Практика

Разбор практических примеров использования базовых элементов языка Java, работа с консолью

[Ввод данных из консоли](#_tyjcwt)

[Полезные примеры](#_3dy6vkm)

[Так делать нельзя](#_tv70etlyje8s)

[Домашнее задание](#_2s8eyo1)

[Дополнительные материалы](#_3rdcrjn)

Используемая литература

# Ввод данных из консоли

Для ввода данных из консоли можно воспользоваться объектом класса Scanner (вопрос, что такое классы и объекты, будет подробно рассмотрен на 5 занятии).

|  |
| --- |
| public static void main(String[] args) {  Scanner sc = new Scanner(System.in); // создание объекта класса Scanner  int a = sc.nextInt(); // чтение целого числа в переменную a  String b = sc.nextLine(); // чтение введенной строки  String c = sc.next(); // слово до следующего пробела  sc.close(); // после завершения работы со сканером его необходимо закрыть, } |

Пример программы, запрашивающей у пользователя ввод целого числа и выводящей в консоль число в 2 раза больше.

|  |
| --- |
| import java.util.Scanner; public class MainClass {  public static void main(String[] args) {  Scanner sc = new Scanner(System.*in*);  System.*out*.println("Введите число: ");  int a = sc.nextInt();  a \*= 2;  System.*out*.println("Введенное вами число, умноженное на 2, равно " + a);  sc.close();  } } |

Как же сделать ввод данных в заданных пределах?

|  |
| --- |
| import java.util.Scanner; public class MainClass {  public static Scanner *sc* = new Scanner(System.*in*);   public static void main(String[] args) {  int d = *getNumberFromScanner*("Введите число в пределах от 5 до 10", 5, 10);  System.*out*.println("d = " + d);  }   public static int getNumberFromScanner(String message, int min, int max) {  int x;  do {  System.*out*.println(message);  x = *sc*.nextInt();  } while (x < min || x > max);  return x;  } }  **Результат:**  Введите число в пределах от 5 до 10  8  d = 8 |

Метод getNumberFromScanner() будет запрашивать у пользователя целое число до тех пор, пока оно не окажется в пределах от min до max включительно. Перед каждым запросом будет выводится сообщение, которое передано в message. Повторный запрос осуществляется с помощью цикла do/while. Мы будем запрашивать у пользователя ввод числа до тех пор, пока он будет пытаться указать число меньше минимального или больше максимального.

# Полезные примеры

Напишем метод, который принимает в качестве параметра одномерный массив и печатает его в консоль. По завершению печати ставится перенос строки. При необходимости можно вместо пробела поставить любой символ-разделитель.

|  |
| --- |
| public static void print1DArray(int[] arr) {  for (int i = 0; i < arr.length; i++) {  System.*out*.print(arr[i] + " ");  }  System.*out*.println(); } |

Печать двумерного прямоугольного массива с нумерацией строк и столбцов.

|  |
| --- |
| public static void print2DArray(int[][] arr) {  for (int i = 0; i <= arr[0].length; i++) {  System.*out*.print(i + " ");  }  System.*out*.println();  for (int i = 0; i < arr.length; i++) {  System.*out*.print(i + 1 + " ");  for (int j = 0; j < arr[i].length; j++) {  System.*out*.print(arr[i][j] + " ");  }  System.*out*.println();  } } |

Первый цикл отвечает за печать шапки таблицы. После него стоит оператор System.out.println() для перевода строки. После этого открывается двойной цикл для печати самого массива, i отвечает за номер строки, j за номер столбца. Сам же цикл j отвечает за печать элементов массива. Перед печатью строки массива прописываем номер этой строки System.*out*.print(i + 1 + " ").

Посчитать сумму элементов в массиве можно с помощью следующего кода.

|  |
| --- |
| public static int arrSum(int[] arr) {  int sum = 0;  for (int i = 0; i < arr.length; i++) {  sum += arr[i];  }  return sum; } |

Для расчёта суммы вводим временную переменную sum, к ней в цикле будем прибавлять значения элементов массива. Как только пройдём по всем элементам массива, в переменной sum будет находиться сумма всех элементов. По аналогии можно решить задачу подсчета элементов массива, удовлетворяющих какому-либо условию, например, количество чисел 5 в массиве – пробегаем по всему массиву и увеличиваем счетчик, если нашли число 5.

Для формирования случайного числа нужно создать объект класса Random и вызвать у него метод nextInt(n), который возвращает случайное целое число в пределах от 0 до n – 1 включительно. В примере ниже в x могут попасть числа 0, 1, 2, 3, ..., 19.

|  |
| --- |
| public class MainClass {  public static void main(String[] args) {  Random rand = new Random();  intx = rand.nextInt(20);  } } |

Можно напечатать текст в консоль с форматированием с помощью метода System.out.printf(). Вначале вводится форматируемая строка с вставками вида %d, %f, %s, %c, на месте которых затем подставляются значения, взятые из аргументов метода.

|  |
| --- |
| public static void main(String[] args) {  System.*out*.printf("Слово: %s, Число с плавающей запятой: %f, Целое число: %d, Символ: %c", "Java", 2.5f, 20, 'e'); }  **Результат:**  Слово: Java, Число с плавающей запятой: 2,500000, Целое число: 20, Символ: e |

Сравнение строк должно осуществляться с помощью метода equals(), как показано в примере ниже. Смысл такого сравнения будет пояснен на занятиях по ООП.

|  |
| --- |
| public static void main(String[] args) {  String str1 = "A";  String str2 = "A";  String str3 = "B";  System.*out*.println(str1.equals(str2));  System.*out*.println(str1.equals(str3)); }  **Результат:**  true  false |

# Так делать нельзя

В данном разделе перечислены мелкие ошибки, встречающиеся у студентов, начинающих изучать язык Java и программирование в целом.

После закрывающейся круглой скобки в операторах **if** и **for** точку с запятой ставить нельзя:

|  |
| --- |
| public static void main(String[] args) {  int x = 10;  if (x < 20); { *// <- вот тут*  System.out.println(1);  }  for (int i = 0; i < 5; i++); { *// <- и вот тут*  System.out.println(i);  } } |

Нельзя объявлять методы внутри методов .

|  |
| --- |
| public static void main(String[] args) {  public static void method2() { *// <-* } } |

При вызове метода внутри скобок нельзя объявлять переменные.

|  |
| --- |
| public static void main(String[] args) {  *method*(int z = 5); *// <-* } public static void method(int x) {  System.*out*.println(x); } |

В приведённом ниже случае и во многих похожих случаях оператор continue не нужен, цикл и без него перейдет на следующий шаг, после того как дойдет до последней строки тела цикла.

|  |
| --- |
| public static void main(String args[]) {  for (int i = 0; i < 5; i++) {  if (i < 3) {  System.*out*.println("e");  } else continue;  } } |

Следите за скобками. Каждая открывающаяся фигурная скобка должна быть закрыта.

|  |
| --- |
| public class MainClass {  public static void main(String[] args) {  // <- тут не хватает закрытой фигурной скобки } |

В методе с возвратом не должно быть ситуаций, при которых ни один return не сработает. Если методу подать число x = 20, мы не сможем выйти из него, поэтому такой код даже не скомпилируется.

|  |
| --- |
| public static boolean wrongReturn(int x) {  if(x < 10) {  return true;  } } |

# Домашнее задание

Делать только одну задачу.

1. Написать программу, которая загадывает случайное число от 0 до 9 и пользователю дается 3 попытки угадать это число. При каждой попытке компьютер должен сообщить, больше ли указанное пользователем число, чем загаданное, или меньше. После победы или проигрыша выводится запрос – «Повторить игру еще раз? 1 – да / 0 – нет»(1 – повторить, 0 – нет).
2. \* Создать массив из слов  
   String[] words = {"apple", "orange", "lemon", "banana", "apricot", "avocado", "broccoli", "carrot", "cherry", "garlic", "grape", "melon", "leak", "kiwi", "mango", "mushroom", "nut", "olive", "pea", "peanut", "pear", "pepper", "pineapple", "pumpkin", "potato"}.  
   При запуске программы компьютер загадывает слово, запрашивает ответ у пользователя, сравнивает его с загаданным словом и сообщает, правильно ли ответил пользователь. Если слово не угадано, компьютер показывает буквы, которые стоят на своих местах.  
   apple – загаданное  
   apricot - ответ игрока  
   ap############# (15 символов, чтобы пользователь не мог узнать длину слова)  
   Для сравнения двух слов посимвольно можно пользоваться:  
   String str = "apple";  
   char a = str.charAt(0); - метод, вернет char, который стоит в слове str на первой позиции  
   Играем до тех пор, пока игрок не отгадает слово.  
   Используем только маленькие буквы.

# Дополнительные материалы

1. К. Сьерра, Б. Бейтс Изучаем Java // Пер. с англ. – М.: Эксмо, 2012. – 720 с.
2. Кей С. Хорстманн, Гари Корнелл Java. Библиотека профессионала. Том 1. Основы // Пер. с англ. – М.: Вильямс, 2014. – 864 с.

# Используемая литература

1. Брюс Эккель Философия Java // 4-е изд.: Пер. с англ. – СПб.: Питер, 2016. – 1168 с.
2. Г. Шилдт Java 8. Полное руководство // 9-е изд.: Пер. с англ. – М.: Вильямс, 2015. – 1376 с.
3. Г. Шилдт Java 8: Руководство для начинающих. // 6-е изд.: Пер. с англ. – М.: Вильямс, 2015. – 720 с.