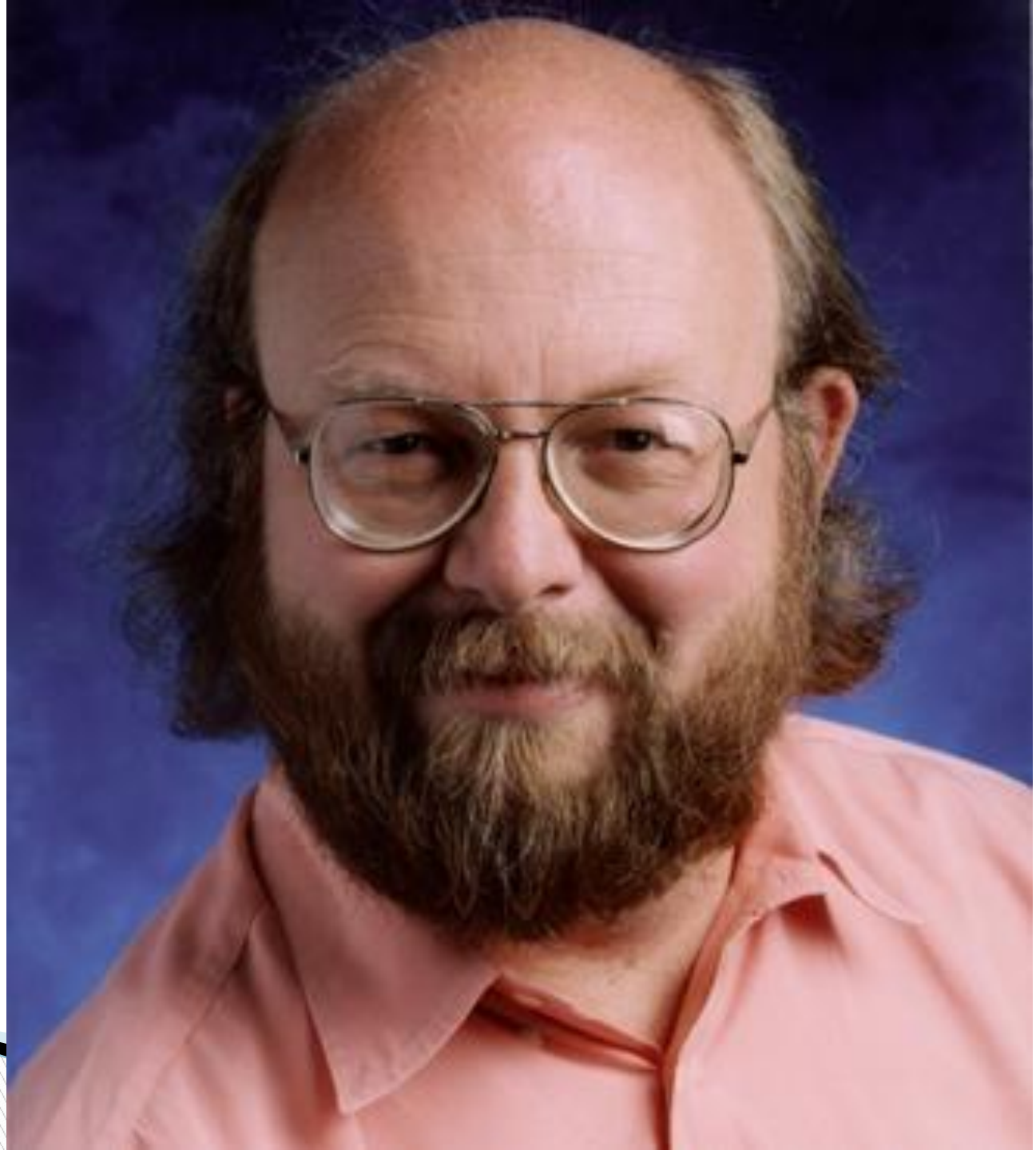
Язык Java



```
class MyProgram {
  public static void main (String args[]) {
  System.out.println ("Это Ваша первая
  программа на Java");
  }}
```

class My Program{

public static void main (String args[])

{

System.out.println ("Это Ваша первая программа на Java");

}}

```
class
   MyProgram
   {
   public
   static
   void
   main
   (
   String
   Args
   [
   ]
```

```
)
    {
    System
    .
    out
    .
    println
    (
    "Это Ваша
    первая программа на Java"
    )
    ;
    }
    }
```

```
class MyProgram { public static
  void main (String args[]) {
  System.out.println ("Эта Ваша
  первая программа на Java"); )}
```

Комментарии бывают трех типов.

• Многострочный комментарий.

*/**

*Комментарий */*

Этот комментарий использовался еще в языке С. Он может располагаться на одной или нескольких строках между группами симоволов / * и * /.

- Однострочный комментарий.

// Комментарий

Этот однострочный комментарий пришел к нам из языка C++. Комментарий может располагаться только на одной строке начиная с пары символов / / и до конца строки.

- Многострочный тип комментариев для документирования кода:

*/** код*
*Комментарий */*

Подобные комментарии располагаются между /** и */ на одной или нескольких строках. С помощью специальной утилиты javadoc на их основе собираются специальные файлы, которые в результате оказываются представлены в виде HTML-страницы.

```
class MyProgram
   {
   /** Выводим на экран строку "Это Ваша первая
   программа
   на языке Java". */
   public static void main (String args[])
   {
   System.out.println ("Эта Ваша первая программа на
   Java");
   }
   } /*
   Конец программы */
```

Работа с переменными

Создадим программу, которая демонстрирует действия с переменными (листинг 1.5).

Листинг 1.5. Использование переменных в программе

```
class Squarel {
public static void main (String args[])
{
int a, b = 4, s;
a = 5;
s = a * b ;
System.out.println ("Площадь прямоугольника со
сторонами 5 см
и 4 см равна " + s + " квадратных сантиметров.");
}
}
```

Из всего текста рассмотрим лишь следующий участок кода.

```
int a, b = 4, s;
  a = 5;
  s = a * b ;
  System.out.println ("Площадь прямоугольника
  со сторонами 5 см
  и 4 см равна " + s + " квадратных
  сантиметров. ");
```

В первой строке создаются три переменные: a, b и s целого типа данных (на это указывает ключевое слово int — указатель целого типа данных), причем переменная b сразу получает значение 4 с помощью оператора (в данном случае этот термин употребляется в значении математических, логических и других знаков) равной".

Слева от этого оператора располагается переменная, которой мы присваиваем значение, а в правой части напишем выражение, вычисляющее значение. Далее присваиваем переменной a значение 5. В следующей строке присваиваем теперь уже переменной s значение произведения значений переменных a и Ь (умножаются эти числа с помощью оператора умножения — "*").

И затем на экран выводится строка Площадь прямоугольника со сторонами 5 см и 4 см равна, далее значение переменной s, а затем строка квадратных сантиметров.

Идентификаторы

Имена переменных, классов, объектов, интерфейсов, методов называются идентификаторами. Названия идентификаторов выбираются по следующим правилам:

- они должны начинаться с буквы или символа подчеркивания;
- они могут содержать латинские буквы, символы подчеркивания или цифры без пробелов;
- названия идентификаторов не должны совпадать с ключевыми словами;

Ключевые слова, которые не должны присутствовать в названиях идентификаторов

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

Приведем десять примеров допустимых имен переменных (листинг 1.6).

Листинг 1.6. Примеры использования допустимых имен переменных

```
_book
new_look
go_home
Russial
pi
_Table_
My_l_l
Myl_Home
Close_Your_Book
Spartak_Moscow
```

А вот десять примеров недопустимых имен (листинг 1.7).

Листинг 1.7. Примеры использования недопустимых имен переменных

```
do
Go Home
lRussia
Banan;Banan
n
"my"
1show_2
TV* TV
abstract
lpower
yes or no
```

Вопросы для самопроверки:

1. Что такое язык Java?
2. Что такое ООП?
3. Что такое процедурное программирование?
4. Где и когда разрабатывали язык Java?
5. Какое отношение имеет дуб к программированию?
6. Что такое IDE?
7. Что такое класс?
8. Что такое объект класса?
9. Что такое переменная?
10. Что такое метод?
11. Что такое поля класса?
12. Что такое комментарии и какими они бывают?
13. Что такое утилита javadoc?
14. Что такое ключевое слово?
15. Что такое возвращение значения?
16. Что такое идентификатор?
17. Что такое полиморфизм?
18. Что такое наследование?
19. Что такое тип данных int?

Контрольные упражнения

1. Напишите программу, которая выводила бы приветствие.
2. Напишите программу, которая бы вычисляла площадь квадрата со стороной 5 сантиметров.

Данные в Java

Переменные и константы

Начнем знакомство с данными в Java с переменных.

Переменная— именованная ячейка памяти для хранения данных определенного типа.

Приведем пример использования переменных с целым типом данных (листинг 2.1).

Листинг 2.1.

```
Пример использования переменных с целым типом данных
class Example
{
public static void main (String args[])
{
int x; // Объявляем переменную x.
x = 10; // Инициализируем ее (присваиваем значение)
int y = 6; // Объявляем и инициализируем переменную y
int z;
z = x * y; // Записываем в переменную z результат умножения x на y
int b;
b = x * x; // Записываем в переменную b результат умножения x на x
System.out.println ("Площадь прямоугольника со сторонами " + x + " см и
" + y + " см равна " + z + " см") ;
System.out.println ("Площадь квадрата со стороной " + x + " см равна " +
b + " см");
}
}
```

Таким образом, мы сначала объявляем переменную x, затем инициализируем ее (т.е. присваиваем значение — в данном случае 10). Переменную y одновременно и объявляем, и инициализируем. Затем объявляем переменную z и в нее записываем результат умножения переменных x и y, т.е. 60.

Далее объявляем переменную b и записываем в нее квадрат числа — значение переменной x (т.е. квадрат числа 10 — 100).

Рассмотрим, что такое константы.

Константа — это именованная ячейка памяти, способная хранить данные, которые потом изменяться не будут.
Константа— это фактически переменная, объявленная с ключевым словом j (модификатором) final (оно как раз и говорит о том, что значение переменной изменяться не будет). Также под понятие константы можно подвести понятие литерала

Приведем пример использования переменных с модификатором final, т.е. именованных констант (листинг 2.2).

Листинг 2.2. Пример использования именованных констант

```
class Length
{
public static void main (String args [ ])
{
final double Pi = 3.1415926536;
// Именованная константа с типом данных для чисел с плавающей точкой.
double lengthl = Pi * 5;
double length2 = Pi * 10;
System.out.println ("Длина окружности с диаметром 5 см разна " + lengthl + ", а с диаметром 10 см - " + length2);
}
}
```

Если попытаться изменить значение константы, то это приведет к ошибке.

Литералы

Литералы — это константы, которые записаны по правилам языка Java. Рассмотрим сначала целочисленные литералы. Приведем примеры целочисленных литералов.

111 — десятичное (десятеричное) число
056 — число 46 в восьмеричной форме
0 x AB — 171 в шестнадцатеричной форме

В языке Java возможны три системы счисления: десятичная (десятеричная), восьмеричная и шестнадцатеричная. Числа в десятичной форме — это числа с основанием 10, числа в восьмеричной форме — это числа с основанием 8, числа в шестнадцатеричной форме — числа с основанием 16. Записи чисел в восьмеричной форме начинаются с 0, далее каждая цифра числа должна быть от 0 до 7. Преобразуются они в десятичные числа так: рассмотрим, например, число 056 в восьмеричной форме. В десятичной форме это число записывается так.

$$5x8 + 6 = 46$$

Запись числа в шестнадцатеричной форме начинается с символов 0x, а последующие цифры должны быть от 0 до 15. Поскольку, например, число — это 2 цифры, а нужно записать его одной цифрой, то числа от 10 до 15 записываются буквами от A до f. Рассмотрим правила перехода шестнадцатеричных чисел в десятичные. Возьмем, например, число 0 x AB. В десятичной форме это число будет записываться так.

10 x 16 + 11 = 171.

Записи числа с типом данных long ("длинные целые числа" — об этом далее) обычно имеют на конце символ 1 (чтобы отличить их от целочисленных констант типа int — простых чисел).

Рассмотрим теперь литералы для чисел с плавающей точкой.

18.01
31.4e-1
0.314e1

Эти литералы, как и целочисленные, могут иметь знаки "+" и (т.е. быть положительными или отрицательными), иметь в записи точку, которая разделяет целую и дробную часть, а также букву e и следом за ней — степень, в которую необходимо возвести число (если степень положительная, знак "+" можно не указывать).

Рассмотрим теперь, как выглядят символьные литералы.

'a'
'b'
'c'

Символьные литералы должны заключаться в одинарные кавычки. Среди символьных литералов есть так называемые escape-последовательности, которые позволяют произвести какую-либо операцию, например, перевести курсор на новую строку или вывести обратную косую черту.
Они представляют собой набор последовательностей вида *uaaaaa*, где вместо a могут быть какие-либо символы или числа. Однако есть специальные символы, которые соответствуют escape-последовательностям.

Таблица 2.1. Специальные символы, соответствующие escape-последовательностям

Специальные символы	Действие
\a	Предупреждение (звонок)
\b	Возврат курсора на шаг
\f	Перевод страницы
\n	Следующая строка (перевод на новую строку)
\r	Возврат каретки
\t	Табуляция
\\	Отображение обратной косой черты
\’	Отображение одинарной кавычки
\”	Отображение двойной кавычки
\aaa	Символ восьмеричного значения (не более 377 — т.е. 255 в десятичной системе)

Рассмотрим теперь строковые литералы.
Они принадлежат объектам типа String и располагаются между двумя кавычками, например:
"Строка"
"Старая строка \n Новая строка"
"До табуляций \t После табуляции"

К *булевым литералам* относятся такие значения, как true (истина) и false (ложь). Они служат для представления логического (или булева) типа данных — boolean.
Наконец последний тип литералов — *ссылочный литерал* — null. Его можно использовать для присвоения значений объекту, т.е. сделать так, чтобы объект не был инициализирован.