

## Дәріс-6. Енгізу-шығару ағындары

### Жоспар:

1. File класы
2. Енгізу-шығару ағындары
3. Алдын ала анықталған ағындар

### File класы

Java қосымшаларында файлдармен жұмыс жасау үшін java.io пакетінің кластары қолданылады.

File класы каталогтар мен файл аттарын объект ретінде сақтап, өңдеуге мүмкіндік береді.

File класының тәсілдері файлдың жасалған уақыты, файлға қол жеткізу (доступ), жаңадан файл ашу, атын өзгерту, файлды жабу, өшіру, файл маршрутын анықтауға мүмкіндік береді.

### File класының тәсілдері

Жаңадан файл ашу – **File fp=new File("demo.txt");**

Файл атын анықтау - **fp.getName();**

Файл маршрутын анықтау - **fp.getPath();**

Файлға баратын абсолютты жолды анықтау - **fp.getAbsolutePath();**

Файл мөлшерін анықтау - **fp.length();**

Файлдың соңғы өңделген уақытын анықтау - **fp.lastModified();**

Файлды оқу үшін ашу - **fp.canRead();**

Файлды жазу үшін ашу - **fp.canWrite();**

Файлды өшіру - **fp.delete();**

Файлды жабу - **fp.close();**

Осыларды қолданып мысал келтірейік. **import java.io.\*;**

**import java.util.\*; public class FileTest {**

```
public static void main(String[] args) throws IOException /*отказ от
обработки исключения в main()*/ {
```

```
    PrintWriter pw=new PrintWriter(new OutputStreamWriter(System.out,
    "Cp866"),true);
```

```
    File fp=new File("C:\\demo.txt"); if(fp.isFile()){//если объект дисковый
    файл
```

```
        pw.println("Имя файла:\t" + fp.getName()); pw.println("Путь к файлу:\t" +
    fp.getPath()); pw.println("Абсолютный путь: \t" + fp.getAbsolutePath());
    pw.println("Размер файла:\t" + fp.length());
```

```
        pw.println("Последняя модификация файла:\t" + fp.lastModified());
    pw.println("Файл доступен для чтения:\t" + fp.canRead()); pw.println("Файл
    доступен для записи:\t" + fp.canWrite()); pw.println("Файл удален:\t" +
    fp.delete()); }
```

```

    if(fp.createNewFile()){
        pw.println("Файл " + fp.getName() + " создан");
    }
    if(fp.exists()){
        pw.println("файл "+ fp.getName() + " существует");
    }
    else

        pw.println("файл " + fp.getName() + " не существует"); //в объект типа File
помещается каталог\директория
        File dir = new File("f:\\myworks\\умқjava");

        if (dir.isDirectory())/*если объект объявлен как каталог на диске*/
pw.println("Директория!");
        if(dir.exists()){//если каталог существует pw.println("Dir " + dir.getName()
+ " существует"); File [] files = dir.listFiles();
        pw.println("");
        for(int i=0; i < files.length; i++){
            Date date = new Date(files[i].lastModified()); pw.println(files[i].getPath() + "
\t| " + files[i].length() + "\t| " + date.toString();//toLocaleString();//toGMTString())
            } } } }

```

### Енгізу-шығару ағындары

Байт тізбектерін енгізу ағындары абстрактты **InputStream** классының ішкі классы, байт тізбектерін шығару ағындары абстрактты **OutputStream** классының ішкі классы болып саналады.

Файлдармен жұмыс жасағанда осы кластардың ағындарды ашып, оны сәйкес файлдармен байланыстыратын **FileInputStream** және **FileOutputStream** ішкі класстарының құрастырушылары қолданылады. Байты немесе байттар массивін оқу үшін **FileInputStream** класының **read()/read(byte[] b)** тәсілдері қолданылады.

```

import java.io.*; public class ReadBytes {
    public static void main(String[] args){ int b;
    try {
        PrintWriter pw=new PrintWriter(new OutputStreamWriter(System.out,
"Ср866"), true); File f = new File("ReadBytes.java");
        FileInputStream is = new FileInputStream(f); while ((b = is.read()) != -1){

            //прочитанные данные выводятся на консоль pw.println("прочитан байт
= " + b);

        }
        is.close(); //заккрытие потока ввода
    }
}

```

```
catch (IOException e){ System.out.println("ошибка файла: " + e);  
} } }
```

Байтты немесе байт массивін файлға жазу үшін шығару ағыны – `FileOutputStream` классының `write()` тәсілі қолданылады.

```
import java.io.*;  
public class WriteBytes {  
    public static void main(String[] args){ int pArray[] = {1, 2, 3, 5, 7, 11, 13,  
17}; try {  
        FileOutputStream os = new FileOutputStream("bytewrite.dat" ); for (int i = 0;  
i < pArray.length; i++)  
            os.write(pArray[i]);  
            os.close();  
        }  
        catch (IOException e) { System.out.println("ошибка файла: " + e); } } }
```

Енгізу-шығару ағындарын сүзбеден өткізуге арналған **`DataInputStream`** классының тәсілдері енгізілген мәндерді сәйкес базалық типтерге түрлендіреді:

**`readBoolean()`, `readByte()`, `readChar()`, `readInt()`, `readLong()`, `readFloat()`, `readDouble()`**

`DataOutputStream` классының тәсілдері базалық типтегі мәндерді шығару ағындарына түрлендіруге мүмкіндік береді: **`writeBoolean()`, `writeChar()`, `writeInt()`, `writeLong()`, `writeFloat()`, `writeLine()`.**

#### **Бақылау сұрақтары:**

1. `File` классы қандай мүмкіндіктер береді?
2. `File` классының тәсілдері
3. Байт немесе байттар массивін оқу үшін `FileInputStream` қандай тәсілдері қолданылады.
4. Байтты немесе байт массивін файлға жазу үшін шығару ағыны – `FileOutputStream` классының қандай тәсілі қолданылады?
5. Енгізу-шығару ағындарын сүзбеден өткізуге арналған `DataInputStream` классының тәсілдері енгізілген мәндерді сәйкес қандай базалық типтерге түрлендіреді?