



ОНЛАЙН-ОБРАЗОВАНИЕ

Подготовительный курс Java

Модуль 2.1

«Объектно–ориентированное программирование»

Виталий Чибриков
преподаватель



Наследование, extends

```
class TimeSupplier{
    public long getTime(){ return System.currentTimeMillis();}
}

class LocalTimeSupplier extends TimeSupplier{
    public String getLocalTime(){ return "Local time";}
}

class Main{
    public void printTime(LocalTimeSupplier timeSupplier){
        System.out.println("POSIX time: " + timeSupplier.getTime());
    }
}
```



Полиморфизм

```
class TimeSupplier{
    public long getTime(){ return System.currentTimeMillis();}
}

class POSIXTimeSupplier extends TimeSupplier{
    public long getTime(){ return System.currentTimeMillis() / 1000;}
}

class Main{
    public void printTime(TimeSupplier timeSupplier){
        System.out.println("POSIX: " + timeSupplier.getTime());
    }
}
```



Абстракция, интерфейс, implements

```
interface TimeSupplierInterface{
    long getTime();
}

class TimeSupplierImpl implements TimeSupplierInterface{
    public long getTime(){ return System.currentTimeMillis() / 1000;}
}

class Main{
    public void printTime(TimeSupplierInterface timeSupplier){
        System.out.println("Time: " + timeSupplier.getTime());
    }
}
```



Ссылки между объектами

```
class TimeHolder{  
    private TimeSupplierInterface timeSupplier;  
  
    public TimeHolder(TimeSupplierInterface supplier){  
        timeSupplier = supplier;  
    }  
  
    public void printTime(){  
        System.out.println("POSIX: " + timeSupplier.getTime());  
    }  
}
```



Инкапсуляция, модификаторы доступа

Access Levels

Modifier	Class	Package	Subclass	World
<code>public</code>	Y	Y	Y	Y
<code>protected</code>	Y	Y	Y	N
<i>no modifier</i>	Y	Y	N	N
<code>private</code>	Y	N	N	N



Ключевые слова `this` и `super`

`this` это ссылка на текущий объект, в конструкторах и методах объекта

`super` это ссылка на родительский объект



Generics

Generic programming это когда тип данных как параметр

1. Облегчает проверку типов данных на стадии компиляции
2. Позволяет писать generic алгоритмы

Не работает с примитивными типами данных



```
public class GenericExample<T> {  
    private T value;  
    public GenericExample(T value){  
        this.value = value;  
    }  
    public T getT(){  
        return value;  
    }  
    public static void main(String[] args) {  
        GenericExample<Integer> intObject = new GenericExample<>(1);  
        Integer valueInteger = intObject.getT();  
        GenericExample<String> stringObject = new GenericExample<>("word");  
        String valueString = stringObject.getT();  
    }  
}
```



```
public class GenericExample {  
  
    public static <T> T getTheFirst(List<T> list){  
        return list.get(0);  
    }  
  
    public static void main(String[] args) {  
        List<Integer> listOfInts = new ArrayList<>();  
        listOfInts.add(0);  
        Integer intValue = getTheFirst(listOfInts);  
  
        List<String> listOfStrings = new ArrayList<>();  
        listOfStrings.add("Java is the best!");  
        String stringValue = getTheFirst(listOfStrings);  
    }  
}
```

