

Data Types

Intro

Problem

Как с помощью языка программирования хранить информацию в оперативной памяти?

Solution

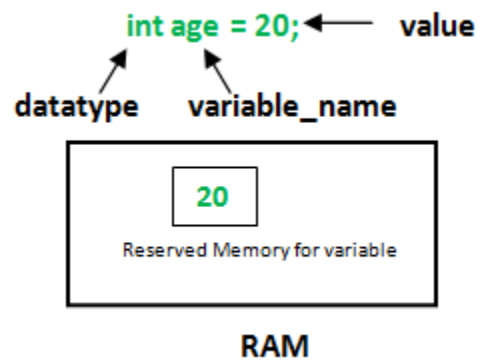
- Data types
- Variables

Variables

Variables

- **Переменная (variable)** - это именованная **ячейка памяти**
- Содержимое **переменной** может изменяться в процессе выполнения программы.
- При объявлении переменной:
 - сначала указывается **data type** (**тип данных**) переменной
 - затем **identifier** (**идентификатор**) задаваемой переменной
 - например: **int age**.

Variables



Variables

Identifier

Memory

myNumber



Address	Value
0012CCGWH80	23

Data Types

Типы данных

Типы данных в языке Java делятся на:

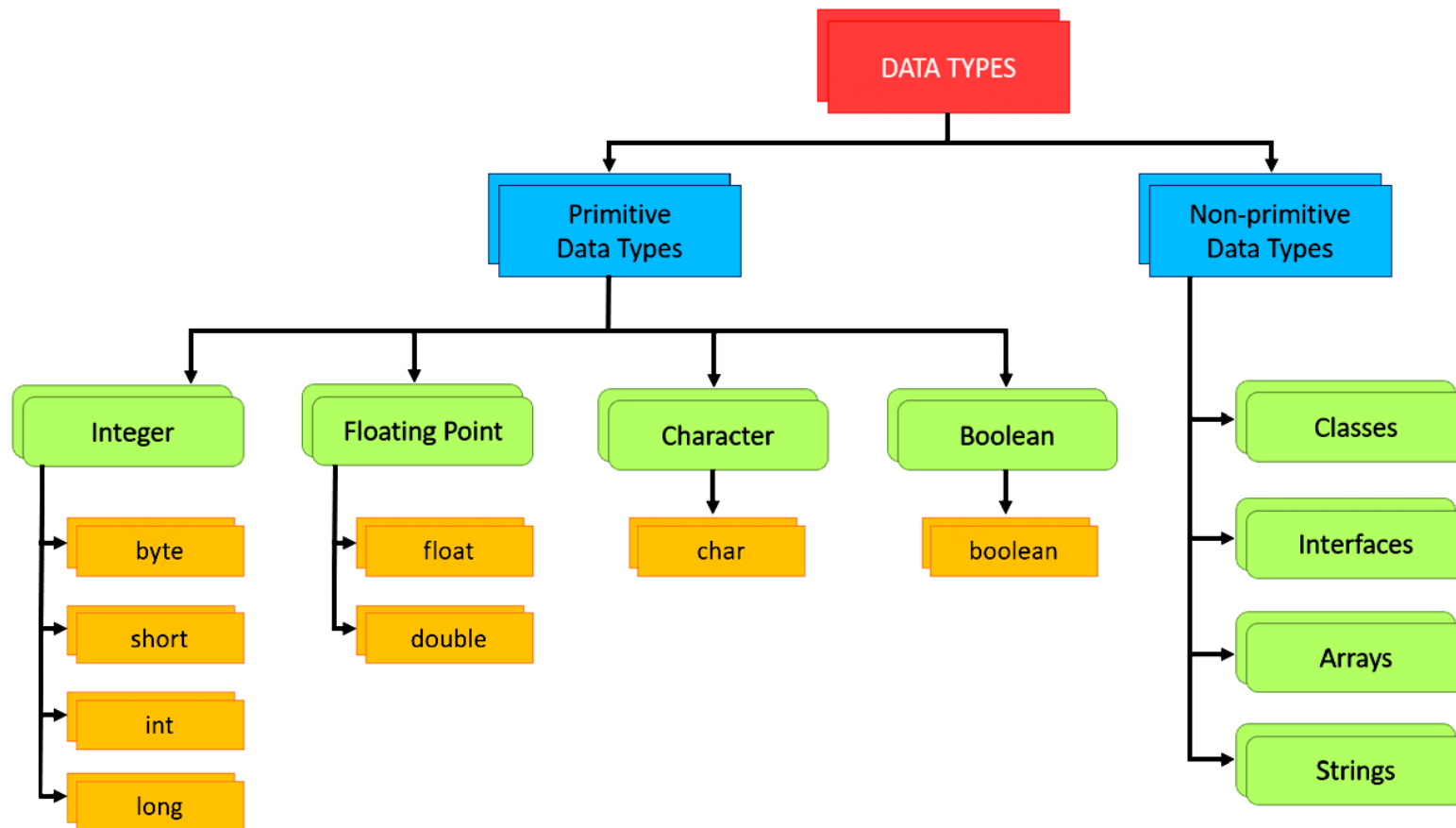
- **primitive** (примитивные)
- **reference/non-primitive** (ссылочные)

Типы данных

Существует несколько predefined примитивных типов:

- **boolean** - логический
- **byte** - целочисленный
- **char** - символьный
- **short** - целочисленный
- **int** - целочисленный
- **long** - целочисленный
- **float** - числовой, с плавающей точкой
- **double** - числовой, с плавающей точкой

Типы данных



Типы данных в языке Java

Все остальные типы - **ссылочные**. Если быть точным, то любой **Object**

Примитивные типы данных

Тип	Размер (байт)	По умолчанию	Мин	Макс
boolean	1	false	false	true
char	2	\u0000	\u0000	\uffff
byte	1	0	-128	127
short	2	0	-32 768	32 767
int	4	0	-2 147 483 648	2 147 483 647
long	8	0	-9 223 372 036 854 775 808	9 223 372 036 854 775 807
float	4	0.0F	1.4E-45	1.4E+38
double	8	0.0d	4.9E-324	1.8E+308

Default value

- Это значение, которое помещается в память
- НО без инициализации переменной будет **compile error**

Why char uses 2 byte in java and what is \u0000 ?

- It is because java uses **Unicode** system not **ASCII** code system.
- The **\u0000** is the lowest range of **Unicode** system.

Definition and Initialization

Definition and Initialization

```
int x; // объявление переменной  
x = 10; // присвоения значения  
System.out.println(x); // 10
```

Definition and Initialization

```
int y = 10; // объявление и инициализация переменной  
System.out.println(y); // 10
```

Definition and Initialization

```
int z;  
System.out.println(z);
```

Compile error: **java: variable x might not have been initialized**

Definition and Initialization

```
int x, y;  
x = 10;  
y = 25;  
System.out.println(x); // 10  
System.out.println(y); // 25
```

Bad

Definition and Initialization

```
int a = 8, b = 15;  
System.out.println(a); // 8  
System.out.println(b); // 15
```

Bad

Definition and Initialization

```
int i = 1;  
System.out.println(i); // 1  
int j = 11;  
System.out.println(j); // 11
```

Well

Literals

Literals

- **Литералы** - это явно заданные значения в коде программы.
- Фактически, *константы определенного типа*, которые *находятся в коде в момент запуска*.

Literals

```
class Test {  
    public static void main(String[] args) {  
        System.out.println("Hello world!");  
    }  
}
```

Типы литералов

- Числовые:
 - Целочисленные;
 - С плавающей точкой;
- Строковые;
- Символьные;
- Логические.

Literal type **long**

```
long a = 1_234_567_890_1; // error: integer number too large  
long b = 1_234_567_890_1L; // Все в порядке  
long c = 1_234_567_890_1l; // Not recommend. Use `L`
```

Literal type **float**

```
float d = 2.718; // error: incompatible types: possible lossy conversion from double to float
float e = 2.718F; // Все в порядке
float f = 2.718f; // Все в порядке
```

Literal type **double**

```
double a = 2.718281828459045; // Тип double в классическом виде  
double d = 4.05E-13; // Тип double в научном виде
```

Literal type **char**

```
char i = 'A'; // A (latin)  
char j = '\u0041'; // A (latin)  
char k = 65; // A (latin)
```

Другие системы счисления

- **Decimal**
- **Binary**
- **Octal**
- **Hexadecimal**

Другие системы счисления

- **Decimal** declaration and possible chars are [0-9]
- **Binary** representation starts with 0B or 0b and possible chars are [0-1]
- **Octal** declaration starts with 0 and possible chars are [0-7]
- **Hexadecimal** declaration starts with 0X or 0x and possible chars are [0-9A-Fa-f]

Другие системы счисления

```
int decimalNumber = 42;  
int binaryNumber = 0b101010;  
int octalNumber = 052;  
int hexadecimalNumber = 0x2A;
```

Поразрядное разделение

```
int x = 123456789;  
int y = 123_456_789;  
int z = 123__456_____789;  
System.out.println(x); // 123456789  
System.out.println(y); // 123456789  
System.out.println(z); // 123456789
```

Ключевое слово `var`

Ключевое слово **var** (@since 10)

```
var x = 10;  
System.out.println(x); // 10
```

Ключевое слово **var** (@since 10)

```
var y; // error: cannot infer type for local variable y  
y = 10;
```

Constants

Constants

- Кроме переменных, в Java для хранения данных можно использовать **константы**.
- Константы позволяют задать такие переменные, которые не должны больше изменяться.
- В отличие от переменных константам можно присвоить значение **ТОЛЬКО** один раз.
- Как правило, константы имеют имена в **ВЕРХНЕМ_РЕГИСТРЕ**.
- Объявляется как и переменные, но только с служебным словом **final**

Example

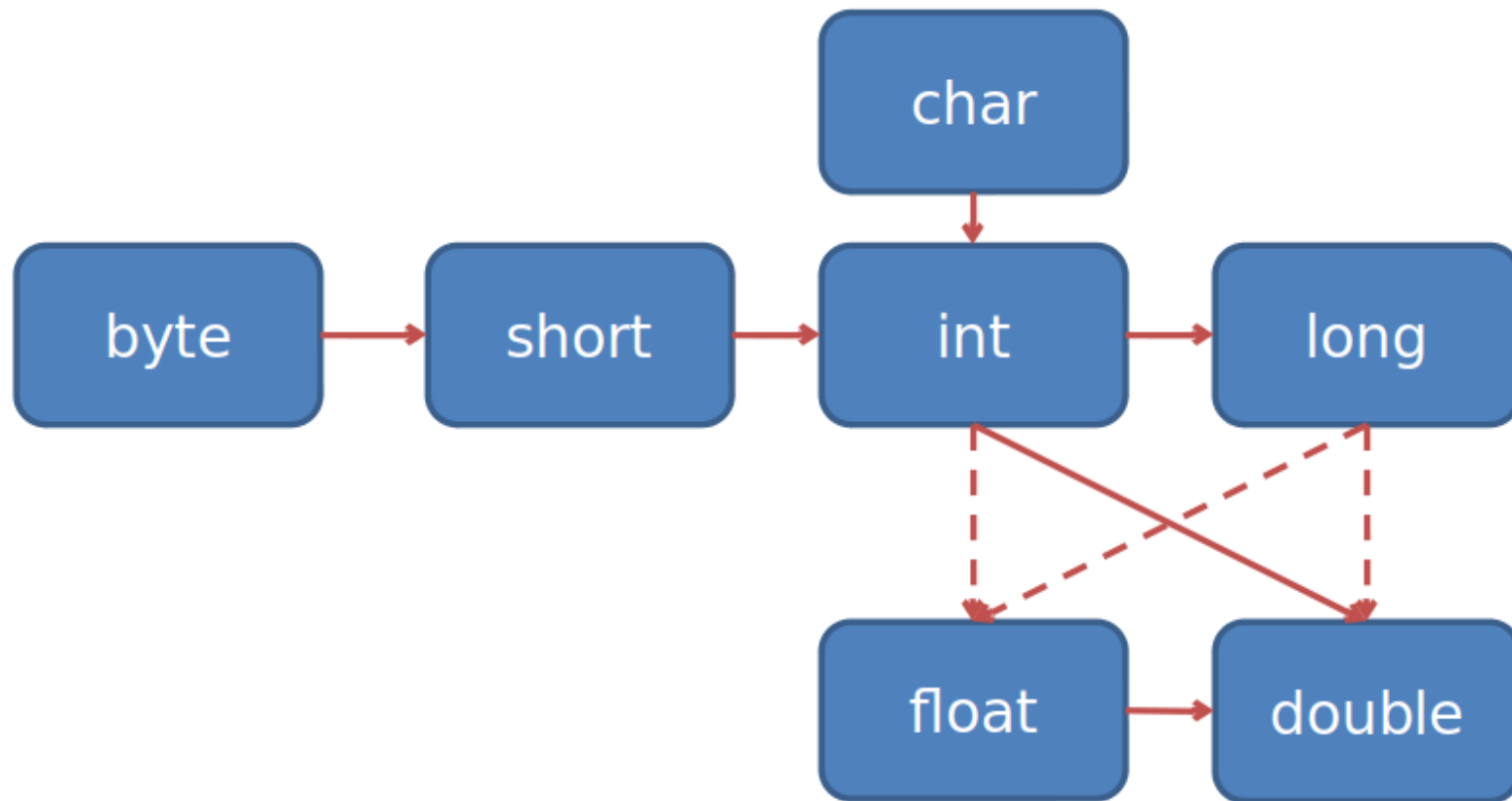
```
final int LIMIT = 5;  
final String NUMBER_SYSTEM = "BINARY";
```

Type Casting

Преобразование типов

- **Widening Casting** (automatically)
- **Narrowing Casting** (manually)

Преобразование типов



Examples: Widening Casting

```
short a = 'Z';  
System.out.println(a); // 90  
double b = 1_234_567_890_123_456L;  
System.out.println(b); // 1.234567890123456E15
```

Examples: Widening Casting with lossy

```
float c1 = 1_234_567_890_123_456L;  
System.out.println(c1); // 1.23456795E15  
float c2 = 123_456_789;  
System.out.println(c2); // 1.23456792E8  
double c3 = 1_234_567_890_123_456_789L;  
System.out.println(c3); // 1.23456789012345677E18
```

Examples: Narrowing Casting

```
byte d = 128; // error: incompatible types: possible lossy conversion from :
char e = 2L; // error: incompatible types: possible lossy conversion from lo
short f = '\uffff'; // error: incompatible types: possible lossy conversion

byte i1 = (byte) 128;
byte i2 = (byte) 129;
System.out.println(i1); // -128
System.out.println(i2); // -127
```

Java

Java

- **Java - строго типизированный** язык программирования.
- т.е. типы переменных должны быть известны до **compile time**
- (@since 10) или могут **ОДНОЗНАЧНО** установлены во время **compile time**