

2. labos, tema 2

Zadatak 1.

U prvom zadatku teme 1 ovog labosa trebali ste napraviti implementaciju klase `SimpleHashtable` koja ima sljedeće metode:

```
public SimpleHashtable();
public SimpleHashtable(int capacity);
public void put(Object key, Object value);
public Object get(Object key);
public int size();
public boolean containsKey(Object key);
public boolean containsValue(Object value);
public void remove(Object key);
public boolean isEmpty();
public String toString();
```

U tom zadatku je bilo objašnjeno kakve smiju biti vrijednosti ključeva i pridruženih vrijednosti te preostalih parametara gore navedenih metoda. Proširite vaše rješenje iz tog zadatka tako što ćete napraviti potpunu obradu i izazivanje iznimki za klasu `SimpleHashtable` i njenu ugniježdenu klasu `TableEntry` te vašu (izvršnu) klasu s `main` metodom.

Zadatak 2.

U zadatku s teme 2 prvog labosa ste trebali implementirati klase `Rectangle`, `Circle` i `Line` koje predstavljaju geometrijske likove. Svaki od tih razreda ste trebali opremiti metodom `drawOnPicture` koja prima referencu na sliku (`Picture` ili `PictureDisplay`) te na njoj paljenjem slikovnih elemenata crta taj lik. Uvedite novo sučelje `DrawableShape` koje se sastoji od metode `drawOnPicture` i koje će implementirati klase ova tri geometrijska lika.

Nakon toga, primijenite oblikovni obrazac metoda tvornice za ove geometrijske likove. Pri tome neka `DrawableShape` bude sučelje objekta koji nastaje metodom tvornicom, a neka njegove implementacije `Rectangle`, `Circle` i `Line` budu konkretni proizvodi. Napišite sučelje `DrawableShapeCreator` koje sadrži metode opisane u nastavku kojim se stvaraju novi objekti tipa `DrawableShape`:

```
DrawableShape[] create(int num)
DrawableShape create(String typeOfShape)
```

Prva metoda treba vratiti polje geometrijskih likova odgovarajuće duljine, a druga metoda treba vratiti jedan geometrijski lik odgovarajućeg tipa na osnovu vrijednosti parametra `typeOfShape` (tj. vrijednosti "rectangle", "circle" ili "line"). Osim toga napišite klasu `RandomShapeFactory` koje će biti konkretna implementacija ove metode tvornice, a vraćat će nasumično generirane geometrijske likove. Pri tome pretpostavite da se u konstruktoru ove klase predaju maksimalna širina i visina lika.

Za testiranje rada ove tvornice klasa napišite (izvršnu) klasu s `main` metodom koja će koristiti ovu metodu tvornica kao što je na primjer prikazano u sljedećem odsječku koda:

```
static final int WIDTH = 100;
static final int HEIGHT = 100;
```

```
public static void main(String[] args) {  
    DrawableShapeCreator factory = new RandomShapeFactory(WIDTH, HEIGHT);  
    DrawableShape[] shapes = factory.create(5);  
    Picture pic = new Picture(WIDTH, HEIGHT);  
    for(DrawableShape s : shapes) {  
        s.drawOnPicture(pic);  
    }  
    PictureDisplay.showPicture(pic);  
}
```

U ovom zadatku također napravite obradu i izazivanje iznimki tako što ćete proći kroz implementacije tvornice i svih geometrijskih likova te iznimkama obraditi sve što je potrebno (npr. stvaranje pravokutnika negativne širine, crtanje geometrijskog lika na slici na koju on ne stane, predaja krivog stringa pod `typeOfShape`, itd.).