Практическая работа №2.4.

1. Работа с файловой системой из Java. Создание, удаление, переименование файлов. Чтение и запись в файл.

1.1. Задача:Изучение теоретического материала. Закрепление полученных знаний путем разработки приложения на языке программирования Java в среде разработки eclipse.

1.2.Тип занятий: исследовательский.

1.3.Содержание: теоретический материал, задания для самостоятельного выполнения, задания со звездочкой.

2.Теоретический материал к уроку

2.1. Создание файла и папки

Создание файла в папке проекта

//Определяем файл

File file = new File(fileName);

//проверяем, что если файл не существует, то создаем его в каталоге нашего проекта

if(!file.exists()){

file.createNewFile();

}

Пример:

File file = new File("sample1.txt");

//проверяем, что если файл не существует, то создаем его в каталоге нашего проекта

if(!file.exists()){

file.createNewFile();

}

Создание папки

//Определяем папку

File dir = new File(путь и имя папки);

//проверяем, что если папка не существует, то создаем ее

if(!dir.exists()){

dir.mkdir();

}

Пример:

File dir =new File("C://SomeDir");

//проверяем, что если папка не существует, то создаем ее

if(!dir.exists()){

dir.mkdir();

}

Создание файла внутри папки

File file2 = new File(путь или имя объекта папки, имя файла);

//проверяем, что если файл не существует, то создаем его созданной папке

if(!file2.exists()){

file2.createNewFile();

}

Пример:

File file2 = new File(dir, "sample2.txt");

//проверяем, что если файл не существует, то создаем его созданной папке

if(!file2.exists()){

file2.createNewFile();

}

2.2. Переименование файла

//Определяем новый файл

File file3 = new File(fileName);

//переименовываем

file.renameTo(file3);

Пример:

File file3 = new File("sample3.txt");

file.renameTo(file3);

2.3. Удаление файла из папки

  //удаление с использованием полного пути в файлу

        File file = new File("/Users/prologistic/file.txt");

        if(file.delete()){

            System.out.println("/Users/prologistic/file.txt файл удален");

        }else System.out.println("Файла /Users/prologistic/file.txt не обнаружено");

        //удаление с использованием только имени файла

        file = new File("file.txt");

        if(file.delete()){

            System.out.println("file.txt файл был удален с корневой папки проекта");

        }else System.out.println("Файл file.txt не был найден в корневой папке проекта");

        //удаление файла с использованием относительного пути к файлу

        file = new File("tmp/file.txt");

        if(file.delete()){

            System.out.println("tmp/file.txt файл был удален с корневой папки проекта");

        }else System.out.println("Файл tmp/file.txt не был найден в корневой папке проекта");

        //удаляем пустую папку

        file = new File("tmp");

        if(file.delete()){

            System.out.println("tmp папка была удалена с корневой директории проекта");

        }else System.out.println("tmp папка не была найдена в корневой директории проекта");

        //пробуем удалить папку с файлами

        file = new File("/Users/prologistic/project");

        if(file.delete()){

            System.out.println("/Users/prologistic/project была найдена c корневой директории проекта");

        }else System.out.println("/Users/prologistic/project не была найдена в корневой директории проекта или не является пустой");

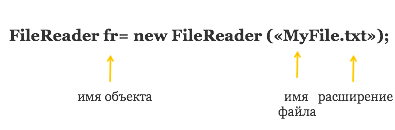
2.4. Чтение и запись в файла с помощью FileWriter и FileReader

**1. Объявление**

Перед тем, как вызывать какие-нибудь методы для работы с файлами, нужно объявить FileWriter/FileReader:



или

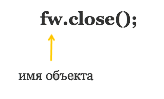


**2. Нужно закрыть поток**

FileWriter/FileReader - это потоки, их нужно не только «открыть» (то-есть объявить), но и **«закрыть»**. Представьте, что Вы открыли кран. Нельзя же уйти из дому, оставив воду литься?

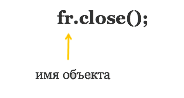
Это **правило** работает и для других потоков - кроме стандартных System.in и System.out.

Закрыть поток можно с помощью **.close()**:



или

|  |  |
| --- | --- |
|  | FileWriter fw = new FileWriter( "sample1.txt" );          fw.close();            FileReader fr = new FileReader( "sample2.txt" );          fr.close();   1. Команда записи в файл: **write();** Используется до закрытия потока.   Пример: fw.write(“привет”);   1. Команда добавления символа в документ: **append();** Используется до закрытия потока.   Пример: fw.append(‘.’);   1. Для чтения из файла используется класс Scanner и построчное считывание в цикле:   Пример:  Scanner scan = new Scanner(fr);  int i = 1;  while (scan.hasNextLine()) { //пока есть следующая строка  System.out.println(scan.nextLine());             i++;  } |



2.5. Чтение и запись в файл с помощью FileInputStream и FileOutputStream

1) Запись с преобразованием строки в байты:

String text = "Hello world!"; // строка для записи

FileOutputStream fos=new FileOutputStream("[C://SomeDir//notes.txt](about:blank)");

// перевод строки в байты

byte[] buffer = text.getBytes();

fos.write(buffer, 0, buffer.length);

fos.close();

2) Посимвольное чтение:

FileInputStream fin=new FileInputStream("[C://SomeDir//notes.txt](about:blank)");

            int i=-1;

            while((i=fin.read())!=-1){

                System.out.print((char)i);

            }

        }

2.6. Используемые библиотеки:

* java.io.File - позволяет получить информацию о файле.
* java.io.FileReader - для чтения текстовых файлов
* java.io.FileWriter - для записи текстовых файлов
* java.io.IOException – обработка ошибок
* java.io.FileInputStream, java.io.FileOutputStream - предназначены прежде всего для записи двоичных файлов, то есть для записи и чтения байтов. И хотя они также могут использоваться для работы с текстовыми файлами, но все же для этой задачи больше подходят другие классы.

3. Задания

3.1. уровень 1

1. Откройте eclipse. Командой File – new – Java Project создайте новый проект, в нем создайте пакет и класс с методом main.
2. Подключите следующие библиотеки:

**import** java.io.File;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.util.Scanner;

1. Добавьте в метод main обработку исключений для работы с текстовыми файлами: **throws** IOException
2. Напишите команды для создания текстового файла внутри вашего каталога проекта с именем 1.txt, проверив сначала не существует ли он уже.
3. Напишите команды для создания каталога (папки) MyDir на диске C из команды для создания текстового файла внутри каталога MyDir с названием 2.txt, проверив сначала не существуют ли они уже.
4. Запустите программу на выполнение, проверьте создались ли файлы открыв папку вашего проекта и папку на диске C.
5. Если файлы создались, переименуйте с помощью JAVA файл 1.txt в myfile.txt.
6. Создайте объект класса FileWriter для вашего файла.
7. Запишите в ваш файл произвольную строку.
8. Добавьте отдельный символ в ваш файл. Не забудьте закрыть поток!!!
9. Запустите программу на выполнение, проверьте появилась ли строка в вашем файле.
10. Создайте объект класса FileReader для вашего файла.
11. Считайте посимвольно текст из вашего файла с выводом на экран консоли. Не забудьте закрыть поток!!!
12. Запустите удаление файла с несуществующим путем, например:

File file = **new** File("/Users/prologistic/file.txt");

**if**(file.delete()){

System.***out***.println("/Users/prologistic/file.txt файл удален");

}**else** System.***out***.println("Файла /Users/prologistic/file.txt не обнаружено");

1. Теперь пробуем удалить нашу папку с диска C:

//пробуем удалить папку с файлами

1. Должно выйти сообщение, что папка не пустая, нужно сначала удалить
2. наш файл 2.txt://удаление файла с использованием относительного пути к файлу

file = **new** File("c://MyDir/Hello2.txt");

**if**(file.delete()){

System.***out***.println("c://MyDir/Hello2.txt файл был удален");

}**else** System.***out***.println("Файл c://MyDir/Hello2.txt не был найден");

1. Теперь нужно еще раз попробовать удалить нашу папку и самостоятельно по примеру напишите удаление вашего файла из папки проекта.
2. \*Задания со звездочкой добавьте чтение из файла и запись в файл, подключив библиотеки:

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

и используя FileOutputStream и FileInputStream.