

# Java

## Ponovitev

---

DIRI 2003 – Programski jeziki

# Robert

---

- Ko Robert nadzira pisni izpit, mu postane dolgčas in se začne igrati naslednjo igrico. Najprej na tablo napise niza "a" in "b":
- "a", "b",
- Nato pa pripisuje nove nize, pri čemer dobi vsak naslednji niz tako, da stakne skupaj prejšnja dva. Od tod dobi zaporedje nizov
- "a", "b", "ab", "bab", "abbab", "bababbab", "abbabbababbab", ...
- Napisi metodo `robert`, ki sprejme pozitivno celo število `n` in vrne `n`-ti niz v zgornjem zaporedju. Na primer, če poženemo ukaz
- `String s = robert(6);`
- potem je niz `s` enak "bababbab".

Matija Lokar,  
Fakulteta za matematiko in fiziko

DIRI 2003

## Robert

---

- Dve spremenljivki tipa `String`
  - Prvi niz, drugi niz
- Sestavimo novi niz
  - `novi = prvi + drugi;`
- Preložimo nize
  - `prvi = drugi;`
  - `drugi = novi;`
- Ustrezno mnogokrat ponovimo
  - Zanka `while`
- Pozor na `n = 1, 2!`

Program

## Najmanjši na začetek

---

- Napiši metodo `najmanjsi_na_zacetek`, ki sprejme tabelo celih števil in v njej zamenja ničti in najmanjši element tabele. Na primer, ko požnemo
  - `int[] b = {10, 4, 7, 18, 2, 7};`
  - `najmanjsi_na_zacetek(b);`
- je tabela `b` enaka `{2, 4, 7, 18, 10, 7}`.

## Najmanjši na začetek

- Tip metode: void
  - Kar počnemo s tabelami, ki so parametri, SE POZNA na tabeli!
- Poiščemo najmanjšega
  - Zapomnimo si tako najmanjšega, kot njegov indeks
  - Kandidat za najmanjšega je 0-ti
    - `int kje = 0;`
    - `int min = t[0];`
  - V zanki pregledamo vse preostale
    - Če je tekoči element manjši od trenutno najmanjšega, si njegovo vrednost in njegov indeks zapomnimo
      - `if (t[i] < min) {`
      - `min = t[i];`
      - `kje = i;`
      - `}`
- Zamenjamo 0-tega in najmanjšega
  - Potrebujemo indeks!
  - `t[kje] = t[0];`
  - `t[0] = min;`

[programi](#)

## Prva številka

- Napiši program `PrvaStevka.java`, ki z ukazne vrstice sprejme pozitivno celo število in izpiše njegovo prvo številko. Na primer:
  - `> java PrvaStevka 675`
  - `6`
  - `> java PrvaStevka 000031`
  - `3`
  - `> java PrvaStevka 1`
  - `1`

## Prva številka

---

- Argument iz ukazne vrstice
  - `arg[0]`
  - Pretvorba z `Integer.parseInt()`
- 3224
  - Znebimo se 4
  - 322
  - Znebimo se 2, ...
- "Režemo" zadnje številke
  - `n = n / 10;`
- Ponavljamo, dokler ne bo število enomestno!

program

## Prva številka

---

- Rekurzivna različica
- Ideja:
  - Prva številka 231545 je enaka prvi številki 23154 (prvotno število brez enice!)
  - Prva številka 23154 je enaka prvi številki 2315
  - Prva številka 2315 je enaka prvi številki 231
  - ...
- Metoda!
- Ustavitveni pogoj
  - Število je enomestno
- Sicer:
  - Klic te metode na številu brez zadnje številke

program

## Vzorec

- Napiši program v Javi, ki iz ukazne vrstice sprejme pozitivno celo število  $n$  in izpiše na zaslon naslednji vzorec (prikazan je primer  $n = 5$ ):

```
.....#####  
.....#####  
.....#####  
.....#####  
.....#####  
#####.....  
#####.....  
#####.....  
#####.....  
#####.....
```

## Vzorec

- Dva dela, naredimo vsakega posebej:

```
.....#####  
.....#####  
.....#####  
.....#####  
.....#####  
  
#####.....  
#####.....  
#####.....  
#####.....  
#####.....
```

## Vzorec

- Za

```
.....#####
.....#####
.....#####
.....#####
.....#####
```
- n vrstic: .....#####
- Vsaka vrstica: n x . in n x #. program
  - `int i = 1;`
  - `while (i <= n) {`
  - `System.out.print(".");`
  - `i++;`
  - `}`
- Zanka v zanki
  - Pravzaprav dve zanki v zanki
  - Prva gre po vrsticah
  - Druga izpiše ustrezno število . (in potem še ena #)

## Naloge 1. 2002

- Izbrali bomo nekaj nalog s sistemom ma strani
  - <http://www.educa.fmf.uni-lj.si/www375/2002/prj/prj-obv.htm>
  - in jih skupaj rešili!
- Teh 300 nalog vsebuje le stvari, ki jih že znamo!