

ALGORITMI IN PODATKOVNE STRUKTURE 1



1. laboratorijske vaje

Rekurzija (lažje naloge)

NALOGA 1

Napišite rekurzivno funkcijo `public static boolean jePalindrom(String word)`, ki preveri, ali je vhodna beseda `word` palindrom.

- Klic `jePalindrom("rotator")` vrne vrednost `true`.
- Klic `jePalindrom("abcda")` vrne vrednost `false`.

Ideja: za palindrom velja, da ima prvi znak enak zadnjemu. Če palindromu odrežemo prvi in zadnji znak, ponovno dobimo palindrom.

Uporabne metode razreda `String`:

- Dolžino niza vrne metoda `length()`.
- Posamezen znak niza izluščimo z metodo `charAt()`.
- Podniz izluščimo z metodo `substring()`.

NALOGA 2



Napišite rekurzivno funkcijo `public static int vsota(int[] array, int index)`, ki vrne vsoto elementov polja `array`, če ga opazujemo od pozicije `index` naprej.

- Za niz `int[] data = {11, 7, 10, 10}` klic `vsota(data, 0)` vrne vrednost 38.

Ideja: vsota celotnega niza je enaka vsoti vrednosti prvega elementa in vsoti preostalih elementov.

NALOGA 3

Napišite rekurzivno funkcijo `public static boolean jePotenca(int a, int b)`, ki ali je število `a` potenca števila `b`.

- Klic `jePotenca(16, 2)` vrne vrednost `true`.
- Klic `jePotenca(5, 2)` vrne vrednost `false`.

Ideja: če je število `a` potenca števila `b`, potem je tudi število `a/b` potenca števila `b`.

Operacije nad celimi števili:

- `a / b` vrne celi del kvocienta.
- `a % b` vrne ostanek pri deljenju.

NALOGA 4

Napišite rekurzivno funkcijo `public static void izpisiBinarno(int n)`, ki vhodno število n izpiše v binarni obliki.

- Klic `izpisiBinarno(0)` izpiše 0.
- Klic `izpisiBinarno(4)` izpiše 100.
- Klic `izpisiBinarno(19)` izpiše 10011.

Ideja: najprej binarno izpišemo vrednost $n/2$, nato na konec dopišemo $n\%2$.

Operacije nad celimi števili:

- a / b vrne celi del kvocienta.
- $a \% b$ vrne ostanek pri deljenju.