

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРА-
ЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики
Кафедра вычислительных технологий

ЛАБОРАТОРНАЯ РАБОТА №2
Дисциплина: Платформо-независимое программирование

Работу выполнил: _____ А. А. Костров

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

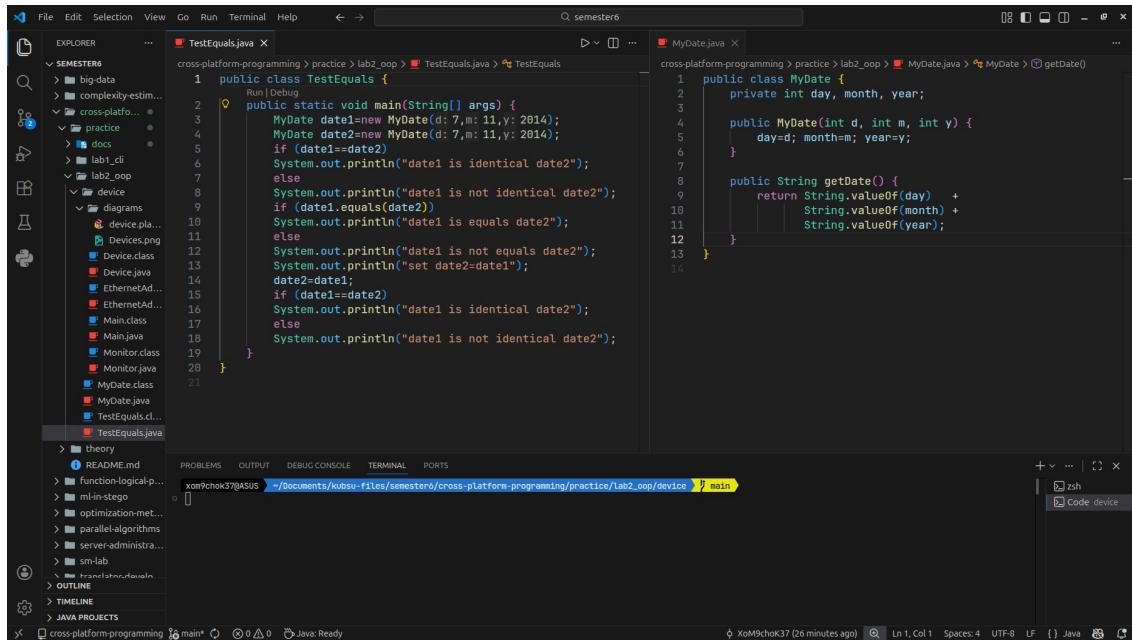
Преподаватель: _____ В. И. Шиян

Краснодар
2026

Тема. Основные концепции ООП, простейшие UML-диаграммы.

Цель. Освоить процесс построения иерархии классов на основе разработанных UML-диаграмм. Изучить синтаксис и возможности переопределённых (overriding) функций. Разработать программу согласно варианту.

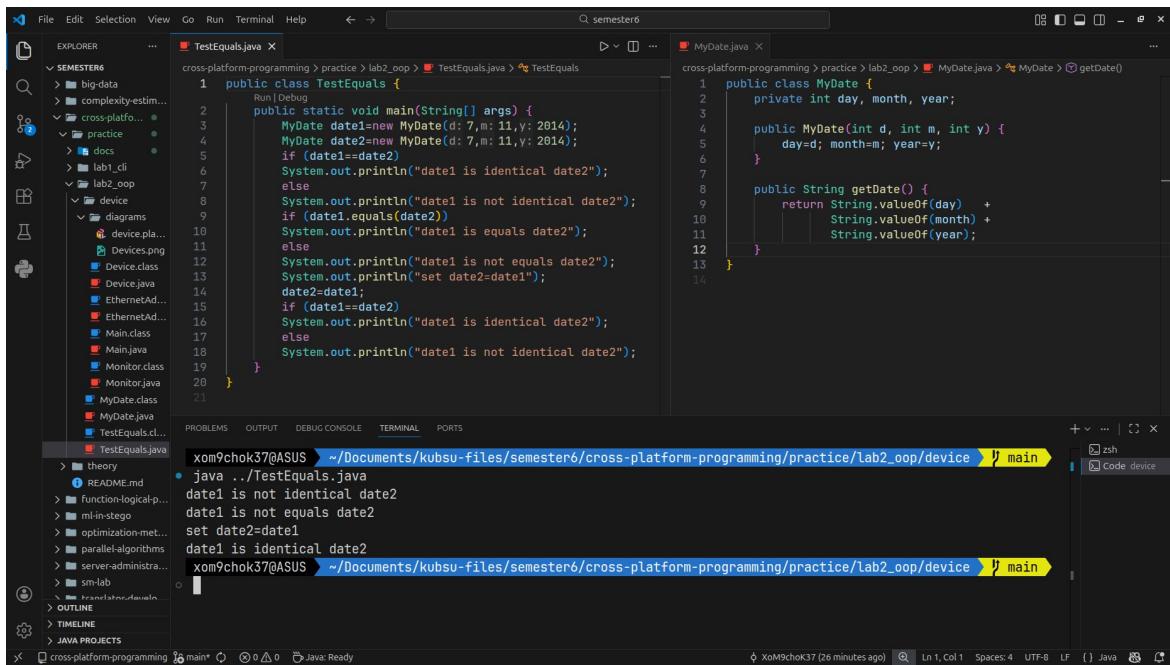
1. Скомпилируем и запустим пример 3.3 без переопределённого метода *equals* (рисунки 1-2).



```
TestEquals.java
public class TestEquals {
    public static void main(String[] args) {
        MyDate date1=new MyDate(d: 7,m: 11,y: 2014);
        MyDate date2=new MyDate(d: 7,m: 11,y: 2014);
        if (date1==date2)
            System.out.println("date1 is identical date2");
        else
            System.out.println("date1 is not identical date2");
        if (date1.equals(date2))
            System.out.println("date1 is equals date2");
        else
            System.out.println("date1 is not equals date2");
        date2=date1;
        if (date1==date2)
            System.out.println("date1 is identical date2");
        else
            System.out.println("date1 is not identical date2");
    }
}

MyDate.java
public class MyDate {
    private int day, month, year;
    public MyDate(int d, int m, int y) {
        day=d; month=m; year=y;
    }
    public String getDate() {
        return String.valueOf(day) +
               String.valueOf(month) +
               String.valueOf(year);
    }
}
```

Рисунок 1 — Напишем код программы



```
TestEquals.java
public class TestEquals {
    public static void main(String[] args) {
        MyDate date1=new MyDate(d: 7,m: 11,y: 2014);
        MyDate date2=new MyDate(d: 7,m: 11,y: 2014);
        if (date1==date2)
            System.out.println("date1 is identical date2");
        else
            System.out.println("date1 is not identical date2");
        if (date1.equals(date2))
            System.out.println("date1 is equals date2");
        else
            System.out.println("date1 is not equals date2");
        date2=date1;
        if (date1==date2)
            System.out.println("date1 is identical date2");
        else
            System.out.println("date1 is not identical date2");
    }
}

MyDate.java
public class MyDate {
    private int day, month, year;
    public MyDate(int d, int m, int y) {
        day=d; month=m; year=y;
    }
    public String getDate() {
        return String.valueOf(day) +
               String.valueOf(month) +
               String.valueOf(year);
    }
}
```

Рисунок 2 — Запустим программу

2. Разработаем иерархию классов (рисунки 3-6).

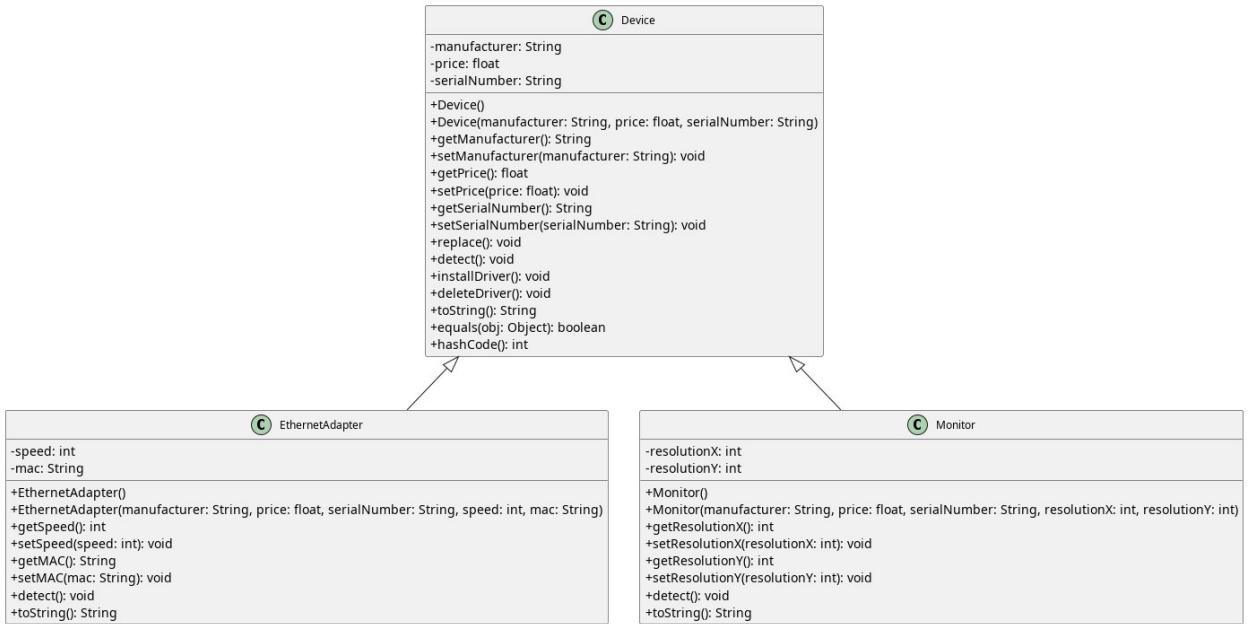


Рисунок 3 — Составим UML-диаграмму

The screenshot shows a dual-pane code editor with two instances of the `Device.java` file. The left pane displays the initial template with basic constructor, getters, and setters. The right pane shows the completed implementation with additional methods like `detect()`, `installDriver()`, `deleteDriver()`, and `toString()`, along with overridden methods `equals()` and `hashCode()`.

```

Device.java (Left)


```

public class Device {
 private String manufacturer;
 private float price;
 private String serialNumber;

 public Device() {}

 public Device(String manufacturer, float price, String serialNumber) {
 this.manufacturer = manufacturer;
 this.price = price;
 this.serialNumber = serialNumber;
 }

 public String getManufacturer() {
 return manufacturer;
 }

 public void setManufacturer(String manufacturer) {
 this.manufacturer = manufacturer;
 }

 public float getPrice() {
 return price;
 }

 public void setPrice(float price) {
 this.price = price;
 }

 public String getSerialNumber() {
 return serialNumber;
 }

 public void setSerialNumber(String serialNumber) {
 this.serialNumber = serialNumber;
 }

 public void replace() {
 System.out.println("Устройство " + manufacturer + " (SN: " + serialNumber + ") заменено");
 }

 public void detect() {
 System.out.println("Устройство " + manufacturer + " обнаружено");
 }

 public void installDriver() {
 System.out.println("Установка драйвера для устройства " + manufacturer);
 }

 public void deleteDriver() {
 System.out.println("Удаление драйвера для устройства " + manufacturer);
 }

 @Override
 public String toString() {
 return "Device [manufacturer=" + manufacturer +
 ", price=" + price +
 ", serialNumber=" + serialNumber + "]";
 }

 @Override
 public boolean equals(Object obj) {
 if (this == obj) return true;
 if (obj == null || getClass() != obj.getClass()) return false;
 Device device = (Device) obj;
 if (Float.compare(device.price, price) != 0) return false;
 if (manufacturer != null ? !manufacturer.equals(device.manufacturer) : device.manufacturer != null) return false;
 return serialNumber != null ? serialNumber.equals(device.serialNumber) : device.serialNumber == null;
 }

 @Override
 public int hashCode() {
 int result = manufacturer != null ? manufacturer.hashCode() : 0;
 result = 31 * result + (price != +0.0f ? Float.floatToIntBits(price) : 0);
 result = 31 * result + (serialNumber != null ? serialNumber.hashCode() : 0);
 return result;
 }
}

```



```

Device.java (Right)


```

public class Device {
    private String manufacturer;
    private float price;
    private String serialNumber;

    public Device() {}

    public Device(String manufacturer, float price, String serialNumber) {
        this.manufacturer = manufacturer;
        this.price = price;
        this.serialNumber = serialNumber;
    }

    public String getManufacturer() {
        return manufacturer;
    }

    public void setManufacturer(String manufacturer) {
        this.manufacturer = manufacturer;
    }

    public float getPrice() {
        return price;
    }

    public void setPrice(float price) {
        this.price = price;
    }

    public String getSerialNumber() {
        return serialNumber;
    }

    public void setSerialNumber(String serialNumber) {
        this.serialNumber = serialNumber;
    }

    public void replace() {
        System.out.println("Устройство " + manufacturer + " (SN: " + serialNumber + ") заменено");
    }

    public void detect() {
        System.out.println("Устройство " + manufacturer + " обнаружено");
    }

    public void installDriver() {
        System.out.println("Установка драйвера для устройства " + manufacturer);
    }

    public void deleteDriver() {
        System.out.println("Удаление драйвера для устройства " + manufacturer);
    }

    @Override
    public String toString() {
        return "Device [manufacturer=" + manufacturer +
               ", price=" + price +
               ", serialNumber=" + serialNumber + "]";
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Device device = (Device) obj;
        if (Float.compare(device.price, price) != 0) return false;
        if (manufacturer != null ? !manufacturer.equals(device.manufacturer) : device.manufacturer != null) return false;
        return serialNumber != null ? serialNumber.equals(device.serialNumber) : device.serialNumber == null;
    }

    @Override
    public int hashCode() {
        int result = manufacturer != null ? manufacturer.hashCode() : 0;
        result = 31 * result + (price != +0.0f ? Float.floatToIntBits(price) : 0);
        result = 31 * result + (serialNumber != null ? serialNumber.hashCode() : 0);
        return result;
    }
}

```


```


```

Рисунок 4 — Создадим класс *Device*

The screenshot shows a Java IDE interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** Standard icons for file operations like Open, Save, Find, etc.
- Search Bar:** A search bar at the top right labeled "semester6".
- Code Editor:** The main area displays the code for `EthernetAdapter.java`. The code is a class that extends `Device`, with methods for speed and MAC address.
- Explorer:** On the left, the "EXPLORER" view shows the project structure. It includes a "SEME..." folder, a "cross-platform-programming" folder containing "big-data", "complexity-estim...", "practice" (with "docs" and "lab1_cli"), "lab2_oop" (with "device" and "diagrams"), and "Devices.png". Inside "lab2_oop/device", there are "EthernetAd...", "EthernetAd...", "Main.class", "Main.java", "Monitor.class", "Monitor.java", "MyDate.class", "MyDate.java", "TestEquals.cl...", and "TestEquals.java".
- Status Bar:** At the bottom, it says "Java: Ready".

Рисунок 5 — Создадим класс *EthernetAdapter*

The screenshot shows a Java IDE interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** Standard icons for file operations like Open, Save, Find, etc.
- Search Bar:** A search bar at the top right labeled "semester6".
- Code Editor:** The main area displays the code for `Monitor.java`. The code is a class that extends `Device`, with methods for resolution X and Y.
- Explorer:** On the left, the "EXPLORER" view shows the project structure. It includes a "SEME..." folder, a "cross-platform-programming" folder containing "big-data", "complexity-estim...", "practice" (with "docs" and "lab1_cli"), "lab2_oop" (with "device" and "diagrams"), and "Devices.png". Inside "lab2_oop/device", there are "EthernetAd...", "EthernetAd...", "Main.class", "Main.java", "Monitor.class", "Monitor.java", "MyDate.class", "MyDate.java", "TestEquals.cl...", and "TestEquals.java".
- Status Bar:** At the bottom, it says "Java: Ready".

Рисунок 6 — Создадим класс *Monitor*

3. Продемонстрируем создание списка объектов классов из нашей иерархии в классе Main (рисунки 7-8).

```

File Edit Selection View Go Run Terminal Help < > semester6
EXPLORER Main.java X
cross-platform-programming > practice > lab2_oop > device > Main.java > ...
1 import java.util.ArrayList;
2
3 public class Main {
4     public static void main(String[] args) {
5         ArrayList<Device> devices = new ArrayList<>();
6
7         Monitor monitor1 = new Monitor(manufacturer: "Samsung", price: 25000.5f, serialNumber: "SN12345", resolutionX: 1920, 1080);
8         Monitor monitor2 = new Monitor(manufacturer: "LG", price: 32000.0f, serialNumber: "SN67890", resolutionX: 2560, 1440);
9         Monitor monitor3 = new Monitor(manufacturer: "Samsung", price: 25000.5f, serialNumber: "SN12345", resolutionX: 1920, 1080);
10
11        EthernetAdapter adapter1 = new EthernetAdapter(manufacturer: "TP-Link", price: 1500.0f, serialNumber: "EA12345", speed: 1000, mac: "00:1A:2B:3C:4D:5E");
12        EthernetAdapter adapter2 = new EthernetAdapter(manufacturer: "D-Link", price: 1200.0f, serialNumber: "EA67890", speed: 100, mac: "00:1B:3C:4D:5E:6F");
13        EthernetAdapter adapter3 = new EthernetAdapter(manufacturer: "TP-Link", price: 1500.0f, serialNumber: "EA12345", speed: 1000, mac: "00:1A:2B:3C:4D:5E");
14
15        addUnique(devices, monitor1);
16        addUnique(devices, monitor2);
17        addUnique(devices, monitor3);
18        addUnique(devices, adapter1);
19        addUnique(devices, adapter2);
20        addUnique(devices, adapter3);
21
22        System.out.println("== Список уникальных устройств ===");
23        System.out.println("Всего устройств: " + devices.size() + "\n");
24
25        for (int i = 0; i < devices.size(); i++) {
26            System.out.print("Устройство " + (i + 1) + ":");
27            Device device = devices.get(i);
28
29            if (device instanceof Monitor) {
30                Monitor monitor = (Monitor) device;
31                System.out.println(" Тип: Монитор");
32                System.out.println(" Производитель: " + monitor.getManufacturer());
33                System.out.println(" Цена: " + monitor.getPrice());
34                System.out.println(" Серийный номер: " + monitor.getSerialNumber());
35                System.out.println(" Разрешение: " + monitor.getResolutionX() + "x" + monitor.getResolutionY());
36
37                monitor.setPrice(monitor.getPrice() * 0.9f);
38                System.out.println(" Цена со скидкой: " + monitor.getPrice());
39
40            } else if (device instanceof EthernetAdapter) {
41                EthernetAdapter adapter = (EthernetAdapter) device;
42                System.out.println(" Тип: Сетевой адаптер");
43                System.out.println(" Производитель: " + adapter.getManufacturer());
44                System.out.println(" Цена: " + adapter.getPrice());
45                System.out.println(" Серийный номер: " + adapter.getSerialNumber());
46                System.out.println(" Скорость: " + adapter.getSpeed() + " Мбит/с");
47                System.out.println(" MAC-адрес: " + adapter.getMac());
48
49            }
50        }
51    }
52
53 }

```

Рисунок 7 — Создадим класс Main

```

File Edit Selection View Go Run Terminal Help < > semester6
EXPLORER Main.java X
cross-platform-programming > practice > lab2_oop > device > Main.java > ...
1 import java.util.ArrayList;
2
3 public class Main {
4     public static void main(String[] args) {
5         ArrayList<Device> devices = new ArrayList<>();
6
7         Monitor monitor1 = new Monitor(manufacturer: "Samsung", price: 25000.5f, serialNumber: "SN12345", resolutionX: 1920, 1080);
8         Monitor monitor2 = new Monitor(manufacturer: "LG", price: 32000.0f, serialNumber: "SN67890", resolutionX: 2560, 1440);
9         Monitor monitor3 = new Monitor(manufacturer: "Samsung", price: 25000.5f, serialNumber: "SN12345", resolutionX: 1920, 1080);
10
11        EthernetAdapter adapter1 = new EthernetAdapter(manufacturer: "TP-Link", price: 1500.0f, serialNumber: "EA12345", speed: 1000, mac: "00:1A:2B:3C:4D:5E");
12        EthernetAdapter adapter2 = new EthernetAdapter(manufacturer: "D-Link", price: 1200.0f, serialNumber: "EA67890", speed: 100, mac: "00:1B:3C:4D:5E:6F");
13        EthernetAdapter adapter3 = new EthernetAdapter(manufacturer: "TP-Link", price: 1500.0f, serialNumber: "EA12345", speed: 1000, mac: "00:1A:2B:3C:4D:5E");
14
15        addUnique(devices, monitor1);
16        addUnique(devices, monitor2);
17        addUnique(devices, monitor3);
18        addUnique(devices, adapter1);
19        addUnique(devices, adapter2);
20        addUnique(devices, adapter3);
21
22        System.out.println("== Список уникальных устройств ===");
23        System.out.println("Всего устройств: " + devices.size() + "\n");
24
25        for (int i = 0; i < devices.size(); i++) {
26            System.out.print("Устройство " + (i + 1) + ":");
27            Device device = devices.get(i);
28
29            if (device instanceof Monitor) {
30                Monitor monitor = (Monitor) device;
31                System.out.println(" Тип: Монитор");
32                System.out.println(" Производитель: " + monitor.getManufacturer());
33                System.out.println(" Цена: " + monitor.getPrice());
34                System.out.println(" Серийный номер: " + monitor.getSerialNumber());
35                System.out.println(" Разрешение: " + monitor.getResolutionX() + "x" + monitor.getResolutionY());
36
37                monitor.setPrice(monitor.getPrice() * 0.9f);
38                System.out.println(" Цена со скидкой: " + monitor.getPrice());
39
40            } else if (device instanceof EthernetAdapter) {
41                EthernetAdapter adapter = (EthernetAdapter) device;
42                System.out.println(" Тип: Сетевой адаптер");
43                System.out.println(" Производитель: " + adapter.getManufacturer());
44                System.out.println(" Цена: " + adapter.getPrice());
45                System.out.println(" Серийный номер: " + adapter.getSerialNumber());
46                System.out.println(" Скорость: " + adapter.getSpeed() + " Мбит/с");
47                System.out.println(" MAC-адрес: " + adapter.getMac());
48
49            }
50        }
51    }
52
53 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

xom9chok37@ASUS ~ /~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab2_oop/device > main && javac Main.java && java Main
==== Проверка уникальности ====
Было создано 6 объектов (3 монитора и 3 адаптера)
В массив добавлено только 4 уникальных объектов
Дубликаты были отсеяны методом equals()
xom9chok37@ASUS ~ /~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab2_oop/device > main

```

Рисунок 8 — Запустим программу