

Други домаћи задатак из Објектно оријентисаног програмирања 2

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

- **Бандит** се ствара са задатим **тимом** ком припада (A, B) и садржи целобројну количину златника (при стварању 50). Тим и количина златника могу да се дохвате. Могуће је променити количину златника за задату количину. Текстуални опис бандита се састоји из ознаке тима и количине златника (нпр. A50).
- **Вагон** садржи произвољан број бандита. При стварању, вагон је празан. Могуће је додати бандита у вагон. Могуће је испитати да ли се у вагону налази задати бандит. Могуће је уклонити задатог бандита из вагона. Могуће је дохватити број бандита у вагону. Могуће је дохватити бандита са задате позиције у вагону. Текстуални опис вагона садржи описе свих бандита одвојене зарезима, унутар пара угластих заграда.
- **Композиција** вагона садржи произвољан број вагона. При стварању, композиција је празна. Могуће је додати задати вагон на крај композиције. Могуће је дохватити вагон у ком се налази задати бандит. Грешка (`GNepostojeciVagon`) је уколико вагон не постоји. Могуће је дохватити суседни вагон задатог вагона у задатом **смеру** (*ISPRED, IZA*). Грешка (`GNepostojeciVagon`) је уколико вагон не постоји. Текстуални опис композиције садржи описе свих вагона одвојене доњим цртама.
- **Акција** се ствара са задатом композицијом и може да се изврши над задатим бандитом.
- **Померање** је акција која се ствара са задатим смером. Померањем, бандит се уклања из вагона у ком се тренутно налази и додаје у суседни вагон композиције у смеру померања. Уколико се бандит не налази ни у једном вагону композиције или суседни вагон не постоји, операција је без ефекта. Текстуални опис померања је облика **Pomeranje: смер**.

Приложена је класа са главном функцијом која испитује основне функционалности пакета класа уз исписивање резултата на стандардном излазу (конзоли).

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 које се оцењује

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

```
package main;
import banditi.*;

public class MainDZ {
    public static void main(String[] args) {

        Vagon vagon1 = new Vagon();
        for (int i = 0; i < 6; i++) {
            Bandit bandit = new Bandit(i % 2 == 0 ? Bandit.Tim.A : Bandit.Tim.B);
            bandit.promeniBrojZlatnika((int) (Math.random() * 20 - 10));
            vagon1.dodajBandita(bandit);
        }

        int zlatniciA = 0, zlatniciB = 0;
        for (int i = 0; i < vagon1.dohvatiBrojBandita(); i++) {
            Bandit trenutni = vagon1.dohvatiBandita(i);
            if (trenutni.dohvatiTim() == Bandit.Tim.A) {
                zlatniciA += trenutni.dohvatiBrojZlatnika();
            } else {
                zlatniciB += trenutni.dohvatiBrojZlatnika();
            }
        }
        System.out.println("Bogatiji tim je tim " + (zlatniciA > zlatniciB ? "A" : "B"));
        if (vagon1.sadrziBandita(vagon1.dohvatiBandita(0))) {
            vagon1.ukloniBandita(vagon1.dohvatiBandita(0));
        }

        Vagon vagon2 = new Vagon(); Kompozicija kompozicija = new Kompozicija();
        kompozicija.dodajVagon(vagon1); kompozicija.dodajVagon(vagon2);

        try {
            Vagon vagon = kompozicija.dohvatiSusedniVagon(vagon2, Smer.ISPRED);
            System.out.println("Postoji levi vagon: " + vagon);
            Akcija akcija = new Pomeranje(kompozicija, Smer.IZA);
            akcija.izvrsi(vagon1.dohvatiBandita(0));
            System.out.println(kompozicija);
        } catch (GNepostojeciVagon e) { System.out.println("Ne postoji vagon."); }

        try {
            Vagon vagon = kompozicija.dohvatiVagon(new Bandit(Bandit.Tim.A));
            System.out.println("Postoji bandit u vagonu: " + vagon);
        } catch (GNepostojeciVagon e) { System.out.println("Ne postoji vagon."); }
    }
}
```

=====
Primer izlaza:
=====

Bogatiji tim je tim A
Postoji levi vagon: [B50,A51,B47,A55,B50]
[A51,B47,A55,B50]_[B50]
Ne postoji vagon.