



Polsemestrálny test praktická časť



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	<input type="text"/>
--------------------	--------------	----------------------

Doplňujúce zdrojové kódy sú na stránke predmetu PAZ1b.

- (5b) Uvažujme triedu SpajanyZoznam z prednášky o spájaných zoznamoch. Do triedy SpajanyZoznam pridajte metódu equals, ktorá vráti, či zoznam referencovaný parametrom zoznam obsahuje rovnakú postupnosť hodnôt ako zoznam, nad ktorým túto metódu voláme.

```
public boolean equals(SpajanyZoznam zoznam)
```

- (5b) Do triedy Osoba z prednášky o stromoch pridajte metódu pocetJedinacikov, ktorá vráti počet všetkých takých potomkov osoby, ktorí sú jedináčikovia (t.j. nemajú iných súrodencov).

```
public int pocetJedinacikov()
```

- (5b) **Uvádzačov algoritmus:** Na ples bolo pozvaných mnoho hostí. Podmienkou pre účasť na bále je, že žiaden pozvaný hosť nesmie prísť s partnerom. Úlohou uvádzača je popárovat' prichádzajúcich mužov a ženy. Aby nenastal zmatok, uvádzač si vymyslel takúto stratégiu: Po príchode dostane každý účastník bálu poradové číslo (1, 2, 3, ...) a bude uvedený do salónika. V prípade, že príde na bál dáma a v salóniku má najvyššie poradové číslo nejaký pán, dáma nepôjde do salónika - títo účastníci hneď vytvoria pár a uvádzač ich oboch uvedie do sály. To isté platí, keď príde pán a najvyššie číslo v salóniku má nejaká dáma. Vytvorte metódu paruuj, ktorá vráti všetky páry, ktoré takýmto párovaním vznikli. Pozor, týmto algoritmom sa nemusí podariť nájsť každému hosťovi partnera a teda vo vrátenom Map-e sa nemusia nachádzať všetky osoby, ktoré prišli na bál.

Zdrojový kód k 3. úlohe

```
import java.util.*;

public class Bal {

    public static class Osoba {
        String meno;
        boolean muz;

        public Osoba(String meno, boolean muz) {
            this.meno = meno;
            this.muz = muz;
        }

        @Override
        public String toString() {
            return meno;
        }
    }

    public static Map<Osoba, Osoba> paruj(List<Osoba>prichadzajuci) {
        return null;
    }

    public static void main(String[] args) {
        List<Osoba> prichadzajuci = new ArrayList<Osoba>();

        prichadzajuci.add(new Osoba("Janka", false));
        prichadzajuci.add(new Osoba("Jozef", true));
        prichadzajuci.add(new Osoba("Jan", true));
        prichadzajuci.add(new Osoba("Jozef", true));
        prichadzajuci.add(new Osoba("Anna", false));

        System.out.println(Bal.paruj(prichadzajuci));
    }
}
```