Univerzitet u Beogradu 29.01.2015.
Mašinski fakultet

Drugi kolokvijum iz Objektno orijentisanog programiranja i Java-e

1. Implementirati klasu

```
CurrencyConverter
```

sa privatnim atributom `conversionRates`, tipa `HashMap<String,Double>`, koji implementira uredjene parove tipa (`String`,`Double`) , gde je prva komponenta naziv valute a druga je odnos vrednosti valute prema vrednosti dinara. Obezbediti da konstruktor klase `CurrencyConverter` ima jedan argument tipa `HashMap<String,Double>` koji inicijalizuje atribut `conversionRates`. Implementirati metodu

```
void updateConversionRates(String currency, Double rate)
```

koja unosi u atribut `conversionRates` par (`currency`,`rate`). Implementirati metodu

```
double getConversionRate(String currency)
```

koja za zadatu vrednost `currency` vraća vrednost odnosa vrednosti valute `currency` prema vrednosti dinara. Implementirati metodu

```
void setConversionRates(HashMap<String,Double>)
```

koja postavlja novu vrednost atributa `conversionRates`.

U klasi `A` implementirati metodu

```
public static void main(String arg[])
```

kreirati promenljivu tipa `HashMap<String,Double>`, dodati parove (`"dollars"`, 100), (`"euros"`,120) i (`"pounds"`, 130) i kreirati objekat klase `CurrencyConverter` sa argumentom koji je objekat klase `HashMap<String, Double>`.

2. U klasi `CurrencyConverter` implementirati unutrašnju klasu `Currency`, privatnog pristupa, koja ima zaštićeni atribut `name` tipa `String`. U klasi `CurrencyConverter` definisati

```
interface Conversion
```

sa javnim pristupom, koji ima metodu `convertToDinars` koja prihvata argument tipa `double` i vraća argument tipa `double`. Implementirati unutrašnje klase, klase `CurrencyConverter`, sa imenima `Dollars`, `Euros` i `Pounds`, sa javnim pristupom, koje implementiraju `interface Conversion` i koje nasledjuju klasu `Currency`. Obezbediti da unutrašnje klase `Dollars`, `Euros` i `Pounds` u svom konstruktoru postavljaju vrednost atributa `name` klase `Currency` na vrednosti `"dollars"`, `"euros"` i `"pounds"`, redom.
U klasi `A` implementirati metodu

```
public static void main(String[] argc)
```

koja kreira objekte klasa `Dollars`, `Euros` i `Pounds` i pozvati metode `converToDinars`.

3. Ako ste rešili prva dva zadatka integrišite rešenja u jedinstvenu klasu `CurrencyConverter` sa sledećim dodacima. Metode

```
convertToDinars(double amount)
```

vrše konverziju na taj način što množe vrednost `amount` sa vrednošću koja se dobija iz atributa `conversionRates` za vrednost atributa `name`.
Obezbediti da metoda `getConversionRate`, klase `CurrencyConverter`, generiše izuzetak `NoCurrencyConversionRate` ako je metoda pozvana sa vrednošću argumetna `currency` koji se ne nalazi u atributu `conversionRates`. Obezbediti da izuzetak

```
NoCurrencyConversionRate
```

ima konstruktor koji prihvata argument tipa `String` koji prihvata vrednost tipa `String` čija se vrednost ne nalazi u atributu `conversionRates`. Obezbediti da metode `convertToDinars` unutrašnjih klasa `Dollars`, `Euros` i `Pounds` generišu izuzetke

```
NoDollarsConversionRate
NoEurosConversionRate
NoPoundsConversionRate
```

redom, u slučaju da atribut `conversionRates` nema odgovarajuće vrednosti. Izuzeci

`NoDollarsConversionRate`
`NoEurosConversionRate`
`NoPoundsConversionRate`

su podklase klase `NoCurrencyConversionRate`, koje imaju konstruktore bez argumenata i koje prosledjuju konstruktoru klase `NoCurrencyConversionRate` vrednosti identične vrednostima atributa `name` odgovarajućih klasa.

3. Ako niste rešili prvi i drugi zadatak objasnite sve konstruktore klase `Exception`.