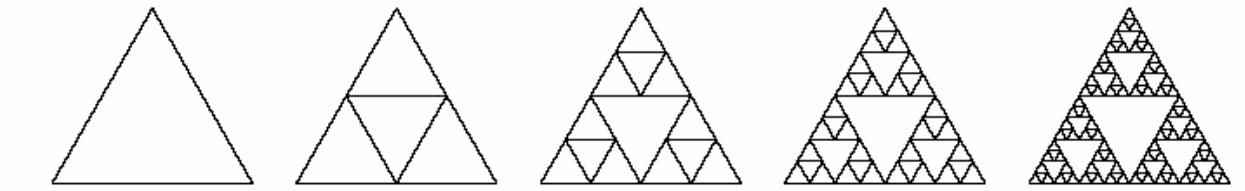
Programski jeziki - JAVA

Ime in priimek: _____

[25 T] Dan je razred `LogoZelva` z objektnimi metodami `public void fd(int d)`, `public void bk(int d)`, `public void right(int kot)` in `public void left(int kot)`. Prva premakne želvo naprej za d enot (pri tem se seveda nariše črta dolžine d), druga jo premakne nazaj za d enot (tudi tu nastane črta dolžine d). Smer želve se pri obeh ukazih ne spremeni. Metoda right spremi smer želve za dani parameter stopinj v desno, metoda left pa obrne smer želve v levo. Pri the dveh metodah se položaj želve ne spremeni, pa tudi ne nariše se nič.

Sestavi statično metodo `trikotnik(zelva, n, d)`, ki kot parametre dobi `LogoZelvo zelva` in naravni števili `n` in `d` ter nariše sliko, kot je prikazano na spodnjih slikah. Prvi parameter je objekt tipa `LogoZelva`, drugi parameter določa red slike (spodnje slike so redov 0, 1, 2, 3 in 4) in tretji dolžino stranice trikotnika.

Namig: preprosta rešitev je rekurzivna (premisli, kako sliko reda *n* dobiš iz slik nižjega reda).



```
public static void trikotnik(LogoZelva z, int n, int d)
{ // narise sliko stopnje n. Po koncu risanja je zelva v prvotnem polozaju
      if (n == 0)
      {
            z.fd(d); z.left(120);
            z.fd(d); z.left(120);
            z.fd(d); z.left(120);
      }
      else
      {   // trije liki st. n - 1
         trikotnik(z, n - 1, d / 2); // spodaj levo
         z.fd(d / 2);
         trikotnik(z, n - 1, d / 2); // spodaj desno
         z.bk(d / 2); // spet v osnovni poziciji
         z.left(60); // proti zgornjemu trikotniku);
         z.fd(d / 2);
         z.right(60);
         trikotnik(z, n - 1, d / 2); // zgornji
         z.left(60);
         z.bk(d / 2);
         z.right(60); // spet v osnovni poziciji in smeri
      }
}
```

[25 T] Napiši *metodo* rotiraj, ki sprejme tabelo nizov in vrne *novo* tabelo nizov, v kateri je prvi element podane tabele prestavljen na zadnje mesto, ostali elementi pa se premaknejo za ena. Na primer, če poženemo

```
String[] t = { "jabolko", "banana", "stiropor", "grozdje" };
String[] s = rotiraj(t);
```

mora biti vrednost tabele s enaka {"banana", "stiropor", "grozdje", "jabolko"}

Ideja:

Najprej bomo naredili novo tabelo, ki bo take velikosti, kot je podana tabela. Nato bomo v zanki preložili i-ti element (i bo šel od 1 do dol_tabele – 1) prvotne tabele v i – 1 element nove tabele. Na koncu bomo kot zadnji element nove tabele dali še 0-tega prvotne tabele.

```
public static String[] rotiraj(String[] tabela) {
      int dolzina = tabela.length;
      String[] nova = new String[dolzina];
      int i = 1;
      while (i < dolzina) {
            nova[i - 1] = tabela[i]; // rotiramo za mesto v desno
            i = i + 1;
      }
      nova[dolzina - 1] = tabela[0]; // prvi postane zadnji
      return nova;
}
```

[25 T] Ravnatelj ves besen pokliče vas, sicer učitelja, a silom prilik tudi vzdrževalca računalniške opreme, k sebi. Ves besen vam kaže na zaslon računalnika, kjer je dopis, ki je resnici na ljubo res čuden. Npr. namesto „šola“ piše „šoollllaaa“ in podobno. Vzrok temu je okvarjena tipkovnica, ki je poljubno ponavljala pritisnjeni znak. Seveda ste za okvaro tipkovnice krivi vi. Ker vam grozi, da boste ob bajno stimulacijo v znesku 1876 SIT, se ponudite, da boste napisali program, ki bo tekst spravil v red. Ena, dba tr in že je tu program v Javi, ki bo opravil le potrebno. Le metoda `brezPonovitev`, ki sprejme niz `a` in vrne nov niz, ki ga dobimo tako, da zaporedne ponovitve znakov v `a` zamenjamo z eno samo kopijo. Na primer, ko pokličemo `brezPonovitev` na nizu

`"Raaaavnnattellj jje nnnnnnajbooooooljšššši nnaaaa sssvetu"`

dobimo niz

`"Ravnatelj je najboljši na svetu"`

Ideja:

Najprej naredimo novo prazen niz

```
String novi = "";
```

Potrebovali bomo indeks, ki bo povedal kje smo v starem nizu in zadnji znak prepisan v novi niz. Prvi zanka niza enostavno prepišemo in si kot zadnji znak zapomnimo prepisani znak. Nato pregledamo preostale znake niza. Če je tekoči niz `(a.charAT(kjeStari())` različen od zadnjega prepisanega (), ga dodamo v novi niz in si ga zapomnimo kot zadnji prepisani znak, drugače ne.

```
public static String brezPonovitev(String stari) {
  // iz niza stari odtrani odvečne pomnožene znake
        int dolzina = stari.length();
        String novi = "";
        char zadnjiPrepisani;
        // prepisemo prvega
        novi = novi + stari.charAt(0);
        zadnjiPrepisani = stari.charAt(0);
        // pregledamo preostale
        int kjeStari = 1;
        while (kjeStari < dolzina) {
               char znak = stari.charAt(kjeStari); // trenutni znak
                if (znak != zadnjiPrepisani) {
                    novi = novi + znak; // prepišemo
                    zadnjiPrepisani = znak;
                }
                kjeStari = kjeStari + 1;
        }
        return novi;
}
```

[25 T] Ljubljanski pekovski ceh je imel predpisano težo hlebcev kruha. Predpis so strogo nadzorovali: vsako leto se je cehovski mojster nenapovedano pojavil v pekarni in stehtal vse hlebce, ki so jih imeli na policah. Če je več kot 15% hlebcev za več kot 20% odstopalo od predpisane teže, so peka namočili v Ljubljanico. Na voljo je program, ki prebere predpisano težo, število hlebcev in težo vsakega hlebca v pekarni ter ugotovi, ali se bo pek namakal ali ne. Žal pa je pekovska miš nagrizla izpis programa ravno na ključnem mestu. Dopolni ta del programa!

```
public class PekovskiCeh {
	public static void main(String[] a) {
		int steviloHlebcev;
		int steviloSlabih = 0;
		int predpisanaTeza;
		int dovoljenOdstopHlebca = 20; // v odstotkih
		int kolikoLahkoOdstopa = 15;   // v odstotkih

		// preberemo osnovne podatke

		String beri = JOptionPane.showInputDialog("Predpisana teza: ");
		predpisanaTeza = Integer.parseInt(beri);
		beri = JOptionPane.showInputDialog("Stevilo hlebcev: ");
		steviloHlebcev = Integer.parseInt(beri);

		// preverimo vse hlebce

		int kateri = 1;
		int tezaHlebca;
		int odstopOdPredpisaneTeze;

		// SLEDOVI MIŠJIH ZOB
		// PAZI NA POIMENOVANJE SPREMENLJIVK



		while (kateri <= steviloHlebcev) {
			beri = JOptionPane.showInputDialog("Teza hlebca: ");
			tezaHlebca = Integer.parseInt(beri);
			odstopOdPredpisaneTeze =
			  Math.abs(predpisanaTeza - tezaHlebca) * 100 / predpisanaTeza;
			if (odstopOdPredpisaneTeze > dovoljenOdstopHlebca)
			{
				steviloSlabih = steviloSlabih + 1;
			}
			kateri = kateri + 1;
		}






		// TU JE PAPIR SPET CEL

		int odstopOdStevila;
		odstopOdStevila = steviloSlabih * 100 / steviloHlebcev;

		if (odstopOdStevila > kolikoLahkoOdstopa)
		{
			System.out.println("V vodo z goljufom! ");
		}
		else
		{
			System.out.println("Tokrat naj mu bo! Se naprej bo smrdel!");
		}
	}
}
```