

# Применение функционального стиля в языке Java

Бухаров Марк  
Крупича Сергей

# Содержание

- Что такое функциональное программирование
- Пример: функция `intToString`
- Пример: работа с коллекциями
- Пример: использование функции-предиката

# Что такое функциональное программирование (ФП)

Основная концепция ФП - всё есть математические функции

$$f : A \rightarrow B$$

Метод становится математической функцией, если:

1. Он ссылочно прозрачен (referentially transparent)
2. Сигнатура метода является "честной"

## Пример: честность сигнатуры метода

```
public int Divide(int x, int y)
{
    return x / y;
}
```

## Пример: честность сигнатуры метода

```
public static int Divide(int x, NonZeroInteger y)
{
    return x / y.Value;
}
```

# Функции в Java с помощью анонимных классов

```
public interface Function<F, T> {  
    T apply(F from);  
}
```

## Пример: функция `intToString`

```
Function <Integer, String> intToString =  
new Function<Integer, String>() {  
    @Override  
    public String apply(Integer from) {  
        return from.toString();  
    }  
};
```

## Пример: работа с коллекциями

Хотим объединить числа в строку с разделителем. Нефункциональное решение:

```
public String joinNumbers(Collection<Integer> numbers) {  
    StringBuilder result = new StringBuilder();  
    boolean first = true;  
    for (Integer number : numbers) {  
        if (first)  
            first = false;  
        else  
            result.append(", ");  
        result.append(number);  
    }  
    return result;  
}
```



## Пример: работа с коллекциями

```
public static <F, T> List<T>
map(Collection<F> from, Function<F, T> transformer) {
    ArrayList<T> result = new ArrayList<T>();
    for (F element : from) //к каждому элементу коллекции применяем transformer
        result.add(transformer.apply(element));
    return result;
}

public static <T> String join(Collection<T> from, String separator) {
    StringBuilder result = new StringBuilder();
    boolean first = true;
    for (T element : from) { //конкатенация в строку
        if (first)
            first = false;
        else
            result.append(separator);
        result.append(element);
    }
    return result.toString();
}
```

## Пример: работа с коллекциями

```
public String joinNumbers(Collection<Integer> numbers) {  
    return join(map(numbers, intToString), ",", "");  
}
```

## Пример: использование функции-предиката

```
public interface Predicate<T> {  
    boolean test(T t);  
}  
  
public static <T> ArrayList<T>  
filter(Collection<T> from, Predicate<T> check) {  
    ArrayList<T> result = new ArrayList<T>();  
    for (T element : from) {  
        if (check.test(element))  
            result.add(element);  
    }  
    return result;  
}
```

## Пример: использование функции-предиката

```
static Predicate<Integer> is_odd = new Predicate<Integer>() {  
    @Override  
    public boolean test(Integer t) {  
        return (t % 2 != 0);  
    }  
};  
  
public static ArrayList<Integer>  
filterOddNumbers(ArrayList<Integer> numbers) {  
    return filter(numbers, is_odd);  
}
```