepartments = new ArrayList();

    public static void addDepartment(Department department) {
        allDepartments.add(department);
    }

    public static List<Department> getAllDepartments() {
        return allDepartments;
    }

    @Override
    public String toString() {

    @Override
    public String toString() {
        return "Firm(" + "Name of firm=" + nameOfFirm + ", Head=" + head + " )";
    }
}

```

```

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

public class Main {
    public static void main(String[] args) {
        CreateClasses();
        printResult();
    }

    public static void CreateClasses() {
        Employee employee1 = new Employee("name1", "surname1", 1000);
        Employee employee2 = new Employee("name2", "surname2", 1100);
        Employee employee3 = new Employee("name3", "surname3", 1200);
        Employee employee4 = new Employee("name4", "surname4", 1300);
        Employee employee5 = new Employee("name5", "surname5", 1400);
        Employee employee6 = new Employee("name6", "surname6", 1500);
        Department department1 = new Department("Department1", "Chief1");
        Department department2 = new Department("Department2", "Chief2");
        Department department3 = new Department("Department3", "Chief3");
        Firm firm1 = new Firm("Firm1", "Head1");
        Firm firm2 = new Firm("Firm2", "Head2");
        Firm firm3 = new Firm("Firm3", "Head3");
        Department.addEmployee(employee1);
        Department.addEmployee(employee2);
        Department.addEmployee(employee3);
        Department.addEmployee(employee4);
        Department.addEmployee(employee5);
        Department.addEmployee(employee6);
        firm1.addDepartment(department1);
        firm2.addDepartment(department2);
        firm3.addDepartment(department3);
        Firm.addFirm(firm1);
        Firm.addFirm(firm2);
        Firm.addFirm(firm3);
    }

    private static void printResult() {
        System.out.println("A list of all company workers : " + ListOfWorkers());
    }

    private static void printResult() {
        System.out.println("A list of all company workers : " + ListOfWorkers());
    }

    private static List<String> ListOfWorkers() {
        List<String> ListOfAllWorkers = new ArrayList<>();
        ArrayList<Employee> allEmployees = (ArrayList<Employee>) Department.getAllEmployees();
        for (Employee employee : allEmployees) {
            ListOfAllWorkers.add(employee.toString());
        }
        ArrayList<Department> allDepartment = (ArrayList<Department>) Firm.getAllDepartments();
        for (Department department : allDepartment){
            ListOfAllWorkers.add(department.getChief());
        }
        ArrayList<Firm> allFirms = (ArrayList<Firm>) Firm.getAllFirms();
        for (Firm firm : allFirms){
            ListOfAllWorkers.add(firm.getHead());
        }

        return ListOfAllWorkers;
    }
}

```

## Результати:

```

A list of all company workers : [name1 surname1, name2 surname2, name3 surname3, name4 surname4, name5 surname5, name6 surname6, Chief1, Chief2, Chief3, Head1, Head2, Head3]
Process finished with exit code 0

```

```
nameó surnameó, Chief1, Chief2, Chief3, Head1, Head2, Head3]
```

**Висновки:** На лабораторній роботі ми ознайомилися з javadoc для таких інтерфейсів та класів як:Collection,List,ArreyList та інше.