Java Fundamentals

Наследование и полиморфизм

Наследование и полиморфизм

Наследование

Парадигма ООП

Наследование —механизм объектно-ориентированного программирования (наряду с инкапсуляцией, полиморфизмом и абстракцией), позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом.

```
class A {
    public int field1;
    public void method() {
        /* … */
    }
}
```

```
class B extends A {
    public int field2;
}
```

```
static void main(String[] args) {
    B b = new B();
    b.field1 = 5;
    b.field2 = 8;
    b.method();
}
```

Вызов конструктора базового класса

Использование ключевого слово super для вызова конструктора базового класса.

```
class BaseClass {
    public BaseClass() {
        System.out.println("Base");
    }
}
```

```
class DerivedClass extends BaseClass {
    public DerivedClass() {
        super();
        System.out.println("Derived");
    }
}
```

Приведение типов

Приведение к базовому типу

Приведение к базовому типу используется для сокрытия реализации членов производного класса.

```
BaseClass instance = new DerivedClass();
```

Переменная instance типа BaseClass хранит ссылку на экземпляр класса DerivedClass.

Приведение типов

UpCast и DownCast

UpCast - приведениеэкземпляра производного класса к базовому типу.

```
BaseClass up = new DerivedClass();
```

DownCast - приведение экземпляра базового типа к производному типу.

```
DerivedClass down = (DerivedClass) up;
```

Полиморфизм

Парадигма ООП

Полиморфизм —возможность объектов с одинаковой спецификацией иметь различную реализацию.

Формы полиморфизма:

1. Ad-hocполиморфизм
2. Классический (принудительный) полиморфизм:
    - использование переопределенных членов (@Override).
    - приведение типов.

В случаеодновременного использования двух форм классического полиморфизма, первая форма нейтрализует вторую (доминирует над второй).

final

Модификатор

При применении к классу, модификатор final запрещает другим классам наследоваться от этого класса.

Модификатор final можно использовать и с методами. Это позволяет запретить переопределять методы в производных классах.

Q&A