

Наследование

Наследование – это процесс, посредством которого один объект может наследовать основные свойства другого объекта и добавлять к ним черты, характерные только для него.

Добавление собственных полей и методов

Изменение методов класса-предка

Ссылка на объект класса может быть приведена к ссылке на объект любого из классов, находящихся выше данного в иерархии наследования

Все классы наследники библиотечного класса
Object

Синтаксис объявления наследования

Базовый класс, класс-предок, суперкласс

```
class Base {}
```

```
class Son extends Base {}
```

ЯВНЫЙ ВЫЗОВ КОНСТРУКТОРА базового класса

```
class Base {  
    private int a;  
    public Base(int a) {  
        this.a=a;  
    }  
}  
class Son extends Base {  
    public Son() {  
        super(0);  
    }  
}
```

Методы класса Object

String toString()

boolean equals(Object obj)

int hashCode()

Пример переопределения методов

```
public String toString() {  
    return "Point(" + x + ", " + y + ")";    }
```

```
public boolean equals(Object obj)  
{  
    if(!(obj instanceof Point)) return false;  
    Point c = (Point)obj;  
    return x == c.x && y == c.y;  
}
```

```
int hashCode() { return x + y; }
```

Полиморфизм

Полиморфизм — концепция в объектно-ориентированном программировании, представляющая из себя возможность добавления в классы виртуальных методов, реализуемых в классах-потомках.

Объекты subclasses можно использовать везде, где используются объекты суперкласса

Интерфейс

Полностью абстрактный класс, не включающий реализацию

Уровень абстракции, описывающий общие действия для решения некоторой задачи

Все методы интерфейса по-умолчанию являются публичными (`public`) и абстрактными (`abstract`), а поля - `public static final`

В Java класс может наследовать (реализовывать) несколько интерфейсов, но только один класс

Синтаксис

Интерфейс

```
public interface Comparable {  
    int compareTo(Object o);  
}
```

Реализация интерфейса

```
public class MyClass implements Comparable {  
    .....  
    int compareTo(Object o) {... return 1;}  
}
```