

# Uvod u objektno orijentirano programiranje

## Iznimke i I/O

# Iznimke

Definicija: Iznimna situacija koja narušava normalni tijek izvršavanja programa

Kako se nositi s mogućim iznimkama?

Try - catch → Slijedi primjer

# I/O

I/O= Input/Output

- Input: tipkovnica ili datoteka
- Output: konzola ili datoteka

Za danas...

- Tekstualne datoteke
- Kreiranje datoteka i direktorija
- Spremanje datoteka
- Čitanje datoteka

## Stream

- Objekt koji ili dohvaća podatke s izvora (tipkovnica, datoteka,...) ili ih dostavlja na željenu lokaciju (konzola, datoteka,...)
- Output: display ili datoteka
- Input stream (`System.in` koji spaja program na tipkovnicu)
- Output stream (`System.out` koji spaja program na konzolu (ekran))

# Međupohrana (Buffering)

Bez međupohrane:

- Svaki byte se čita/piše sa diska čim je dostupan
- Veliki overhead, malo čekanje za svaki byte

Sa međupohranom:

- Čitanje i pisanje u komadićima
- Manji overhead, duže čekanje za neke byte-ove

# Tekstualne vs binarne datoteke

## Tekstualne datoteke

Bitovi reprezentiraju karaktere koji se mogu "printati"

Jedan byte za ASCII karakter

Svaka datoteke stvorena s

"tekstualnim editorom"

## Binarne datoteke

Bitovi reprezentiraju kodirane informacije kao što su instrukcije programa

Može ih čitati kompjuter ali ne i čovjek

Ne mogu se "printati"

\*Pod printanje misli se na to da su razumljive kada ih se prikaže

# Tekstualne datoteke

Klase bitne za tekst output:

- `PrintWriter`
- `FileOutputStream` (ili `FileWriter`)

Klase bitne za tekst input:

- `BufferedReader`
- `FileReader`

- `FileOutputStream` i `FileReader` kao argumente primaju ime datoteke
- `PrintWriter` i `BufferedReader` imaju korisne metode za lakše čitanje i pisanje
- Najčešće se koriste kombinacije dvije klase
- `import java.io.*`; nova knjižica `import java.nio.*`

# Primjer

Iz datoteke čitamo liniju po liniju. U novu datoteku zapisujemo pročitano u obliku /BrLinije/ProcLinija

Npr:

Ulaz:

Danas je lijep dan.

Vani pada kiša.

Mogu puno učiti.

Izlaz:

/1/Danas je lijep dan.

/2/Vani pada kiša.

/3/Mogu puno učiti.



# Iznimke

U predhodnom primjeru, čim pokušamo stvoriti čitač datoteke dobivamo grešku u IDE kako nismo ispravno baratali(handle) s mogućom iznimkom. Moramo riješiti zadatak na način da uhvatimo grešku odmah kod događaja (Try-Catch) ili ju propagiramo dalje(Throws exception kod metode).

# String, StringBuffer i StringBuilder

String je objekt koji kad ga stvorimo se ne mijenja(immutable). Ono što se dešava svaki put kad ga mi mijenjamo je to da stari String pokupi GarbageCollector a stvara se novi String.

To stvara puno nepotrebnog 'smeća' i zato postoji mogućnost korištenja StringBuffer-a i StringBuildera ako očekujemo promjene u našim stringovima.

Ako nemamo višedretveni sustav i ne moramo brinuti o sigurnosti naših dretva, koristimo String Builder zato što je brži