



Пример 2: Программа "Math Operations"

- Цель примера

Продemonстрировать работу нескольких математических операций для примитивных числовых типов данных, а также закрепить навыки работы с выводом текста на экран. После завершения, программа должна вывести в консоль следующий текст:

```
Welcome to the 'Math Operations'!  
  
1) Addition '+':  
a = 100, b = 45  
a + b = 100 + 45 = 145  
  
2) Subtraction '-':  
x = 40, y = 18  
x - y = 40 - 18 = 22  
  
3) Division '/':  
m = 15, n = 3  
m / n = 15 / 3 = 5
```

- Поэтапное создание программы

- Шаг 1.

Необходимо создать Java класс с именем `MathOperations` и методом `main()`, который является входной точкой в программу:

```
public class MathOperations {  
  
    public static void main(String[] args) {  
  
    }  
  
}
```

- Шаг 2.

В методе `main()` добавить строку – приветствие:

```
public class MathOperations {  
  
    public static void main(String[] args) {  
        System.out.println("Welcome to the 'Math Operations'!");  
    }  
  
}
```

Обратите внимание, что текст, который выводится в консоль, обязательно нужно писать между двойными кавычками: `" "`. В кавычках можно писать не только буквы, но и любой другой доступный символ для вывода на экран.

⚠ **Важно:** Есть несколько доступных методов для вывода текста на экран, а именно: `print()`, `println()` и `printf()`. Принцип их работы похож, но все таки есть разница. Например, первый метод просто выводит текст в консоль, а второй метод дополнительно переносит каретку на новую строку. Третий метод помимо вывода позволяет задать форматирование для текста. В рамках данной задачи будут продемонстрированы только методы `print()` и `println()`.

- Шаг 3.

Необходимо создать две переменные, которые будут хранить числа для демонстрации операции сложения:

```
public class MathOperations {  
  
    public static void main(String[] args) {  
        System.out.println("Welcome to the 'Math Operations'!");  
  
        int a = 100;  
        int b = 45;  
    }  
  
}
```

Для демонстрации используется две переменные типа `int`, которые позволяют хранить целые числа в диапазоне между `-2,147,483,648` : `2,147,483,647`. Значения `100` и `45` хранятся в переменных с названием `a` и `b` соответственно.

⚠ **Важно:** Тип `int` считается самым удобным типом данных для работы с целыми числами, поскольку он является типом "по-умолчанию", согласно спецификации языка Java, а следовательно не требует дополнительных уточняющих символов. Более того, у него сравнительно большой диапазон, приемлемый для большинства повседневных задач.

– Шаг 4.

Добавить вывод названия математической операции и значения переменных `a` и `b` в консоль:

```
public class MathOperations {  
  
    public static void main(String[] args) {  
        System.out.println("Welcome to the 'Math Operations'!");  
  
        int a = 100;  
        int b = 45;  
  
        System.out.println("1) Addition '+:"); // 1  
        System.out.print("a = ");           // 2  
        System.out.print(a);                 // 3  
        System.out.print(", ");              // 4  
        System.out.print("b = ");           // 5  
        System.out.print(b);                 // 6  
    }  
}
```

▶ Результат:

```
Welcome to the 'Math Operations'!  
1) Addition '+:  
a = 100, b = 45
```

Кроме текста, в качестве параметра в методах `print()` и `println()` можно указывать имя переменной. В таком случае в консоль будет выводиться **значение** этой переменной.

Это самый простой и одновременно объемный способ вывести текст в одну строку. Каждый вызов метода `print()` (строки 2 – 6) выводит на экран отдельный кусочек текста, который в последствии выглядит как одна большая строка:

```
a = 100, b = 45
```

Обратите внимание на каждый пробел внутри кавычек, благодаря которому текст выглядит красиво отформатированным.

⚠ **Важно:** Такой подход для вывода текста на экран используется крайне редко из-за своей громоздкости.

⚠ **Важно:** Поясняющий текст (напр. `a =`) может быть произвольным и не обязан совпадать с именем переменной. В рамках данной задачи это носит иллюстрационный характер.

– Шаг 5.

Выполнить операцию сложения над переменными `a` и `b` и присвоить результат в переменную с названием `sum`, затем вывести результат в консоль:

```
public class MathOperations {  
  
    public static void main(String[] args) {  
        System.out.println("Welcome to the 'Math Operations'!");
```

```

int a = 100;
int b = 45;

System.out.println("1) Addition '+:");
System.out.print("a = ");
System.out.print(a);
System.out.print(", ");
System.out.print("b = ");
System.out.print(b);

int sum = a + b;           // 1

System.out.println();      // 2
System.out.print("a + b = "); // 3
System.out.print(a);       // 4
System.out.print(" + ");   // 5
System.out.print(b);       // 6
System.out.print(" = ");   // 7
System.out.print(sum);     // 8
}
}

```

Результат:

```

Welcome to the 'Math Operations'!
1) Addition '+:
a = 100, b = 45
a + b = 100 + 45 = 145

```

Результат сложения (строка 1) присваивается в переменную с названием `sum`. Для вывода подробной информации об операции сложения на экран используется тот же подход, что и в предыдущем шаге (строки 3 – 8). Обратите внимание, что для перевода каретки на новую строку вызывается метод `println()` без параметров (строка 2).

- Шаг 6.

!! Примечание: В дальнейшем, исходный код с предыдущими операциями будет игнорироваться, чтобы подробно рассматривать только конкретную математическую операцию. Полный исходный код программы можно посмотреть в конце документа.

Для демонстрации операции вычитания, необходимо вывести информацию об операции, создать переменные `x` и `y` типа `long` со значениями `40` и `18` соответственно, а также вывести их значения на экран:

```
public class MathOperations {

    public static void main(String[] args) {
        System.out.println("2) Subtraction '-:");

        long x = 40;
        long y = 18;

        System.out.println("x = " + x + ", y = " + y); // 1
    }

}
```

Результат:

```
2) Subtraction '-:
x = 40, y = 18
```

Альтернативный подход к выводу сложных строк в консоль является склейка нескольких строк в одну (строка 1), которая является параметром метода `println()`. Операция склейки строк называется конкатенацией и в самом простом случае выполняется с помощью оператора `+`, который представляет собой не только операцию сложения для чисел, но и оператором сложения строк.

Обратите внимание, насколько сильно уменьшился объем записи текста в консоль. Такой подход является предпочтительным на начальном этапе обучения.

- Шаг 7.

Присвоим результат вычитания в переменную `difference` и сформируем строку с выводом демонстрации вычитания:

```
public class MathOperations {

    public static void main(String[] args) {
        System.out.println("2) Subtraction '-:");

        long x = 40;
        long y = 18;

        System.out.println("x = " + x + ", y = " + y);

        long difference = x - y;

        System.out.println("x - y = " + x + " - " + y + " = " + difference);
    }

}
```

Результат:

```
2) Subtraction '-:
x = 40, y = 18
x - y = 40 - 18 = 22
```

Ничего отличного от предыдущих шагов тут нет, нужно лишь привыкнуть к такой форме записи результата.

– Шаг 8.


Для демонстрации деления используется такой же подход, как и в предыдущих шагах:

```
public class MathOperations {  
  
    public static void main(String[] args) {  
        System.out.println("3) Division '/':");  
  
        short m = 15;  
        short n = 3;  
  
        System.out.println("m = " + m + ", n = " + n);  
  
        System.out.println("m / n = " + m + " / " + n + " = " + (15 / 3)); // 1  
    }  
}
```

Результат:

```
3) Division '/':  
m = 15, n = 3  
m / n = 15 / 3 = 5
```

Стоит заметить, что выводить в консоль можно сразу результат математических операций без промежуточного объявления переменной для хранения результата (строка 1).

 **Важно:** Скобки определяют порядок выполнения математических операций в соответствии с математическими правилами, а также улучшают восприятие кода.

• Рекомендации:

- Запустить программу, заменить значения переменных и сравнить результаты работы программы
- Рассмотреть другие доступные математические операции (умножение и остаток от деления)
- Рассмотреть другие доступные примитивные типы данных (`byte` , `float` , `double`)

- Полный код программы

```
public class MathOperations {

    public static void main(String[] args) {
        System.out.println("Welcome to the 'Math Operations'!");

        int a = 100;
        int b = 45;

        System.out.println("1) Addition '+:");
        System.out.print("a = ");
        System.out.print(a);
        System.out.print(", ");
        System.out.print("b = ");
        System.out.print(b);

        int sum = a + b;

        System.out.println();
        System.out.print("a + b = ");
        System.out.print(a);
        System.out.print(" + ");
        System.out.print(b);
        System.out.print(" = ");
        System.out.print(sum);

        System.out.println();
        System.out.println();
        System.out.println("2) Subtraction '-:");

        long x = 40;
        long y = 18;

        System.out.println("x = " + x + ", y = " + y);

        long difference = x - y;

        System.out.println("x - y = " + x + " - " + y + " = " + difference);

        System.out.println();
        System.out.println("3) Division '/:");

        short m = 15;
        short n = 3;

        System.out.println("m = " + m + ", n = " + n);

        System.out.println("m / n = " + m + " / " + n + " = " + 15 / 3);
    }
}
```