Работа с файлами. Основные классы.

import java.io.\*;

**Файловые объекты:**

File

Байтовый ввод/вывод:

InputStream

FileInputStream

DataInputStream

OutputStream

FileOutputStream

DataOutputStream

Символьный ввод/вывод:

Reader

Whiter

**Об исключениях:**

! Используется семейство исключений IOException

! Любой метод работы с файлами может выбросить исключение и требует, чтобы оно было обработано, например, закрыть файл при аварии:

try{

//некоторые подготовительные действия, указатели на потоки, файлы

try{

//открытие/закрытие файла на чтение/запись, чтение и запись

//ошибки чтения/записи желательно обработать без исп.исключений

например, если файл не удалось открыть, ни читать\не писать в него.

}catch(IOException){//действия при ошибке чтения, непредусмотренной

}finally{например, мы должны закрыть файл в любом случае}

}catch(IOException e){обработка исключений} // здесь файл закрывать нельзя – при ошибке будет исключение, которое обработать уже некому.

! можно поймать исключение и заменить его своим, более конкретным.

**Класс File**

File file = new File("filename"); //по умолчанию в корневом каталоге проекта

getAbsoluteFile(); //полный путь + имя файла

FileInputStream

FileInputStream in=null; - создание ссылки вне try и её зануление

in=new FileInputStream(file) – (внутри try) открыть файл, вернёт указатель in.close() – (внутри finally, можно ещё внутри try) закрыть файл, после закрытия файла попытка чтения вернёт исключения

int read() – читаем 1 байт

int read(b) – чтение в byte[] b до конца массива или файла

int read(b,i,len) – заполнение части массива, чтение максимум len байт

byte[] in.readAllBytes()–чтение всего потока ввода, сохранение в виде массива

byte[] in.readNBytes(n)-чтение n байт из потока, сохранение в виде массива.

FileOutputStream

FileOutputStream out=null;// здесь и далее все аналогично FileInputStream.

out=new FileOutputStream(file);

out.close();

out.write(a) – запись в файл, где а это int или массив byte[]

out.white(a,i,n) – запись части массива, i – индекс, n-число байт

out.write(s1.getBytes(), 0, s1.length());//пишем s1, строку String

out.flush()-очищает все буферы вывода

DataOutputStream, DataInputStream:

in=new DataOutputStream(FileInputStream(file)); - создание О

Класс DataOutputStream позволяет писать элементарные данные в поток через интерфейс DataOutput, который определяет методы, преобразующие элементарные значения в форму последовательности байтов.

writeBoolean(boolean v) : записывает в поток булевое однобайтовое значение

writeDouble(double v): записывает в поток 8-байтовое значение double

writeInt(int v): записывает в поток целочисленное значение int

writeUTF(String str): записывает в поток строку в кодировке UTF-8

+byte,short,char,float,long

Класс DataInputStream действует противоположным образом.

В методах выше меняем write на read,например, readShort() возвращает short

int skipBytes(int n): пропускает при чтении из потока n байтов