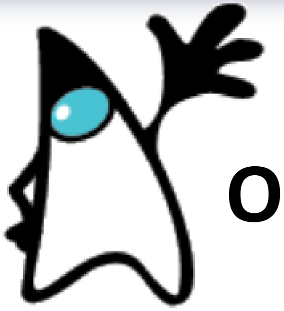
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА



Object-Oriented Programming in the Java language

Java fundamentals

Yevhen Berkunskyi, NUoS

eugeny.berkunsky@gmail.com

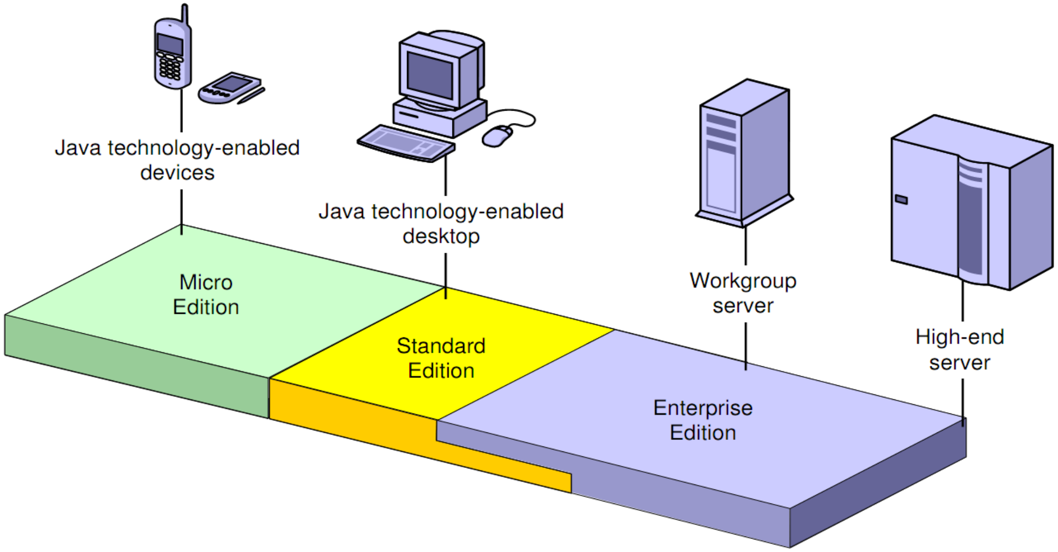
http://www.berkut.mk.ua

Tools

- Compiler and SDK:
  - JDK 21: Oracle JDK OpenJDK Liberica JDK – choose version 11, 17 or 21 with (full) or without (standard) JavaFX
- IDEs
  - Apache NetBeans 20: http://netbeans.apache.org
  - JetBrains IntelliJ IDEA 2023.3.x jetbrains.com/idea/
  - Eclipse and other

What Java is?

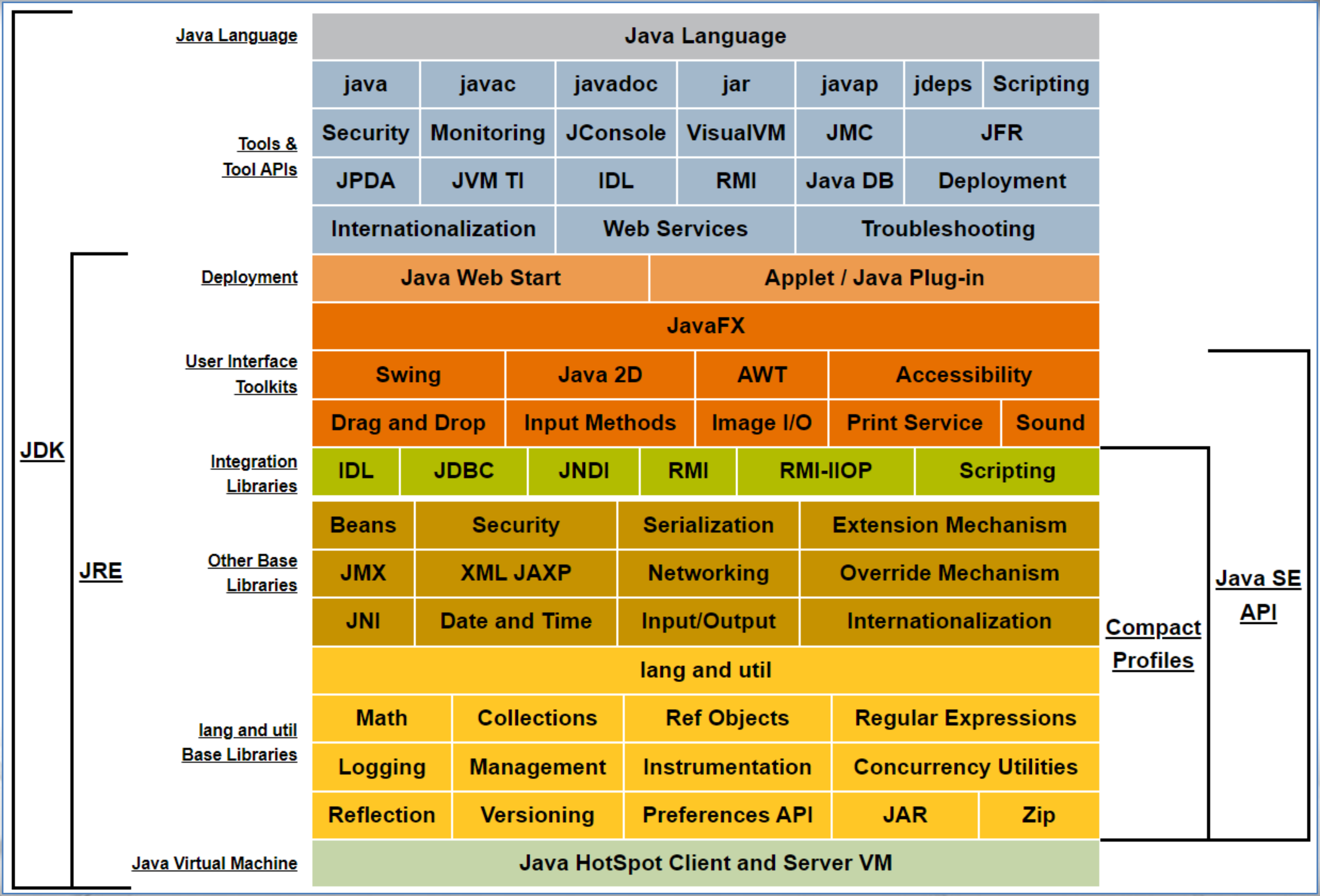
- Programming language
- Platform:
  - Hardware
  - Software OS: Windows, Linux, Solaris, MacOS etc.
- Developer’s community
- Technologies



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



JDK

JRE

Java Language: Java Language

Tools & Tool APIs:
java | javac | javadoc | jar | javap | jdeps | Scripting
Security | Monitoring | JConsole | VisualVM | JMC | JFR
JPDA | JVM TI | IDL | RMI | Java DB | Deployment
Internationalization | Web Services | Troubleshooting

Deployment: Java Web Start | Applet / Java Plug-in

User Interface Toolkits:
JavaFX
Swing | Java 2D | AWT | Accessibility
Drag and Drop | Input Methods | Image I/O | Print Service | Sound

Integration Libraries: IDL | JDBC | JNDI | RMI | RMI-IIOP | Scripting

Other Base Libraries:
Beans | Security | Serialization | Extension Mechanism
JMX | XML JAXP | Networking | Override Mechanism
JNI | Date and Time | Input/Output | Internationalization

lang and util Base Libraries:
lang and util
Math | Collections | Ref Objects | Regular Expressions
Logging | Management | Instrumentation | Concurrency Utilities
Reflection | Versioning | Preferences API | JAR | Zip

Java Virtual Machine: Java HotSpot Client and Server VM

Compact Profiles

Java SE API

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

Very old versions

… … … … … … … … … … … …

Old versions

- 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**

New history

Java 9 2017/9/27

Java 10 2018/3/20

Java 11 2018/9/25

Java 12 2019/03/19

Java 13 2019/09/17

…….

Java 17 2021/09/14

…….

Java 20 2023/03/21

Java 21 2023/09/19

Java 22 2024/03/15

Portable code in Java

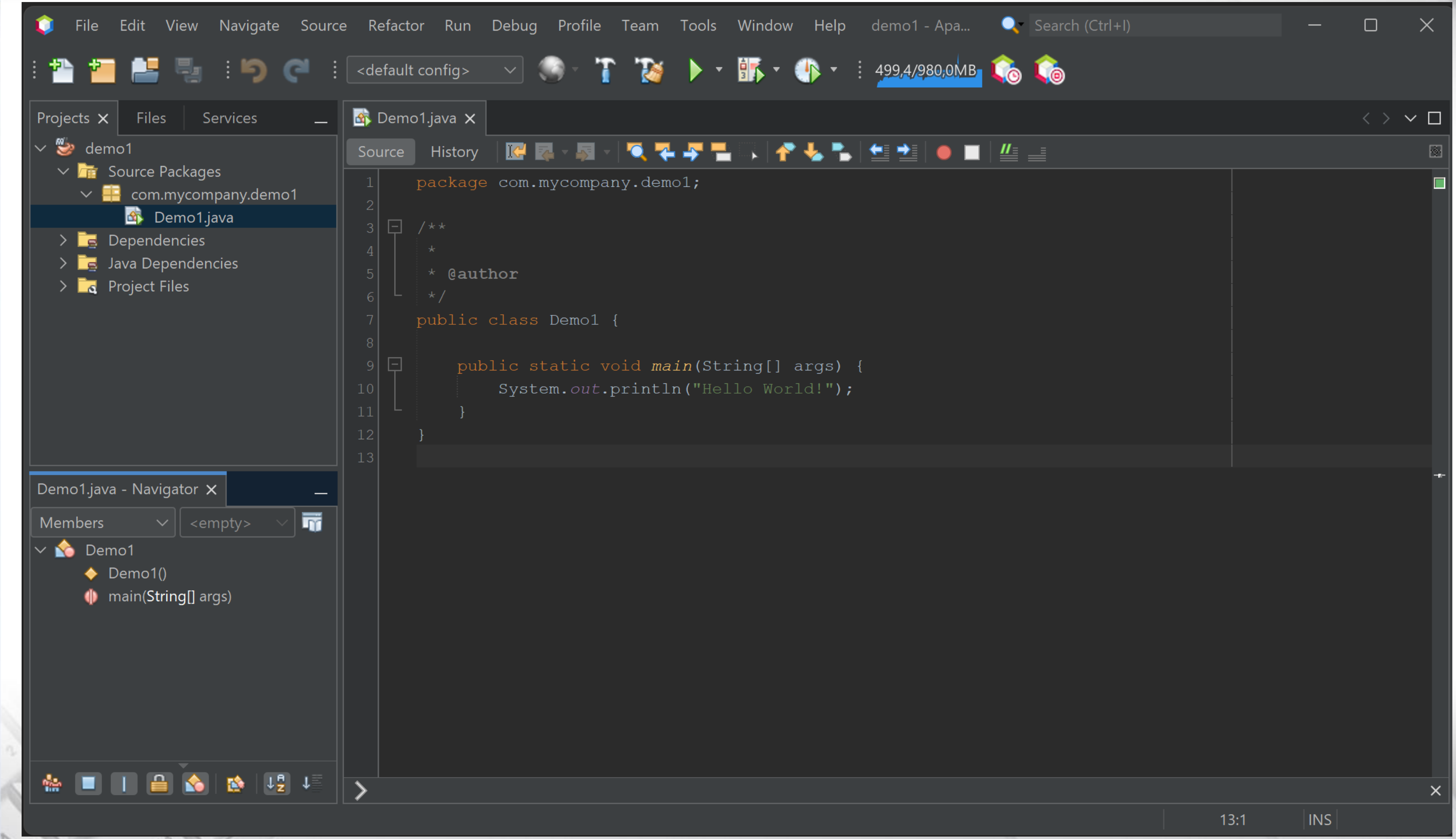
- Programs distribute as class-files or as jar-packages.
- Class-file contains intermediate code (bytecode).
- Bytecode – is set of data and statement sequence for JVM.
- Class-files execute by JVM.
- Class-file structure can be changed with changing of JVM.

JDK

JDK contains set of tools for create Java Apps.

Utility	Description
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



- demo1
  - Source Packages
    - com.mycompany.demo1
      - Demo1.java
  - Dependencies
  - Java Dependencies
  - Project Files

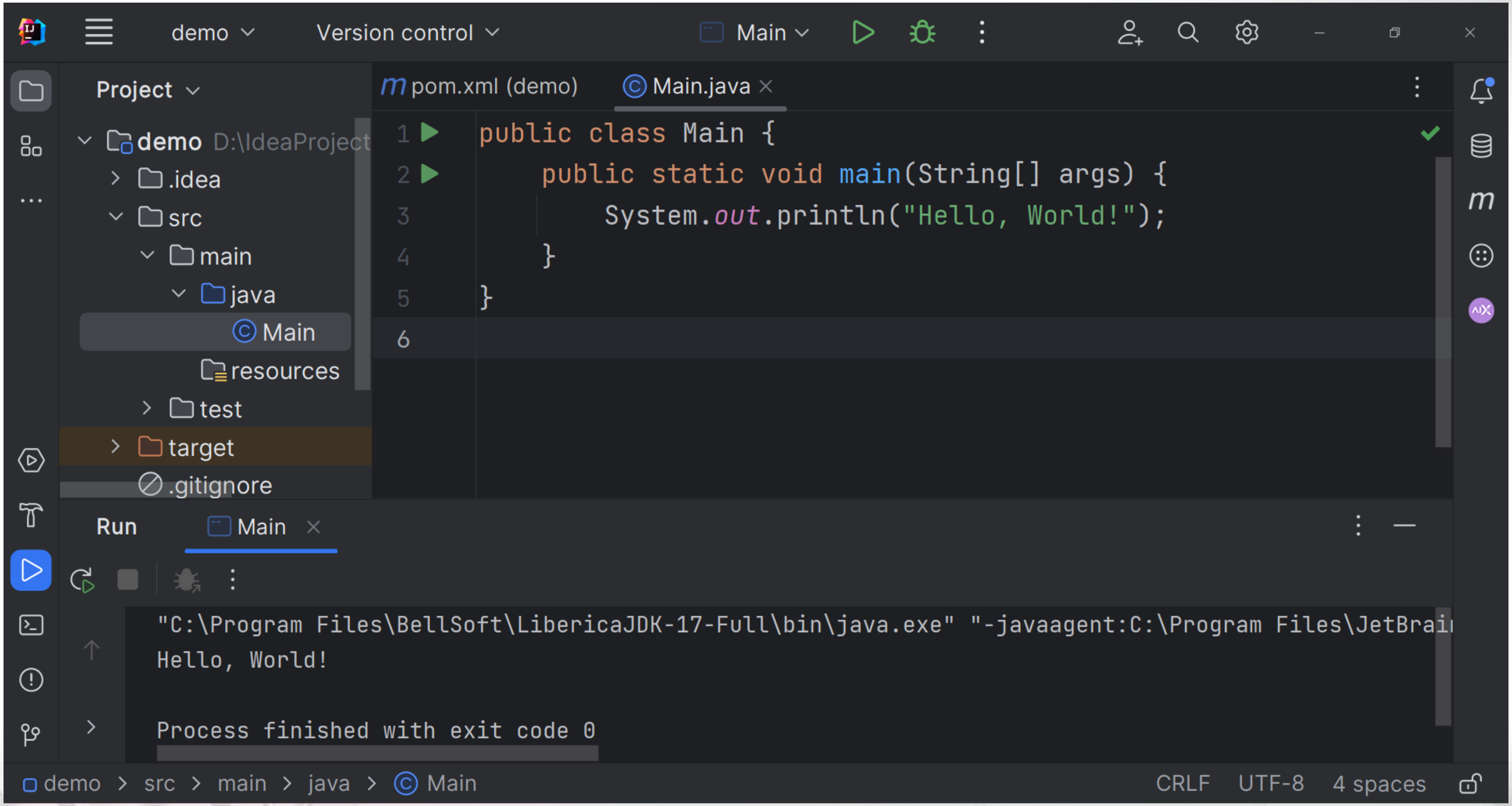
```java
package com.mycompany.demo1;

/**
 *
 * @author
 */
public class Demo1 {

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

- Demo1
  - Demo1()
  - main(String[] args)

JetBrains IntelliJ IDEA



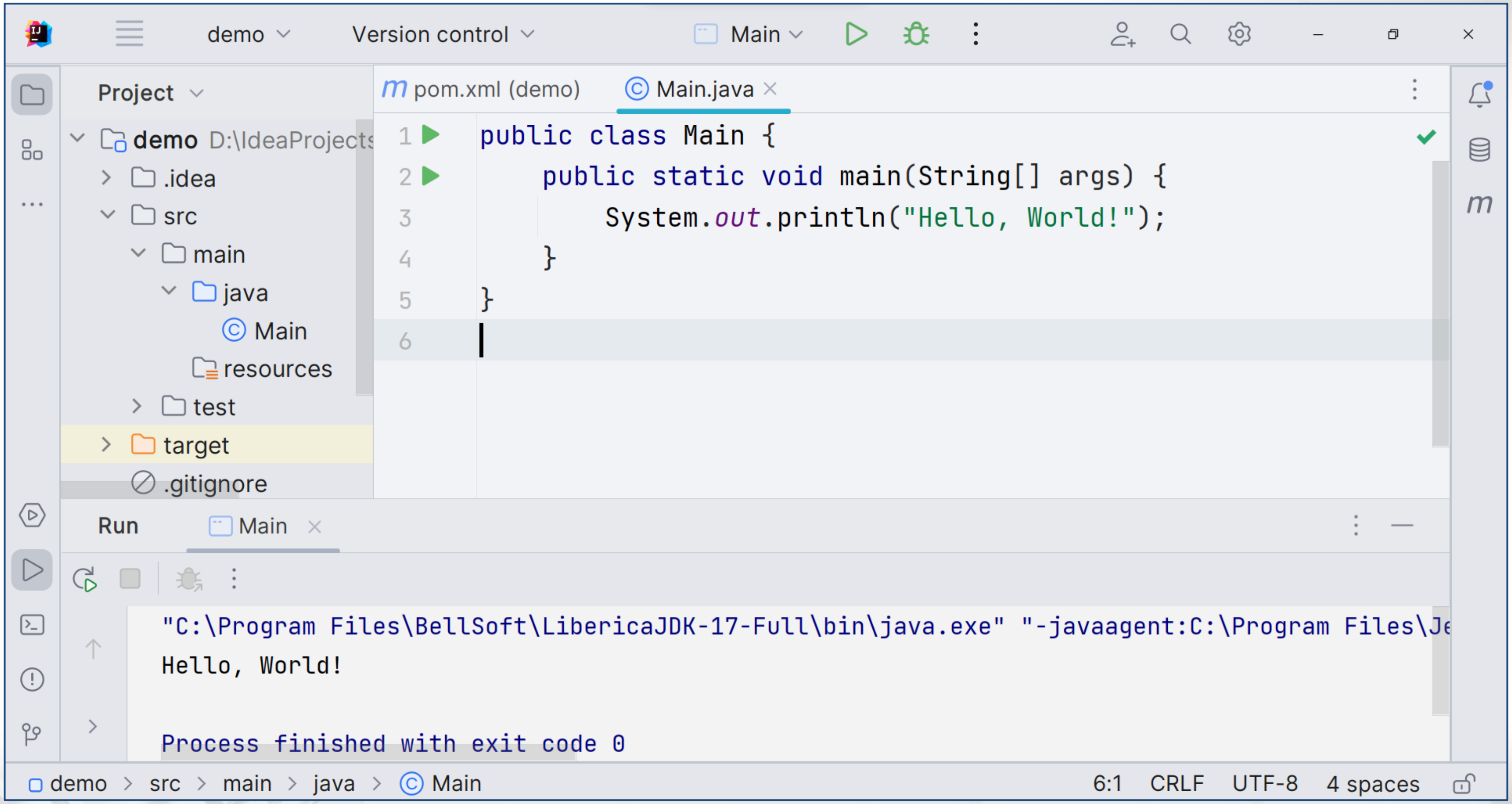
```java
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

```
"C:\Program Files\BellSoft\LibericaJDK-17-Full\bin\java.exe" "-javaagent:C:\Program Files\JetBrai
Hello, World!

Process finished with exit code 0
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

JetBrains IntelliJ IDEA

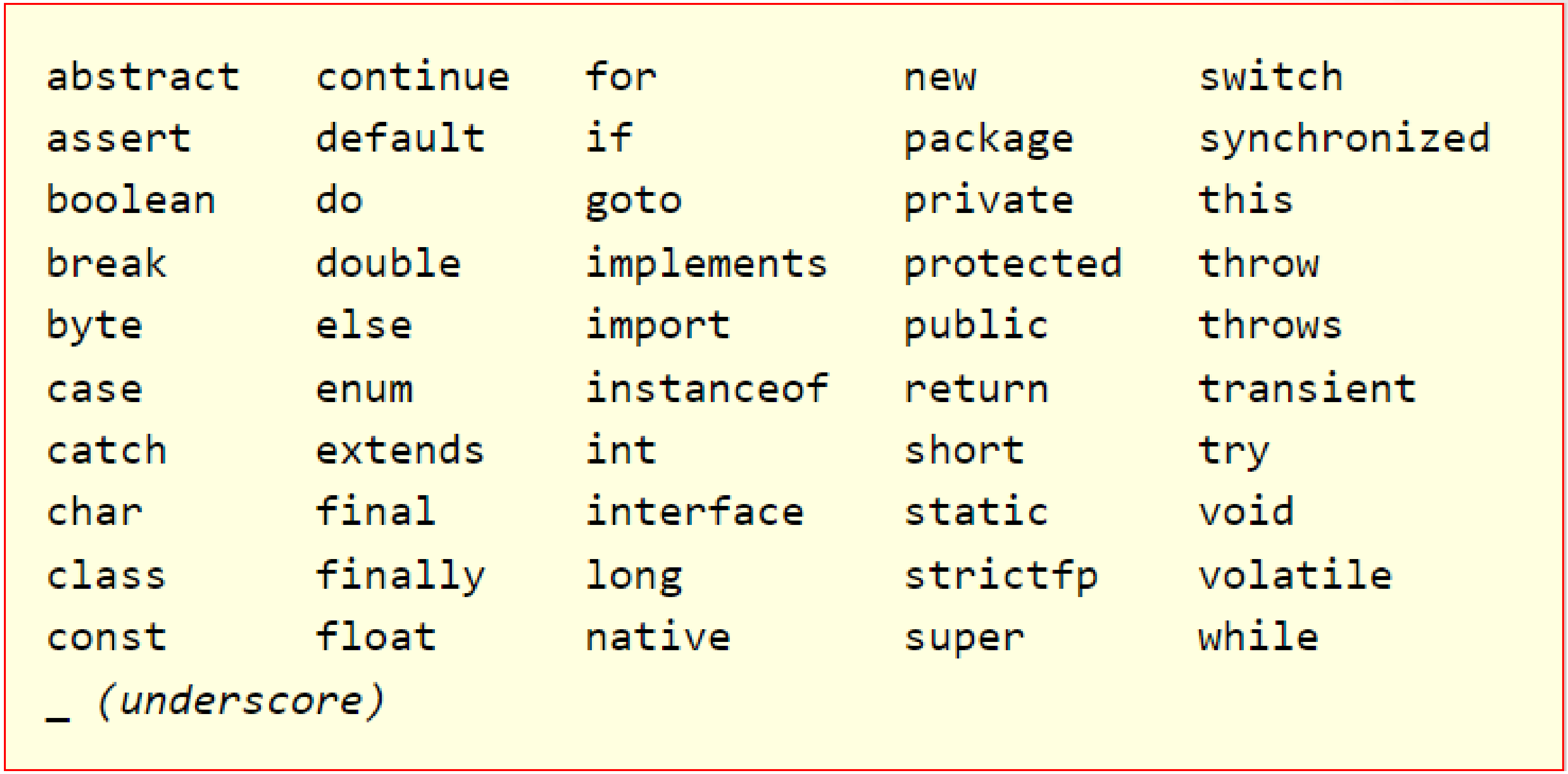


```java
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

```
"C:\Program Files\BellSoft\LibericaJDK-17-Full\bin\java.exe" "-javaagent:C:\Program Files\Je
Hello, World!

Process finished with exit code 0
```

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
_ (underscore)
```

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'	'\u00d1'	Ñ
'A'	'\u0041'	A	'å'	'\u00e5'	å
'B'	'\u0042'	B	'ß'	'\u00df'	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!"


"""
  First line.
  Second line.
  Last line."""
```

White Spaces

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

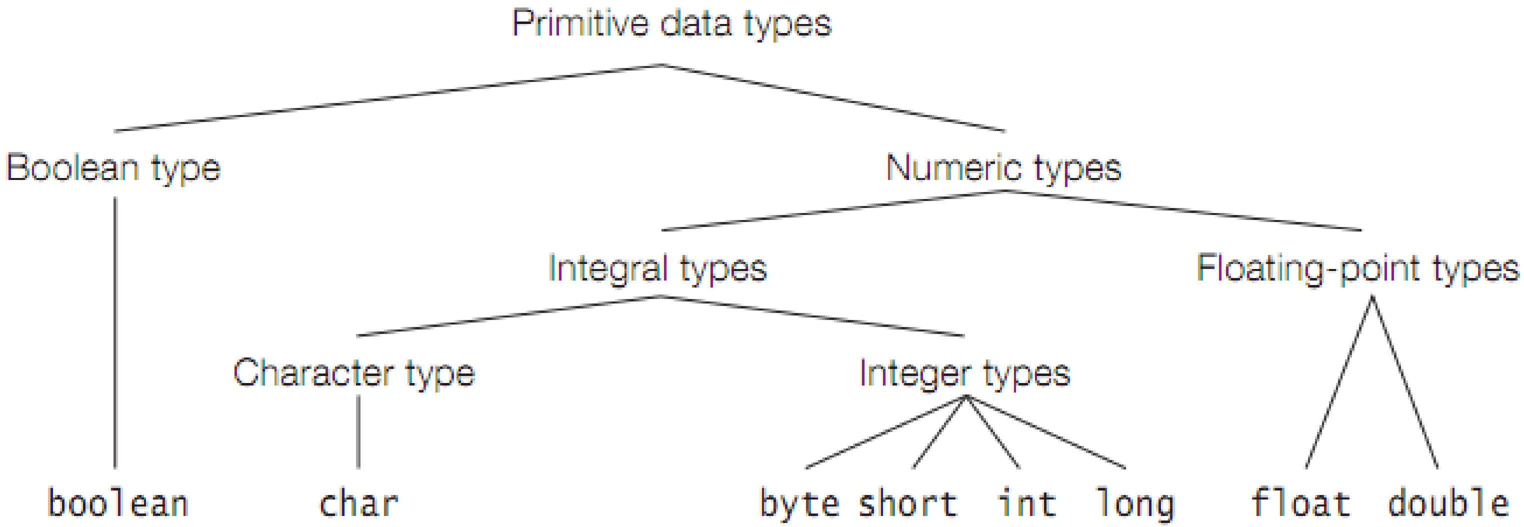
Line terminators can be:

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

Comments

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

Primitive Data Types



Integer Types

type	size	min value	max value
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}	$2^{63}-1$

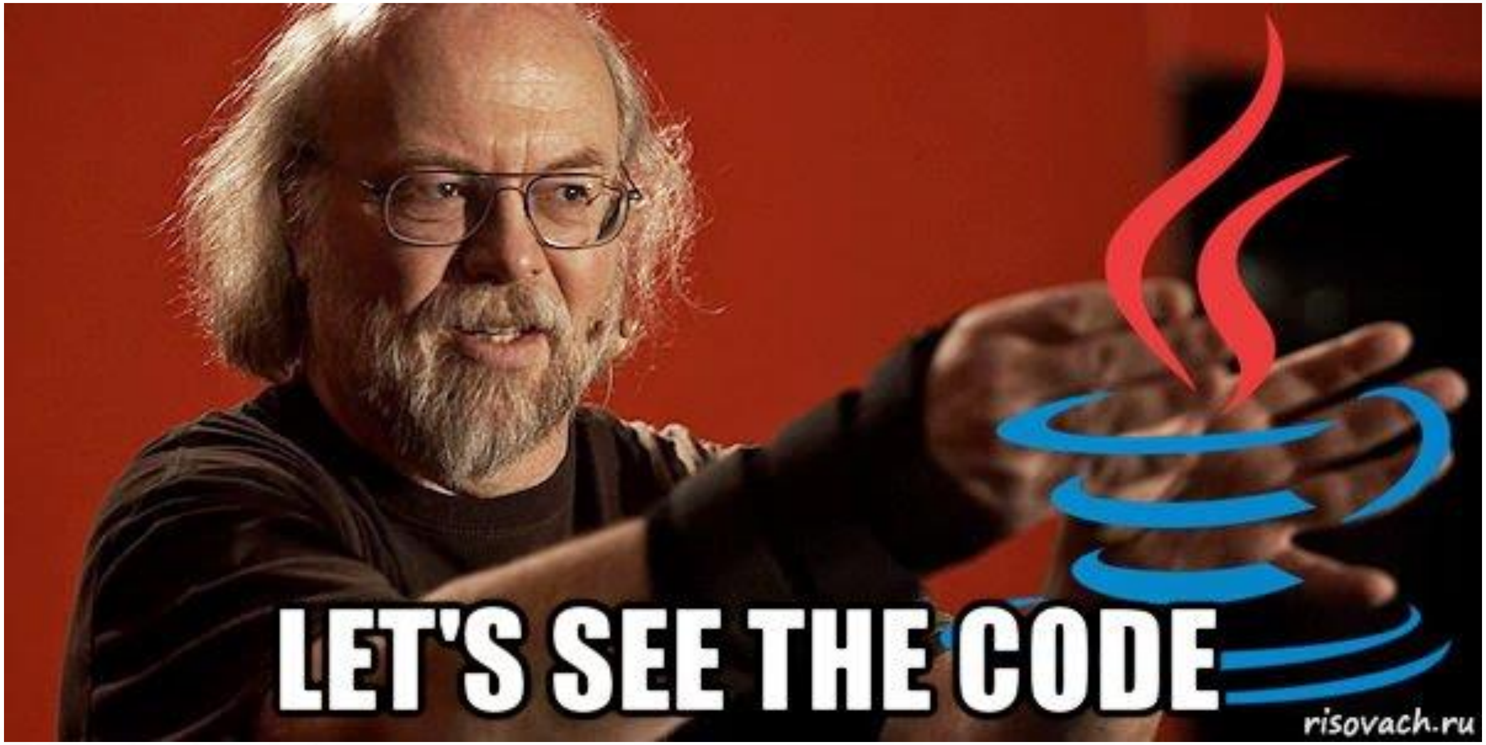
The char Type

type	size	min value	max value
char	16	0x0 (\u0000)	0xffff (\uffff)

The Floating-Point Types

type	size	min value	&	max value
float	32	1.401298464324817E-45f		3.40282347663852886 0e+38f
double	64	4.9406564584124654 4e-324		1.797693134862315 70e+308

Example



Next slide…

Questions?