

1. Основные элементы языка Java

1.1. Идентификаторы

Идентификаторы используются для именования переменных, классов, методов. Идентификатор может быть любой последовательностью букв, символов подчеркивания, цифр. Первым символом должна быть буква, чтобы отличать идентификатор от чисел.

1.2. Константы

Константа – это конкретное значение, не изменяемое во время работы программы. Например, 100 – целое число; 3.14 – число с плавающей точкой; 'h' – символьная константа; "Это строка" – строка.

1.3. Комментарии

Возможны три вида комментариев.

Однострочный начинается двумя знаками // и заканчивается концом строки.

Многострочный комментарий начинается символами /* и заканчивается символами */.

Документационный начинается символами /** и заканчивается символами */. Эти комментарии используются при создании HTML-файла, который документирует программу.

1.4. Ключевые слова Java

Ключевые слова приведены в таблице 1.

Таблица 1. Ключевые слова

abstract	const	finally	int	public	this
boolean	continue	float	interface	return	throw
break	default	for	long	short	throws
byte	do	goto	native	static	transient
case	double	if	new	strictfp	try
catch	else	implement	package	super	void
char	extends	import	private	switch	volatile
class	final	instanceof	protected	synchronized	while

1.5. Библиотеки классов языка Java

Библиотеки классов позволяют разрабатывать программы самого различного назначения. Базовые возможности обеспечивает библиотека классов System, которая автоматически включается в создаваемые программы. Другие библиотечные классы надо подключать к программам инструкцией import.

2. Типы данных, переменные и массивы

2.1. Целые типы

Целые типы, их размеры и диапазон значений приведены в таблице 2.

Таблица 2. Целые типы данных

Тип	Ширина в битах	Диапазон значений
long	64	от $-2^{63} = -9223372036854775808$ до $2^{63}-1 = 9223372036854775807$
int	32	от $-2^{31} = -2147483648$ до $2^{31}-1 = 2147483647$
short	16	от $-2^{15} = -32768$ до $2^{15}-1 = 32767$

byte	8	от $-2^7 = -128$ до $2^7-1 = 127$
------	---	-----------------------------------

Ширина определяет диапазон значений соответствующего типа. Размер памяти, выделяемый под переменные, может отличаться от размера памяти, выраженной в байтах и определяется исполнительной средой Java.

Если в выражение входят переменные или константы типов `int`, `short` и `byte`, то перед проведением вычислений `short` и `byte` расширяются до `int`.

В тексте программ целые числа могут представляться в десятичной, восьмеричной и шестнадцатеричной системах счисления. Представление чисел – как в языке C++. Например, 65, 0101, 0x41 – это три представления для одного и того же числа – “шестьдесят пять”.

Восьмеричные числа имеют приставку ‘0’ – цифру ноль и могут включать только цифры от ‘0’ до ‘7’. То есть запись 09 ошибочна.

2.2. Типы с плавающей точкой

Типы чисел с плавающей точкой приведены в таблице 3.

Таблица 3. Типы чисел с плавающей точкой

Тип	Ширина в битах	Диапазон значений модуля
double	64	от $1.7 \cdot 10^{-308}$ до $1.7 \cdot 10^{+308}$
float	32	от $1.7 \cdot 10^{-38}$ до $1.7 \cdot 10^{+38}$

В программах числа с плавающей точкой могут записываться в форме с фиксированной точкой и в форме с плавающей точкой. Например, 3.14159265 – запись с фиксированной точкой; 2718281828.0e-9 – запись с плавающей точкой.

2.3. Символьный тип

Для обозначения символьного типа используется ключевое слово `char`. Его ширина составляет 16 битов, что позволяет хранить числовые значения от 0 до 65535. Этим числовым значениям поставлены в соответствие символы в соответствии со стандартом Unicode, который обеспечивает кодирование всех, используемых людьми символов.

Символы заключаются в апострофы, например, '1', '@', 'Y'.

Символы, не имеющие графического изображения, представляются эскейп-последовательностями, например, '\n' – это символ новой строки. Возможно использование числовых кодов символов в восьмеричной или шестнадцатеричной системах счисления. Для восьмеричной формы используют обратную наклонную черту (\) за которой следуют три восьмеричные цифры кода символа, например, '\101' – это латинская буква 'A'. Для шестнадцатеричного представления после обратной наклонной черты стоит символ `u` и четыре шестнадцатеричные цифры, например, '\u0041' – это латинская 'A'.

Следующая программа выводит большие и малые латинские и русские буквы и их коды.

Программа 3. Символы и их коды

```
public class Alphabet {
    public static void main(String args[])
    {
        char c;
        int code, i;
        System.out.println("LARGE LATIN LETTERS");
        i = 0;
        for(c = 'A'; c <= 'Z'; c++){
            System.out.print(c + "=" + (code = c) + " ");
            i++; // Счетчик символов
        }
    }
}
```

```

        if(i % 10 == 0)           // После вывода 10 букв,
            System.out.println(); // переход на новую строку
    }
    System.out.println('\n');
    System.out.println("small latin letters");
    i = 0;
    for(c = 'a'; c <= 'z'; c++){
        System.out.print(c + "=" + (code = c) + " ");
        i++;
        if(i % 10 == 0)
            System.out.println();
    }
    System.out.println("\n");
    System.out.println("БОЛЬШИЕ РУССКИЕ БУКВЫ");
    i = 0;
    for(c = 'А'; c <= 'Я'; c++){
        System.out.print(c + "=" + (code = c) + " ");
        i++;
        if(i % 10 == 0)
            System.out.println();
    }
    System.out.println('\n');
    System.out.println("малые русские буквы");
    i = 0;
    for(c = 'а'; c <= 'я'; c++){
        System.out.print(c + "=" + (code = c) + " ");
        i++;
        if(i % 10 == 0)
            System.out.println();
    }
    System.out.println();
}
}
}

```

Программа выводит следующее:

LARGE LATIN LETTERS

A=65 B=66 C=67 D=68 E=69 F=70 G=71 H=72 I=73 J=74
 K=75 L=76 M=77 N=78 O=79 P=80 Q=81 R=82 S=83 T=84
 U=85 V=86 W=87 X=88 Y=89 Z=90

small latin letters

a=97 b=98 c=99 d=100 e=101 f=102 g=103 h=104 i=105 j=106
 k=107 l=108 m=109 n=110 o=111 p=112 q=113 r=114 s=115 t=116
 u=117 v=118 w=119 x=120 y=121 z=122

БОЛЬШИЕ РУССКИЕ БУКВЫ

A=1040 Б=1041 В=1042 Г=1043 Д=1044 Е=1045 Ж=1046 З=1047 И=1048 Й=1049
 К=1050 Л=1051 М=1052 Н=1053 О=1054 П=1055 Р=1056 С=1057 Т=1058 У=1059
 Ф=1060 Х=1061 Ц=1062 Ч=1063 Ш=1064 Щ=1065 Ъ=1066 Ы=1067 Ь=1068 Э=1069
 Ю=1070 Я=1071

малые русские буквы

a=1072 б=1073 в=1074 г=1075 д=1076 е=1077 ж=1078 з=1079 и=1080 й=1081
 к=1082 л=1083 м=1084 н=1085 о=1086 п=1087 р=1088 с=1089 т=1090 у=1091
 ф=1092 х=1093 ц=1094 ч=1095 ш=1096 щ=1097 ъ=1098 ы=1099 ь=1100 э=1101
 ю=1102 я=1103

Обсудим программу.

Аргументом методов print() и println() является строка. При выводе числовых значений происходит их преобразование в строки. Если нужно вывести сразу несколько значений можно производить склеивание соответствующих строк оператором +. В

выражении `c + "=" + (code = c) + " "` сначала вычисляется содержимое скобок. Целой переменной `code` присваивается символьная переменная `c`. Значением `code` будет числовой код, соответствующий символу `c`. Затем число `code` преобразуется в строку, которая склеивается с остальными строками и выводится.

Из результатов видно, что латинские буквы имеют в кодировке UNICODE такие же коды, как в однобайтовой кодировке ANSI.

2.4. Логический тип `boolean`

Для логических значений используется тип `boolean`, который может иметь два значения: `true` (истина) и `false` (ложь). Вот программа, иллюстрирующая использование этого типа. Следующая программа иллюстрирует использование типа `boolean`.

Программа 4. Использование типа `boolean`

```
public class DemoBoolean {
    public static void main(String args[])
    {
        boolean b;
        b = true;
        System.out.println("b = " + b);
        b = false;
        System.out.println("b = " + b);
        System.out.println("(10 > 9) = " + (10 > 9));
        System.out.println("(10 < 9) = " + (10 < 9));
        System.out.println("(10 == 9) = " + (10 == 9));
        System.out.println("(10 != 9) = " + (10 != 9));
    }
}
```

Программа выводит:

```
b = true
b = false
(10 > 9) = true
(10 < 9) = false
(10 == 9) = false
(10 != 9) = true
```

Задачи 1 - 4. Кодировка символов

1. Напишите программу, выводющую коды арабских цифр от 1 до 9.
2. Напишите программу, выводющую код русской буквы ё и Ё.
3. Выведите коды символов между 'Z' и латинской 'a'.
4. Найти греческие буквы.