

Машинная математика. Переменные и типы данных.

№ урока: 2 Курс: JAVA Starter

Средства обучения: Компьютер с установленной IntelliJ IDEA

Обзор, цель и назначение урока

Рассмотрение двоичной и шестнадцатеричной систем счисления.

Рассмотрение понятия переменной и типов данных.

Изучив материал данного занятия, учащийся сможет:

- Понимать двоичную и шестнадцатеричную системы счисления.
- Понимать когда и какие типы использовать при создании переменной.
- Понимать отличие между вещественными и десятичными типами.

Содержание урока

1. Обзор устройства оперативной памяти ОЗУ (RAM). Адресация памяти.
2. Обзор позиционных, непозиционных, смешанных и алфавитных систем счисления.
3. Детальное рассмотрение двоичной и шестнадцатеричной систем счисления.
4. Понятие знака для двоичного числа. Двоичное дополнение.
5. Представление данных в ОЗУ с использованием двоичной и шестнадцатеричной систем счисления.
6. Представление бита, тетрады, байта, машинного слова, двойного машинного слова, учетверенного машинного слова, параграфа памяти.
7. Килобайт, мегабайт, гигабайт, терабайт.
8. Понятие вещественного числа. Точность числа.
9. Роль модуля FPU в обработке чисел с плавающей точкой.
10. Рассмотрение примера: создание переменной в JAVA
11. Правила именования локальных переменных в JAVA (Code convention).
12. Рассмотрение примера: простые типы и их псевдонимы.
13. Рассмотрение примера: целые типы, вещественные типы, десятичный тип, логический тип, символьный и строковый типы. Суффиксы. Диапазоны допустимых значений чисел со знаком и без знака.
14. Рассмотрение примера: Значения по умолчанию для локальных переменных.

Резюме

- **ОЗУ (Оперативное запоминающее устройство)** – RAM(RandomAccessMemory, память с произвольным доступом) – энергозависимая память в которой хранятся данные и команды необходимые процессору для выполнения им операций.
- **CPU (Central processing unit – ЦПУ, центральное обрабатывающее устройство)** – исполнитель машинных инструкций (кода программ).
- **ALU (Arithmetic and logic unit – АЛУ, Арифметико-логическое устройство)** – блок процессора, который служит для выполнения арифметических и логических преобразований над данными.
- **Система счисления** – символический метод записи чисел.
- **Позиционная система счисления (позиционная нумерация)** – система счисления, в которой значение каждого числового знака (цифры) в записи числа зависит от его позиции (разряда).
- **Двоичная система счисления** — это позиционная система счисления с основанием 2. В этой системе счисления числа записываются с помощью двух символов (0 и 1).
- **Шестнадцатеричная система счисления** — позиционная система счисления по целочисленному основанию 16. Обычно в качестве шестнадцатеричных цифр используются десятичные цифры от 0 до 9 и латинские буквы от A до F, то есть (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).

- **Дополнительный код (two's complement, или twos-complement)** — способ представления знаковых (положительных и отрицательных) целых чисел.
- **Бит (Bit - binary digit)** – единица измерения информации - один двоичный разряд в двоичной системе счисления. Впервые слово bit, было использовано Клодом Шенноном для логарифмической единицы информации в 1948 г. В вычислительной технике, слово «бит» часто применяется в значении «двоичный разряд».
- **Тетрада** (от греч. tetrás, родительный падеж tetrádos — четвёрка), совокупность 4 бит.
- **Байт (byte)** — единица хранения и обработки цифровой информации. В настольных вычислительных системах байт считается равным восьми битам, в этом случае он может принимать одно из $256 (2^8)$ различных значений.
- **Разрядность процессора**- способность одновременно обрабатывать какое-то количество бит. Часто, разрядностью компьютера называют разрядность его машинного слова.
- **Машинное слово** — машинно-зависимая и платформенно зависимая величина, измеряемая в битах или байтах (тритах или трайтах – машина Сетунь-70), равная разрядности регистров процессора и/или разрядности шины данных.
- Для 32-битных процессоров x86: исторически машинным словом считается 16 бит, реально — 32 бита. Это правило распространяется на двойные слова (32 бита – 64 бита), учетверенные слова (64 бита – 128 бит) и параграф (128 бит – 256 бит).
- Килобайт (KB): $1 \text{ KB} = 1024 \text{ B} = 2^{10} \text{ B}$, где B - байт
- Мегабайт (MB): $1 \text{ MB} = 1024 \text{ KB} = 1024^2 \text{ B} = 2^{20} \text{ B} = 1048576 \text{ B}$
- Гигабайт (GB): $1 \text{ GB} = 1024 \text{ MB} = 1024^3 \text{ B} = 2^{30} \text{ B} = 1\,073\,741\,824 \text{ B}$
- Терабайт (TB): $1 \text{ TB} = 1024 \text{ GB} = 1024^4 \text{ B} = 2^{40} \text{ B} = 1\,099\,511\,627\,776 \text{ B}$
- **Вещественное число или действительное число** (от лат. realis — действительный) - в информатике - тип данных, содержащий числа, записанные с десятичной точкой и/или с десятичным порядком.
- **Плавающая запятая** — форма представления действительных чисел, в которой число хранится в форме мантиссы и показателя степени. При этом число с плавающей запятой имеет фиксированную относительную точность и изменяющуюся абсолютную. Наиболее часто используемое представление утверждено в стандарте IEEE 754. Реализация математических операций с числами с плавающей запятой в вычислительных системах может быть, как аппаратная, так и программная.
- **Число одинарной точности (англ. Single precision)** — компьютерный формат представления чисел, занимающий в памяти одну ячейку (машинное слово; в случае 32-битного компьютера — 32 бита или 4 байта). Как правило, обозначает формат числа с плавающей точкой стандарта IEEE 754. Числа одинарной точности с плавающей точкой обеспечивают относительную точность 7-8 десятичных цифр и масштабы в диапазоне от 10^{38} до примерно 10^{-38} . Числа одинарной/двойной/расширенной точности (32, 64 и 80 бит) поддерживаются на аппаратном уровне сопроцессором (FPU).
- **Число двойной точности (англ. Double precision)** — компьютерный формат представления чисел, занимающий в памяти две последовательных ячейки (компьютерных слова; в случае 32-битного компьютера — 64 бита или 8 байт). Как правило, обозначает формат числа с плавающей запятой стандарта IEEE 754. Числа двойной точности с плавающей точкой обеспечивают относительную точность около 16 десятичных цифр и масштабы в диапазоне от 10^{308} до примерно 10^{-308} . В компьютерах, которые имеют 64-разрядные с плавающей точкой арифметические единицы, большинство численных вычислений осуществляется в двойной точности с плавающей точкой, поскольку использование чисел одинарной точности обеспечивает почти такую же производительность.
- **Математический сопроцессор** — сопроцессор для расширения командного множества центрального процессора и обеспечивающий его функциональностью модуля операций с плавающей запятой, для процессоров, не имеющих интегрированного модуля.
- **Модуль операций с плавающей запятой** (или с плавающей точкой; англ. floating point unit (FPU)) — часть процессора для выполнения математических операций над вещественными числами.
- **Переменная (Variable)** – это область памяти, которая хранит в себе некоторое значение, которое можно изменить.

- **Инициализация переменной** – это первое присвоение ей значения. Все последующие присвоения новых значений этой переменной, не считаются инициализацией.
- **Правило: При создании переменной обязательно указать ее тип, а при дальнейшем ее использовании, тип указывать не нужно.**
- В языке JAVA нет базовых типов - все типы реализуются как классы библиотеки NET Framework, но в нем есть набор встроенных типов, которые рассматриваются как **псевдонимы (алиасы–aliases, короткие и упрощенные формы записи)** типов из пространства имен System.
- **Псевдоним** - альтернативный тип, который можно использовать вместо системных типов. Например: `boolean` - это псевдоним класса `Boolean`, а тип `int` - псевдоним класса `Integer`.
- **JAVA является строго типизированным языком.** Каждая переменная должна иметь четко определенный тип.
- При создании переменной, используйте название-псевдоним, когда это возможно, а не полное имя класса.
- **Идентификатор** - последовательность символов, которые используются для именования членов, таких как переменные, методы, параметры, а также множество других программных конструкций, которые будут рассмотрены позже. Иными словами: **идентификатор** переменной – **это имя** этой переменной.
- Спецификация языка Java рекомендует придерживаться определенных правил (casing conventions) при создании идентификаторов (выбора имен для ваших переменных, методов и т.д.).
- К таким правилам можно отнести:
Camel casing – каждое слово, исключая первое, в идентификаторе начинается с большой буквы;
Uppercase – идентификатор состоит из букв написанных в верхнем регистре (все буквы большие)
- Технически, имена переменных могут начинаться со знака «`_`» - нижнее подчеркивание и с любого алфавитного символа. (Имена не могут начинаться с цифр и других символов.)
- Для именования локальных переменных в JAVA, рекомендуется использовать **соглашение Camel Casing**. Чтобы выделить слова в идентификаторе, первые буквы каждого слова (кроме первого) сделайте заглавными. Например: `myAge`, `myName`.
- Язык JAVA чувствительный к регистру (case sensitivity) Например: `MyName` и `myName` – это разные имена.
- Не используйте символы подчеркивания, дефисы и любые другие неалфавитно-цифровые символы для разделения слов в идентификаторе.
- Не используйте венгерскую нотацию. Суть **венгерской нотации** сводится к тому, что имена идентификаторов предваряются заранее оговорёнными префиксами, состоящими из одного или нескольких символов. Например: `string sClientName`; `int iSize`;
- Имена переменных должны быть понятны и передавать смысл каждого элемента.
- В редких случаях, если у идентификатора нет точного семантического значения, используйте общие названия. Например: `value`, `item`.
- Символы Юникода — это 16-разрядные символы, которые используются для представления большинства известных письменных языков мира.

Закрепление материала

- Что такое переменная?
- Где и для чего используются переменные?
- Назовите основные типы данных.
- Какие типы данных подходят для хранения значений чисел с плавающей запятой?
- В каком формате должны задаваться значения для строковых переменных?

Дополнительное задание

Задание

Используя IntelliJ IDEA, создайте проект. Создайте необходимое количество переменных типа `char`, каждой переменной присвойте значение одного символа в формате UNICODE. Выведите в консоль фразу «Здравствуй, ВАШЕ_ИМЯ!».

Самостоятельная деятельность учащегося

Задание 1

Выучите основные акронимы и понятия, рассмотренные на уроке.

Выучите все типы данных, рассмотренные на уроке и диапазоны значений типов: `byte`, `short`, `int`, `long`, `boolean`, `double`, `float`, `char`.

Запомните значения по умолчанию для всех простых типов данных.

Задание 2

Используя IntelliJ IDEA, создайте проект.

Дано значение числа π , которое равно 3,141592653 и значение числа Эйлера e , которое равно 2,7182818284590452. Создайте две переменные, присвойте им значения числа π и числа e и выведите их на экран без потери точности.

Задание 3

Используя IntelliJ IDEA, создайте проект.

Создайте три строковые переменные и присвойте им значения:

```
"\nмоя строка 1"
```

```
"\tmоя строка 2"
```

```
"\амоя строка 3"
```

Выведите значение каждой переменной на экран. Какие отличия вы увидели. Сделайте выводы.

Рекомендуемые ресурсы

Переменные в Java

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html>

Базовые типы (Primitive Data Types)

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>