**Лабораторная работа №16**

**Основы программирования на Java: проектирование иерархии классов. Наследование и полиморфизм**

**Цель работы:** Освоить основные способы создания Java-программ либо с помощью обычного редактора, либо с помощью среды разработки. Приобрести навыки проектирования иерархии классов в Java, следуя принципам наследования и полиморфизма.

**Задание 1**

17. Создать объект класса Дерево, используя классы Лист, Ветка. Методы: зацвести, опасть листьям, покрыться инеем, пожелтеть листьям.

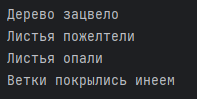
// Класс Лист  
class Leaf {  
 public void turnYellow() {  
 System.*out*.println("Листья пожелтели");  
 }  
  
 public void fall() {  
 System.*out*.println("Листья опали");  
 }  
  
 public void blossom() {  
 System.*out*.println("Дерево зацвело");  
 }  
}

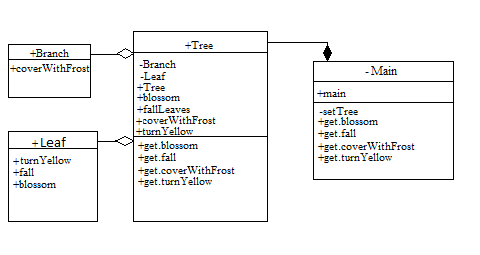
// Класс Ветка  
class Branch {  
 public void coverWithFrost() {  
 System.*out*.println("Ветки покрылись инеем");  
 }  
}

// Класс Дерево  
class Tree {  
 private Branch branch;  
 private Leaf leaf;  
  
 public Tree() {  
 this.branch = new Branch();  
 this.leaf = new Leaf();  
 }  
  
 public void blossom() {  
 leaf.blossom();  
 }  
  
 public void fallLeaves() {  
 leaf.fall();  
 }  
  
 public void coverWithFrost() {  
 branch.coverWithFrost();  
 }  
  
 public void turnYellow() {  
 leaf.turnYellow();  
 }  
}

public class Main {  
 public static void main(String[] args) {  
 Tree tree = new Tree();  
 tree.blossom();  
 tree.turnYellow();  
 tree.fallLeaves();  
 tree.coverWithFrost();  
 }  
}

Вывод





**Контрольные вопросы**

**1.** Класс и объект

* Класс - это шаблон, который определяет структуру и поведение объектов. Он содержит свойства (данные) и методы (действия) объектов.
* Объект - это экземпляр класса. Он имеет конкретные значения свойств и может выполнять методы, определенные в классе.

**2.** Наследование и полиморфизм

Наследование

* Позволяет создавать новые классы (дочерние классы) на основе существующих классов (родительских классов).
* Дочерние классы наследуют свойства и методы родительских классов, а также могут добавлять свои собственные.

Полиморфизм

* Способность объектов разных классов вести себя по-разному в ответ на один и тот же метод.
* Достигается за счет переопределения методов в дочерних классах.
* Позволяет обрабатывать объекты разных типов как объекты одного общего типа (например, родительского класса).

**3.** Дан класс:

package ch04.q01;

class Quest41 {}

Укажите правильные варианты наследования от этого класса (2):

1) package ch04.q01; class Quest4 extends Quest41 {}

2) package ch04.q01.\_2; public class Quest42 extends Quest41 {}

3) package ch04.q01; public class Quest43 implements Quest41 {}

4) package ch04.q01\_\_2; import ch04.q01.Quest41;

5) public class Quest44 extends Quest41 {}

6) package ch04.q01; public class Quest45 extends Quest41 {}

Ответ: 1, 3

**4.** Выберите правильные утверждения (3):

1) Класс может быть использован в качестве суперкласса для себя самого.

2) В конструкторе класса можно совместно использовать вызовы this и super.

3) Статические методы можно определять в подклассах с той же сигнатурой,

что и в базовом классе.

4) Статические методы можно перегружать в подклассах.

5) Динамическое связывание определяет версию вызываемого метода на этапе выполнения

Ответ: 1, 2, 5

**5.** Дан код:

package ch04.q03;

public class Quest43 {

private final void method () {} //1

}

class Quest431 extends Quest43 {

public void method () {} //2

}

Что произойдет в результате компиляции этого кода (1)?

1. ошибка компиляции в строке 1
2. ошибка компиляции в строке 2
3. компиляция без ошибок

Ответ: 3

**6.**

Дан код двух классов:

//класс 1

package ch04.q04;

public class Quest41 {}

//класс 2

package ch04.q04.\_2;

import ch04.q04.Quest41;

public class Quest43 extends Quest4l {

Public Quest43() {

super();

}}

С каким атрибутом доступа объявлен конструктор по умолчанию в базовом классе Quest41 (1)

1) public

2) private

3) protected

4) friendly

Ответ: 4

**7.**

Дан код:

package ch04.q05;

public class Quest51 {

public String toString () {

return getClass () getSimpleName ();

}

public static void main (String () args) {

Quest53 q = mew Quest53();

System. out printIn (q.toString());

}}

class QuestS2 extends Quest51 {}

class Quest53 extends Quest52 {}

Что выведется на консоль в результате компиляции и запуска программы (1)?

1) Quest52

2) Quest53

3) Quest51

4) ошибка компиляции

Ответ: 4