

Робота з файлами

Файли

Файл - це набір байтів, які зберігаються на фізичному носії. Файли мають назву та шлях. Шлях до файлу складається з послідовності всіх директорій, що до нього ведуть.

Абсолютний шлях починається з кореневої директорії на диску: для Windows це, наприклад, “C:\” або “D:”, для Linux/MacOS - просто “/”.

Відносний шлях рахується від поточної директорії. Так, якщо абсолютний шлях до файлу - “C:\Users\Me\My Documents\some-file.txt”, а поточна директорія - “C:\Users”, то відносний шлях до файлу буде “Me\My Documents\some-file.txt”.

Типи файлів

Всі файли - це просто набір байтів. Пам'яті комп'ютера всеодно що всередині: просто текст, чи картинка jpeg або png, чи аудіофайл чи будь що - воно буде зберігатись просто як байти.

З точки зору зберігання файлів в пам'яті - ніяких типів файлів не існує, а розширення файла є просто частиною його назви. Однак розширення важливе для програм які відкривають файли - воно слугує підказкою про те як саме інтерпретувати байти з файлу.

Клас File

Об'єкт класу File – це абстрактне представлення файлу або директорії з файлової системи комп'ютера.

За допомогою цього класу можна створювати, видаляти і перейменовувати файли та директорії, а також отримувати інформацію про файл: права читання/запису/виконання, час створення/останньої зміни файлу і т.д.

Цей клас НЕ ДАЄ можливості змінювати вміст файлу.

Детальніше:

<https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/io/File.html>

```
1 private static void fileShowcase() throws IOException {
2
3     File testRootDir = new File("/home/max/way2it/lesson10_files/files");
4     File childDir = new File(testRootDir, "inner_directory");
5     File childFile = new File(testRootDir, "some_file");
6
7     System.out.println("testRootDir exists: " + testRootDir.exists()); // false при першому запуску
8
9     childDir.mkdirs(); // створить всі необхідні директорії, включаючи inner_directory
10
11     System.out.println("testRootDir is a directory: " + testRootDir.isDirectory()); // true
12
13     if (childFile.createNewFile()) { // створить файл some_file всередині lesson10_files/files/
14         System.out.println("File created: " + childFile.getAbsolutePath());
15         System.out.println("Modified at (ms): " + childFile.lastModified());
16     } else {
17         System.out.println("File not created: " + childFile.getAbsolutePath());
18     }
19
20     System.out.println(Arrays.toString(testRootDir.list())); // виведе [inner_directory, some_file]
21 }
```

```
1  FileOutputStream fos1 = new FileOutputStream("/home/max/way2it/lesson10_files/file1");
2
4  String s = "1234567890";
5  byte[] byteString = s.getBytes();
6  fos1.write(byteString);
7
8  byte[] bytes = new byte[4];
9  int a = 1_234_567_890; // 01001001 | 10010110 | 00000010 | 11010010
10 bytes[3] = (byte) a; // 11010010
11 a = a >> 8; // 00000000 | 01001001 | 10010110 | 00000010
12 bytes[2] = (byte) a; // 00000010
13 a = a >> 8; // 00000000 | 00000000 | 01001001 | 10010110
14 bytes[1] = (byte) a; // 10010110
15 a = a >> 8; // 00000000 | 00000000 | 00000000 | 01001001
16 bytes[0] = (byte) a; // 01001001
17
18 FileOutputStream fos2 = new FileOutputStream("/home/max/way2it/lesson10_files/file2");
19 fos2.write(bytes);
20
21 fos1.close();
22 fos2.close();
```

Запис тексту у файл

```
1 private static void printStreamShowcase() throws IOException {
2
3
4     // створення потоку запису у файл
5     // якщо файлу нема - він створиться, якщо немає директорії files - програма викине Exception
6     PrintStream ps = new PrintStream("/home/max/way2it/lesson10_files/files/text_file");
7
8     // запис у файл інформації в ТЕКСТОВОМУ форматі
9     ps.println("Hello, the long number is: ");
10    ps.println(122234534534567890L);
11
12    // закриття потоку виводу
13    ps.close();
14 }
```

Запис байтів у файл

```
1 private static void fileOutputStreamShowcase() throws IOException {  
2     // створення простого потоку виводу байтів у файл  
4     FileOutputStream fos = new  
5     FileOutputStream("/home/max/way2it/lesson10_files/files/bytes_file");  
6  
7     // створення допоміжного об'єкту, що вміє конвертувати різні типи в масиви байтів при записі  
8     DataOutputStream dos = new DataOutputStream(fos);  
9  
10    // запис рядка в кодуванні UTF  
11    dos.writeUTF("Hello, the long number is: ");  
12  
13    // запис байтів числа  
14    dos.writeLong(122234534534567890L);  
15  
16    dos.close();  
17    fos.close();  
18 }
```


Читання тексту з файлу

```
1 private static void scannerShowcase() throws IOException {
2     File file = new File("/home/max/way2it/lesson10_files/files/text_file");
3
4     // створення об'єкту Scanner, що буде читати з файлу
5     Scanner sc = new Scanner(file);
6
7     // читання тексту
8     String text = sc.nextLine();
9
10    // читання тексту і конвертація його в long
11    long number = sc.nextLong();
12
13    System.out.println(text + number); // Hello, the long number is: 122234534534567890
14
15    // закриття потоку
16    sc.close();
17 }
```

Читання байтів з файлу

```
1 private static void fileInputStreamShowcase() throws IOException {  
2     // створення простого потоку вводу байтів з файлу  
4     FileInputStream fis = new FileInputStream("/home/max/way2it/lesson10_files/files/bytes_file");  
5  
6     // створення допоміжного об'єкту, що вміє конвертувати прочитані байти в типи даних  
7     DataInputStream dis = new DataInputStream(fis);  
8  
9     // зчитування байтів і перетворення їх у рядок  
10    String text = dis.readUTF();  
11  
12    // зчитування 8 байтів і перетворення їх на long  
13    long number = dis.readLong();  
14  
15    System.out.println(text + number);  
16  
17    dis.close();  
18    fis.close();  
19 }
```