

Задания из четвертой Главы учебника

**Упражнение 1.** Цель: Создание Java-программы, которая осуществляет конвертацию переменных в различные типы данных.

### Convert.java

```
class Convert{
    public static void main ( String[] args ) {
        float daysFloat = 365.25f ;
        String weeksString = "52" ;
        int daysInt = (int) daysFloat ;
        int weeksInt = Integer.parseInt( weeksString ) ;
        int week = ( daysInt / weeksInt ) ;
        System.out.println( "Дней в неделе: " + week ) ;
    }
}
```

Результат

Дней в неделе: 7

**Упражнение 2.** Цель: Создание Java-программы, для обработки строковых и целочисленных массивов.

### Array.java

```
class Array {
    public static void main ( String[] args ) {
        String[] str = { "Java ", "Хороший", " Язык" } ;
        int[] num = new int[3] ;
        num[0] = 100 ;
        num[1] = 200 ;
        str[1] = "Лучший" ;
        System.out.println( "Размер строкового массива " + str.length );
        System.out.println( "Размер целочисленного массива "+ num.length);
        System.out.println( num[0] + "," + num[1]+ ","+num[2] ) ;
        System.out.println( str[0] + str[1] + str[2] ) ;
    }
}
```

Результат

Размер строкового массива 3

Размер целочисленного массива 3

100,200,0

Java Лучший Язык

**Упражнение 3.** Цель: Создание Java-программы, использующей командную строку для передачи аргументов. В коде был использован массив аргументов args, который позволяет пользователю указывать опции при запуске программы. Программа проверяет наличие аргументов, и если они присутствуют, она анализирует первый аргумент и выводит соответствующее сообщение. Если аргументы отсутствуют, программа выдаст сообщение о том, что опция не была указана.

**Option.java**

```

class Option {
    public static void main(String[] args) {
        if (args.length > 0) {
            if (args[0].equals("-en")) {
                System.out.println("Опция для Английского языка");
            } else if (args[0].equals("-es")) {
                System.out.println("Опция для Испанского языка");
            } else {
                System.out.println("Неизвестная опция");
            }
        } else {
            System.out.println("Не указана опция");
        }
    }
}

```

Результат

```

PS C:\Users\abzal\Documents\q-java\5.1> java Option -en
Опция для Английского языка

```

```

PS C:\Users\abzal\Documents\q-java\5.1> java Option -es
Опция для Испанского языка

```

```

PS C:\Users\abzal\Documents\q-java\5.1> java Option -ez
Неизвестная опция

```

```

PS C:\Users\abzal\Documents\q-java\5.1> java Option
Не указана опция

```

**Упражнение 4.** Цель: Создание Java-программы, выполняющей математические операции над тремя аргументами, введенными через командную строку. Программа проверяет, что количество аргументов равно 3, затем выполняет операцию (сложение, вычитание, умножение или деление) в зависимости от второго аргумента, выводя результат или сообщение об ошибке.

**Args.java**

```

class Args {
    public static void main(String[] args) {
        if (args.length != 3) {
            System.out.println("Неверное число аргументов");
            return;
        }
        int num1 = Integer.parseInt(args[0]);
        int num2 = Integer.parseInt(args[2]);
        String msg = args[0] + args[1] + args[2] + "=";
        if (args[1].equals("+"))
            msg += (num1 + num2);
        else if (args[1].equals("-"))
            msg += (num1 - num2);
        else if (args[1].equals("x"))
            msg += (num1 * num2);
        else if (args[1].equals("/"))
            msg += (num1 / num2);
        else msg = "Неправильный оператор";
    }
}

```

```

        System.out.println( msg ) ;
    }
}

```

## Результат

```
PS C:\Users\abzal\Documents\q-java\5.1> java Args 16 + 4
16+4=20
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Args 16 - 4
16-4=12
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Args 16 x 4
16x4=64
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Args 16 / 4
16/4=4
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Args 16 % 4
Неправильный оператор
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Args 16 +
Неверное число аргументов
```

**Упражнение 5.** Цель: Создание Java-программы, которая принимает аргументы командной строки и выводит их значения, а также выполняет итерацию по элементам массивов с использованием различных типов циклов (for, while, do-while).

## Loops.java

```

class Loops {
    public static void main ( String[] args ) {
        if ( args.length > 0 ) { }
        for ( int i = 0 ; i < args.length ; i++ )
        {
            System.out.println( "args[" +i+ "] - | "+ args[i] ) ;
        }
        String[] htm = { "HTML5", "in", "easy", "steps" } ;
        int j = 0 ;
        while ( j < htm.length )
        {
            System.out.println( "htm[" +j+ "] - | " + htm[j] ) ;
            j++ ;
        }
        String[] xml = { "XML", "in", "easy", "steps" } ;
        int k = 0 ;
        if ( xml.length > 0 ) do
        {
            System.out.println( "\t\txml["+k+"] - | "+xml[k] ) ;
            k++ ;
        } while ( k < xml.length ) ;
    }
}

```

## Результат

```
PS C:\Users\abzal\Documents\q-java\5.1> java Loops Java in easy steps
```

```
args[0] - | Java
args[1] - | in
args[2] - | easy
args[3] - | steps
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Loops
```

```
htm[0] - | HTML5
htm[1] - | in
htm[2] - | easy
htm[3] - | steps

xml[0] - | XML
xml[1] - | in
xml[2] - | easy
xml[3] - | steps
```

**Упражнение 6.** Цель: Программа на языке Java, созданная для анализа объемов продаж в различных месяцах года. В коде используются массивы для хранения данных о продажах в киосках (kiosk) и точках продаж (outlet) за четыре квартала. Затем программа вычисляет общий объем продаж за год, а также выводит информацию о продажах в каждом месяце.

**Elements.java**

```
class Elements
{
    public static void main ( String[] args ) {
        int[] kiosk_q1 = { 42000 , 48000 , 50000 } ;
        int[] kiosk_q2 = { 52000 , 58000 , 60000 } ;
        int[] kiosk_q3 = { 46000 , 49000 , 58000 } ;
        int[] kiosk_q4 = { 50000 , 51000 , 61000 } ;
        int[] outlet_q1 = { 57000 , 63000 , 60000 } ;
        int[] outlet_q2 = { 70000 , 67000 , 73000 } ;
        int[] outlet_q3 = { 67000 , 65000 , 62000 } ;
        int[] outlet_q4 = { 72000 , 69000 , 75000 } ;
        int[] sum = new int[ 12 ] ;
        int total = 0 ;
        for ( int i = 0 ; i < kiosk_q1.length ; i++ )
        {
            sum[ i ] = kiosk_q1[i] + outlet_q1[i] ;
            sum[i+3] = kiosk_q2[i] + outlet_q2[i] ;
            sum[i+6] = kiosk_q3[i] + outlet_q3[i] ;
            sum[i+9] = kiosk_q4[i] + outlet_q4[i] ;
        }
        for ( int i = 0 ; i < sum.length ; i++ )
        {
            System.out.println( "Месяц " + ( i+1 ) + " объем продаж:\t" + sum[i] ) ;
            total += sum[i] ;
        }
        System.out.println( "ОБЩИЙ ОБЪЕМ ПРОДАЖ ЗА ГОД\t" + total ) ;
    }
}
```

## Результат

```
PS C:\Users\abzal\Documents\q-java\5.1> javac Elements.java
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Elements
```

```
Месяц 1 объем продаж: 99000
Месяц 2 объем продаж: 111000
Месяц 3 объем продаж: 110000
Месяц 4 объем продаж: 122000
Месяц 5 объем продаж: 125000
Месяц 6 объем продаж: 133000
Месяц 7 объем продаж: 113000
Месяц 8 объем продаж: 114000
Месяц 9 объем продаж: 120000
Месяц 10 объем продаж: 122000
Месяц 11 объем продаж: 120000
Месяц 12 объем продаж: 136000
ОБЩИЙ ОБЪЕМ ПРОДАЖ ЗА ГОД 1425000
```

**Упражнение 7.** Цель: Создание Java-программы, использующей двумерный массив для отображения точек 'X' и '-' в виде паттерна. Каждая строка массива представляет собой набор булевых значений, где 'true' соответствует 'X', а 'false' - '-' в выводе программы.

### Dimensions.java

```
class Dimensions {
    public static void main ( String[] args ) {
        boolean[][] points = new boolean[5][20] ;
        points[0][5] = true ;
        points[1][6] = true ;
        points[2][7] = true ;
        points[3][8] = true ;
        points[4][9] = true ;
        for ( int i = 0 ; i < points.length ; i++ )
        {
            System.out.print( "\n" ) ;
            for ( int j = 0 ; j < points[0].length ; j++ ) {
                char mark = ( points[i][j] ) ? 'X' : '-' ;
                System.out.print( mark ) ;}
        }
    }
}
```

## Результат

```
PS C:\Users\abzal\Documents\q-java\5.1> javac Dimensions.java
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Dimensions
```

```
-----X-----
-----X-----
-----X-----
-----X-----
-----X-----
```

**Упражнение 8.** Цель: Создание Java-программы для обработки исключений. Принимает целое число из командной строки и выводит его, с обработкой ошибок ввода и выводом завершения программы..

### Exceptions.java

```
class Exceptions {
    public static void main ( String[] args ) {
        try
        {
            int num = Integer.parseInt( args[0] ) ;
            System.out.println( "Вы ввели: "+ num ) ;
        }
        catch( ArrayIndexOutOfBoundsException e )
        { System.out.println( "Требуется целочисленный аргумент." ) ; }
        catch( NumberFormatException e )
        { System.out.println( "Неверный формат аргумента." ) ; }
        finally { System.out.println( "Программа завершила работу.");}
    }
}
```

Результат

```
PS C:\Users\abzal\Documents\q-java\5.1> javac Exceptions.java
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Exceptions
```

```
Требуется целочисленный аргумент.
```

```
Программа завершила работу.
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Exceptions TWENTY
```

```
Неверный формат аргумента.
```

```
Программа завершила работу.
```

```
PS C:\Users\abzal\Documents\q-java\5.1> java Exceptions 20
```

```
Вы ввели: 20
```

```
Программа завершила работу.
```

## Заключение

- Любые числовые значения можно конвертировать в другие типы числовых данных с помощью приведения типов, а также к строковому типу, используя метод `toString()`.
- Значение строкового типа может быть сконвертировано в целочисленное (`int`) с помощью метода `Integer.parseInt()`, а также в значение плавающего типа (`float`), используя метод `Float.parseFloat()`.
- Массив — это переменная, которая может содержать множественные значения, которые инициализируются списком внутри фигурных скобок.
- Пустой массив может быть создан при помощи ключевого слова `new`.
- Свойство массива `length` содержит целочисленную величину, равную числу элементов этого массива.
- К любому элементу массива можно обращаться по его индексу.
- Главный метод программы создает строковый массив с именем `args`, в котором сохраняются аргументы командной строки.
- Первый аргумент командной строки автоматически сохраняется в элементе `args[0]`.
- Множественные аргументы, передаваемые в программу из командной строки, должны быть разделены пробелом.
- Циклы являются идеальным способом для того, чтобы прочитать значение, хранящееся в элементах массива.
- Данные из нескольких массивов можно комбинировать и формировать из них новый массив объединенных данных.
- Многомерные массивы могут содержать множественные наборы значений элементов, каждый из которых имеет свою размерность.
- Блок операторов `try catch` используется для обработки исключений, возникающих на этапе исполнения.
- Класс `Exception` перехватывает все исключения, включая `NumberFormatException` и `ArrayIndexOutOfBoundsException`.
- Блок `try catch` может быть расширен оператором `finally`, содержащим код, который будет выполняться в любом случае.