

- ☑ Локализация — адаптация программных продуктов для определенного языка и/или определенной местности
 - Перевод текста
 - Использование соответствующих форматов данных
 - Замена звуковой и визуальной информации
 - localization = l10n

| | | | | | | | | |
|---------|-----|----|----|----|---|---|---|----|
| 0005610 | 001 | \0 | \0 | \0 | h | e | l | l |
| 0005620 | o | , | w | o | r | l | d | \0 |
| 0005610 | 001 | \0 | \0 | \0 | п | р | и | в |
| 0005620 | е | т | , | | м | и | р | \0 |

- ☑ Интернационализация — проектирование программ таким образом, чтобы их локализация была возможна без конструктивных изменений
 - выделение текстовых данных из кода
 - отображение данных с учетом местных форматов
 - internationalization = i18n

application.exe

messages.txt

```
[hello]  
en=hello
```

✓ application.exe

✓ messages.txt

```
[hello]
en=hello
ru=привет
it=ciao
es=hola
fr=salut
de=hallo
zh=你好
```

- ✓ Локаль — совокупность характеристик, определяющих географический, политический или культурный регион
- ✓ Класс `java.util.Locale`
- ✓ Примеры: `it`, `en_UK`, `ru_RU.CP1251`
- ✓ Элементы локали:
 - язык — 2 строчные буквы (иногда 3) (`ru`)
 - страна (регион) — 2 заглавные буквы (`RU`) или 3 цифры
 - вариант (например, кодировка для русской локали)
 - письменность — 4 буквы, первая заглавная (`Cyrl`)
 - расширение

☑ Конструкторы класса Locale

- `new Locale(String lang)`
- `new Locale(String lang, String country)`
- `new Locale(String lang, String country, String variant)`

☑ Класс Locale.Builder

- `new Locale.Builder().setLanguage("ru").setRegion("RU").build()`

☑ Метод forLanguageTag()

- `Locale.forLanguageTag("ru-RU");`

☑ Константы класса Locale

- `Locale.FRENCH`
- для русского константы нет

- ☑ Получение списка локалей и локали по умолчанию
 - `static Locale[] getAvailableLocales()`
 - `static Locale getDefault()`
- ☑ Преобразование в строку
 - `String toString() // ru_RU`
 - `String getDisplayName() // Russian (Russia)`

- ☑ Класс ResourceBundle
- ☑ Формат "ключ-значение"
 - ключ - условная строка, значение - перевод
- ☑ 2 варианта:
 - Текстовый файл свойств *.property (PropertyResourceBundle)
 - Класс со списком (ListResourceBundle)

☑ `GuiLabels_en.properties`

`s1 = Yes`

`s2 = No`

`GuiLabels_ru.properties`

`s1 = Да`

`s2 = Нет`

```
ResourceBundle r = ResourceBundle.getBundle("GuiLabels");
```

```
JButton b1 = new JButton(r.getString("s1"));
```

```
JButton b2 = new JButton(r.getString("s2"));
```

☑ + отдельные текстовые файлы

☑ - только String

☑ Кодировка **ISO-8859-1** — необходима обработка с помощью `native2ascii` (до Java 9)

```
public class GuiLabels_en extends ResourceBundle {
    public Object[][] getContents() { return contents; }
    private Object[][] contents = { {"s1", "Yes"}, {"s2", "No"} };
}

public class GuiLabels_ru extends ResourceBundle {
    public Object[][] getContents() { return contents; }
    private Object[][] contents = { {"s1", "Да"}, {"s2", "Нет"} };
}

ResourceBundle r = ResourceBundle.getBundle("GuiLabels");
JButton b1 = new JButton(r.getString("s1"));
JButton b2 = new JButton(r.getString("s2"));
```

+ любые типы объектов

- нужна компиляция файлов



- ☑ Построение списка кандидатов

```
Locale loc = Locale.US; // Locale.getDefault() = ru_RU;
```

```
ResourceBundle rb = ResourceBundle.getBundle("GuiLabels", loc);
```

- 1) GuiLabels_en_US // полная текущая локаль
- 2) GuiLabels_en // сокращенная текущая локаль
- 3) GuiLabels_ru_RU // полная системная локаль
- 4) GuiLabels_ru // сокращенная системная локаль
- 5) GuiLabels // ресурсы по умолчанию (если нет подходящих)

- Формирование набора (bundle) - .class .property
- MissingResourceException

- ✓ `getString(key)`
- ✓ `getStringArray(key)`
- ✓ `getObject(key)`

- ✓ Просматривается список ресурсов
- ✓ Возвращается первая найденная строка
- ✓ `MissingResourceException`

☑ Класс `NumberFormat` — абстрактный

- `NumberFormat nf = NumberFormat.getNumberInstance();`
- `NumberFormat cf = NumberFormat.getCurrencyInstance();`
- `NumberFormat pf = NumberFormat.getPercentInstance();`
- `nf.format(new Float(999.8));`

☑ Класс `DecimalFormat`

- `df = (DecimalFormat) nf;`
- `df.applyPattern("##,##0.00");`
- `df.format(new Float(888.7));`

☑ Класс `DecimalFormatSymbols`

- `ds.setDecimalSeparator('=');`
- `df.setDecimalFormatSymbols(ds);`
- `df.format(new Float(777.6));`

0 — цифра, 0 отображается

— цифра, 0 не отображается

. — разделитель десятичной дроби

, — разделитель групп разрядов

E — разделитель мантиссы и порядка

- — знак минус

; — разделитель подшаблонов

% — умножить на 100 и отобразить как процент

‰ — умножить на 1000 и отобразить как промилле

¤ — символ валюты



☑ Класс DateFormat — абстрактный

- `DateFormat df = DateFormat.getDateInstance(DateFormat.FULL)`
- `DateFormat tf = DateFormat.getTimeInstance(DateFormat.LONG)`
- `DateFormat dtf = DateFormat.getDateTimeInstance(DateFormat.SHORT)`
- `df.format(new Date());`

☑ Класс SimpleDateFormat

- `sdf = (SimpleDateFormat) df;`
- `sdf.applyPattern("yyyy-MM-dd");`
- `sdf.format(new Date());`

☑ Класс DateFormatSymbols

- `ds.setShortWeekdays("пнд", "втр", "срд", "чтв", "птн", "сбт", "вск");`
- `sdf.setDateFormatSymbols(ds);`
- `sdf.format(new Date());`

Более 4 символов — полный формат, 3 — сокращенный, 2 - число

G — эра

y — год

M — месяц после числа

L — месяц (название)

d — число

E — название дня недели

H — часы

m — минуты

s — секунды

S — миллисекунды

z — временная зона



- ☑ 10 октября 2023 в 18:00 будет кольцеобразное солнечное затмение.
- ☑ Annular solar eclipse will happen at 18:00AM on October 10, 2023.

Eclipse_en.properties

```
msg = {0} solar eclipse will happen  
at {1,time,short} on {1,date,short}.
```

```
full = Total  
part = Partial  
ring = Annular
```

Eclipse_ru.properties

```
msg = {1,date,short} в {1,time,short}  
будет {0} солнечное затмение.
```

```
full = полное  
part = частное  
ring = кольцеобразное
```

```
ResourceBundle r = ResourceBundle.getBundle("Eclipse");  
MessageFormat mf = new MessageFormat(r.getString("msg");  
Date date = Date.from(Instant.parse("2023-10-14T18:00"));  
Object[] args = {r.getString("ring", date.getTime());  
mf.format(args);
```

☑ Класс ChoiceFormat extends NumberFormat

- 0 friends like it
- 1 friend likes it
- 1000 friends like it

Like_en.properties

```
msg = {0} it  
one = {0,number} friend likes  
many = {0,number} friends like
```

```
ResourceBundle r = ResourceBundle.getBundle("Like");  
MessageFormat mf = new MessageFormat(r.getString("msg");  
double[] lims = { 0, 1, 2 };  
String one = r.getString("one");  
String many = r.getString("many");  
String[] msgs = { many, one, many };  
ChoiceFormat cf = new ChoiceFormat(lims, msgs);  
mf.setFormatByArgumentIndex(0, cf);  
Object[] args = { new Integer(15) };  
mf.format(args);
```

15 friends like it

- ✓ Класс Collator - абстрактный
- ✓ Collator getInstance()
- ✓ int compare()
- ✓ RuleBasedCollator

```
List<String> lst = Arrays.asList({"Fluor", "Chlor", "Brom", "Jod"});  
Collator c1 = Collator.getInstance(Locale.EN);  
Collator c2 = Collator.getInstance(new Locale("cz", "CZ"));  
lst.sort(c1);    // Brom, Chlor, Flour, Jod  
lst.sort(c2);    // Brom, Fluor, Chlor, Jod  
// A, Á, B, C, Č, D, Ď, E, É, Ě, F, G, H, Ch, I, Í, J, K, L, M, N,  
// Ň, O, Ó, P, Q, R, Ř, S, Š, T, Ť, U, Ú, Ů, V, W, X, Y, Ý, Z, Ž
```

☑ Класс BreakIterator — поиск границ

☑ Методы

- `getCharacterInstance()`
- `getWordInstance()`
- `getSentenceInstance()`
- `getLineInstance()`
- `int first()`
- `int last()`
- `int next()`
- `int previous()`



УНИВЕРСИТЕТ ИТМО

Программирование. 2 семестр

Новое в Java 9+



☑ Классы

- `public class` - доступен всем
- объединяются в пакеты (доступ внутри пакета)
- упаковываются в JAR-архивы

☑ Пакеты ≈ каталоги

- пакет и JAR - разные сущности
- файл `package-info.java` / `.class`



| |
|----|
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |

☑ classpath

- просмотр всех путей, загрузка первого найденного класса

☑ конфликт версий

- не та версия класса, классы разных версий

☑ кастомные загрузчики классов

☑ нет зависимостей на уровне JVM

☑ JRE - Java Runtime Environment

- rt.jar - содержит всю стандартную библиотеку



| |
|----|
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |

- ✓ Явные зависимости (requires и exports)
- ✓ модульная библиотека
 - базовый модуль java.base
 - платформенные модули (все остальные)
 - новый формат артефактов - jmod
- ✓ jlink - для сборки кастомного образа
 - rt.jar удален в Java 11
- ✓ модульный JAR (+module-info.class)



| |
|----|
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |

✓ Декларация модуля - module-info.java / .class

```
module my.mod {  
    requires java.base;  
    requires java.sql;  
    exports my.package;  
}
```

✓ Опции javac и java

```
javac --module-path mods:libs -d ./out  
  
java --module-path mods:libs --module my.mod
```



- ✓ **module m0** - наш модуль
- ✓ **requires m1** - m0 требует m1 при компиляции и выполнении
- ✓ **requires transitive m2** - m0 требует m2 и модули, зависящие от m0, тоже требуют m2
- ✓ **requires static m3** - m0 требует m3 при компиляции, но не обязательно при выполнении

8

9

10

11

12

13

14

15

16

17

18

- ✓ `module m0` - наш модуль
- ✓ `exports r1` - экспортирует пакет `r1` всем (доступ во время компиляции к **открытым** сущностям пакета)
- ✓ `exports r1 to m1` - доступ только для `m1`
- ✓ `opens r2` - открывает пакет `r2` всем (доступ с помощью рефлексии во время выполнения ко **всем** сущностям)
- ✓ `opens r2 to m2` - открыт только для `m2`

8

9

10

11

12

13

14

15

16

17

18

- ✓ **module** m0 - наш модуль
- ✓ **uses** i1 - использует сервис с интерфейсом i1 (может загружать его с помощью ServiceLoader)
- ✓ **provides** i2 with c1, c2 - предоставляет интерфейс сервиса i2, который реализуют конкретные классы c1 и c2

8

9

10

11

12

13

14

15

16

17

18

☑ Безымянный модуль

- автоэкспорт всех своих пакетов
- имеет доступ ко всем другим модулям
- используется classpath



| |
|----|
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |

☑ Вывод типа

- для локальных переменных

```
var s = "Hello";  
var count = 10;
```

- для λ-выражений (Java 11)

```
(var s) -> s.toUpperCase();
```

| |
|----|
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |

☑ Выражение с выбором

```
int days = switch(month) {  
    case FEBRUARY -> 28;  
    case APRIL, JUNE, SEPTEMBER, NOVEMBER -> 30;  
    default -> 31;  
}
```

8

9

10

11

12

13

14

15

16

17

18

☑ Неизменяемый класс данных

- неявные геттеры
- неявный конструктор
- поля неявно финальные
- класс неявно финальный

```
record Point(int x, int y) { }  
  
var p = new Point(42, 24);  
var x0 = p.x();  
var y0 = p.y();
```

☑ extends java.lang.Record;

☑ Сопоставление с типом

- instanceof

```
List<String> lst = new LinkedList<>();  
if (obj instanceof String str) {  
    lst.add(str.toLowerCase());  
}
```

- switch (Java 16 - preview)

```
String s = switch(obj) {  
    case Person p -> p.getName();  
    case Money m -> m.format("%5.2f");  
    case default -> obj.toString();  
}
```

| |
|----|
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |

☑ До Java 8

- все методы - public abstract

☑ Java 8

- static - не наследуется, вместо классов-утилит (Path/Paths)
- default - реализация в классах по умолчанию

☑ Java 9

- private - вспомогательные методы для static и default

8

9

10

11

12

13

14

15

16

17

18

- ☑ Контроль наследования классов и реализации интерфейсов

```
public abstract sealed class Shape
    permits Square, Circle { ... }

public final class Square extends Shape { ... }

public final class Circle extends Shape { ... }
```

8

9

10

11

12

13

14

15

16

17

18

☑ Интерактивная оболочка REPL (run-eval-print loop)

```
| Welcome to JShell -- Version 11.0.5
| For an introduction type: /help intro

jshell> 2 + 3
$1 ==> 5

jshell> var a = "Hello"
a ==> "Hello"

jshell> System.out.println(a);
Hello

jshell>
```

8

9

10

11

12

13

14

15

16

17

18

- ☑ Компиляция для любой версии с опцией `--release`
 - `javac --release 9`
- ☑ Запуск однофайловых программ без явной компиляции
 - `java Hello.java`

8

9

10

11

12

13

14

15

16

17

18



☑ java.net.http.*

- class HttpClient
- class HttpRequest
- interface HttpResponse<T>
- interface WebSocket

8

9

10

11

12

13

14

15

16

17

18

- ✓ `jwebserver -p 8080`
- ✓ `com.sun.net.httpserver.*`
 - `class SimpleFileServer`
 - ◆ `static createFileServer`
 - ◆ `static createFileHandler`
 - ◆ `static createOutputFilter`



| |
|----|
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |

- ✓ Новое внутреннее представление кодировки Latin-1
 - byte[] вместо char[] - в 2 раза меньше места
- ✓ Java 11 - isBlank(), lines(), repeat(), strip()
- ✓ Java 12 - indent(), transform()
- ✓ Java 15 - блок текста

```
var html = """
    <html>
      <body>
        <p>Hello world</p>
      </body>
    </html>
""";
```

- ✓ ResourceBundle загружается в UTF-8!
- ✓ Java 15 - Unicode 13.0
- ✓ форматтеры:
 - CompactNumberFormat (Java 12)
 - appendDayPeriodText (аналог AM/PM) (Java 16)
 - HexFormat (Java 17)
- ✓ Кодировка по умолчанию - UTF-8 (Java 18)

| |
|----|
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |

- ✓ Set.of(), List.of(), Map.of()
- ✓ Optional.stream()
- ✓ copyOf(Collection) (Java 10)
- ✓ Collectors.toUnmodifiable() (Java 10)
- ✓ Optional.orElseThrow() (Java 10)
- ✓ Collection.toArray() (Java 11)
- ✓ Collectors.teeing() (Java 12)
- ✓ Stream.mapMulti(), Stream.toList() (Java 16)



| |
|----|
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |

✓ Понятное сообщение об ошибке

```
var x = a.b.c.i;
```

```
Exception in thread "main" java.lang.NullPointerException  
    at Prog.main(Prog.java:5)
```

```
Exception in thread "main" java.lang.NullPointerException:  
    Cannot read field "c" because "a.b" is null  
    at Prog.main(Prog.java:5)
```

- ☑ Record patterns (21)
- ☑ Pattern matching for switch (21)
- ☑ Sequenced Collections (21)
- ☑ Virtual Threads (21)
- ☑ Foreign Functions and Memory API
- ☑ String Templates
- ☑ Unnamed Variables
- ☑ Unnamed Classes
- ☑ Vector API
- ☑ JDK 21 - сентябрь 2023
- ☑ Preview
- ☑ Incubator



- ✓ **Record patterns (21)**
- ✓ Pattern matching for switch (21)
- ✓ Sequenced Collections (21)
- ✓ Virtual Threads (21)
- ✓ Foreign Functions and Memory API
- ✓ String Templates
- ✓ Unnamed Variables
- ✓ Unnamed Classes
- ✓ Vector API

```
record Point(int x, int y) {}

if (o instanceof Point p) {
    int x = p.x();
    int y = p.y();
    System.out.print(x+y);
}

if (o instanceof
    Point(int x, int y)) {
    System.out.print(x+y);
}
```

- ✓ Record patterns (21)
- ✓ **Pattern matching for switch (21)**
- ✓ Sequenced Collections (21)
- ✓ Virtual Threads (21)
- ✓ Foreign Functions and Memory API
- ✓ String Templates
- ✓ Unnamed Variables
- ✓ Unnamed Classes
- ✓ Vector API

```
String s = switch(obj) {  
    case Person p -> p.getName();  
    case Money m   -> m.format("%5.2f");  
    case default   -> obj.toString();  
}
```


- ✓ Record patterns (21)
- ✓ Pattern matching for switch (21)
- ✓ **Sequenced Collections (21)**
- ✓ Virtual Threads (21)
- ✓ Foreign Functions and Memory API
- ✓ String Templates
- ✓ Unnamed Variables
- ✓ Unnamed Classes
- ✓ Vector API
- ✓ Почти как Deque
 - add/get/remove: first/last
 - + **reversed()**
- ✓ SequencedCollection
 - List, Deque
 - SequencedSet / **Map**
 - ◆ SortedSet / **Map**
 - ◆ NavigableSet / **Map**
 - ◆ LinkedHashSet / **Map**



- ✓ Record patterns (21)
- ✓ Pattern matching for switch (21)
- ✓ Sequenced Collections (21)
- ✓ **Virtual Threads (21)**
- ✓ Foreign Functions and Memory API
- ✓ String Templates
- ✓ Unnamed Variables
- ✓ Unnamed Classes
- ✓ Vector API
- ✓ Java Thread – OS thread
- ✓ до Java 19 – 1:1
- ✓ "green threads" – M:1
- ✓ Virtual Threads – M:N
 - goroutines (Go)
 - processes (Erlang)

- ✓ Record patterns (21)
- ✓ Pattern matching for switch (21)
- ✓ Sequenced Collections (21)
- ✓ Virtual Threads (21)
- ✓ **Foreign Functions and Memory API**
- ✓ String Templates
- ✓ Unnamed Variables
- ✓ Unnamed Classes
- ✓ Vector API
- ✓ Взаимодействие с кодом и памятью вне JVM
- ✓ выделение памяти
- ✓ управление памятью
- ✓ управление ЖЦ ресурсов
- ✓ вызов внешних функций

- ✓ Record patterns (21)
- ✓ Pattern matching for switch (21)
- ✓ Sequenced Collections (21)
- ✓ Virtual Threads (21)
- ✓ Foreign Functions and Memory API
- ✓ **String Templates**
- ✓ Unnamed Variables
- ✓ Unnamed Classes
- ✓ Vector API

```
x = 2; y = 3;           "2 + 3 = 5"  
  
print(x + " + " + y + " = " + (x+y))  
  
new StringBuilder().append(x)  
    .append("+").append(y)...  
  
format("%d + %d = %d", x,y,x+y)  
  
print(STR."\{x} + \{y} = \{x+y}")  
print(FMT."\{x} + \{y} = %5.2f\{x+y}")  
  
java.lang.StringTemplate     STR.  
java.util.FormatProcessor   FMT.
```

- ✓ Record patterns (21)
- ✓ Pattern matching for switch (21)
- ✓ Sequenced Collections (21)
- ✓ Virtual Threads (21)
- ✓ Foreign Functions and Memory API
- ✓ String Templates
- ✓ **Unnamed Variables**
- ✓ Unnamed Classes
- ✓ Vector API

```
try {
    int i = Integer.parseInt(s);
} catch (NumberFormatException ex) {
    System.out.println("Error!");
}

try {
    int i = Integer.parseInt(s);
} catch (NumberFormatException _) {
    System.out.println("Error!");
}
```

- ✓ Record patterns (21)
- ✓ Pattern matching for switch (21)
- ✓ Sequenced Collections (21)
- ✓ Virtual Threads (21)
- ✓ Foreign Functions and Memory API
- ✓ String Templates
- ✓ Unnamed Variables
- ✓ **Unnamed Classes**
- ✓ Vector API

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

```
void main() {  
    System.out.println("Hello world");  
}
```

- ✓ Record patterns (21)
- ✓ Pattern matching for switch (21)
- ✓ Sequenced Collections (21)
- ✓ Virtual Threads (21)
- ✓ Foreign Functions and Memory API
- ✓ String Templates
- ✓ Unnamed Variables
- ✓ Unnamed Classes
- ✓ **Vector API**
- ✓ Векторные вычисления с поддержкой аппаратного ускорения

```
float[] a = new float[] {0.1F, 0.2F, 0.3F, 0.4F};  
float[] b = new float[] {0.5F, 0.6F, 0.7F, 0.8F};  
  
FloatVector va = fromArray(SPECIES_128, a, 0);  
FloatVector vb = fromArray(SPECIES_128, b, 0);  
  
var result = va.add(vb).div(4F).pow(2F).neg();
```