



Object-Oriented Programming in the Java language



Part 0. Java fundamentals

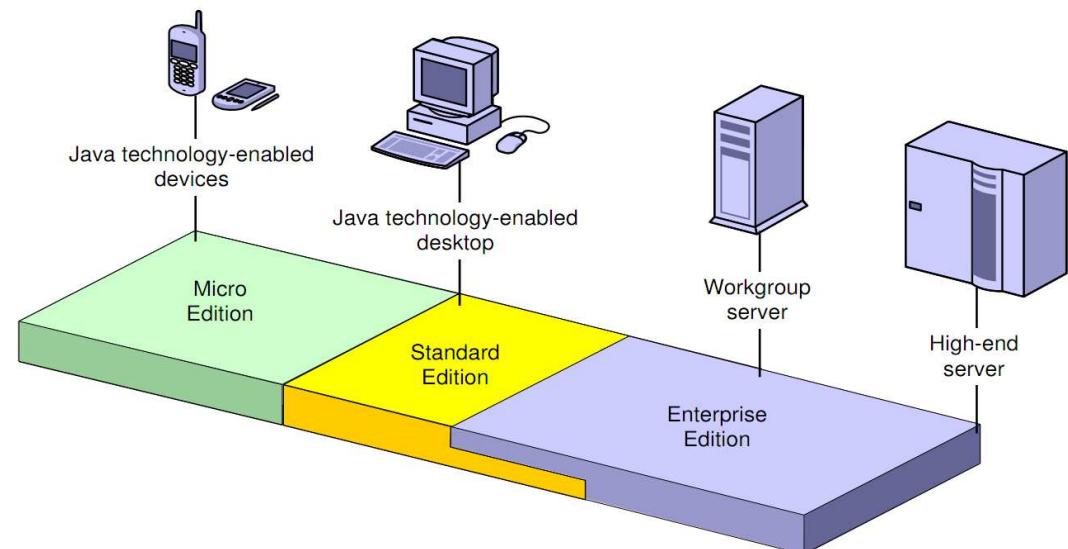
Yevhen Berkunskyi, NUoS
eugeny.berkunsky@gmail.com
<http://www.berkut.mk.ua>





What Java is?

- Programming language
- Platform:
 - Hardware
 - Software OS: Windows, Linux, Solaris, MacOS etc.
- Developer's community
- Technologies

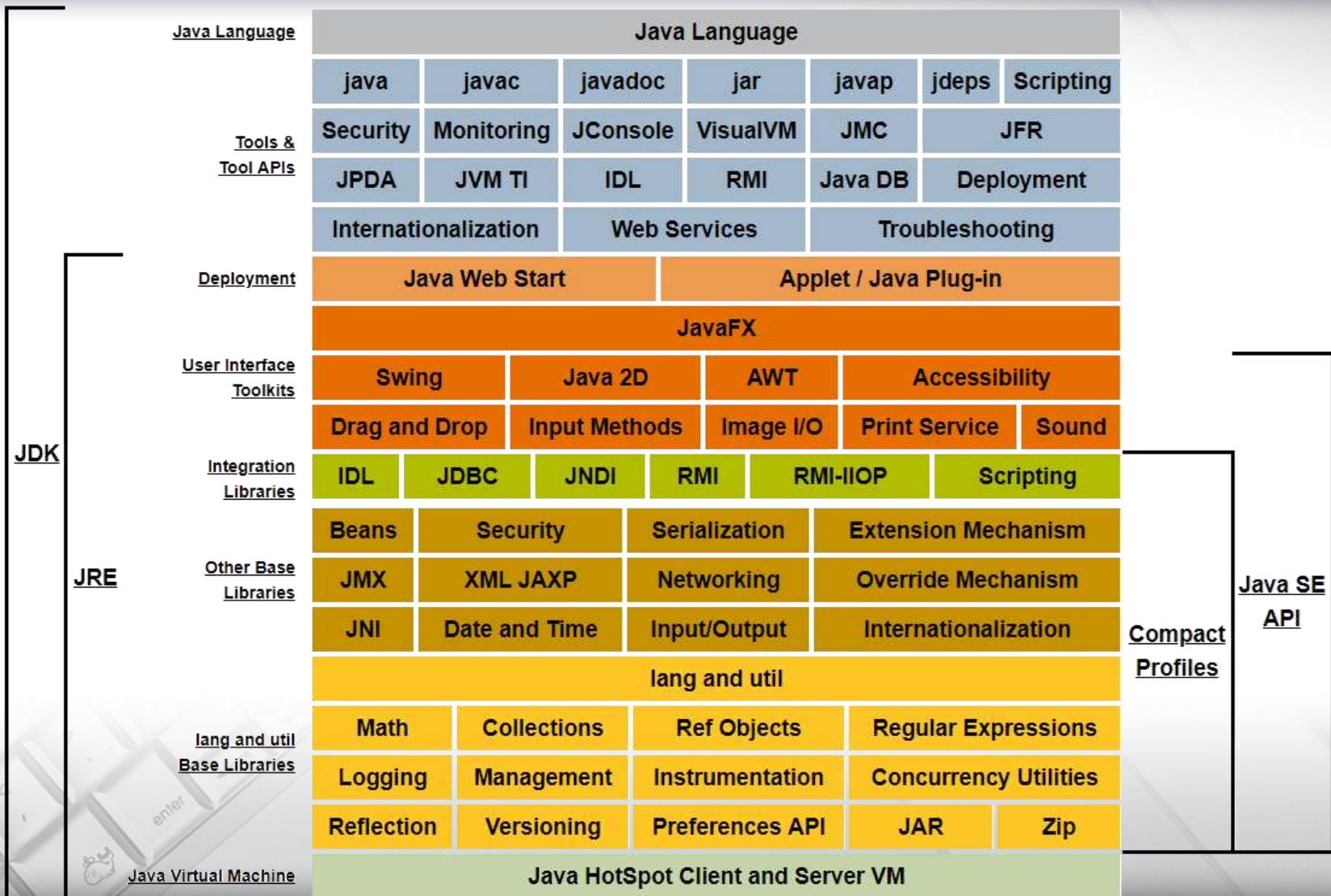


Java Platform

- Developer tools are for any platform.
- Java Virtual Machine, JVM ensures uniformity of the interface with the operating system.
- Portability: «Write once, run everywhere».
- Provided with rich class library JDK (Java Development Kit).
- JRE (Java Runtime Environment) – environment that allows you to run the Java programs

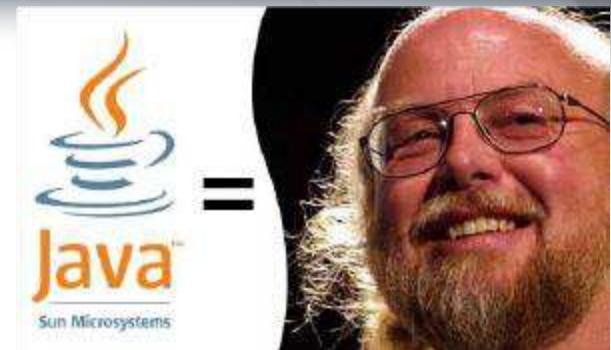


Java SE Technologies



Brief history of Java

- Was created in 1991-1995 by James Gosling group
- First name was “Oak”
 - Renamed to Java, because language Oak was exist.
- Official birthday – May 23, 1995
- Main reason for create
 - The need for platform-free language to embed in appliances
- Possibility of using for WWW





Development of Java: releases

- 1.4.0 Merlin 2002/2/13
- 1.4.1 Hopper 2002/10/16
- 1.4.2 Mantis 2003/5/29
- 5.0 Java SE 5 2004/9/30
- Java SE 6 2006/12/15
- Java SE 7 2011/7/7
- **Java SE 8 2014/3/18**

Java SE 9 2017/9/27

Java SE 10 2018/3/20

Java SE 11 2018/9/25

Java SE 12 2019/03/19

Java SE 13 2019/09/17

.....

Java SE 17 2021/09/14

Java SE 18 2022/03/22

Java vs C++ differences

- Operator overloading
- Multiply inheritance
- Automated type casting
- Address arithmetic
- Destructors
- ...

All this out!

Google: “java c++ differences”

Portable code in Java

- Programs distribute as class-files or as jar-packages.
- Class-file contains intermediate code (bytecode).
- Bytecode – is set of data and statement sequence for JVM.
- Class-files execute by JVM.
- Class-file structure can be changed with changing of JVM.



Course organisation

- Objectives
- Program
- Tools
- Summarizing
- Term paper



Course objectives

- Learn Java basics
- Using modern IDE for Java
- Learn of Object-Oriented principles of program design
- Learn of standard libraries

Program

- Installing Java and IDE
- Structure of Java program
- Flow Control in Java
- OOP basics
- Arrays, strings, as Java objects
- Collections and Maps
- Files. Input and Output
- Exceptions and handling exceptions
- New possibilities in Java SE 8/11/17

Tools

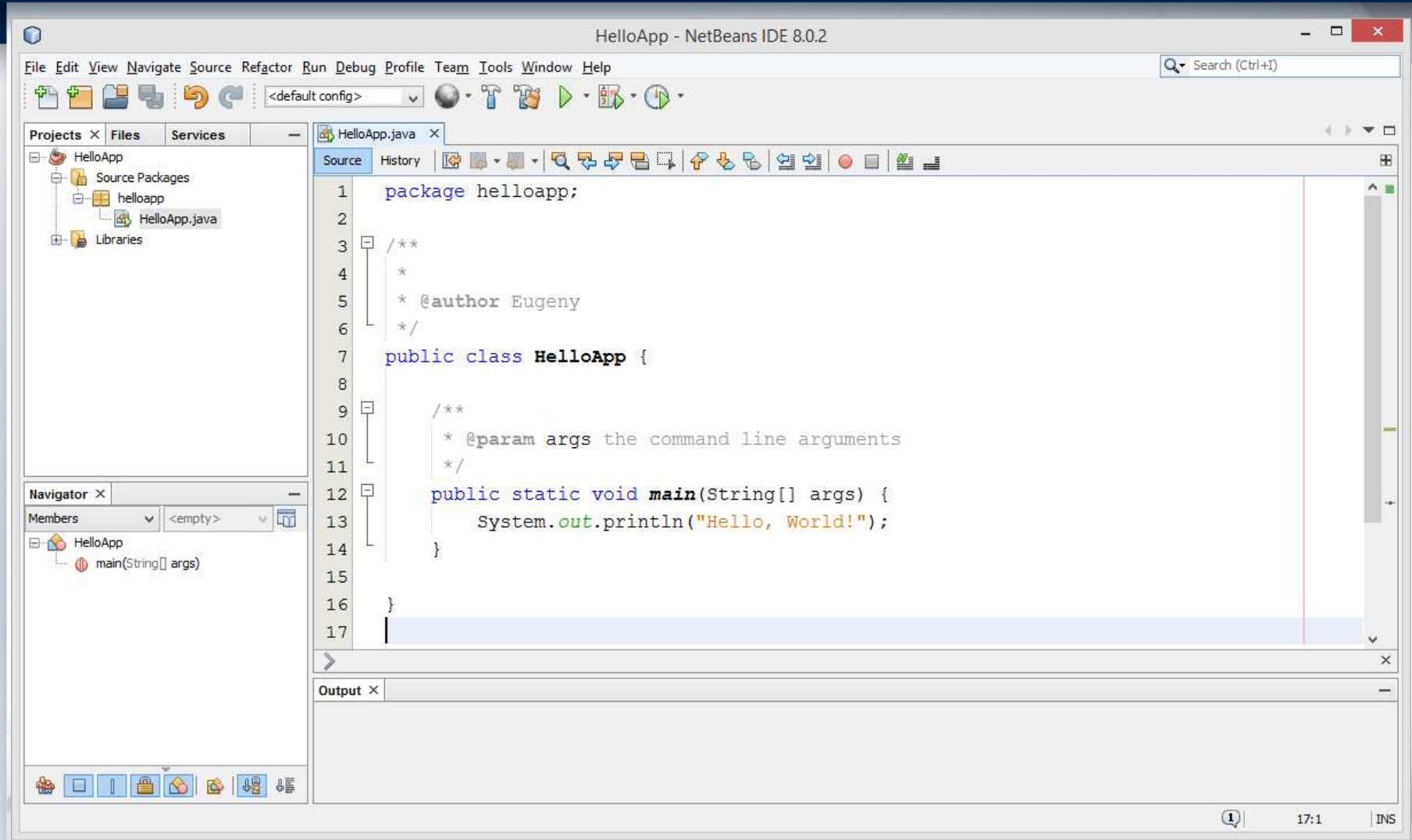
- Compiler and SDK:
 - JDK 17: [Oracle JDK OpenJDK](#)
[Liberica JDK](#) – choose version 8, 11 or 17
with (full) or without (standard) JavaFX
- IDEs
 - Apache NetBeans 12: <http://netbeans.apache.org>
 - JetBrains IntelliJ IDEA 2021.3.x [jetbrains.com/idea/](https://www.jetbrains.com/idea/)
 - Eclipse and other

JDK

JDK contains set of tools for create Java Apps.

Утилита	Описание
javac	Java Compiler. Compile source code to intermediate bytecode
java	Bytecode interpreter. Executes class
javadoc	Tool for creating standard documentation JavaDoc
javah	Tool for header creation for C/C++ integration
jar	Tool for create distributing jars for Java programs
javap	Disassembler

NetBeans IDE





JetBrains IntelliJ IDEA

The screenshot shows the IntelliJ IDEA interface with a Java project named "first". The "Main.java" file is open in the editor, displaying the following code:

```
package first;  
  
public class Main {  
    public static void main(String[] args) {  
        // write your code here  
    }  
}
```

The code editor has syntax highlighting for Java, with "package", "public", "class", "static", "void", and "main" keywords in orange, and strings in blue. A yellow circle highlights the placeholder comment "// write your code here". The IntelliJ interface includes a Project tool window on the left, a toolbar at the top, and various panels like Structure, Bookmarks, Version Control, and Event Log on the right.



JetBrains IntelliJ IDEA

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** first > src > first > Main > main
- Editor:** The Main.java file is open, displaying the following code:

```
package first;
public class Main {
    public static void main(String[] args) {
        // write your code here
    }
}
```
- Sidebar:** Project (first), External Libraries, Scratches and Consoles.
- Right Sidebar:** Embedded Web Browser, alxCoder, Database.
- Bottom:** Version Control, TODO, Problems, Terminal, Profiler, Event Log.
- Status Bar:** 6:28 CRLF UTF-8 4 spaces.



Keywords

abstract	continue	for	new	switch
assert	default	if	package	synchronized
boolean	do	goto	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while
<i>_ (underscore)</i>				

Keywords not currently in use: const goto

New keyword in Java SE 9: _



Reserved Literals

null true false
var (since JDK 10/11)
yield (since JDK 16)

Literals

Examples:

Integer 2000 0 -7

Floating-point 3.14 -3.14 .5 0.5

Character 'a' 'A' '0' ':' '-' ')

Boolean true false

String "abba" "3.14" "for" "a piece of the action"



Integer Literals

Decimal	10235 104L
Octal	01234
Hexadecimal	0x12F
Binary	0b101



Floating-Point

Examples of double Literals

0.0	0.0d	0D		
0.49	.49	.49D		
49.0	49.	49D		
4.9E+1	4.9E+1D	4.9e1d	4900e-2	.49E2

Examples of float Literals

0.0F	0f	
0.49F	.49F	
49.0F	49.F	49F
4.9E+1F	4900e-2f	.49E2F

Character Literals

A character literal is quoted in single-quotes (').

All character literals have the primitive data type char.

A Unicode character can always be specified as a four-digit hexadecimal number (i.e., 16 bits) with the prefix \u.

Character Literals examples

'	'	'	'\u0020'	Space	'a'	'	'\u0061'	a
'0'	'	'\u0030'	0		'b'	'	'\u0062'	b
'1'	'	'\u0031'	1		'z'	'	'\u007a'	z
'9'	'	'\u0039'	9		'Ñ'	'	'\u0084'	Ñ
'A'	'	'\u0041'	A		'å'	'	'\u008c'	å
'B'	'	'\u0042'	B		'ß'	'	'\u00a7'	ß
'Z'	'	'\u005a'	Z					



String Literals

Examples:

"Here comes a tab.\t And here comes another one\u0009!"

"What's on the menu?"

"\"String literals are double-quoted.\\""

"Left!\nRight!"

"Don't split me up!"



White Spaces

A white space is a sequence of spaces, tabs, form feeds, and line terminator characters in a Java source file.

Line terminators can be:

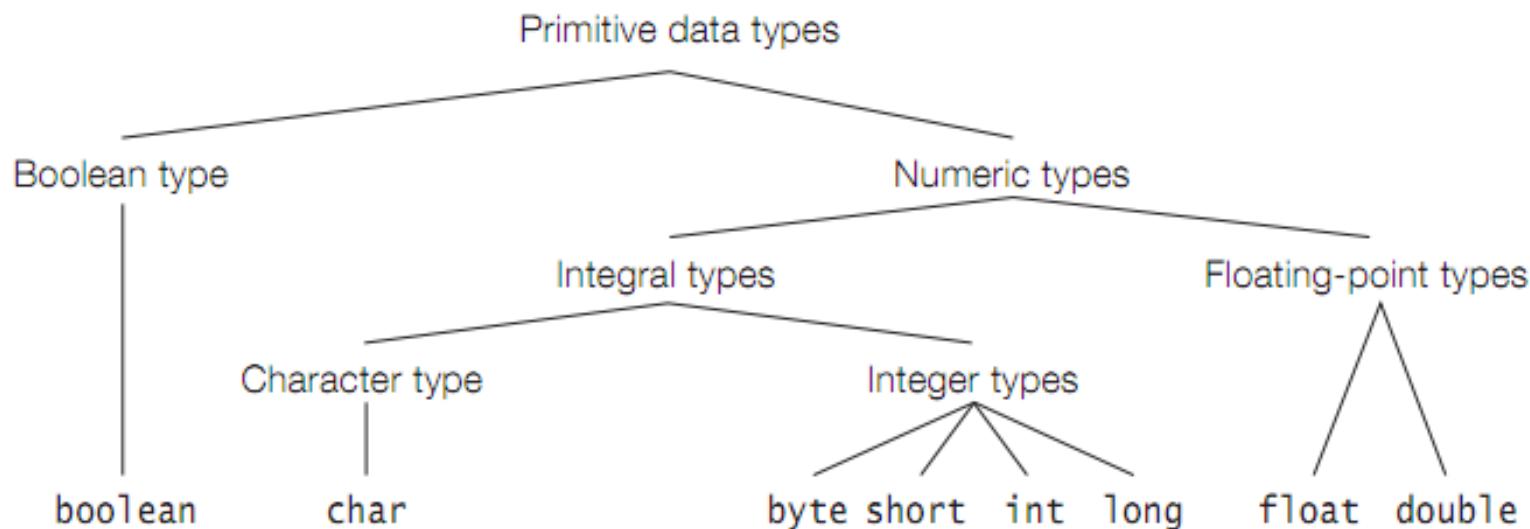
- newline,
- carriage return,
- carriage return - newline sequence.

Comments

- Single-Line Comment //
- Multiple-Line Comment /* */
- Documentation Comment /** */



Primitive Data Types





Integer Types

<u>type</u>	<u>size</u>	<u>min value</u>	<u>max value</u>
byte	8	-2^7 (-128)	2^7-1 (+127)
short	16	-2^{15} (-32768)	$2^{15}-1$ (+32767)
int	32	-2^{31} (-2147483648)	$2^{31}-1$ (+2147483647)
long	64	-2^{63}	$2^{63}-1$ (-9223372036854775808L) (9223372036854775807L)



The char Type

<u>type</u>	<u>size</u>	<u>min value</u>	<u>max value</u>
char	16	0x0 (\u0000)	0xffff (\uffff)



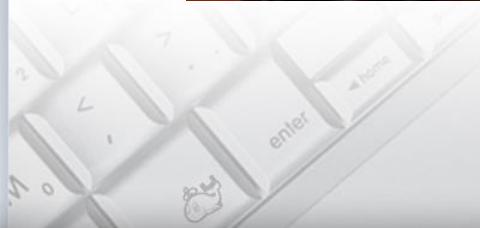
The Floating-Point Types

<u>type</u>	<u>size</u>	<u>min value</u>	&	<u>max value</u>
float	32	1.401298464324817E-45f		3.402823476638528860e+38f
double	64	4.94065645841246544e-324		1.79769313486231570e+308



НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА

Example





Questions?

