

Программирование на Java

13. Java 13 и другие

Глухих Михаил Игоревич
mailto: glukhikh@mail.ru

“Древние” версии Java

- ▶ Когда что появилось?
 - Strings in switch, try with resources, <>:
Java 7 aka 1.7 (2011)
 - Generics, Enums, Varargs, for-each:
Java 5 aka 1.5 (2004)
 - Collections framework, Swing:
Java 1.2 (1998)
 - Inner classes:
Java 1.1 (1997)
 - Initial version:
Java 1.0 (Jan 1996)

Java 1.8 aka 8 (2014)

- ▶ “Прошлое”
- ▶ Long-term support! (until 2030)

- ▶ Streams / lambdas / SAMs
- ▶ Default method in interfaces
- ▶ Annotations on types
- ▶ JavaFX in JDK

Изменение графика релизов

- ▶ Начиная с Java 9 Oracle переходит на график «1 релиз в полгода»
- ▶ При этом далеко не все релизы являются LTS (Long-term support)

Java 9–10–11 (2017–18)

- ▶ “Настоящее”
- ▶ Java 11 is long-term support! (until 2026)
- ▶ Remove JavaFX, Java EE
- ▶ More powerful garbage collector API
- ▶ `java.net.HttpClient`
- ▶ private interface methods

Java 9–10–11 (2017–18)

- ▶ “Настоящее”
- ▶ Java 11 is long-term support! (until 2026)

- ▶ Local variable type inference
 - `var x = ...`
 - `for (var x: ...) { ... }`
 - `try (var x = BufferedReader(...)) { ... } catch ...`
 - `(var x, var y) -> x.foo(y)`

Java 12–13–14 (2019–20)

- ▶ Non-LTS (2020 support only)
- ▶ “Будущее”
- ▶ Switch expressions
 - `int x = switch (someEnum) {
 case FIRST -> 1;
 case SECOND -> {
 System.out.println("!!!");
 yield 2;
 }
 default -> 3;
};`
 - `->` allows to omit break

Java 12-13-14 (2019-20)

- ▶ NB: Yet preview!
- ▶ Instanceof “pattern matching”
 - `if (someObj instanceof String ss && ss.length() > 0)`

Java 12-13-14 (2019-20)

- ▶ NB: Yet preview!
- ▶ Records
 - `record` `Point(int x, int y) { ... }`
 - ~ Kotlin data classes
 - constructor
 - equals, hashCode
 - Getters
 - toString

Java 12-13-14 (2019-20)

- ▶ Three-quoted multi-line strings
aka Text Blocks

- String s = `"""`

- There is the first line.

- There is the second line.

- `""";`
;

Java 17 (not yet)

- ▶ Next long-term support version
- ▶ ETA 2021–2022

- ▶ Possible features
 - Release preview stuff
 - More powerful pattern matching
 - Sealed types (like Kotlin sealed classes)
 - Value types (like Kotlin inline classes)

Bcë!