**Как создать новый файл в Java?**

Сегодня мы рассмотрим простейший способ создания файла в Java, узнаем какие исключения могут нас подстерегать при работе с файлами и рассмотрим разные способы создания файла.

Для создания нового файла в Java чаще всего используется класс java.io.File. Во время инициализации объекта File мы должны предоставить ему имя файла, а затем создать сам файл вызовом метода createNewFile(). Этот метод возвращает булево true**,**если новый файл создан успешно, и false, если файл уже существует. Этот метод также бросает java.io.IOException, когда он не может создать файл.

**Немного теории по созданию файла в Java**

Создать файл в Java можно одним из трех способов, передав в объект File:

* абсолютный путь
* только указать имя файла
* указать относительный путь (в этом случае объект файла пытается найти файлы в корневой директории проекта)

При создании пути к файлу мы должны использовать свойство системы File.separator**,**чтобы сделать нашу программу платформонезависимой.

Давайте рассмотрим различные способы создания файла в простой Java программе:

|  |
| --- |
| package ua.com.prologistic;    import java.io.File;  import java.io.IOException;    public class CreateNewFile {        //В этом классе представлены способы создания файла в Java      public static void main(String[] args) throws IOException {          // получаем разделитель пути в текущей операционной системе          String fileSeparator = System.getProperty("file.separator");            //создаем абсолютный путь к файлу          String absoluteFilePath = fileSeparator + "Users" + fileSeparator + "prologistic" + fileSeparator + "file.txt";            File file = new File(absoluteFilePath);          if(file.createNewFile()){              System.out.println(absoluteFilePath + " Файл создан");          } else {               System.out.println("Файл " + absoluteFilePath + " уже существует");          //создаем файл только с указанием имени файла          file = new File("file.txt");          if(file.createNewFile()){              System.out.println("file.txt файл создан в корневой директории проекта");          }else System.out.println("file.txt файл уже существует в корневой директории проекта");            //создаем файл с указанием относительного пути к файлу          String relativePath = "tmp" + fileSeparator + "file.txt";          file = new File(relativePath);          if(file.createNewFile()){              System.out.println(relativePath + " файл создан в корневой директории проекта");          }else System.out.println("Файл " + relativePath + " уже существует в директории проекта");      }    } |

Когда мы запустим эту программу первый раз, то увидим следующий результат:

|  |
| --- |
| /Users/prologistic/file.txt Файл создан  file.txt файл создан в корневой директории проекта  Exception in thread "main"  java.io.IOException: No such file or directory      at java.io.UnixFileSystem.createFileExclusively(Native Method)      at java.io.File.createNewFile(File.java:947)      at ua.com.prologistic.CreateNewFile.main(CreateNewFile.java:32) |

В том месте, где мы использовали относительный путь, программа бросает IOException, поскольку **tmp** каталога просто нет в корневой папке проекта. Поэтому метод createNewFile() просто пытается создать файл в корневом каталоге проекта.

Теперь давайте создадим каталог tmp в корне проекта и снова выполнил программу. Вот результат:

|  |
| --- |
| Файл /Users/prologistic/file.txt уже существует  Файл file.txt уже существует в директории проекта  tmp/file.txt файл создан в корневой директории проекта |

Здесь мы видим, что **файл с указанием полного пути** уже существует (был создан, когда мы первый раз запускали программу) и не был создан;

**файл с указанием только названия** уже существует и не был создан;

**файл с указанием относительного пути** был успешно создан (в первом запуске этого файла не было, так как папки tmp не существовало).