

I/O в Java

Юрий Литвинов
yurii.litvinov@gmail.com

23.01.2019г

I/O в Java

- ▶ Интерфейсы
 - ▶ InputStream, OutputStream — работа с байтами
 - ▶ Reader, Writer — работа с символами
- ▶ Реализации
 - ▶ File* — работа с файлами
 - ▶ Buffered* — оптимизация с помощью буферизации
 - ▶ Data* — работа с данными
 - ▶ ByteArray* — потоки в оперативной памяти
- ▶ Исключения
 - ▶ IOException — общий предок
 - ▶ FileNotFoundException, EOFException и т.д.
- ▶ Пример:
`new BufferedReader(new FileReader("path/to/file"));`
- ▶ Потоки надо закрывать (close(), try-with-resource)

Пример чтения из файла

```
FileInputStream in = null;
```

```
try {  
    in = new FileInputStream("test.txt");  
    int c;  
  
    while ((c = in.read()) != -1) {  
        doSomething(c);  
    }  
} finally {  
    if (in != null) {  
        in.close();  
    }  
}
```

Или (try-with-resource)

```
try (var in = new FileInputStream("test.txt")) {  
    int c;  
  
    while ((c = in.read()) != -1) {  
        doSomething(c);  
    }  
}
```

- ▶ Работает для классов, реализующих `java.lang.AutoCloseable`

Ещё пример

```
static String readFirstLineFromFile(String path)
    throws IOException {
    try (var br = new BufferedReader(new FileReader(path))) {
        return br.readLine();
    }
}
```

Scanner

```
try (var s = new Scanner(  
    new BufferedReader(new FileReader("test.txt")))) {  
    while (s.hasNextDouble()) {  
        System.out.println(s.nextDouble());  
    }  
}
```

Задача на остаток пары

- ▶ Реализовать консольную программу, работающую в трёх режимах:
 - ▶ архивирует каталог (путь до него – первый аргумент командной строки) с заданным именем (второй аргумент)
 - ▶ разархивирует заданный архив (первый аргумент) в заданное место (второй аргумент)
 - ▶ распечатывает содержимое архива (первый аргумент): имена файлов, размер сжатый/несжатый, метод архивации и т.п.
- ▶ Могут быть полезны классы `ZipFile`, `ZipEntry`, `ZipOutputStream`, `ZipInputStream`