

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”  
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Отчёт  
по лабораторной работе №5

Выполнил:  
студент группы ПО-9  
Качаловский Данил Сергеевич

Проверил:  
Крощенко А. А.

Брест 2024

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

### **Вариант 7**

#### **Задание 1**

Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов

interface Корабль ← abstract class Военный Корабль ← class Авианосец

#### **Код программы (файл Ship.java)**

```
package Lab5_1;

public interface Ship {

    String getName();

    String getType();

    void move(int x, int y);

    void attack(Ship target);

}
```

#### **Код программы (файл WarShip.java)**

```
package Lab5_1;

public abstract class WarShip implements Ship {

    private String name;
    private String type;

    public WarShip(String name, String type) {
        this.name = name;
        this.type = type;
    }

    @Override
    public String getName() {
        return name;
    }

    @Override
    public String getType() {
        return type;
    }

    @Override
    public abstract void move(int x, int y);

    @Override
    public abstract void attack(Ship target);

}
```

**Код программы (файл AircraftCarrier.java)**

```
package Lab5_1;

public class AircraftCarrier extends WarShip {

    private int numberOfAircraft;

    public AircraftCarrier(String name, int numberOfAircraft) {
        super(name, "Aircraft Carrier");
        this.numberOfAircraft = numberOfAircraft;
    }

    @Override
    public void move(int x, int y) {
        System.out.println("Авианосец " + getName() + " перемещается в (" + x
+ "," + y + ")");
    }

    @Override
    public void attack(Ship target) {
        System.out.println("Авианосец " + getName() + " атакует " +
target.getName());
    }

    public int getNumberOfAircraft() {
        return numberOfAircraft;
    }

}
```

**Код программы (файл Destroyer.java)**

```
package Lab5_1;

public class Destroyer extends WarShip {

    private String name;

    public Destroyer(String name) {
        super(name, "Destroyer");
        this.name = name;
    }

    @Override
    public String getName() {
        return name;
    }

    @Override
    public String getType() {
        return "Destroyer";
    }

    @Override
    public void move(int x, int y) {
```

```

        System.out.println("Уничтожитель " + getName() + " перемещается в ("
+ x + "," + y + ")");
    }

    @Override
    public void attack(Ship target) {
        System.out.println("Уничтожитель " + getName() + " атакует " +
target.getName());
    }
}

```

**Код программы (файл Lab5\_1.java)**

```

package Lab5_1;

public class Lab5_1 {

    public static void main(String[] args) {
        AircraftCarrier carrier = new AircraftCarrier("Корабль", 90);

        carrier.move(10, 20);
        Destroyer enemy = new Destroyer("Враг");
        carrier.attack(enemy);

        System.out.println("Имя: " + carrier.getName());
        System.out.println("Тип: " + carrier.getType());
        System.out.println("Количество авианосцев: " +
carrier.getNumberOfAircraft());
    }
}

```

**Вывод**

D:\Libs\openjdk-21.0.2\bin\java.exe "-jav

Авианосец Корабль перемещается в (10,20)

Авианосец Корабль атакует Враг

Имя: Корабль

Тип: Aircraft Carrier

Количество авианосцев: 90

## Задание 2

В следующих заданиях требуется создать суперкласс (абстрактный класс, интерфейс) и определить общие методы для данного класса. Создать подклассы, в которых добавить специфические свойства и методы. Часть методов переопределить. Создать массив объектов суперкласса и заполнить объектами подклассов. Объекты подклассов идентифицировать конструктором по имени или идентификационному номеру. Использовать объекты подклассов для моделирования реальных ситуаций и объектов.

Создать базовый класс Садовое дерево и производные классы Яблоня, Вишня, Груша и другие. С помощью конструктора автоматически становить номер каждого дерева. Принять решение о пересадке каждого дерева в зависимости от возраста и плодоношения.

### Код программы (GardenTree.java)

```
package Lab5_2;

public abstract class GardenTree {

    private static int nextId = 1;
    private int id;
    private int age;
    private boolean isFruiting;

    public GardenTree() {
        this.id = nextId++;
        this.age = 0;
        this.isFruiting = false;
    }

    public int getId() {
        return id;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public boolean isFruiting() {
        return isFruiting;
    }

    public void setFruiting(boolean isFruiting) {
        this.isFruiting = isFruiting;
    }

    public abstract void grow();

    public abstract boolean needsReplanting();

    @Override
    public String toString() {
```

```

        return "Садовое дерево #" + id + " (" + getAge() + " лет, " +
(isFruiting ? "плодоносит" : "не плодоносит") + ")";
    }

}

```

#### Код программы (AppleTree.java)

```

package Lab5_2;

public class AppleTree extends GardenTree {

    private String variety;

    public AppleTree(String variety) {
        super();
        this.variety = variety;
    }

    @Override
    public void grow() {
        setAge(getAge() + 1);
        if (getAge() >= 5) {
            setFruiting(true);
        }
    }

    @Override
    public boolean needsReplanting() {
        return getAge() >= 25;
    }

    @Override
    public String toString() {
        return super.toString() + " - Яблоня (" + variety + ")";
    }

}

```

#### Код программы (CherryTree.java)

```

package Lab5_2;

public class CherryTree extends GardenTree {

    public CherryTree() {
        super();
    }

    @Override
    public void grow() {
        setAge(getAge() + 1);
        if (getAge() >= 4) {
            setFruiting(true);
        }
    }

}

```

```

    }

    @Override
    public boolean needsReplanting() {
        return getAge() >= 20;
    }

    @Override
    public String toString() {
        return super.toString() + " - Вишня";
    }
}

```

#### Код программы (PearTree.java)

```

package Lab5_2;

public class PearTree extends GardenTree {

    private String variety;

    public PearTree(String variety) {
        super();
        this.variety = variety;
    }

    @Override
    public void grow() {
        setAge(getAge() + 1);
        if (getAge() >= 6) {
            setFruiting(true);
        }
    }

    @Override
    public boolean needsReplanting() {
        return getAge() >= 30;
    }

    @Override
    public String toString() {
        return super.toString() + " - Груша (" + variety + ")";
    }
}

```

#### Код программы (Lab5\_2.java)

```

package Lab5_2;

public class Lab5_2 {
    public static void main(String[] args) {
        GardenTree[] trees = new GardenTree[3];
        trees[0] = new AppleTree("Антоновка");
        trees[1] = new CherryTree();
    }
}

```

```

trees[2] = new PearTree("Дюшес");

for (int i = 1; i < 10; i++) {
    for (GardenTree tree : trees) {
        tree.grow();
        System.out.println(tree);
    }
}

System.out.println("Деревья, которые нужно пересадить:");
for (GardenTree tree : trees) {
    if (tree.needsReplanting()) {
        System.out.println(tree);
    }
}
}
}

```

#### Вывод

[D:\Libs\openjdk-21.0.2\bin\java.exe](#) "-javaagent:D:\JetBrains\Int

```

Садовое дерево #1 (1 лет, не плодоносит) - Яблоня (Антоновка)
Садовое дерево #2 (1 лет, не плодоносит) - Вишня
Садовое дерево #3 (1 лет, не плодоносит) - Груша (Дюшес)
Садовое дерево #1 (2 лет, не плодоносит) - Яблоня (Антоновка)
Садовое дерево #2 (2 лет, не плодоносит) - Вишня
Садовое дерево #3 (2 лет, не плодоносит) - Груша (Дюшес)
Садовое дерево #1 (3 лет, не плодоносит) - Яблоня (Антоновка)
Садовое дерево #2 (3 лет, не плодоносит) - Вишня
Садовое дерево #3 (3 лет, не плодоносит) - Груша (Дюшес)
Садовое дерево #1 (4 лет, не плодоносит) - Яблоня (Антоновка)
Садовое дерево #2 (4 лет, плодоносит) - Вишня
Садовое дерево #3 (4 лет, не плодоносит) - Груша (Дюшес)
Садовое дерево #1 (5 лет, плодоносит) - Яблоня (Антоновка)
Садовое дерево #2 (5 лет, плодоносит) - Вишня
Садовое дерево #3 (5 лет, не плодоносит) - Груша (Дюшес)

```