

Naloga 1

```
public class Potenca {
    public static int potencia(int osnova, int eksponent) {
        if (eksponent == 0) {
            return 1;
        }
        else {
            int t = potencia(osnova, eksponent/2);
            if (eksponent % 2 == 0) {
                return t * t;
            }
            else {
                return t * t * osnova;
            }
        }
    }
}
```

Formatted: Font: (Default)
Times New Roman, 12 pt

Formatted: Default, Indent:
Left: 0 cm, Space After: 0 pt

Naloga 2 [12 + 13 točk]

(a) Kaj dejansko program izpiše na ekran? Odgovor nariši čitljivo in jasno označi presledke (označeni so z ~)

```
/....\
~/... \
~~/. . \
~~~/ . \
```

(b) Popravi program tako, da bo pravilno delal.

```
public class Vzorec {
    public static void main(String[] args) {
        final int N = 5;
        for (int i = 0; i < N; i = i + 1) {
            for (int j = 0; j < i; j = j + 1) {
                System.out.print("~");
            }
            System.out.print("\n");
            for (int j = 0; j < 2 * (N - i); j = j + 1) {
                System.out.print(".");
            }
            System.out.println("/");
        }
    }
}
```

Deleted: 4

Deleted: /

Deleted: -

Deleted: \n

Naloga 3 [25 točk]

Napiši *program* `PrvaStevka.java`, ki z *ukazne vrstice* sprejme *pozitivno celo število* in izpiše njegovo *prvo števko* (cifro). Na primer:

```
> java PrvaStevka 675
6
> java PrvaStevka 000031
3
> java PrvaStevka 1
1
```

[Nasvet: niz pretvorimo v celo število z metodo `int Integer.parseInt(String s)` .]

```
public class PrvaStevka{
    public static void main(String[] a) {
        int n = Integer.parseInt(a[0]);
        // podatek dobimo ob klicu iz ukazne vrstice
        while (n > 10) { // dokler ne odstranimo vseh stevk razen prve!
            n = n / 10;
        }
        System.out.println(n); // v n je ostala le se prva stevka
    }
}
```

in še rekurzivna različica

```
public static int prvaStevka(int n)
{ // z rekurzijo ugotovimo prvo stevko števila
  // predpostavka: parameter je pozitiven!
  if (n < 10)
    return n; // enomestno število
  return prvaStevka(n / 10); // rekurzivni klic brez zadnje številke!
}
```

Naloga 4 [30 točk]

Napiši *metodo* `najmanjsi_na_zacetek`, ki sprejme tabelo celih števil in v njej zamenja *ničti* in *najmanjši* element tabele. Na primer, ko požanemo

```
int[] b = {10, 4, 7, 18, 2, 7};
najmanjsi_na_zacetek(b);
```

je tabela `b` enaka `{2, 4, 7, 18, 10, 7}`.

```
public static void najmanjsi_na_zacetek(int[] t)
{
    int dol = t.length;
    if (dol != 0) { // ce so nam slucajno dali prazno tabelo
        // poisci minimum
        int kje = 0;
        int min = t[0];
        int i = 1;
        while (i < dol) {
            if (t[i] < min) {
                // nasli smo boljsega kandidata za najman.
                min = t[i];
                kje = i;
            }
            i++;
        }
        // zamenjava
        t[kje] = t[0];
        t[0] = min;
    }
}
```