

OOP – Auditorne vježbe – tjedan 6

Zadatak 1:

Stvorite klasu `Prodavac` koja sadrži metodu `zabiljezi(String zapis)`. Ta metoda prima zapis prodaje dobiven OCR-om sa pisanog računa u obliku teksta npr. "12345.45, 4, 0.12" gdje:

- prvi element jest ukupni plaćeni iznos za istu vrstu artikla,
- drugi element jest broj artikala,
- zadnji element jest vraćeni iznos.

Metoda mora parsirati primljeni zapis pretpostavljajući navedeni format zapisa. Zatim metoda mora izračunati jediničnu cijenu prodanog proizvoda i ispisati ju na ekran. Naravno, krhko je znanje pa treba biti oprezan pri parsiranju i loviti sva kršenja pretpostavljenog formata te ispisati uzrok pogreške.

Za pomoć, mogući primljeni zapisi su (poput testova na Edgaru):

- "a,2,2.3"
- "1,2"
- null
- "12345.45, 4, 0.12"
- " 4, 0,1"
- " 4, 0,4"
- "44, 3, 100"
- "102,12,3,4,5,6,7,8,9,0"
- " , , "

Isprobati dvije alternative:

- Nastaviti sa izvođenjem programa nakon ispisa poruke pogreške.
- Proslijediti pogrešku nakon ispisa poruke.

Za samostalnu vježbu

Što ako se primi zapis sa negativnim brojevima, poput "-10,1,-5" ili "-11,-2,-3"?

```

public class Prodavac {
    public static void zabiljezi(String zapis)
    {
        try
        {
            String[] elementi = zapis.split(",");
            float[] podatci=new float[3];

            //arbitrarno: ignoriramo visak, gledamo samo prva 3
            for (int i=0; i<3; i++)
            {
                podatci[i]=Float.parseFloat(elementi[i]);
            }

            float jedinicaCijena=(podatci[0]-podatci[2])/podatci[1];

            // dijeljenje s float 0 ne izaziva "Div By 0" iznimku

            // TODO #3: isprobajte kod ispod
            //if (podatci[1]==0)
            //    throw new ArithmeticException("Dijeljenje s 0!");

            // ovo je specifična greska vezana uz domenu problema
            if (jedinicaCijena<0)
                throw new IllegalArgumentException("Negativna jedinica cijena!");

            System.out.println("Jedinica cijena: " + jedinicaCijena);

            //TODO #4 (zadaca): sto ako su predani negativni brojevi?
        }
        /*
        // TODO #1: poredak exceptiona? isprobajte kod ispod
        catch(Exception ex)
        {
            System.err.println("Neka druga greska u zapisu" + ex.getMessage());
        }
        */
        //TODO #2: isprobajte zakomentirana prosljedjivanja iznimki
        catch(NullPointerException ex)
        {
            System.err.println("Nije poslan zapis");

            // omatanje i prosljedjivanje iznimke
            // sto ako se prosljedi bez omatanja?
            //throw new IllegalArgumentException("Nije poslan zapis",ex);
        }
        catch(NumberFormatException ex)
        {
            System.err.println("Zapis ne sadrzi brojeve");
            //throw new IllegalArgumentException("Zapis ne sadrzi brojeve", ex);
        }
        catch(ArrayIndexOutOfBoundsException ex)
        {
            System.err.println("Zapis nema barem 3 elementa");
            //throw new IllegalArgumentException("Zapis nema barem 3 elementa",ex);
        }
        catch(IllegalArgumentException ex)
        {
            System.err.println(ex.getMessage());
            //throw ex;
        }
    }
}

```

```

        catch(Exception ex)
        {
            System.err.println("Neka druga greska u zapisu: " + ex.getMessage());
            //throw new IllegalArgumentException("Neka druga greska u zapisu",ex);
        }
    }

    public static void main(String[] args) {
        Prodavac.zabiljezi("12345.45,4,0.12");
        Prodavac.zabiljezi("a,2,2.3");
        Prodavac.zabiljezi("1,2");
        Prodavac.zabiljezi(null);
        Prodavac.zabiljezi(" 4, 0,1");
        Prodavac.zabiljezi(" 4, 0,4");
        Prodavac.zabiljezi("44, 3, 100");
        Prodavac.zabiljezi("102,12,3,4,5,6,7,8,9,0");
        Prodavac.zabiljezi(",,");
        //Prodavac.zabiljezi("-7,2,-10");
    }
}

```

Zadatak 2:

Klasa Bankar u metodi pohlepnoGrabljenje() neka stvori polje znakova veličine Integer.MAX_VALUE koje sadrži znak '\$'. Unutar metode main, isprobati poziv pohlepnoGrabljenje unutar try-catch bloka koji lovi **sve** iznimke i ispisuje njihov opis. Što se dogodi kod izvođenja? Zašto?

a) Kakve se klase može hvatati catch blokom? Kakve se klase može bacati kao iznimke?

```
import java.util.Arrays;

public class Bankar {

    public static void pohlepnoGrabljenje()
    {
        char [] gimme=new char [Integer.MAX_VALUE];
        Arrays.fill(gimme, '$');
        System.out.println("Uspjeh!");
    }

    public static void main(String[] args) {
        try
        {
            pohlepnoGrabljenje();
        }
        catch (Exception ex)
        {
            System.err.println("Ha! Uhvacen na djelu!" + ex.getMessage());
        }
        /*
        //TODO #1: sto se sve moze uhvatiti?
        catch(Bankar ex)
        {
        }
        */
    }
}
```

Zadatak 3:

Stvorite dvije vlastite iznimke:

- UpException koja je **neprovjeravana**
- TomahawkException koja je **provjeravana**

Stvorite klasu ExtremeWays koja sadrži pet statičkih metoda:

- method1 – poziva method2, te potom method4
- method2 – poziva method3, pri čemu hvata ArithmeticException
- method3 – izaziva TomahawkException
- method4 – izaziva UpException
- main – poziva method1, pri čemu hvata TomahawkException i UpException

Za vježbu isprobajte hvatati ili nehvatati vlastite iznimke na različitim mjestima u kodu.

```
public class UpException extends RuntimeException {  
  
    private static final long serialVersionUID = 1L;  
  
    public UpException(String message) {  
        super(message);  
    }  
  
}
```

```
public class TomahawkException extends Exception {  
  
    private static final long serialVersionUID = 1L;  
  
    public TomahawkException(String message) {  
        super(message);  
    }  
  
}
```

```

public class Extremeways {

    public static void method1() throws TomahawkException
    {
        method2();
        method4();
    }

    public static void method2() throws TomahawkException
    {
        try
        {
            method3();
        }
        catch(ArithmeticException ex)
        {
            System.err.println("Stranded arithmetic error!");
        }
    }

    public static void method3() throws TomahawkException
    {
        throw new TomahawkException("Swoosh!");
    }

    public static void method4() // throws UpException
    {
        throw new UpException("?!");
    }

    public static void main(String[] args)
    {
        //method1();

        try
        {
            method1();
        }
        catch(TomahawkException | UpException ex)
        {
            System.out.println(ex.getMessage());
        }
    }
}

```