

Занятие 9

Anastasiya Solodkaya, Denis Stepulenok

LevelUP

8 ноября 2016 г.

- 1 Рекурсия
- 2 Стек вызовов
- 3 Решение задач с помощью рекурсии

- 1 Рекурсия
- 2 Стек вызовов
- 3 Решение задач с помощью рекурсии

Что такое рекурсия?

Рекурсия - это когда функция вызывает сама себя. Такая функция называется рекурсивной.

Пример рекурсии

```
public static int m(int n) {  
    if(n == 0){  
        return 0  
    }  
  
    return n + m(n - 1);  
}  
  
public static void main(String[] args) {  
    int y = m(5);  
    System.out.println(y);  
}
```

Ответ

Ответ: 15

Пример рекурсии

Как вы думаете, что произойдет в этом случае?

```
public static int m(int n) {  
    return n + m(n - 1);  
}  
  
public static void main(String[] args) {  
    int y = m(5);  
    System.out.println(y);  
}
```

Ответ

Ответ: бесконечная рекурсия

Пример рекурсии

```
public static int m(int n) {  
    if(n <= 1){  
        return 1;  
    }  
    return n + m(n - 1) * m(n - 2);  
}  
  
public static void main(String[] args) {  
    int y = m(3);  
    System.out.println(y);  
}
```

Ответ

Ответ: 6

Чем отличаются примеры

```
public static int m0(int n) {  
    if(n == 0){  
        return 0;  
    }
```

```
    return n + m0(n - 1);  
}
```

```
public static int m1(int n) {  
    if(n <= 1){  
        return 1;  
    }
```

```
    return n + m1(n - 1) * m1(n - 2);  
}
```

Чем отличаются примеры

- В одном случае рекурсию легко заменить на цикл
- Это называется "хвостовой рекурсией"
- Если рекурсию легко заменить на цикл, то лучше использовать цикл.

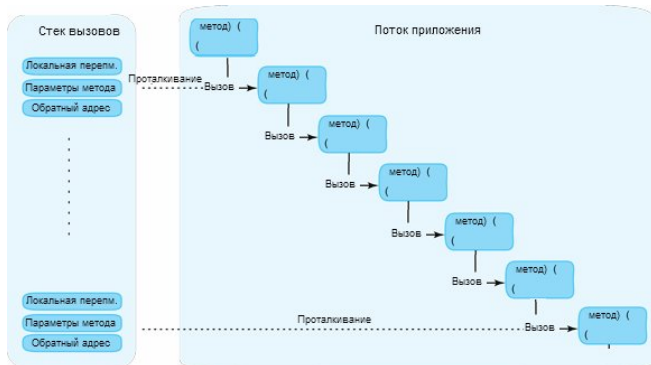
Сравните этот код

```
public static int m0(int n) {  
    if(n == 0){  
        return 0;  
    }  
  
    return n + m0(n - 1);  
}  
  
public static int m1(int n) {  
    int result = 0;  
    for(int i = 0; i <= n; i++) {  
        result += i;  
    }  
    return result  
}
```

- 1 Рекурсия
- 2 **Стек вызовов**
- 3 Решение задач с помощью рекурсии

Почему лучше использовать цикл?

Стек вызовов. Каждый вызов метода добавляет один кадр стека.



Задача поиска суммы

$n!$

$$\sum_{x=0}^n x = 1 + 2 + 3 + \dots + n$$

$$\sum_{x=0}^n x = n + \sum_{x=0}^{n-1} x$$

```
public static long summ(int n) {  
    if(n == 0) return 0;  
    return n + summ(n - 1);  
}  
  
public static void main(String[] args) {  
    System.out.println(summ(100000));  
}
```


Результат

```
Exception in thread "main" java.lang.StackOverflowError  
at com.levelp.app.mine.Test.summ(Test.java:12)  
at com.levelp.app.mine.Test.summ(Test.java:12)  
at com.levelp.app.mine.Test.summ(Test.java:12)  
...
```

- 1 Рекурсия
- 2 Стек вызовов
- 3 Решение задач с помощью рекурсии

Задача поиска факториала числа

$n!$

$$n! = 1 \times 2 \times 3 \times \dots \times n$$

$$n! = n \times (n - 1)!$$

```
public static long factorial(int n) {  
    if(n == 0){  
        return 1;  
    }  
  
    return n * factorial(n - 1);  
}  
  
public static void main(String[] args) {  
    System.out.println(factorial(10));  
}
```

Быстрая сортировка

