

# Praktikum Programski Sistemi UI sistem



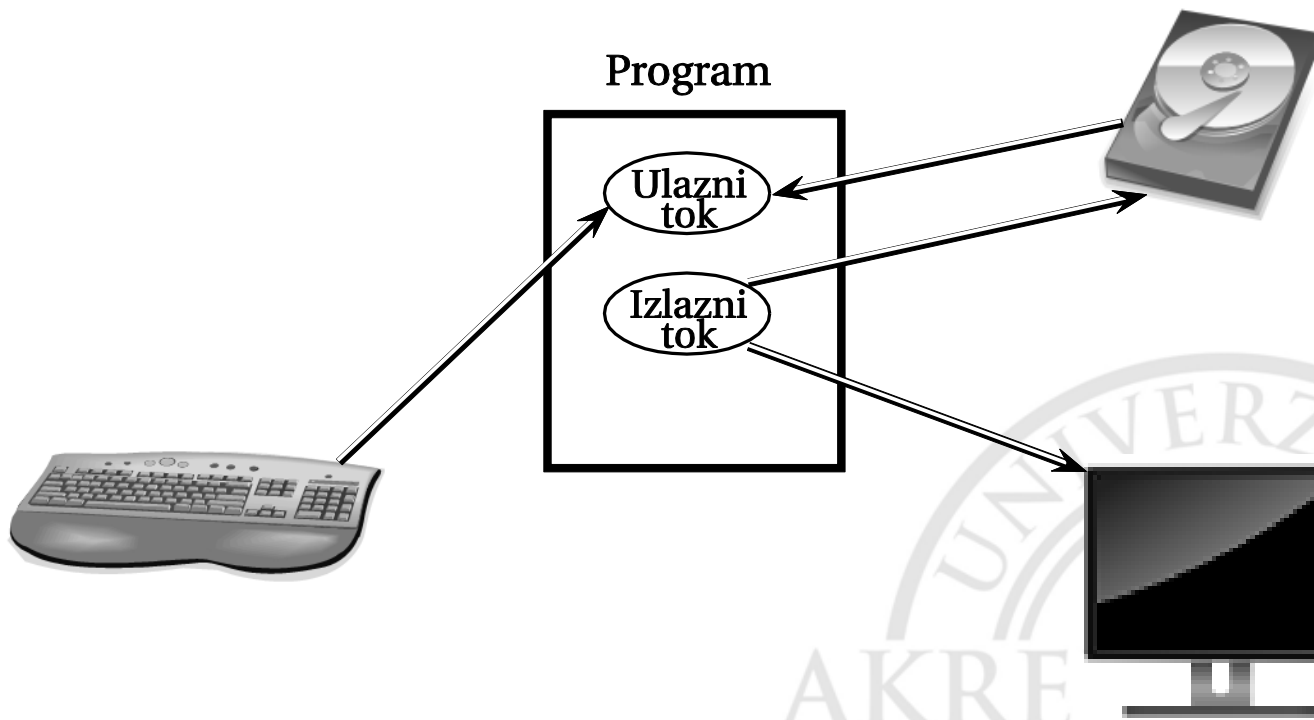
# Programski ulaz/izlaz

- **Ulaz**: dobijanje (čitanje) podataka iz okruženja
- **Izlaz**: predavanje (pisanje) podataka u okruženje
- Osnovne U/I klase u paketima: java.io, java.nio, java.nio.channels
- Ulaz/izlaz se jednoobrazno predstavlja tokovima podataka

# Tokovi podataka

- Tok (*stream*) podataka: jednosmerni i sekvencijalni protok bajtova u program ili iz njega
- Izlaz programa: podaci se u programu upisuju u **izlazni tok**
- Ulaz programa: podaci se u programu čitaju iz **ulaznog toka**

# Tokovi podataka

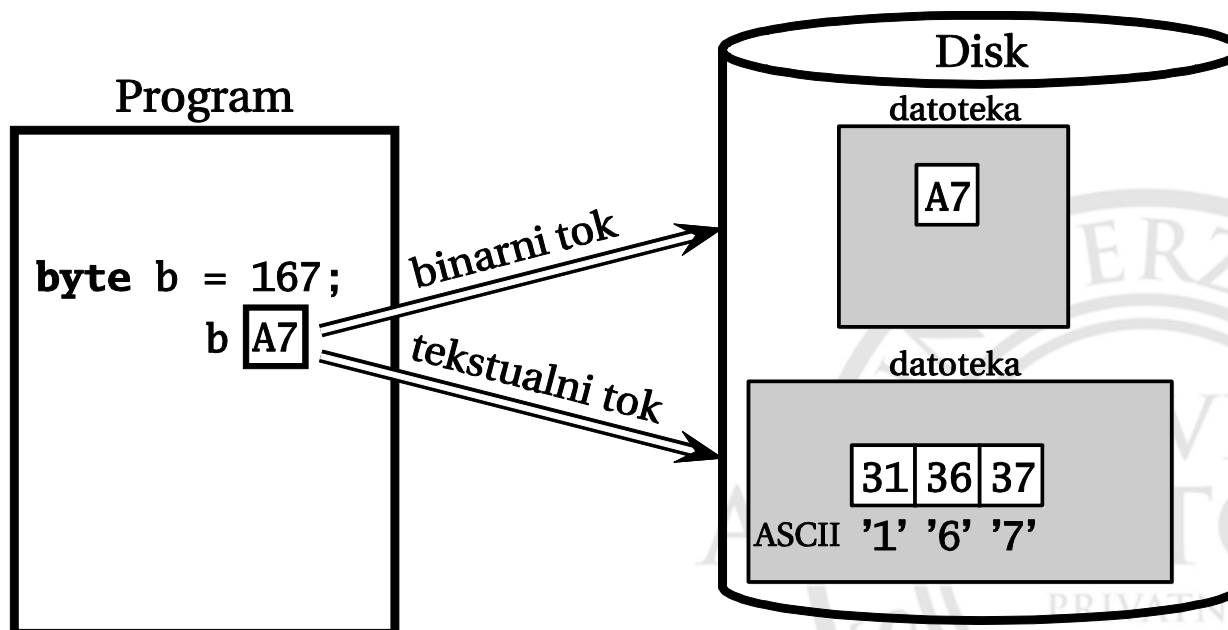


# Binarni i tekstualni tokovi

- **Binarni podaci:** podaci predstavljeni na isti način na koji se zapisuju unutar računara
- **Tekstualni podaci:** podaci predstavljeni nizovima znakova
- **Binarni i tekstualni:** U/I tokovi podataka za prenos binarnih i tekstualnih podataka
- Tekstualni tokovi obezbeđuju automatsko konvertovanje bajtova prilikom ulaza/izlaza

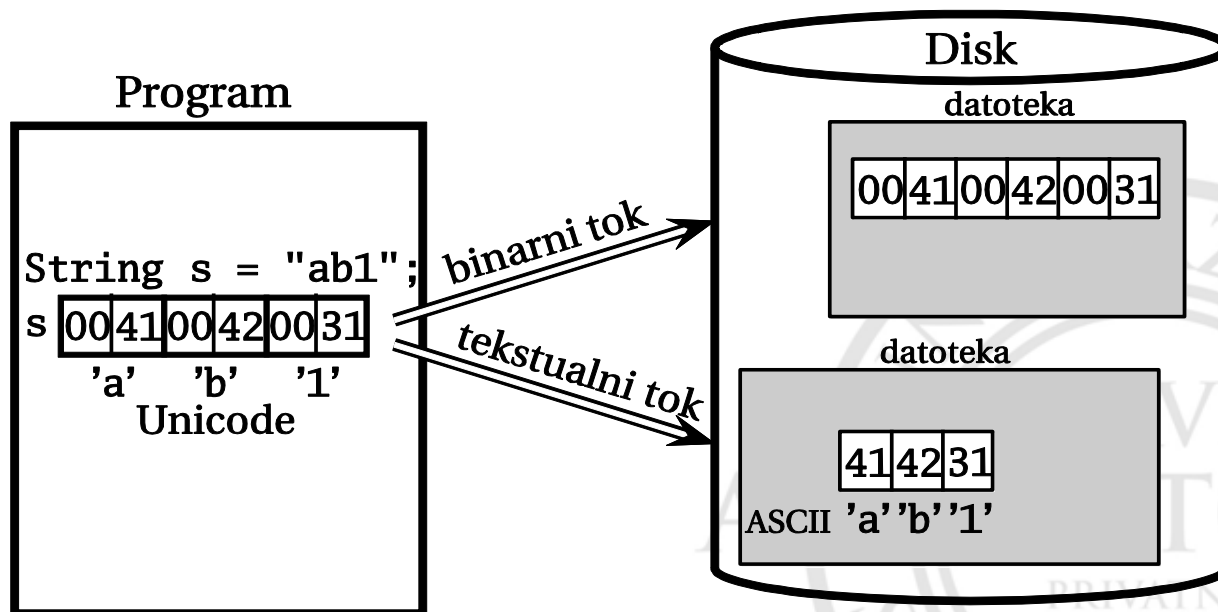
# Binarni i tekstualni tokovi

- Primer: upisivanje broja 167 u datoteku



# Binarni i tekstualni tokovi

- Primer: upisivanje stringa "ab1" u datoteku



# Hijerarhija klasa za ulaz/izlaz

- Četiri apstraktne klase na vrhu
  - **InputStream** za ulazne binarne tokove
  - **OutputStream** za izlazne binarne tokove
  - **Reader** za ulazne tekstualne tokove
  - **Writer** za izlazne tekstualne tokove
- Primitivne U/I operacije: čitanje i pisanje jednog bajta (znaka)

```
public int read() throws IOException
```

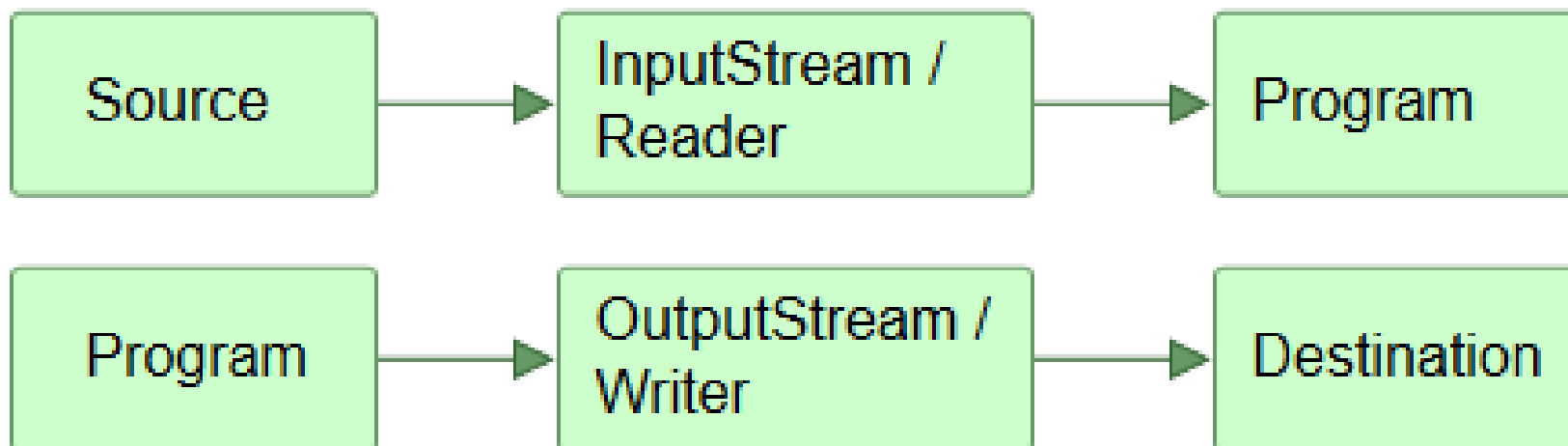
```
public void write(int b) throws IOException
```



# Hijerarhija klasa za ulaz/izlaz

- Klase koje nasleđuju osnovne U/I klase proširuju osnovne mogućnosti rada sa tokovima podataka
- Čitanje i pisanje binarnih podataka
  - **DataInputStream** i **DataOutputStream**
- Čitanje i pisanje tekstualnih podataka
  - **Scanner** i **PrintWriter**

# Tokovi podataka



# Svrha U/I klasa

- Rad sa fajlovima
- Pristup mreži
- Pristup internim memorijskim baferima
- Komunikacija između niti (pipe)
- Baferisanje
- Filtriranje
- Parsiranje
- Čitanje i upis teksta, primitivnih podataka
- Serijalizacija

# Pregled U/I klasa

	Byte Based		Character Based	
	Input	Output	Input	Output
Basic	InputStream	OutputStream	Reader InputStreamReader	Writer OutputStreamWriter
Arrays	ByteArrayInputStream	ByteArrayOutputStream	CharArrayReader	CharArrayWriter
Files	FileInputStream RandomAccessFile	FileOutputStream RandomAccessFile	FileReader	FileWriter
Pipes	PipedInputStream	PipedOutputStream	PipedReader	PipedWriter
Buffering	BufferedInputStream	BufferedOutputStream	BufferedReader	BufferedWriter
Filtering	FilterInputStream	FilterOutputStream	FilterReader	FilterWriter
Parsing	PushbackInputStream StreamTokenizer		PushbackReader LineNumberReader	
Strings			StringReader	StringWriter
Data	DataInputStream	DataOutputStream		
Data - Formatted		PrintStream		PrintWriter
Objects	ObjectInputStream	ObjectOutputStream		
Utilities	SequenceInputStream			

# Hijerarhija klasa za ulaz/izlaz

- Klase Scanner/PrintWriter i DataInputStream/DataOutputStream omogućavaju jednostavan ulaz/izlaz za vrednosti primitivnih tipova podataka
- Čitanje i pisanje objekata?
- Serijalizacija i deserijalizacija objekata: predstavljanje objektima vrednostima primitivnih tipova radi ulaza/izlaza
- Za kompleksne objekte nije jednostavan zadatak (primer: dugme tipa JButton)

# Hijerarhija klasa za ulaz/izlaz

- Automatska serijalizacija i deserijalizacija objekata
  - **ObjectInputStream** i **ObjectOutputStream**
  - `readObject()` i `writeObject(Object o)`
- Samo za objekte klasa koje implementiraju specijalni interfejs **Serializable**

```
public class NekaKlasa implements Serializable {  
    . . .  
}
```

# Hijerarhija klasa za ulaz/izlaz

- Interfejs Serializable ne sadrži nijedan metod
- Interfejs Serializable služi samo kao marker da kompajler za objekte implementirajuće klase doda ugrađeni mehanizam za automatsku serijalizaciju i deserijalizaciju
- ObjectInputStream i ObjectOutputStream predstavljaju binarne tokove
- Tekstualni tokovi (XML format)

# Datoteke

- **Datoteka**: grupa podataka kao celina na spoljašnjem uređaju računara (disk, USB fleš, DVD, ...)
- Sistem datoteka organizovan po direktorijumima (folderima) koji sadrže datoteke, ali i druge direktorijume
- Binarni ulaz/izlaz: **FileInputStream** i **FileOutputStream**
- Tekstualni ulaz/izlaz: **FileReader** i **FileWriter**



# Datoteke

- Primer:

```
FileReader ulaz; // ulazni tok

// konstruisanje ulaznog toka za datoteku ulaz.txt
try {
    ulaz = new FileReader("ulaz.txt");
}
catch (FileNotFoundException e) {
    // Rukovanje izuzetkom
    . . .
}
```

# Datoteke

- Primer:

```
Scanner ulaz; // ulazni tok

// Konstruisanje omotača ulaznog toka za datoteku
// ulaz.txt
try {
    ulaz = new Scanner(new FileReader("ulaz.txt"));
}
catch (FileNotFoundException e) {
    // Rukovanje izuzetkom
    . . .
}
```

# Datoteke

- Primer:

```
PrintWriter izlaz; // izlazni tok

// Konstruisanje omotača izlaznog toka za datoteku
// izlaz.txt
try {
    izlaz = new PrintWriter(new
        FileWriter("izlaz.txt"));
}
catch (IOException e) {
    // Rukovanje izuzetkom
    . . .
}
```

# Imena datoteka

- **Ime** i **sadržaj** datoteke su dva odvojena pojma
- Sadržaj datoteke čine podaci koji se mogu čitati i upisivati U/I operacijama
- Ime datoteke služi za manipulisanje datotekom u sistemu datoteka na disku
- Apsolutno ime: C:\tmp\java\podaci.txt
- Relativno ime: java\podaci.txt

# Imena datoteka

- Klasa **File** predstavlja ime datoteke (direktorijuma)

`File` datoteka;

- `datoteka.exists()`
- `datoteka.isDirectory()`
- `datoteka.delete()`
- `datoteka.list()`
- `datoteka.mkdir()`

