Отчет по контрольной работе

По дисциплине «Разработка программных модулей»

на тему «Объектно ориентированное программирование на языке Java»

Работу выполнил:

Студент группы ИС-33

Костин Д. Г.

Руководитель:

Кошкина А. А.

**Содержание**

[**1. Цель. 3**](#_3najzqinafrz)

[**2. Постановка задачи. 3**](#_sewsoyi477g3)

[**3. Структура проекта. 4**](#_bfj85bqnqojd)

[**4. Описание проекта. 5**](#_8ra2nnmd3equ)

[Интерфейсы: 5](#_awwbjlu8jerw)

[Абстрактные классы: 5](#_40x87oqx86pt)

[Классы 6](#_5xou55x7iurv)

[Класс Main 11](#_mb02nbetgm8m)

[**5. Код программы. 11**](#_ru0zpau9zman)

[Абстрактный класс Building: 11](#_17mn1jbfznsp)

[Абстрактный класс Person: 12](#_g1r1vv26xj6o)

[Интерфейс IHumanActions: 14](#_ie5n6lhb9h90)

[Интерфейс IPrintable: 14](#_5mdlpvibhspo)

[Класс Employee: 14](#_9bg7ohjkcrnj)

[Класс Student: 16](#_8won73r3x3d5)

[Класс Inventory: 18](#_l0bzj9eylcs9)

[Класс Schedule: 19](#_w85389idhux4)

[Класс University: 20](#_swnt9vjoeo07)

[Класс Main: 24](#_3s8l4ud2v0ta)

[**6. Вывод программы: 27**](#_or87892o088m)

[**7. Общий вывод. 43**](#_6er2uz54e722)

## Цель.

Моя цель в выполнении контрольной работы состоит в том, чтобы углубленно изучить и продемонстрировать свое понимание принципов объектно-ориентированного программирования на языке Java. Я стремлюсь к освоению ключевых концепций, таких как создание классов, использование наследования и полиморфизма, а также применение интерфейсов и обработка исключений. Моя цель - не только успешно выполнить задачи контрольной работы, но и научиться применять эти знания в будущем для разработки эффективных и надежных Java-приложений с использованием ООП-подхода.

## Постановка задачи.

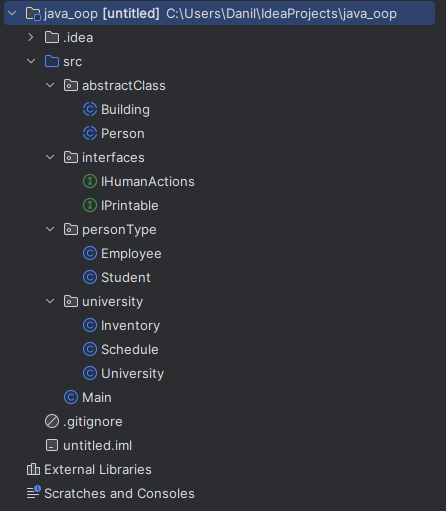
Обязательно должен присутствовать класс Main. Используйте технологии: инкапсуляцию, наследование, абстрактные классы, интерфейсы, final, static, полиморфизм (перегрузка и переопределение методов), конструкторы и инициализаторы.

Представьте также файл с описанием вашего проекта и системы классов.

В качестве ответа необходимо предоставить описание проекта и системы классов скрины кода каждого класса, скрин вывода программы, ссылку на репозиторий с кодом.

Создайте структуру классов, которые могли бы использоваться в работе университета. В университете должны храниться списки инвентаря, сотрудников, учеников, информация о расписании и т.д.

## Структура проекта.



## Описание проекта.

### Интерфейсы:

Представлены простые интерфейсы, отвечающие за вывод данных экземпляра класса (IPrintable) и за методы, описывающие человеческие действия в тех или иных ситуациях (IHumanActions).

### Абстрактные классы:

Класс "**Building**" является абстрактным базовым классом, который реализует интерфейс "IPrintable" и представляет собой общую модель здания или строительного объекта. Вот описание класса "Building":

* Атрибуты класса "Building":
  + name (название здания).
  + address (адрес здания).
  + customer (заказчик строительства).
  + building\_permit (разрешение на строительство).
* Конструктор:
  + Конструктор класса "Building" принимает следующие атрибуты:
    - name (название здания).
    - address (адрес здания).
    - customer (заказчик строительства).
    - building\_permit (разрешение на строительство).
  + В конструкторе устанавливаются значения этих атрибутов.
* Методы класса "Building":
  + getName() и setName(): методы для доступа к названию здания.
  + getAddress() и setAddress(): методы для доступа к адресу здания.
  + getCustomer() и setCustomer(): методы для доступа к заказчику строительства.
  + getBuilding\_permit() и setBuilding\_permit(): методы для доступа к разрешению на строительство.
  + Метод print(), реализующий интерфейс IPrintable, который позволяет выводить информацию о здании.

Класс "Building" предоставляет общие атрибуты и функциональность, которые могут быть унаследованы и дополнены подклассами, представляющими конкретные типы зданий или строительных объектов. Этот класс служит в качестве основы для организации и структурирования информации о строительных объектах.

Класс "**Person**" является абстрактным базовым классом, который реализует интерфейсы "IHumanActions" и "IPrintable" и представляет собой общую модель человека. Вот описание класса "Person":

* Атрибуты класса "Person":
  + name (имя человека).
  + surname (фамилия человека).
  + age (возраст человека).
* Конструктор:
  + Конструктор класса "Person" принимает следующие атрибуты:
    - name (имя человека).
    - surname (фамилия человека).
    - age (возраст человека).
  + В конструкторе устанавливаются значения этих атрибутов.
* Методы класса "Person":
  + getName() и setName(): методы для доступа к имени человека.
  + getSurname() и setSurname(): методы для доступа к фамилии человека.
  + getAge() и setAge(): методы для доступа к возрасту человека.
  + Метод toString(), который предоставляет строковое представление объекта класса "Person".

Абстрактный класс"Person" является базовой моделью для представления конкретных типов людей (например, студентов, сотрудников и т. д.). Он предоставляет общие атрибуты и функциональность, которые могут быть унаследованы и дополнены в подклассах, чтобы представлять разные категории людей.

### Классы

Класс "**Employee**" является подклассом класса "Person" и представляет модель сотрудника. Он расширяет базовые характеристики, описанные в классе "Person", добавляя дополнительные атрибуты, методы и функциональность. Вот описание класса "Employee":

* Атрибуты класса "Employee":
  + specialization (специализация сотрудника).
  + experience (опыт работы сотрудника).
  + salary (заработная плата сотрудника).
  + employee\_id (уникальный идентификатор сотрудника).
  + counter (статический счетчик для генерации уникальных идентификаторов).
* Конструктор:
  + Конструктор класса "Employee" принимает следующие атрибуты:
    - name (имя сотрудника).
    - surname (фамилия сотрудника).
    - age (возраст сотрудника).
    - specialization (специализация сотрудника).
    - experience (опыт работы сотрудника).
    - salary (заработная плата сотрудника).
  + В конструкторе вызывается конструктор суперкласса "Person" с атрибутами name, surname и age.
  + Устанавливается значение employee\_id, инкрементируя counter для создания уникального идентификатора сотрудника.
* Методы класса "Employee":
  + getSpecialization() и setSpecialization(): методы для доступа к специализации сотрудника.
  + getExperience() и setExperience(): методы для доступа к опыту работы сотрудника.
  + getSalary() и setSalary(): методы для доступа к заработной плате сотрудника.
  + getEmployee\_id(): метод для доступа к уникальному идентификатору сотрудника.
  + activity(), work(), rest(): методы, реализующие интерфейс IHumanActions и представляющие действия сотрудника.
  + print(): метод, реализующий интерфейс IPrintable, для вывода информации о сотруднике.
  + remark(Student s) и remark(ArrayList<Student> s): методы для замечания студентам.
  + promotion(): метод для повышения заработной платы сотрудника.

Класс "Employee" позволяет представлять информацию о сотрудниках, их действиях и взаимодействиях со студентами. Класс также обладает возможностью вывода информации о сотруднике и управлением его заработной платой.

Класс "**Student**" представляет модель студента и является подклассом класса "Person". Он добавляет специфичные для студента атрибуты, методы и функциональность. Вот описание класса "Student":

* Атрибуты класса "Student":
  + group (номер группы, в которой учится студент).
  + specialization (специализация студента).
  + social\_scholarship (наличие социальной стипендии).
  + student\_id (уникальный идентификатор студента).
  + counter (статический счетчик для генерации уникальных идентификаторов).
* Конструктор:
  + Конструктор класса "Student" принимает следующие атрибуты:
    - name (имя студента).
    - surname (фамилия студента).
    - age (возраст студента).
    - group (номер группы).
    - specialization (специализация).
    - social\_scholarship (наличие социальной стипендии).
  + В конструкторе вызывается конструктор суперкласса "Person" с атрибутами name, surname и age.
  + Устанавливается значение student\_id, инкрементируя counter для создания уникального идентификатора студента.
* Методы класса "Student":
  + getGroup() и setGroup(): методы для доступа к номеру группы студента.
  + getSpecialization() и setSpecialization(): методы для доступа к специализации студента.
  + isSocial\_scholarship() и setSocial\_scholarship(): методы для доступа к наличию социальной стипендии.
  + toString(): метод для предоставления строкового представления объекта класса "Student".
  + activity(), work(), rest(): методы, реализующие интерфейс IHumanActions и представляющие действия студента.
  + print(): метод, реализующий интерфейс IPrintable, для вывода информации о студенте.
  + getStudent\_id(): метод для доступа к уникальному идентификатору студента.

Класс "Student" позволяет представлять информацию о студентах, их действиях и характеристиках. Класс также обладает возможностью вывода информации о студенте и управлением его атрибутами.

Класс "**Inventory**" представляет собой модель инвентаря или предмета в университете. Он реализует интерфейс "IPrintable" для вывода информации об инвентаре. Вот описание класса "Inventory":

* Атрибуты класса "Inventory":
  + name (название инвентаря или предмета).
  + count (количество данного предмета в инвентаре).
* Конструктор:
  + Конструктор класса "Inventory" принимает следующие атрибуты:
    - name (название инвентаря).
    - count (количество данного предмета в инвентаре).
  + В конструкторе устанавливаются значения этих атрибутов.
* Методы класса "Inventory":
  + getName() и setName(): методы для доступа к названию инвентаря.
  + getCount() и setCount(): методы для доступа к количеству данного предмета в инвентаре.
  + toString(): метод для предоставления строкового представления объекта класса "Inventory".
  + print(): метод, реализующий интерфейс "IPrintable", который выводит информацию об инвентаре.

Класс "Inventory" позволяет представлять информацию о предметах в университетском инвентаре и выводить эту информацию. Этот класс может использоваться для отслеживания и учета инвентаря университета.

Класс "**Schedule**" представляет собой модель расписания в университете. Он реализует интерфейс "IPrintable" для вывода информации о расписании. Вот описание класса "Schedule":

* Атрибуты класса "Schedule":
  + information (информация о событии или занятии в расписании).
  + date (дата и время события или занятия).
* Конструктор:
  + Конструктор класса "Schedule" принимает следующие атрибуты:
    - information (информация о событии или занятии).
    - date (дата и время события или занятия).
  + В конструкторе устанавливаются значения этих атрибутов.
* Методы класса "Schedule":
  + getInformation() и setInformation(): методы для доступа к информации о событии или занятии в расписании.
  + getDate() и setDate(): методы для доступа к дате и времени события или занятия.
  + toString(): метод для предоставления строкового представления объекта класса "Schedule".
  + print(): метод, реализующий интерфейс "IPrintable", который выводит информацию о событии или занятии в расписании, включая дату и информацию о событии.

Класс "Schedule" позволяет представлять информацию о расписании в университете и выводить эту информацию, включая дату и детали события. Этот класс может использоваться для отслеживания расписания занятий в университете.

Класс "**University**" представляет университет и расширяет абстрактный класс "Building". Он содержит списки инвентаря, студентов и сотрудников, а также расписание. Вот описание класса "University":

* Атрибуты класса "University":
  + inventoryList (список инвентаря или предметов в университете).
  + students (список студентов в университете).
  + employees (список сотрудников в университете).
  + schedule (расписание университета).
* Конструктор:
  + Конструктор класса "University" принимает следующие атрибуты:
    - name (название университета).
    - address (адрес университета).
    - customer (заказчик университета).
    - building\_permit (разрешение на строительство университета).
    - inventoryList (список инвентаря).
    - students (список студентов).
    - employees (список сотрудников).
    - schedule (расписание).
  + В конструкторе вызывается конструктор суперкласса "Building" с атрибутами name, address, customer и building\_permit.
* Методы класса "University":
  + getInventoryList() и setInventoryList(): методы для доступа к списку инвентаря.
  + getStudents() и setStudents(): методы для доступа к списку студентов.
  + getEmployees() и setEmployees(): методы для доступа к списку сотрудников.
  + getSchedule() и setSchedule(): методы для доступа к расписанию.
  + addStudent(Student student): метод для добавления студента в список студентов.
  + addEmployee(Employee employee): метод для добавления сотрудника в список сотрудников.
  + addInventory(Inventory inventory): метод для добавления инвентаря в список инвентаря.
  + removeInventory(Inventory inventory): метод для удаления инвентаря из списка инвентаря.
  + removeStudent(Student student): метод для удаления студента из списка студентов.
  + removeEmployee(Employee employee): метод для удаления сотрудника из списка сотрудников.
  + countList(): метод для подсчета количества студентов, сотрудников и инвентаря.
  + printStudentsWithScholarship(): метод для вывода информации о студентах, получающих социальную стипендию.
  + printSpecialtiesWithEmployees(): метод для вывода специальностей и связанных с ними сотрудников.
  + printSpecialtiesWithStudents(): метод для вывода специальностей и связанных с ними студентов.
  + printStudentById(int id): метод для вывода информации о студенте по его уникальному идентификатору.
  + printEmployeeById(int id): метод для вывода информации о сотруднике по его уникальному идентификатору.
  + printSchedule(): метод для вывода информации о расписании университета.
  + print(): метод, реализующий интерфейс "IPrintable", который выводит информацию об университете.
  + toString(): метод для предоставления строкового представления объекта класса "University"

Класс "University" представляет университет и управляет информацией о студентах, сотрудниках, инвентаре и расписании. Этот класс позволяет добавлять, удалять и анализировать данные в университете, а также выводить информацию о них.

### Класс Main

В классе Main cозданы объекты сотрудников и студентов, а также инвентарь и расписание.

Создан объект класса "University" с использованием созданных данных.

В цикле вывели информацию о студентах и вызвали методы для симуляции их действий.

В цикле вывели информацию о сотрудниках и вызвали методы для симуляции их действий.

Вывели информацию о "RKSI" (университете) и вызвали различные методы, такие как вывод студентов с социальной стипендией, вывод сотрудника по уникальному идентификатору, вывод специальностей с соответствующими студентами и сотрудниками, вывод расписания и подсчет количества студентов, сотрудников и инвентаря.

Добавлен новый студент и удален один из сотрудников.

## Код программы.

### Абстрактный класс Building:

package abstractClass;

import interfaces.IPrintable;

public abstract class Building implements IPrintable {

private String name;

private String address;

private String customer;

private String building\_permit;

public Building(String name, String address, String customer, String building\_permit) {

this.name = name;

this.address = address;

this.customer = customer;

this.building\_permit = building\_permit;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getCustomer() {

return customer;

}

public void setCustomer(String customer) {

this.customer = customer;

}

public String getBuilding\_permit() {

return building\_permit;

}

public void setBuilding\_permit(String building\_permit) {

this.building\_permit = building\_permit;

}

}

### Абстрактный класс Person:

package abstractClass;

import interfaces.IHumanActions;

import interfaces.IPrintable;

public abstract class Person implements IHumanActions, IPrintable {

private String name;

private String surname;

private int age;

public Person(String name, String surname, int age) {

this.name = name;

this.surname = surname;

this.age = age;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getSurname() {

return surname;

}

public void setSurname(String surname) {

this.surname = surname;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

@Override

public String toString() {

return "Person{" +

"name='" + name + '\'' +

", surname='" + surname + '\'' +

", age=" + age +

'}';

}

}

### Интерфейс IHumanActions:

package interfaces;

public interface IHumanActions {

void activity();

void work();

void rest();

}

### Интерфейс IPrintable:

package interfaces;

public interface IPrintable {

void print();

}

### Класс Employee:

package personType;

import abstractClass.Person;

import java.util.ArrayList;

public class Employee extends Person {

private String specialization;

private String experience;

private int salary;

private final int employee\_id;

private static int *counter*;

static{

*counter* = 200001;

}

public Employee(String name, String surname, int age, String specialization, String experience, int salary) {

super(name, surname, age);

this.specialization = specialization;

this.experience = experience;

this.salary = salary;

employee\_id = *counter*++;

}

public String getSpecialization() {

return specialization;

}

public void setSpecialization(String specialization) {

this.specialization = specialization;

}

public String getExperience() {

return experience;

}

public void setExperience(String experience) {

this.experience = experience;

}

public int getSalary() {

return salary;

}

public void setSalary(int salary) {

this.salary = salary;

}

@Override

public void activity() {

System.*out*.println("Talking with colleagues.");

}

@Override

public void work() {

System.*out*.println("Teaching...");

}

@Override

public void rest() {

System.*out*.println("Drinking coffee.");

}

@Override

public void print() {

System.*out*.println("Name: " + this.getName() + "\nSurname: " + this.getSurname() + "\nAge: " + this.getAge() + "\nSpecialization: " + this.specialization + "\nExperience: " + this.experience + "\nSalary: " + this.salary + "\nID:" + this.getEmployee\_id());

}

@Override

public String toString() {

return "humans.Employee{" +

"specialization='" + specialization + '\'' +

", experience='" + experience + '\'' +

", salary=" + salary +

'}';

}

public int getEmployee\_id() {

return employee\_id;

}

public void remark(Student s){

System.*out*.println("Сотрудник " + this.getName() + " " + this.getSurname() + " сделал зачемание студенту: " + s.getName() + " " + s.getSurname());

}

public void remark(ArrayList<Student> s){

for(Student i: s){

System.*out*.println("Сотрудник " + this.getName() + " " + this.getSurname() + " сделал зачемание студенту: " + i.getName() + " " + i.getSurname());

}

}

public void promotion(){

this.salary += 20000;

}

}

### Класс Student:

package personType;

import abstractClass.Person;

public class Student extends Person {

private int group;

private String specialization;

private boolean social\_scholarship;

private final int student\_id;

private static int *counter*;

static {

*counter* = 100001;

}

public Student(String name, String surname, int age, int group, String specialization, boolean social\_scholarship) {

super(name, surname, age);

this.group = group;

this.specialization = specialization;

this.social\_scholarship = social\_scholarship;

student\_id = *counter*++;

}

public int getGroup() {

return group;

}

public void setGroup(int group) {

this.group = group;

}

public String getSpecialization() {

return specialization;

}

public void setSpecialization(String specialization) {

this.specialization = specialization;

}

public boolean isSocial\_scholarship() {

return social\_scholarship;

}

public void setSocial\_scholarship(boolean social\_scholarship) {

this.social\_scholarship = social\_scholarship;

}

@Override

public String toString() {

return "humans.Student{" +

"group=" + group +

", specialization='" + specialization + '\'' +

", social\_scholarship=" + social\_scholarship +

"} " + super.toString();

}

@Override

public void activity() {

System.*out*.println(this.getName() + "Talking with friends.");

}

@Override

public void work() {

System.*out*.println(this.getName() + "Studying...");

}

@Override

public void rest() {

System.*out*.println(this.getName() + "Scrolling TikToks.");

}

@Override

public void print() {

System.*out*.println("-----------------------------------------");

System.*out*.println("Student ID: " + student\_id);

System.*out*.println("Name: " + getName());

System.*out*.println("Surname: " + getSurname());

System.*out*.println("Age: " + getAge());

System.*out*.println("Group: " + group);

System.*out*.println("Specialization: " + specialization);

System.*out*.println("Social scholarship: " + social\_scholarship);

System.*out*.println("-----------------------------------------");

}

public int getStudent\_id() {

return student\_id;

}

}

### Класс Inventory:

package university;

import interfaces.IPrintable;

public class Inventory implements IPrintable {

private String name;

private int count;

public Inventory(String name, int count) {

this.name = name;

this.count = count;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getCount() {

return count;

}

public void setCount(int count) {

this.count = count;

}

@Override

public String toString() {

return "university.Inventory{" +

"name='" + name + '\'' +

", count=" + count +

'}';

}

@Override

public void print() {

System.*out*.println(this.toString());

}

}

### Класс Schedule:

package university;

import interfaces.IPrintable;

import java.util.Date;

public class Schedule implements IPrintable {

private String information;

private Date date;

public Schedule(String information, Date date) {

this.information = information;

this.date = date;

}

public String getInformation() {

return information;

}

public void setInformation(String information) {

this.information = information;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

@Override

public String toString() {

return "university.Schedule{" +

"information='" + information + '\'' +

", date=" + date +

'}';

}

@Override

public void print() {

System.*out*.println(this.getDate());

System.*out*.println(this.getInformation());

}

}

### Класс University:

package university;

import abstractClass.Building;

import personType.Employee;

import personType.Student;

import java.util.ArrayList;

public final class University extends Building {

private ArrayList<Inventory> inventoryList;

private ArrayList<Student> students;

private ArrayList<Employee> employees;

private Schedule schedule;

public University(String name, String address, String customer, String building\_permit, ArrayList<Inventory> inventoryList, ArrayList<Student> students, ArrayList<Employee> employees, Schedule schedule) {

super(name, address, customer, building\_permit);

this.inventoryList = inventoryList;

this.students = students;

this.employees = employees;

this.schedule = schedule;

}

public ArrayList<Inventory> getInventoryList() {

return inventoryList;

}

public void setInventoryList(ArrayList<Inventory> inventoryList) {

this.inventoryList = inventoryList;

}

public ArrayList<Student> getStudents() {

return students;

}

public void setStudents(ArrayList<Student> students) {

this.students = students;

}

public ArrayList<Employee> getEmployees() {

return employees;

}

public void setEmployees(ArrayList<Employee> employees) {

this.employees = employees;

}

public Schedule getSchedule() {

return schedule;

}

public void setSchedule(Schedule schedule) {

this.schedule = schedule;

}

@Override

public String toString() {

return "university.University{" +

"inventoryList=" + inventoryList +

", students=" + students +

", employees=" + employees +

", schedule=" + schedule +

'}';

}

@Override

public void print() {

System.*out*.println("university.University Name: " + getName());

System.*out*.println("university.University Address: " + getAddress());

System.*out*.println("university.University Customer: " + getCustomer());

System.*out*.println("university.University abstractClass.Building Permit: " + getBuilding\_permit());

System.*out*.println("university.University university.Inventory List: " + inventoryList);

System.*out*.println("university.University Students: " + students);

System.*out*.println("university.University Employees: " + employees);

System.*out*.println("university.University university.Schedule: " + schedule);

System.*out*.println();

}

public void addStudent(Student student) {

students.add(student);

}

public void addEmployee(Employee employee) {

employees.add(employee);

}

public void addInventory(Inventory inventory) {

inventoryList.add(inventory);

}

public void removeInventory(Inventory inventory) {

inventoryList.remove(inventory);

}

public void removeStudent(Student student) {

students.remove(student);

}

public void removeEmployee(Employee employee) {

employees.remove(employee);

}

public void countList() {

System.*out*.println("Количество студентов: " + students.size());

System.*out*.println("Количество работников: " + employees.size());

System.*out*.println("Количество инвентаря: " + inventoryList.size());

}

public void printStudentsWithScholarship() {

for (Student student : students) {

if (student.isSocial\_scholarship()) {

student.print();

}

}

}

public void printSpecialtiesWithEmployees() {

for (Employee employee : employees) {

System.*out*.println(employee.getSpecialization());

}

for (Employee employee : employees) {

System.*out*.println("------------------");

System.*out*.println("Специальность: " + employee.getSpecialization());

System.*out*.println("Работники:");

for (Employee emp : employees) {

if (emp.getSpecialization().equals(employee.getSpecialization())) {

emp.print();

}

}

System.*out*.println("------------------");

}

}

public void printSpecialtiesWithStudents() {

for (Student student : students) {

System.*out*.println(student.getSpecialization());

}

for (Student student : students) {

System.*out*.println("------------------");

System.*out*.println("Специальность: " + student.getSpecialization());

System.*out*.println("Студенты:");

for (Student std : students) {

if (std.getSpecialization().equals(student.getSpecialization())) {

std.print();

}

}

System.*out*.println("------------------");

}

}

public void printStudentById(int id) {

for (Student student : students) {

if (student.getStudent\_id() == id) {

student.print();

return;

}

}

System.*out*.println("Студент с id " + id + " не найден");

}

public void printEmployeeById(int id) {

for (Employee employee : employees) {

if (employee.getEmployee\_id() == id) {

employee.print();

return;

}

}

System.*out*.println("Работник с id " + id + " не найден");

}

public void printSchedule(){

System.*out*.println(schedule.getDate());

System.*out*.println(schedule.getInformation());

}

}

### Класс Main:

import personType.Employee;

import personType.Student;

import university.Inventory;

import university.Schedule;

import university.University;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Date;

public class Main {

public static void main(String[] args) {

Employee employee1 = new Employee("Иван", "Иванов", 30, "Преподаватель", "5 лет", 100000);

Employee employee2 = new Employee("Елена", "Смирнова", 35, "Менеджер по продажам", "8 лет", 120000);

Employee employee3 = new Employee("Алексей", "Петров", 28, "Преподаватель", "3 года", 80000);

Employee employee4 = new Employee("Ольга", "Сидорова", 32, "Бухгалтер", "6 лет", 110000);

Employee employee5 = new Employee("Сергей", "Козлов", 27, "Дизайнер", "4 года", 90000);

Employee employee6 = new Employee("Наталья", "Морозова", 33, "Преподаватель", "7 лет", 100000);

Employee employee7 = new Employee("Дмитрий", "Волков", 29, "Преподаватель", "2 года", 80000);

Employee employee8 = new Employee("Мария", "Тихонова", 31, "Бухгалтер", "6 лет", 95000);

Employee employee9 = new Employee("Григорий", "Николаев", 34, "Маркетолог", "4 года", 110000);

Employee employee10 = new Employee("Екатерина", "Павлова", 26, "Преподаватель", "3 года", 100000);

Student student1 = new Student("Иван", "Иванов", 20, 1, "Информатика", false);

Student student2 = new Student("Алексей", "Смирнов", 21, 2, "Экономика", true);

Student student3 = new Student("Мария", "Петрова", 19, 1, "Математика", false);

Student student4 = new Student("Екатерина", "Сидорова", 20, 3, "Физика", true);

Student student5 = new Student("Александр", "Кузнецов", 21, 2, "Экономика", false);

Student student6 = new Student("Ольга", "Васильева", 19, 1, "Информатика", true);

Student student7 = new Student("Дмитрий", "Петров", 20, 3, "Физика", false);

Student student8 = new Student("Анна", "Смирнова", 21, 2, "Экономика", true);

Student student9 = new Student("Николай", "Иванов", 19, 1, "Математика", false);

Student student10 = new Student("Елена", "Козлова", 20, 3, "Физика", true);

Schedule schedule = new Schedule("8:00 - 9:30 \n9:40 - 11:10 \n11:30 - 13:00 \n13:10 - 14:40 \n", new Date());

ArrayList<Student> students = new ArrayList<Student>(Arrays.*asList*(student1, student2, student3, student4, student5, student6, student7, student8, student9, student10));

ArrayList<Employee> employees = new ArrayList<Employee>(Arrays.*asList*(employee1, employee2, employee3, employee4, employee5, employee6, employee7, employee8, employee9, employee10));

Inventory item1 = new Inventory("Компьютер", 100);

Inventory item2 = new Inventory("Мяч", 33);

Inventory item3 = new Inventory("Клавиатура", 100);

Inventory item4 = new Inventory("Компьютерная мышь", 100);

Inventory item5 = new Inventory("Доска", 50);

ArrayList<Inventory> inventoryArrayList = new ArrayList<Inventory>(Arrays.*asList*(item1, item2, item3, item4, item5));

University RKSI = new University("RKSI", "Тургеневская 10/6", "ООО Победа", "e314wk1-3123-r1243", inventoryArrayList, students, employees, schedule);

for (Student s :

students) {

s.print();

}

student3.rest();

student2.activity();

student4.work();

for (Employee e :

employees) {

e.print();

}

employee2.rest();

employee5.activity();

employee6.work();

employee5.remark(student1);

employee4.remark(students);

System.*out*.println(RKSI);

RKSI.print();

RKSI.printStudentsWithScholarship();

RKSI.printEmployeeById(200005);

RKSI.printSpecialtiesWithStudents();

RKSI.printSpecialtiesWithEmployees();

RKSI.printSchedule();

RKSI.countList();

RKSI.addStudent(new Student("Егор", "Погром", 33, 2, "Программист", true));

RKSI.removeEmployee(employee2);

}

}

## Вывод программы:

C:\Users\Danil\.jdks\openjdk-21.0.1\bin\java.exe -javaagent:C:\Users\Danil\AppData\Local\JetBrains\Toolbox\apps\IDEA-C\ch-0\232.10203.10\lib\idea\_rt.jar=56073:C:\Users\Danil\AppData\Local\JetBrains\Toolbox\apps\IDEA-C\ch-0\232.10203.10\bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath C:\Users\Danil\IdeaProjects\java\_oop\out\production\untitled Main

-----------------------------------------

Student ID: 100001

Name: Иван

Surname: Иванов

Age: 20

Group: 1

Specialization: Информатика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100002

Name: Алексей

Surname: Смирнов

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100003

Name: Мария

Surname: Петрова

Age: 19

Group: 1

Specialization: Математика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100004

Name: Екатерина

Surname: Сидорова

Age: 20

Group: 3

Specialization: Физика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100005

Name: Александр

Surname: Кузнецов

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100006

Name: Ольга

Surname: Васильева

Age: 19

Group: 1

Specialization: Информатика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100007

Name: Дмитрий

Surname: Петров

Age: 20

Group: 3

Specialization: Физика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100008

Name: Анна

Surname: Смирнова

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100009

Name: Николай

Surname: Иванов

Age: 19

Group: 1

Specialization: Математика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100010

Name: Елена

Surname: Козлова

Age: 20

Group: 3

Specialization: Физика

Social scholarship: true

-----------------------------------------

МарияScrolling TikToks.

АлексейTalking with friends.

ЕкатеринаStudying...

-----------------------------------------

Name: Иван

Surname: Иванов

Age: 30

Specialization: Преподаватель

Experience: 5 лет

Salary: 100000

ID:200001

-----------------------------------------

-----------------------------------------

Name: Елена

Surname: Смирнова

Age: 35

Specialization: Менеджер по продажам

Experience: 8 лет

Salary: 120000

ID:200002

-----------------------------------------

-----------------------------------------

Name: Алексей

Surname: Петров

Age: 28

Specialization: Преподаватель

Experience: 3 года

Salary: 80000

ID:200003

-----------------------------------------

-----------------------------------------

Name: Ольга

Surname: Сидорова

Age: 32

Specialization: Бухгалтер

Experience: 6 лет

Salary: 110000

ID:200004

-----------------------------------------

-----------------------------------------

Name: Сергей

Surname: Козлов

Age: 27

Specialization: Дизайнер

Experience: 4 года

Salary: 90000

ID:200005

-----------------------------------------

-----------------------------------------

Name: Наталья

Surname: Морозова

Age: 33

Specialization: Преподаватель

Experience: 7 лет

Salary: 100000

ID:200006

-----------------------------------------

-----------------------------------------

Name: Дмитрий

Surname: Волков

Age: 29

Specialization: Преподаватель

Experience: 2 года

Salary: 80000

ID:200007

-----------------------------------------

-----------------------------------------

Name: Мария

Surname: Тихонова

Age: 31

Specialization: Бухгалтер

Experience: 6 лет

Salary: 95000

ID:200008

-----------------------------------------

-----------------------------------------

Name: Григорий

Surname: Николаев

Age: 34

Specialization: Маркетолог

Experience: 4 года

Salary: 110000

ID:200009

-----------------------------------------

-----------------------------------------

Name: Екатерина

Surname: Павлова

Age: 26

Specialization: Преподаватель

Experience: 3 года

Salary: 100000

ID:200010

-----------------------------------------

ЕленаDrinking coffee.

СергейTalking with colleagues.

НатальяTeaching...

Сотрудник Сергей Козлов сделал зачемание студенту: Иван Иванов

Сотрудник Ольга Сидорова сделал зачемание студенту: Иван Иванов

Сотрудник Ольга Сидорова сделал зачемание студенту: Алексей Смирнов

Сотрудник Ольга Сидорова сделал зачемание студенту: Мария Петрова

Сотрудник Ольга Сидорова сделал зачемание студенту: Екатерина Сидорова

Сотрудник Ольга Сидорова сделал зачемание студенту: Александр Кузнецов

Сотрудник Ольга Сидорова сделал зачемание студенту: Ольга Васильева

Сотрудник Ольга Сидорова сделал зачемание студенту: Дмитрий Петров

Сотрудник Ольга Сидорова сделал зачемание студенту: Анна Смирнова

Сотрудник Ольга Сидорова сделал зачемание студенту: Николай Иванов

Сотрудник Ольга Сидорова сделал зачемание студенту: Елена Козлова

university.University{inventoryList=[university.Inventory{name='Компьютер', count=100}, university.Inventory{name='Мяч', count=33}, university.Inventory{name='Клавиатура', count=100}, university.Inventory{name='Компьютерная мышь', count=100}, university.Inventory{name='Доска', count=50}], students=[humans.Student{group=1, specialization='Информатика', social\_scholarship=false} Person{name='Иван', surname='Иванов', age=20}, humans.Student{group=2, specialization='Экономика', social\_scholarship=true} Person{name='Алексей', surname='Смирнов', age=21}, humans.Student{group=1, specialization='Математика', social\_scholarship=false} Person{name='Мария', surname='Петрова', age=19}, humans.Student{group=3, specialization='Физика', social\_scholarship=true} Person{name='Екатерина', surname='Сидорова', age=20}, humans.Student{group=2, specialization='Экономика', social\_scholarship=false} Person{name='Александр', surname='Кузнецов', age=21}, humans.Student{group=1, specialization='Информатика', social\_scholarship=true} Person{name='Ольга', surname='Васильева', age=19}, humans.Student{group=3, specialization='Физика', social\_scholarship=false} Person{name='Дмитрий', surname='Петров', age=20}, humans.Student{group=2, specialization='Экономика', social\_scholarship=true} Person{name='Анна', surname='Смирнова', age=21}, humans.Student{group=1, specialization='Математика', social\_scholarship=false} Person{name='Николай', surname='Иванов', age=19}, humans.Student{group=3, specialization='Физика', social\_scholarship=true} Person{name='Елена', surname='Козлова', age=20}], employees=[humans.Employee{specialization='Преподаватель', experience='5 лет', salary=100000}, humans.Employee{specialization='Менеджер по продажам', experience='8 лет', salary=120000}, humans.Employee{specialization='Преподаватель', experience='3 года', salary=80000}, humans.Employee{specialization='Бухгалтер', experience='6 лет', salary=110000}, humans.Employee{specialization='Дизайнер', experience='4 года', salary=90000}, humans.Employee{specialization='Преподаватель', experience='7 лет', salary=100000}, humans.Employee{specialization='Преподаватель', experience='2 года', salary=80000}, humans.Employee{specialization='Бухгалтер', experience='6 лет', salary=95000}, humans.Employee{specialization='Маркетолог', experience='4 года', salary=110000}, humans.Employee{specialization='Преподаватель', experience='3 года', salary=100000}], schedule=university.Schedule{information='8:00 - 9:30

9:40 - 11:10

11:30 - 13:00

13:10 - 14:40

', date=Sat Oct 28 22:48:04 MSK 2023}}

university.University Name: RKSI

university.University Address: Тургеневская 10/6

university.University Customer: ООО Победа

university.University abstractClass.Building Permit: e314wk1-3123-r1243

university.University university.Inventory List: [university.Inventory{name='Компьютер', count=100}, university.Inventory{name='Мяч', count=33}, university.Inventory{name='Клавиатура', count=100}, university.Inventory{name='Компьютерная мышь', count=100}, university.Inventory{name='Доска', count=50}]

university.University Students: [humans.Student{group=1, specialization='Информатика', social\_scholarship=false} Person{name='Иван', surname='Иванов', age=20}, humans.Student{group=2, specialization='Экономика', social\_scholarship=true} Person{name='Алексей', surname='Смирнов', age=21}, humans.Student{group=1, specialization='Математика', social\_scholarship=false} Person{name='Мария', surname='Петрова', age=19}, humans.Student{group=3, specialization='Физика', social\_scholarship=true} Person{name='Екатерина', surname='Сидорова', age=20}, humans.Student{group=2, specialization='Экономика', social\_scholarship=false} Person{name='Александр', surname='Кузнецов', age=21}, humans.Student{group=1, specialization='Информатика', social\_scholarship=true} Person{name='Ольга', surname='Васильева', age=19}, humans.Student{group=3, specialization='Физика', social\_scholarship=false} Person{name='Дмитрий', surname='Петров', age=20}, humans.Student{group=2, specialization='Экономика', social\_scholarship=true} Person{name='Анна', surname='Смирнова', age=21}, humans.Student{group=1, specialization='Математика', social\_scholarship=false} Person{name='Николай', surname='Иванов', age=19}, humans.Student{group=3, specialization='Физика', social\_scholarship=true} Person{name='Елена', surname='Козлова', age=20}]

university.University Employees: [humans.Employee{specialization='Преподаватель', experience='5 лет', salary=100000}, humans.Employee{specialization='Менеджер по продажам', experience='8 лет', salary=120000}, humans.Employee{specialization='Преподаватель', experience='3 года', salary=80000}, humans.Employee{specialization='Бухгалтер', experience='6 лет', salary=110000}, humans.Employee{specialization='Дизайнер', experience='4 года', salary=90000}, humans.Employee{specialization='Преподаватель', experience='7 лет', salary=100000}, humans.Employee{specialization='Преподаватель', experience='2 года', salary=80000}, humans.Employee{specialization='Бухгалтер', experience='6 лет', salary=95000}, humans.Employee{specialization='Маркетолог', experience='4 года', salary=110000}, humans.Employee{specialization='Преподаватель', experience='3 года', salary=100000}]

university.University university.Schedule: university.Schedule{information='8:00 - 9:30

9:40 - 11:10

11:30 - 13:00

13:10 - 14:40

', date=Sat Oct 28 22:48:04 MSK 2023}

-----------------------------------------

Student ID: 100002

Name: Алексей

Surname: Смирнов

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100004

Name: Екатерина

Surname: Сидорова

Age: 20

Group: 3

Specialization: Физика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100006

Name: Ольга

Surname: Васильева

Age: 19

Group: 1

Specialization: Информатика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100008

Name: Анна

Surname: Смирнова

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100010

Name: Елена

Surname: Козлова

Age: 20

Group: 3

Specialization: Физика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Name: Сергей

Surname: Козлов

Age: 27

Specialization: Дизайнер

Experience: 4 года

Salary: 90000

ID:200005

-----------------------------------------

Информатика

Экономика

Математика

Физика

Экономика

Информатика

Физика

Экономика

Математика

Физика

------------------

Специальность: Информатика

Студенты:

-----------------------------------------

Student ID: 100001

Name: Иван

Surname: Иванов

Age: 20

Group: 1

Specialization: Информатика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100006

Name: Ольга

Surname: Васильева

Age: 19

Group: 1

Specialization: Информатика

Social scholarship: true

-----------------------------------------

------------------

------------------

Специальность: Экономика

Студенты:

-----------------------------------------

Student ID: 100002

Name: Алексей

Surname: Смирнов

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100005

Name: Александр

Surname: Кузнецов

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100008

Name: Анна

Surname: Смирнова

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: true

-----------------------------------------

------------------

------------------

Специальность: Математика

Студенты:

-----------------------------------------

Student ID: 100003

Name: Мария

Surname: Петрова

Age: 19

Group: 1

Specialization: Математика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100009

Name: Николай

Surname: Иванов

Age: 19

Group: 1

Specialization: Математика

Social scholarship: false

-----------------------------------------

------------------

------------------

Специальность: Физика

Студенты:

-----------------------------------------

Student ID: 100004

Name: Екатерина

Surname: Сидорова

Age: 20

Group: 3

Specialization: Физика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100007

Name: Дмитрий

Surname: Петров

Age: 20

Group: 3

Specialization: Физика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100010

Name: Елена

Surname: Козлова

Age: 20

Group: 3

Specialization: Физика

Social scholarship: true

-----------------------------------------

------------------

------------------

Специальность: Экономика

Студенты:

-----------------------------------------

Student ID: 100002

Name: Алексей

Surname: Смирнов

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100005

Name: Александр

Surname: Кузнецов

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100008

Name: Анна

Surname: Смирнова

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: true

-----------------------------------------

------------------

------------------

Специальность: Информатика

Студенты:

-----------------------------------------

Student ID: 100001

Name: Иван

Surname: Иванов

Age: 20

Group: 1

Specialization: Информатика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100006

Name: Ольга

Surname: Васильева

Age: 19

Group: 1

Specialization: Информатика

Social scholarship: true

-----------------------------------------

------------------

------------------

Специальность: Физика

Студенты:

-----------------------------------------

Student ID: 100004

Name: Екатерина

Surname: Сидорова

Age: 20

Group: 3

Specialization: Физика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100007

Name: Дмитрий

Surname: Петров

Age: 20

Group: 3

Specialization: Физика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100010

Name: Елена

Surname: Козлова

Age: 20

Group: 3

Specialization: Физика

Social scholarship: true

-----------------------------------------

------------------

------------------

Специальность: Экономика

Студенты:

-----------------------------------------

Student ID: 100002

Name: Алексей

Surname: Смирнов

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100005

Name: Александр

Surname: Кузнецов

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100008

Name: Анна

Surname: Смирнова

Age: 21

Group: 2

Specialization: Экономика

Social scholarship: true

-----------------------------------------

------------------

------------------

Специальность: Математика

Студенты:

-----------------------------------------

Student ID: 100003

Name: Мария

Surname: Петрова

Age: 19

Group: 1

Specialization: Математика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100009

Name: Николай

Surname: Иванов

Age: 19

Group: 1

Specialization: Математика

Social scholarship: false

-----------------------------------------

------------------

------------------

Специальность: Физика

Студенты:

-----------------------------------------

Student ID: 100004

Name: Екатерина

Surname: Сидорова

Age: 20

Group: 3

Specialization: Физика

Social scholarship: true

-----------------------------------------

-----------------------------------------

Student ID: 100007

Name: Дмитрий

Surname: Петров

Age: 20

Group: 3

Specialization: Физика

Social scholarship: false

-----------------------------------------

-----------------------------------------

Student ID: 100010

Name: Елена

Surname: Козлова

Age: 20

Group: 3

Specialization: Физика

Social scholarship: true

-----------------------------------------

------------------

Преподаватель

Менеджер по продажам

Преподаватель

Бухгалтер

Дизайнер

Преподаватель

Преподаватель

Бухгалтер

Маркетолог

Преподаватель

------------------

Специальность: Преподаватель

Работники:

-----------------------------------------

Name: Иван

Surname: Иванов

Age: 30

Specialization: Преподаватель

Experience: 5 лет

Salary: 100000

ID:200001

-----------------------------------------

-----------------------------------------

Name: Алексей

Surname: Петров

Age: 28

Specialization: Преподаватель

Experience: 3 года

Salary: 80000

ID:200003

-----------------------------------------

-----------------------------------------

Name: Наталья

Surname: Морозова

Age: 33

Specialization: Преподаватель

Experience: 7 лет

Salary: 100000

ID:200006

-----------------------------------------

-----------------------------------------

Name: Дмитрий

Surname: Волков

Age: 29

Specialization: Преподаватель

Experience: 2 года

Salary: 80000

ID:200007

-----------------------------------------

-----------------------------------------

Name: Екатерина

Surname: Павлова

Age: 26

Specialization: Преподаватель

Experience: 3 года

Salary: 100000

ID:200010

-----------------------------------------

------------------

------------------

Специальность: Менеджер по продажам

Работники:

-----------------------------------------

Name: Елена

Surname: Смирнова

Age: 35

Specialization: Менеджер по продажам

Experience: 8 лет

Salary: 120000

ID:200002

-----------------------------------------

------------------

------------------

Специальность: Преподаватель

Работники:

-----------------------------------------

Name: Иван

Surname: Иванов

Age: 30

Specialization: Преподаватель

Experience: 5 лет

Salary: 100000

ID:200001

-----------------------------------------

-----------------------------------------

Name: Алексей

Surname: Петров

Age: 28

Specialization: Преподаватель

Experience: 3 года

Salary: 80000

ID:200003

-----------------------------------------

-----------------------------------------

Name: Наталья

Surname: Морозова

Age: 33

Specialization: Преподаватель

Experience: 7 лет

Salary: 100000

ID:200006

-----------------------------------------

-----------------------------------------

Name: Дмитрий

Surname: Волков

Age: 29

Specialization: Преподаватель

Experience: 2 года

Salary: 80000

ID:200007

-----------------------------------------

-----------------------------------------

Name: Екатерина

Surname: Павлова

Age: 26

Specialization: Преподаватель

Experience: 3 года

Salary: 100000

ID:200010

-----------------------------------------

------------------

------------------

Специальность: Бухгалтер

Работники:

-----------------------------------------

Name: Ольга

Surname: Сидорова

Age: 32

Specialization: Бухгалтер

Experience: 6 лет

Salary: 110000

ID:200004

-----------------------------------------

-----------------------------------------

Name: Мария

Surname: Тихонова

Age: 31

Specialization: Бухгалтер

Experience: 6 лет

Salary: 95000

ID:200008

-----------------------------------------

------------------

------------------

Специальность: Дизайнер

Работники:

-----------------------------------------

Name: Сергей

Surname: Козлов

Age: 27

Specialization: Дизайнер

Experience: 4 года

Salary: 90000

ID:200005

-----------------------------------------

------------------

------------------

Специальность: Преподаватель

Работники:

-----------------------------------------

Name: Иван

Surname: Иванов

Age: 30

Specialization: Преподаватель

Experience: 5 лет

Salary: 100000

ID:200001

-----------------------------------------

-----------------------------------------

Name: Алексей

Surname: Петров

Age: 28

Specialization: Преподаватель

Experience: 3 года

Salary: 80000

ID:200003

-----------------------------------------

-----------------------------------------

Name: Наталья

Surname: Морозова

Age: 33

Specialization: Преподаватель

Experience: 7 лет

Salary: 100000

ID:200006

-----------------------------------------

-----------------------------------------

Name: Дмитрий

Surname: Волков

Age: 29

Specialization: Преподаватель

Experience: 2 года

Salary: 80000

ID:200007

-----------------------------------------

-----------------------------------------

Name: Екатерина

Surname: Павлова

Age: 26

Specialization: Преподаватель

Experience: 3 года

Salary: 100000

ID:200010

-----------------------------------------

------------------

------------------

Специальность: Преподаватель

Работники:

-----------------------------------------

Name: Иван

Surname: Иванов

Age: 30

Specialization: Преподаватель

Experience: 5 лет

Salary: 100000

ID:200001

-----------------------------------------

-----------------------------------------

Name: Алексей

Surname: Петров

Age: 28

Specialization: Преподаватель

Experience: 3 года

Salary: 80000

ID:200003

-----------------------------------------

-----------------------------------------

Name: Наталья

Surname: Морозова

Age: 33

Specialization: Преподаватель

Experience: 7 лет

Salary: 100000

ID:200006

-----------------------------------------

-----------------------------------------

Name: Дмитрий

Surname: Волков

Age: 29

Specialization: Преподаватель

Experience: 2 года

Salary: 80000

ID:200007

-----------------------------------------

-----------------------------------------

Name: Екатерина

Surname: Павлова

Age: 26

Specialization: Преподаватель

Experience: 3 года

Salary: 100000

ID:200010

-----------------------------------------

------------------

------------------

Специальность: Бухгалтер

Работники:

-----------------------------------------

Name: Ольга

Surname: Сидорова

Age: 32

Specialization: Бухгалтер

Experience: 6 лет

Salary: 110000

ID:200004

-----------------------------------------

-----------------------------------------

Name: Мария

Surname: Тихонова

Age: 31

Specialization: Бухгалтер

Experience: 6 лет

Salary: 95000

ID:200008

-----------------------------------------

------------------

------------------

Специальность: Маркетолог

Работники:

-----------------------------------------

Name: Григорий

Surname: Николаев

Age: 34

Specialization: Маркетолог

Experience: 4 года

Salary: 110000

ID:200009

-----------------------------------------

------------------

------------------

Специальность: Преподаватель

Работники:

-----------------------------------------

Name: Иван

Surname: Иванов

Age: 30

Specialization: Преподаватель

Experience: 5 лет

Salary: 100000

ID:200001

-----------------------------------------

-----------------------------------------

Name: Алексей

Surname: Петров

Age: 28

Specialization: Преподаватель

Experience: 3 года

Salary: 80000

ID:200003

-----------------------------------------

-----------------------------------------

Name: Наталья

Surname: Морозова

Age: 33

Specialization: Преподаватель

Experience: 7 лет

Salary: 100000

ID:200006

-----------------------------------------

-----------------------------------------

Name: Дмитрий

Surname: Волков

Age: 29

Specialization: Преподаватель

Experience: 2 года

Salary: 80000

ID:200007

-----------------------------------------

-----------------------------------------

Name: Екатерина

Surname: Павлова

Age: 26

Specialization: Преподаватель

Experience: 3 года

Salary: 100000

ID:200010

-----------------------------------------

------------------

Sat Oct 28 22:48:04 MSK 2023

8:00 - 9:30

9:40 - 11:10

11:30 - 13:00

13:10 - 14:40

Количество студентов: 10

Количество работников: 10

Количество инвентаря: 5

Process finished with exit code 0

## Общий вывод.

В процессе выполнения данной программы были реализованы классы и интерфейсы для моделирования университетской среды. Были созданы объекты студентов, сотрудников, инвентаря и расписания. Взаимодействие между классами и интерфейсами позволяет управлять данными университета и выполнять различные операции, такие как вывод информации о студентах, сотрудниках и университете в целом.

Программа также демонстрирует использование наследования, интерфейсов, полиморфизма и коллекций в Java. Каждый класс выполняет свою уникальную роль в моделировании университетской среды, а интерфейсы обеспечивают стандартизацию методов для вывода информации и выполнения действий.

Программа успешно создает объекты, выполняет операции с ними и выводит информацию на экран. Таким образом, она демонстрирует основные принципы объектно-ориентированного программирования и структурирования данных в Java

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды и отчет выложены на GitHub.