

## Други домаћи задатак из Објектно оријентисаног програмирања 2

1) Саставити на језику *Java* следећи пакет класа:

- **Деоница** се ствара са задатом реалном дужином израженом у метрима и реалним нагибом израженим у степенима, који могу да се дохвате. Могу да се дохвате једнословна ознака деонице и њено реално убрзање. Могуће је одредити брзину у  $\frac{m}{s}$  на крају деонице за задату реалну почетну брзину користећи формулу  $V_k = \sqrt{2 * a * s + V_p^2}$ , где су  $V_k$  крајња брзина,  $V_p$  почетна брзина у  $\frac{m}{s}$ ,  $a$  убрзање у  $\frac{m}{s^2}$ ,  $s$  дужина у метрима, као и време потребно за прелажење деонице за задату реалну почетну брзину користећи формулу  $t = \frac{(V_k - V_p)}{a}$ . Могуће је саставити текстуални опис деонице у облику *ознака (дужина, нагиб)*.
- **Тешка деоница** има ознаку **T** и дефинише убрзање формулом  $a = g * \sin(u)$ , где су  $g$  гравитационо убрзање ( $9.81 \frac{m}{s^2}$ ),  $u$  нагиб у степенима.
- **Стаза** садржи произвољан број деоница. Ствара се празна, са задатим називом, након чега се деонице могу додавати на крај једна по једна. Може да се одреди тренутни број деоница, укупна реална дужина свих деоница, као и максимални реални нагиб од свих садржаних деоница. Могуће је одредити једнословну ознаку стазе као најзаступљенију ознаку садржаних деоница, односно прве додате деонице уколико је две или више ознака подједнако заступљено (грешка `GOznaka` је ако стаза не садржи ниједну деоницу). Може да се одреди крајња брзина стазе у  $\frac{m}{s}$  за задату почетну реалну брзину, као и време у секундама потребно да се пређе цела стаза за задату почетну реалну брзину. Текстуални облик стазе је *назив|број\_деоница|дужина|макс\_нагиб*, након чега се у наредном реду у угластим заградама исписују све деонице одвојене знаком `,`.

---

### НАПОМЕНЕ:

- Други домаћи задатак је основа за израду друге лабораторијске вежбе.
- Студент треба да преда своја решења, сходно упутствима које добије преко мејлинг листе предмета. Предата решења биће доступна студенту и користиће их као полазну тачку за израду лабораторијске вежбе.
- Решење домаћег задатка се не оцењује, али улази у састав решења лабораторијске вежбе која се оцењује.

=====

Imena klasa i metoda navedenih u main metodi ne smeju se menjati!

=====

```
package main;

import skijanje.Teska;
import skijanje.Deonica;
import skijanje.GOznaka;
import skijanje.Staza;

public class MainDz {

    public static void main(String[] args) {
        // DZ main
        Deonica deo1 = new Teska(100., 10.), deo2 = new Teska(500., 5.);
        System.out.println(deo1);
        System.out.println(deo2);
        System.out.println("deo1 | oznaka: " + deo1.oznaka());
        System.out.println("deo1 | duzina: " + deo1.duzina());
        System.out.println("deo1 | nagib: " + deo1.nagib());
        System.out.println("deo1 | a: " + deo1.ubrzanje());
        System.out.println("deo1 | Vk: " + deo1.brzina(0));
        System.out.println("deo1 | t: " + deo1.vreme(0));
        Staza gradac = new Staza("Gradac");
        gradac.dodaj(deo1);
        gradac.dodaj(deo2);
        System.out.println(gradac);
        System.out.println("staza | broj delova: " + gradac.broj());
        System.out.println("staza | uk duzina: " + gradac.duzina());
        try {
            System.out.println("staza | oznaka: " + gradac.oznaka());
        } catch (GOznaka o) {
            System.out.println(o);
        }
        System.out.println("staza | maks. nagib: " + gradac.nagib());
        System.out.println("staza | Vk: " + gradac.brzina(0));
        System.out.println("staza | t: " + gradac.vreme(0));
    }
}
```

Primer izlaza:

```
=====
T(100.0,10.0)
T(500.0,5.0)
deo1 | oznaka: T
deo1 | duzina: 100.0
deo1 | nagib: 10.0
deo1 | a: 1.7034886229125867
deo1 | Vk: 18.457998932238493
deo1 | t: 10.835410747081726
Gradac|2|600.0|10.0
[T(100.0,10.0),T(500.0,5.0)]
staza | broj delova: 2
staza | uk duzina: 600.0
staza | oznaka: T
staza | maks. nagib: 10.0
staza | Vk: 34.57883111004541
staza | t: 29.690232937675404
```