**Как дописать файл в Java?**

**Добавляем в конец, обновляем файл**

Сегодня мы будет файлы дописывать или обновлять. Мы можем добавлять данные в существующий файл (обновлять содержимое файла), используя FileWriter, BufferedWriter или FileOutputStream.

**Как дописать файл в Java? Теория**

* Если вы работаете с **небольшими объемами текстовой информации**, используйтеFileWriter с флагом true в конструкторе (подробнее смотрите ниже в пункте**Практика**).
* Если **количество операций с файлами очень большое**, то желательно использоватьBufferedWriter.
* Чтобы **дописать в файл бинарные данные или данные с открытого потока**, нужно использовать предназначенный для этого FileOutputStream.

**Как обновить файл. Практика**

Ниже представлен пример программы, показывающий в работе каждый из рассмотренных способов добавить в файл (обновить файл) на Java:

package ua.com.prologistic;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileOutputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.io.OutputStream;

public class UpdateFile {

    public static void main(String[] args) {

        String filePath = "/Users/prologistic/update.txt";

        String appendText = "Этой строкой мы будем обновлять существующий файл";

        // обновляем файл с помощью FileWriter

        appendUsingFileWriter(filePath, appendText);

        // обновляем файл с помощью BufferedWriter

        appendUsingBufferedWriter(filePath, appendText, 100);

        // обновляем файл с помощью OutputStream

        appendUsingOutputStream(filePath, appendText);

    }

    // обновляем файл с помощью FileWriter

    private static void appendUsingOutputStream(String fileName, String data) {

        OutputStream os = null;

        try {

            //в конструкторе FileOutputStream используем флаг true, который обозначает обновление содержимого файла

            os = new FileOutputStream(new File(fileName), true);

            os.write(data.getBytes(), 0, data.length());

        } catch (IOException e) {

            e.printStackTrace();

        }finally{

            try {

                os.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

        }

    }

    // обновляем файл с помощью BufferedWriter

    private static void appendUsingBufferedWriter(String filePath, String text, int noOfLines) {

        File file = new File(filePath);

        FileWriter fr = null;

        BufferedWriter br = null;

        try {

            //для обновления файла нужно инициализировать FileWriter с помощью этого конструктора

            fr = new FileWriter(file,true);

            br = new BufferedWriter(fr);

            for(int i = 0; i < noOfLines; i++){

                br.newLine();

                //теперь мы можем использовать метод write или метод append

                br.write(text);

            }

        } catch (IOException e) {

            e.printStackTrace();

        } finally{

            try {

                br.close();

                fr.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

        }

    }

    // добавляем информацию в файл с помощью FileWriter

    private static void appendUsingFileWriter(String filePath, String text) {

        File file = new File(filePath);

        FileWriter fr = null;

        try {

            fr = new FileWriter(file,true);

            fr.write(text);

        } catch (IOException e) {

            e.printStackTrace();

        }finally{

            try {

                fr.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

        }

    }

}

Используйте нужный вам метод в зависимости от особенностей вашего проекта.