**FILE**

public static void main(String[] args) **throws Exception**{ … }

**File … = new File(“directory or file”)** – creating file or directory

**… .isDirectory()** – checking if it is directory

**… .isFile()** – checking if it is file

**… .exists()** – checking if it exists

**… .mkdir()** – create directory

**… .createNewFile()** – creating new file in your directory

**FileWriter** filewrite **= new FileWriter(**file**);** - creating variable to write into the filefilewrite**.write("**parasha1 \n **");** - writing textfilewrite**.flush();** - pushing new text into the filefilewrite**.close();** - closing file **FileReader** fileread **= new FileReader(**file**);** - creating variable to read file **char[]** ch **= new char[…];** fileread**.read(ch);** - copying message from file to the variable  **BufferedWriter** bufwr **= new BufferedWriter(**filewrite**); -** creating clever variable to write into the filebufwr**.write("**jopka**");** - writing into the variablebufwr**.newLine();** - make new linebufwr**.flush();** - push the variable into the filebufwr**.close();** - close file **BufferedReader** bufrd **= new BufferedReader(**fileread**);** - creating clever reader  **while (**bufrd**.ready()){ System.*out*.println(**bufrd**.readLine());**  } – if file is ready read a line

**Serializing (recording objects)**

You have to add **implements Serializable** to the class if you want to be able to save it to the file

**public class … implements Serializable { … }**

Creating object *car* with parameters **FileOutputStream** *fileOutputStream* **= new FileOutputStream("**first/temp**");** -getting location **ObjectOutputStream** *objectOutputStream* **= new ObjectOutputStream(***fileOutputStream***);** - taking location to the variable of class, which can record objects

*objectOutputStream***.writeObject(***car***);** - recording object*objectOutputStream***.close();** - closing file **FileInputStream** *fileInputStream* **= new FileInputStream("**first/temp**");** - getting directory to read **ObjectInputStream** *objectInputStream* **= new ObjectInputStream(***fileInputStream***);** - taking location to the variable of class, which can read objects

*creating object … to record similar object from the file* **= (***object***)***objectInputStream***.readObject();***objectInputStream***.close();**

**mostly used because of easy to record**

this class records objects in the file, which is very easy to change, also you can easily read them

**Properties** def **= new Properties();** creating default (works like main) **def.setProperty("**key**","**value**");** - set value with key **Properties** properties **= new Properties(**def**);** - create variable with default values (works like map)properties**.setProperty("**key**","**value**");  
FileOutputStream** outputStream **= new FileOutputStream("**…**");** - new variable, where you can record object, it will record with name … **properties.store(**outputStream**,"comments");** - record object **FileInputStream** inputStream **= new FileInputStream("**…**");** - read file …properties**.load(**inputStream**);** - load file into properties properties**.getProperty("**key**");** - get value