

Лабораторная работа №2

Консольный ввод/вывод

Обобщенное понятие источника ввода относится к различным способам получения информации: к чтению дискового файла, символов с клавиатуры, либо получению данных из сети. Аналогично, под обобщенным понятием вывода также могут пониматься дисковые файлы, сетевое соединение и т.п. Эти абстракции дают удобную возможность для работы с вводом-выводом (I/O), не требуя при этом, чтобы каждая часть вашего кода понимала разницу между, скажем, клавиатурой и сетью. В Java эта абстракция называется потоком (stream) и реализована в нескольких классах пакета java.io. Ввод инкапсулирован в классе InputStream, вывод — в OutputStream. В Java есть несколько специализаций этих абстрактных классов, учитывающих различия при работе с дисковыми файлами, сетевыми соединениями и даже с буферами в памяти.

Для получения данных, введенных пользователем, а также для вывода сообщений используется и класс System. Этот класс располагается в пакете *java.lang*, который автоматически подключается в программу, поэтому нам не надо дополнительно импортировать данный пакет и класс.

Для создания потока вывода в класс System определен объект out. В этом объекте определен метод println, который позволяет вывести на консоль некоторое значение с последующим переводом консоли на следующую строку:

```
System.out.println("Hello world");
```

Форматирование вывода

Для форматированного вывода можно использовать оператор *printf()* (унаследованный от языка C):

```
int printf(char *format, <список вывода>);
```

Первый параметр - символьная строка, которая задает спецификацию формата. *Список вывода* - перечень переменных и выражений, значения которых выводятся.

Если после вывода строки не нужно переводить курсор на следующую строку, можно использовать метод

```
System.out.print
```

С помощью printf можем вывести на экран два числа следующим образом:

```
int x=5;
int y=6;
```

```
System.out.printf("x=%d; y=%d \n", x, y);
```

В данном случае символы %d – это спецификатор, который используется для вывода целочисленных значений типа int.

Кроме спецификатора %d мы можем использовать еще ряд спецификаторов для других типов данных:

- %x: для вывода шестнадцатеричных чисел
- %f: для вывода чисел с плавающей точкой
- %e: для вывода чисел в экспоненциальной форме, например, 1.3e+01
- %c: для вывода одиночного символа
- %s: для вывода строковых значений

Например:

```
String name = "Иван";
int age = 30;
float height = 1.7f;
```

```
System.out.printf("Имя: %s    Возраст: %d лет    Рост: %.2f метров \n", name, age, height);
```

При выводе чисел с плавающей точкой мы можем указать количество знаков после запятой, для этого используем спецификатор на %.2f, где .2 указывает, что после запятой будет два знака. В итоге мы получим следующий вывод:

```
Имя: Иван    Возраст: 30 лет    Рост: 1,70 метров
```

Для форматирования вывода можно использовать и класс Number Format в пакете java.text, который имеет три метода, предоставляющих стандартные средства форматирования для

- чисел;
- денежных сумм;
- процентов.

Чтобы получить средства форматирования для национального стандарта по умолчанию, используется один из трех методов.

```
NumberFormat.getNumberInstance()
NumberFormat.getCurrencyInstance()
NumberFormat.getPercentInstance()
```

Каждый из этих методов возвращает объект, имеющий тип `NumberFormat`. Этот объект можно использовать для форматирования одного или нескольких чисел. Затем, чтобы получить строку, содержащую отформатированное число, к объекту типа `NumberFormat` нужно применить метод `format`. Получив отформатированную строку, возможно, вы просто выведете ее на экран.

```
double x = 10000.0/3.0;
NumberFormat formatter = NumberFormat.getNumberInstance();
String s = formatter.format(x); // Строка "3,333.33".
System.out.println(s);
```

Если вы захотите задать минимальное и максимальное количество цифр в целой или дробной части числа, можете воспользоваться методами `setMinimumIntegerDigits`, `setMinimumFractionDigits`, `setMaximumIntegerDigits` и `setMaximumFractionDigits` из класса `NumberFormat`.

```
double x = 10000.0/3.0;
NumberFormat formatter = NumberFormat.getNumberInstance();
formatter.setMaximumFractionDigits(4);
formatter.setMinimumIntegerDigits(6);
String s = formatter.format(x); // Строка "003,333.3333".
```

Консольный ввод

Для реализации консольного ввода в классе `System` определен объект `in`. Однако использование только объекта `System.in` часто сопряжено с трудностями, поэтому, как правило, используют класс `Scanner`, который, в свою очередь использует `System.in`. Ниже показана программа, которая осуществляет ввод массива пяти целых чисел в цикле и вывод их на экран:

```
import java.util.Scanner; // класс Scanner находится в пакете java.util

public class FirstApp {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);
        int[] nums = new int[5];
        for(int i=0; i < nums.length; i++){
            nums[i]=in.nextInt();
        }

        for(int i=0; i < nums.length; i++){
            System.out.print(nums[i]);
        }
        System.out.println();
    }
}
```

Класс `Scanner` имеет еще ряд методов, которые позволяют получить введенные пользователем значения:

- `next()`: считывает введенную строку до первого пробела
- `nextLine()`: считывает всю введенную строку
- `nextInt()`: считывает введенное число `int`
- `nextDouble()`: считывает введенное число `double`
- `hasNext()`: проверяет, было ли введено слово
- `hasNextInt()`: проверяет, было ли введено число `int`
- `hasNextDouble()`: проверяет, было ли введено `double`

Кроме того, класс `Scanner` имеет еще ряд методов `nextByte/nextShort/nextFloat/nextBoolean`, которые считывают данные соответствующих типов данных.

В классе `String` существуют методы, которые можно применять к строкам (перед именем метода указан тип того значения, которое он возвращает):

1. `int length()` — возвращает длину строки (количество символов в ней);
2. `boolean isEmpty()` — проверяет, пустая ли строка;
3. `String replace(a, b)` — возвращает строку, где символ `a` (литерал или переменная типа `char`) заменён на символ `b`;
4. `String toLowerCase()` — возвращает строку, где все символы исходной строки преобразованы к строчным;
5. `String toUpperCase()` — возвращает строку, где все символы исходной строки преобразованы к прописным;
6. `boolean equals(s)` — возвращает истинну, если строка к которой применён метод, совпадает со строкой `s` указанной в аргументе метода (с помощью оператора `==` строки сравнивать нельзя, как и любые другие объекты);
7. `int indexOf(ch)` — возвращает индекс символа `ch` в строке (индекс это порядковый номер символа, но нумероваться символы начинают с нуля). Если символ совсем не будет найден, то возвратит `-1`. Если символ встречается в строке несколько раз, то возвратит индекс его первого вхождения.
8. `int lastIndexOf(ch)` — аналогичен предыдущему методу, но возвращает индекс последнего вхождения, если символ встретился в строке несколько раз.
9. `int indexOf(ch,n)` — возвращает индекс символа `ch` в строке, но начинает проверку с индекса `n` (индекс это порядковый номер символа, но нумероваться символы начинают с нуля).
10. `char charAt(n)` — возвращает код символа, находящегося в строке под индексом `n` (индекс это порядковый номер символа, но нумероваться символы начинают с нуля).

Пример программы:

```
public class Main {
    public static void main(String[] args) {
        String s1 = "firefox";
        System.out.println(s1.toUpperCase()); // виведет «FIREFOX»
        String s2 = s1.replace('o', 'a');
        System.out.println(s2); // виведет «firefax»
        System.out.println(s2.charAt(1)); // виведет «i»
        int i;
        i = s1.length();
        System.out.println(i); // виведет 7
        i = s1.indexOf('f');
        System.out.println(i); // виведет 0
        i = s1.indexOf('r');
        System.out.println(i); // виведет 2
        i = s1.lastIndexOf('f');
        System.out.println(i); // виведет 4
        i = s1.indexOf('t');
        System.out.println(i); // виведет -1
        i = s1.indexOf('r',3);
        System.out.println(i); // виведет -1
    }
}
```

Пример программы, которая выведет на экран индексы всех пробелов в строке, введенной пользователем с клавиатуры:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        for(int i=0; i < s.length(); i++) {
            if(s.charAt(i) == ' ') {
                System.out.println(i);
            }
        }
    }
}
```

Нижe показана программа, которая осуществляет ввод информации о трех школах и вывод ее на экран в виде таблицы. Данные каждой строки таблицы нужно вводить в одной строке через пробел.

```
package pack712;
import java.util.Scanner;
public class Hello712 {

    public static void main (String args [])throws Exception {

        Scanner in = new Scanner(System.in);
        String name1, name2, name3, sc1, sc2, sc3;
        int cnt1, cnt2, cnt3;
        double sq1, sq2, sq3;

        /* Введення даних*/
        System.out.printf("1. Введіть: назву, школу, кількість, площу >");

        name1 = in.next();
        sc1= in.next();
        cnt1= in.nextInt();
        sq1= in.nextDouble();

        System.out.printf("2. Введіть: назву, школу, кількість, площу >");

        name2 = in.next();
        sc2= in.next();
        cnt2 = in.nextInt();
        sq2= in.nextDouble();

        System.out.printf("3. Введіть: назву, школу, кількість, площу >");
        name3=in.next();
        sc3= in.next();
        cnt3= in.nextInt();
        sq3= in.nextDouble();

        /* Виведення таблиці */
        /* Виведення заголовків */
        System.out.printf("-----\n");
        System.out.printf("|                Школи                |\n");
        System.out.printf("|-----|\n");
        System.out.printf("|   Назва   | Школа | Кількість |   Площа   |\n");
        System.out.printf("|           |       |  учнів   |   (м2)    |\n");
        System.out.printf("|-----|-----|-----|-----|\n");

        /* виведення рядків даних */
        System.out.printf("| %8s | %5s | %9d | %-12.1f |\n", name1, sc1, cnt1, sq1);
        System.out.printf("| %8s | %5s | %9d | %-12.1f |\n", name2, sc2, cnt2, sq2);
        System.out.printf("| %8s | %5s | %9d | %-12.1f |\n", name3, sc3, cnt3, sq3);

        /* виведення приміток */
        System.out.printf("|-----|\n");
        System.out.printf("| Примітка:   Л - лінгвістичні;                |\n");
        System.out.printf("|              М - фізико-математичні;          |\n");
        System.out.printf("|-----|\n");
    }
}
```

```
    }  
  
}
```

Результат работы программы:

Школи			
Назва	Школа	Кількість	Площа (м²)
Лінгвіст	Л	220	368.8
Математ	М	50	54.7
Фіз-мат	М	10	12.2
Примітка: Л – лінгвістичні			
М – фізико-математичні			

Задание 1.

Ввести данные с консоли и вывести их на экран в виде таблицы 1.

Таблица 1

Породи собак			
Назва	Група	Мешкання	Чисельність популяції
Пудель	А	Європа	30000
Буль- тер'єр	Б	Європа	560000
Той-тер'єр	А	Європа, Америка	2500
Групи: А – неагресивні породи, Б – агресивні породи			