



Пример 4: Программа "Calculator"

- Цель примера

Продемонстрировать работу с методами с классе без состояния.

- Поэтапное создание программы

В рамках данного примера рассмотрим класс – калькулятор `Calculator`, который обладает следующими характеристиками:

- Методы:
 - Вычисление суммы двух целых чисел (операция сложения)
 - Вычисление разницы двух целых чисел (операция вычитания)

В предыдущих примерах мы рассматривали те классы, у которых есть состояние и поведение, но на практике встречаются случаи, когда состояние может отсутствовать вовсе.

i Информация: Аналогично объектно-ориентированному подходу, также существует функциональное программирование. В функциональном программировании считается обязательным проектировать системы так, чтобы состояние не влияло на результат вычислений функции (метода). Например, функция которая считает сумму двух чисел должна всегда возвращать соответствующий результат, независимо от того сколько раз и с какими параметрами ее вызывали.

– Шаг 1.

Создадим пустой класс `Calculator` :

```
public class Calculator {  
  
}
```

– Шаг 2.

Добавим метод (функцию) для расчета суммы двух чисел. Сигнатура такого метода выглядит следующим образом: `public int sum(int a, int b)` – метод возвращает целое число `int`, называется `sum` и принимает два входных параметра: `a` и `b` :

```
public class Calculator {  
  
    public int sum(int a, int b) {  
  
    }  
  
}
```

⚠ **Важно:** Любой метод, который в соответствии с сигнатурой возвращает результат, должен иметь ключевое слово `return`.

- Шаг 3.

В тело метода (функции) необходимо логику расчета суммы и возврат результата вычисления:

```
public class Calculator {  
  
    public int sum(int a, int b) {  
        int result = a + b;  
        return result;  
    }  
  
}
```

Сначала выполняется операция сложения чисел `a` и `b`, затем результат присваивается переменной `result`. Ключевое слово `return` с переменной `result` возвращает ее результат туда, откуда был произведен вызов метода.

- Шаг 4.

Продemonстрируем работу калькулятора в демонстрационном классе `CalculatorDemo`:

```
public class CalculatorDemo {  
  
    public static void main(String[] args) {  
  
        Calculator calculator = new Calculator();  
  
        int sumResult = calculator.sum(3, 8);  
        System.out.println("sumResult = " + sumResult);  
  
    }  
  
}
```

Результат работы программы:

```
■ sumResult = 11
```

- Шаг 5.

Добавим в класс `Calculator` метод для вычисления разницы двух чисел `diff` :

```
public class Calculator {  
  
    public int sum(int a, int b) {  
        int result = a + b;  
        return result;  
    }  
  
    public int diff(int x, int y) {  
        return x - y;  
    }  
  
}
```

Обратите внимание, что для простых вычислений создавать промежуточную переменную необязательно. Часто, с целью улучшения читаемости, эту переменную игнорируют и сразу возвращают результат вычислений.

- Шаг 6.

Продemonстрируем работу метода `diff` выводя результат вычислений сразу на экран, без присвоения промежуточной переменной:

```
public class CalculatorDemo {  
  
    public static void main(String[] args) {  
  
        Calculator calculator = new Calculator();  
  
        int sumResult = calculator.sum(3, 8);  
        System.out.println("sumResult = " + sumResult);  
  
        System.out.println("calculator.diff(10, 4) = " + calculator.diff(10, 4));  
  
    }  
  
}
```

Результат работы программы:

```
sumResult = 11 calculator.diff(10, 4) = 6
```

• Рекомендации:

- Запустить программу и сравнить результаты;
- Попробовать передать другие параметры и посмотреть на результат;
- Добавить методы для вычисления произведения и частного двух чисел;