

Задания из второй Главы учебника

Упражнение 1. Цель: Создание Java-программы, выполняющей некоторые арифметические операции.

Arithmetic.java

```
class Arithmetic
{
    public static void main( String[] args ) {
        int num = 100 ;
        int factor = 20 ;
        int sum = 0 ;
        sum = num + factor ; // 100 + 20
        System.out.println( "Результат сложения: " + sum ) ;
        sum = num - factor ; // 100 - 20
        System.out.println( "Результат вычитания: " + sum ) ;
        sum = num * factor ; // 100 x 20
        System.out.println( "Результат умножения: " + sum ) ;
        sum = num / factor ; // 100 ÷ 20
        System.out.println( "Результат деления: " + sum ) ;
    }
}
```

Результат

```
Результат сложения: 120
Результат вычитания: 80
Результат умножения: 2000
Результат деления: 5
```

Упражнение 2. Цель: Создание Java-программы, в которой выполняются различные операторы присваивания.

Assignment.java

```
class Assignment {
    public static void main( String[] args ) {
        String txt = "Fantastic " ;
        String lang = "Java" ;
        txt += lang ; // Присваивание с объединением строк
        System.out.println( "Складываем и присваиваем строки: " + txt ) ;
        int sum = 10 ;
        int num = 20 ;
        sum += num ; // Присваиваем результат ( 10 + 20 = 30 )
        System.out.println( "Складываем и присваиваем целые числа: " + sum ) ;
        int factor = 5 ;
        sum *= factor ; // Присваиваем результат ( 30 x 5 = 150 )
        System.out.println( "Результат умножения " + sum ) ;
        sum /= factor ; // Присваиваем результат ( 150 ÷ 5 = 30 )
        System.out.println( "Результат деления: " + sum ) ;
    }
}
```

Результат

Складываем и присваиваем строки: **Fantastic Java**

Складываем и присваиваем целые числа: **30**

Результат умножения **150**

Результат деления: **30**

Упражнение 3. Цель: Создание Java-программы, которая демонстрирует использование операторов сравнения для сравнения строк и чисел. В коде были использованы операторы сравнения для определения равенства, неравенства, больше и меньше между строками и числами..

Comparison.java

```
class Comparison
{
    public static void main( String[] args ) {
        String txt = "Fantastic " ;
        String lang = "Java" ;
        boolean state = ( txt == lang ) ; // Присваиваем результат проверки
        System.out.println( "Проверка строк на равенство: " + state ) ;
        state = ( txt != lang ) ; // Присваиваем результат
        System.out.println( "Проверка строк на неравенство: " + state ) ;
        int dozen = 12 ;
        int score = 20 ;
        state = ( dozen > score ) ; // Присваиваем результат
        System.out.println( "Проверка на больше: " + state ) ;
        state = ( dozen < score ) ; // Присваиваем результат
        System.out.println( "Проверка на меньше: " + state ) ;
    }
}
```

Результат

Проверка строк на равенство: **false**

Проверка строк на неравенство: **true**

Проверка на больше: **false**

Проверка на меньше: **true**

Упражнение 4. Цель: Создание Java-программы, использующей три логических оператора.

Logic.java

```
class Logic {
    public static void main( String[] args ) {
        boolean yes = true ;
        boolean no = false ;
        System.out.println( "Результат выражения yes И yes: " + ( yes && yes ) ) ;
        System.out.println( "Результат выражения yes И no: " + ( yes && no ) ) ;
        System.out.println( "Результат выражения yes ИЛИ yes: " + ( yes || yes ) ) ;
        System.out.println( "Результат выражения yes ИЛИ no: " + ( yes || no ) ) ;
        System.out.println( "Результат выражения no ИЛИ no: " + ( no || no ) ) ;
        System.out.println( "Первоначальное значение переменной yes: " + yes ) ;
        System.out.println( "Инвертированная переменная yes: " + !yes ) ;
    }
}
```

```
}
```

Результат

```
Результат выражения yes И yes: true
Результат выражения yes И no: false
Результат выражения yes ИЛИ yes: true
Результат выражения yes ИЛИ no: true
Результат выражения no ИЛИ no: false
Первоначальное значение переменной yes: true
Инвертированная переменная yes: false
```

Упражнение 5. Цель: Java-программа для определения четности или нечетности чисел 1357 и 2468 с использованием условных операторов. Результат выводится на консоль.

Condition.java

```
class Condition {
    public static void main( String[] args ) {
        int num1 = 1357 ;
        int num2 = 2468 ;
        String result ;
        result = ( num1 % 2 != 0 ) ? "Нечетное" : "Четное" ;
        System.out.println( num1 + " - " + result ) ;
        result = ( num2 % 2 != 0 ) ? " Нечетное " : " Четное " ;
        System.out.println( num2 + " - " + result ) ;
    }
}
```

Результат

```
1357 - Нечетное
2468 - Четное
```

Упражнение 6. Цель: Программа "Precedence" на языке Java иллюстрирует различия в порядке выполнения математических операций с использованием арифметических операторов (-, +, *). Выводит результаты для трех разных порядков действий.

Precedence.java

```
class Precedence {
    public static void main( String[] args ) {
        int sum = 32 - 8 + 16 * 2 ; // 16 x 2 = 32, + 24 = 56
        System.out.println( "Порядок действий по умолчанию: " + sum ) ;
        sum = ( 32 - 8 + 16 ) * 2 ; // 24 + 16 = 40, x 2 = 80
        System.out.println( "Указанный порядок действий: " + sum ) ;
        sum = ( 32 - (8 + 16) ) * 2 ; // 32 - 24 = 8, * 2 = 16
        System.out.println( "Специфичный порядок действий: " + sum ) ;
    }
}
```

Результат

```
Порядок действий по умолчанию: 56
Указанный порядок действий: 80
Специфичный порядок действий: 16
```

Упражнение 7. Цель: Создание Java-программы, использующей управляющие последовательности для форматирования вывода.

Escape.java

```
class Escape {
    public static void main( String[] args ) {
        String header = "\n\tНЬЮ-ЙОРК ПРОГНОЗ НА 3 ДНЯ:\n" ;
        header += "\n\tДень\t\tМакс\tМин\tОсадки\n" ;
        header += "\t---\t\t\t---\t\t\t-----\n" ;
        String forecast = "\tВоскресенье\t68F\t48F\tЯсно\n" ;
        forecast += "\tПонедельник\t69F\t57F\tЯсно\n" ;
        forecast += "\tВторник\t\t71F\t50F\tОблачность\n" ;
        System.out.print( header + forecast ) ;
    }
}
```

Результат

НЬЮ-ЙОРК ПРОГНОЗ НА 3 ДНЯ:

День	Макс	Мин	Осадки
---	----	---	-----
Воскресенье	68F	48F	Ясно
Понедельник	69F	57F	Ясно
Вторник	71F	50F	Облачность

Упражнение 8. Создание Java-программы, использующей побитовые операции для проверки состояния флагов в числе 53 (00110101). Она выводит состояние флагов и указывает, включен ли каждый из них ("ВКЛ") или выключен ("выкл").

Bitwise.java

```
class Bitwise {
    public static void main( String[] args ) {
        int fs = 53 ; // Двоичное представление 00110101
        System.out.println("Флаг 1: "+(( fs&1)>0) ? "ВКЛ" : "выкл"));
        System.out.println("Флаг 2: "+(( fs&2)>0) ? "ВКЛ" : "выкл"));
        System.out.println("Флаг 3: "+(( fs&4)>0) ? "ВКЛ" : "выкл"));
        System.out.println("Флаг 4: "+(( fs&8)>0) ? "ВКЛ" : "выкл"));
        System.out.println("Флаг 5: "+(( fs&16)>0) ? "ВКЛ" : "выкл"));
        System.out.println("Флаг 6: "+(( fs&32)>0) ? "ВКЛ" : "выкл"));
        System.out.println("Флаг 7: "+(( fs&64)>0) ? "ВКЛ" : "выкл"));
        System.out.println("Флаг 8: "+(( fs&128)>0) ? "ВКЛ" : "выкл"));
    }
}
```

Результат

Флаг 1: ВКЛ
 Флаг 2: выкл
 Флаг 3: ВКЛ
 Флаг 4: выкл
 Флаг 5: ВКЛ
 Флаг 6: ВКЛ
 Флаг 7: выкл
 Флаг 8: выкл

Заключение

- Арифметические операторы можно использовать для формирования выражений с двумя операндами для сложения (+), вычитания(-), умножения (*), деления (/) и деления по модулю (%).
- Операторы инкремента (++) и декремента (--) изменяют единственный операнд на единицу.
- Оператор присваивания (=) можно комбинировать с арифметическими операторами, чтобы выполнять арифметические вычисления, а затем присваивать результат.
- Операторы сравнения можно использовать для формирования выражений, оценивающих два операнда на равенство (==), неравенство (!=), больше (>) или меньше (<).
- Оператор присваивания может быть скомбинирован с операторами > и <, и он будет возвращать значение **true** при равенстве операндов.
- Логические операторы И (&&) и ИЛИ (||) формируют выражения, оценивающие два операнда и возвращающие логические значения **true** или **false**.
- Логический оператор ! возвращает обратную логическую величину от значения единственного операнда.
- Условный оператор ? : оценивает заданное логическое выражение и возвращает один из двух операндов в зависимости от результата.
- При оценке логических выражений на предмет истинности можно опускать знак **== true**.
- В сложных арифметических выражениях важно явно указывать порядок операторов с помощью добавления скобок.
- Во избежание синтаксических ошибок кавычки в строковых переменных нужно предварять управляющим оператором \.
- Управляющие последовательности \n (новая строка) и \t (табуляция) организуют простое форматирование вывода.
- В особых ситуациях для выполнения двоичных арифметических операций полезно использовать побитовые операторы.