

2. pisni izpit pri Programiranje 2 (Praktična matematika)

Čas reševanja pisnega izpita je **120 minut**. Pisni izpit lahko rešujete na fakultetnih računalnikih ali lastnih prenosnikih. Reševanje izven učilnice ni dovoljeno! Dovoljena je uporaba poljubnega gradiva, literature, spletnih virov ter zapiskov in rešitev iz predavanj ali vaj. **Prepovedana je kakršnakoli komunikacija v realnem času!**

Pisni izpit je sestavljen iz **štirih enakovrednih nalog**. Prvi dve nalogi zahtevata rešitev v **programskem jeziku Python 3**, zadnji dve nalogi pa zahtevata rešitev v **programskem jeziku Java 8**. Sestavljeni programi ne smejo uporabljati modulov, ki niso skladni s programskim jezikom Python 3, oziroma knjižnic, ki niso del programskega jezika Java 8.

Vso **gradivo potrebno za reševanje nalog** je dostopno na [spletni učilnici](#). Sestavljene **programe oddate na spletni učilnici** kot je zavedeno v razdelkih **Kaj in kako oddam?**. Pazite, da se natančno držite navodil oddaje!

1. Evklidov algoritem (≈ 5 vrstic Python kode)

V programskega jezika Python 3 sestavite program za **izračun največjega skupnega delitelja dveh števil** z uporabo Evklidovega algoritma. Naj bosta a in b naravni števili in naj velja $a > b$. Evklidov algoritem temelji na opazki, da je največji skupni delitelj števil a in b enak največjemu skupnemu delitelju števil $a - b$ in b .

Sestavite funkcijo `gcd(a, b)` za izračun največjega skupnega delitelja kot sledi.

- Če sta števili a in b enaki, potem naj funkcija vrne a ali b .
- Če sta števili a in b različni, potem večje izmed števil npr. a zamenjajte z razliko $a - b$.

Funkcija `gcd(a, b)` naj ponavlja 2. korak, dokler ne velja pogoj iz 1. koraka. Primer uporabe sestavljenega programa je prikazan spodaj.

```
>>> gcd(14, 63)
7
>>> gcd(48, 128)
16
>>> gcd(64, 107)
1
```

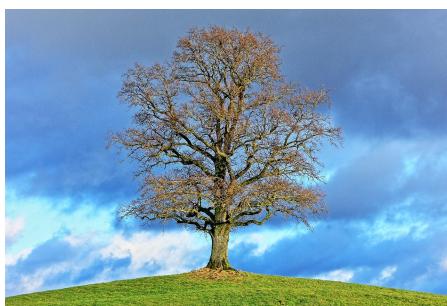
Python

Kaj in kako oddam?

Sestavljen program shranite v **datoteko** `euclid.py`, ki jo **oddate na** [spletni učilnici](#). Ne pričakuje se, da je programska koda opremljena s komentarji. Pazite pa, da program ne vsebuje napak, kar pomeni, da se ukaz `python euclid.py` uspešno izvede!

2. Zlivanje slik drevesa (≈ 15 vrstic Python kode)

Na naslovu <https://lovro.fri.uni-lj.si/pro2/exams/trees.zip> je dostopna mapa z **dvema rastrskima slikama** v formatu PNG, ki **prikazujeta isto drevo** v različnih letnih časih (glej prvi dve slike spodaj). Obe slike sta enake velikosti in približno poravnani. Mapo s slikama si najprej prenesite na lasten računalnik. Nato v programskem jeziku Python 3 sestavite **program**, ki **zlije slike dreves** v eno sliko. Pričakovani izgled ustvarjene "zlite" slike je prikazan spodaj desno.



"Zlita" slika naj bo razdeljena na **tri enako velike dele**, ki si sledijo od leve proti desni (tj. leva, srednja in desna tretjina). Leva tretjina naj bo enaka prvi sliki. Desna tretjina naj bo enaka drugi sliki. Srednja tretjina naj prikazuje **linearno zlivanje prve slike v drugo sliko**. Natančneje, naj bodo $[r_1, g_1, b_1]$ barvne komponente piksla na prvi sliki in $[r_2, g_2, b_2]$ barvne komponente enakoležnega piksla na drugi sliki. Potem naj bodo barve komponente piksla na "zlitii" sliki enake linearnej kombinaciji

$$(1 - \alpha) \cdot [r_1, g_1, b_1] + \alpha \cdot [r_2, g_2, b_2],$$

kjer se parameter $\alpha \in [0, 1]$ enakomerno povečuje od levega proti desnemu robu srednje tretjine. Na primer, $\alpha = 0$ na levem robu srednje tretjine, $\alpha = 1$ na desnem robu, $\alpha = 0,5$ na sredini slike ipd.

Ustvarjeno "zlito" sliko shranite v datoteko `tree.png` v formatu PNG.

Kaj in kako oddam?

Sestavljen program shranite v **datoteko** `blend.py`, ki jo **oddate na** [spletni učilnici](#). Ni potrebno oddajati datoteke `tree.png`, poleg tega se ne pričakuje, da je programska koda opremljena s komentarji. Pazite pa, da program ne vsebuje napak, kar pomeni, da se ukaz `python blend.py` uspešno izvede!

3. Spletno naročanje hrane (\approx 40 vrstic Java kode)

V programskem jeziku Java 8 sestavite razrede `Restaurant`, `Item` in `Order` za **podporo spletnemu naročanju hrane**.

Razred `Restaurant` naj predstavlja **posamezno restavracijo** (npr. McDonald's), za katero hranimo le njeno ime. Razredu dodajte osnovni konstruktor in smiselno implementacijo funkcije `String toString()`.

Razred `Item` naj predstavlja **posamezen izdelek** (npr. Big Mac), za katerega hranimo njegovo ime, ceno v evrih in pa restavracijo, ki ga ponuja. Razredu dodajte osnovni konstruktor in smiselno implementacijo funkcije `String toString()`.

Razred `Order` naj predstavlja **konkretno naročilo**, ki vsebuje seznam naročenih izdelkov. Pri tem omogočamo, da naročilo vsebuje izdelke iz več različnih restavracij. Razredu dodajte konstruktor, ki ustvari prazno naročilo, metodo `void addItem(Item item)`, ki naročilu doda izdelek `item`, in funkcijo `String toString()`, ki vrne niz znakov, ki vključuje:

- skupno ceno naročila,
- ceno naročila za vsako restavracijo posebej in
- seznam izdelkov iz vsake restavracije.

Delovanje razredov lahko preizkusite s pomočjo spodnjega programa, ki ga vključite v metodo

```
void main(String[] args)
```

Java

```
Order order = new Order();

Restaurant restaurant = new Restaurant("McDonalds");
order.addItem(new Item("Big Mac", 3.9, restaurant));
order.addItem(new Item("Hamburger", 1.7, restaurant));
order.addItem(new Item("Fries", 1.7, restaurant));

restaurant = new Restaurant("HoodBurger");
order.addItem(new Item("Le Brie", 7.3, restaurant));
order.addItem(new Item("Krompirček", 3.7, restaurant));

System.out.println(order);
```

Primer izpisa zgornjega programa je prikazan spodaj.

```
McDonalds: 7.30€ [Big Mac, Hamburger, Fries]
HoodBurger: 11.00€ [Le Brie, Krompirček]
Skupaj: 18.30€
```

Bash

Kaj in kako oddam?

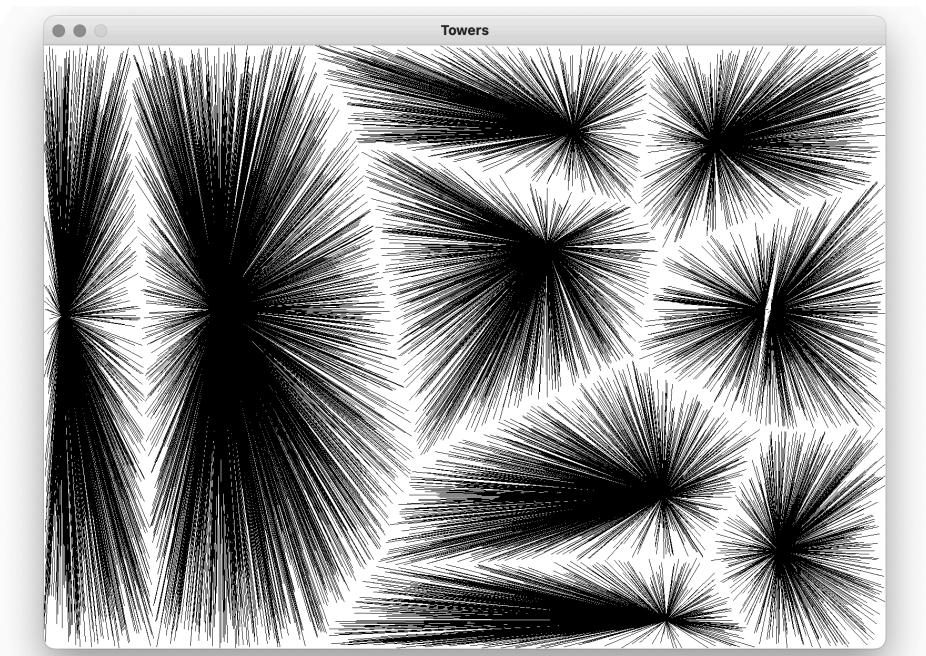
Sestavljeni programe shranite v **datoteke** `Restaurant.java`, `Item.java` in `Order.java`, ki jih **oddate na spletni učilnici**. Ne pričakuje se, da je programska koda opremljena s komentarji. Pazite pa, da program ne vsebuje napak, kar pomeni, da se ukaza `javac Order.java` in `java Order` uspešno izvedeta!

4. Simulacija baznih postaj (≈ 20 vrstic Java kode)

V programskem jeziku Java 8 sestavite **enostaven grafični vmesnik**, ki naj vsebuje zgolj en panel fiksne velikosti. Le-ta naj bo namenjen **simulaciji povezovanja mobilnih naprav** z baznimi postajami. Pri tem predpostavimo, da se mobilna naprava vedno poveže z najbližjo bazno postajo. Mobilne naprave in bazne postaje najpreprosteje predstavite s **seznamom točk v ravnini** `List<Point> points`, ki predstavljajo njihove lokacije na panelu.

Na začetku simulacije ustvarite **10 baznih postaj** na naključno izbranih lokacijah. Le-teh tekom simulacije ne spreminjajte. Nato na naključno izbranih lokacijah ustvarite še **10000 mobilnih naprav**. Lokacije mobilnih naprav **posodobite enkrat na sekundo** tako, da odstranite stare naprave in ustvarite 10000 novih naprav.

Zaradi preglednosti ne prikazujte mobilnih naprav in baznih postaj, vendar zgolj **povežite vsako mobilno napravo z najbližjo bazno postajo** glede na Evklidsko razdaljo. Primer izgleda grafičnega vmesnika je prikazan spodaj. Pri tem je izgled grafičnega vmesnika popolnoma poljuben dokler le-ta zadošča zahtevam naloge.



Priporoča se, da kot osnovo za razvoj uporabite spodnji program.

```
public class Towers {  
  
    public static void main(String[] args) throws Exception {  
        JFrame frame = new JFrame("Towers");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(new Dimension(800, 600));  
        frame.setResizable(false);  
  
        JPanel panel = new JPanel() {  
            private static final long serialVersionUID = 1L;  
            @Override  
            public void paint(Graphics g) {  
                super.paint(g);  
                Graphics2D graphics = (Graphics2D)g;  
  
                ... // izris povezovanja mobilnih naprav  
            }  
        };  
        panel.setBackground(Color.WHITE);  
        frame.add(panel);  
  
        frame.setVisible(true);  
  
        ... // inicializacija baznih postaj  
  
        while (true) {  
            ... // posodabljanje mobilnih naprav  
  
            panel.repaint();  
            Thread.sleep(1000);  
        }  
    }  
}
```

Kaj in kako oddam?

Sestavljen program shranite v **datoteko** `Towers.java`, ki jo **oddate na** [spletni učilnici](#). Ne pričakuje se, da je programska koda opremljena s komentarji. Pazite pa, da program ne vsebuje napak, kar pomeni, da se ukaza `javac Towers.java` in `java Towers` uspešno izvedeta!