

Ushtrimi 1

```
import java.io.*;
```

```
class SuperClass {
```

```
    // SuperClass doesn't declare any exception
```

```
    void method()
```

```
    {
```

```
        System.out.println("SuperClass");
```

```
    }
```

```
}
```

```
// SuperClass inherited by the SubClass
```

```
class Dokument extends SuperClass {
```

```
    // method() declaring Unchecked Exception ArithmeticException
```

```
    void method() throws ArithmeticException
```

```
    {
```

```
        // ArithmeticException is of type Unchecked Exception
```

```
        // so the compiler won't give any error
```

```
        System.out.println("SubClass");
```

```
    }
```

```
// Driver code
```

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

```
{
```

```
    Dokument s = new Dokument();
```

```
    s.method();
```

```
}
```

```
}
```

Ushtrimi 2.

```
class Building {
```

```
    void color()
```

```
    {
```

```
        System.out.println("Blue");
```

```
    }
```

```
}
```

```
class Dokument extends Building{
```

```
    //It throws an unchecked exception
```

```
    void color() throws NullPointerException
```

```
    {
```

```

        System.out.println("White");
    }
    public static void main(String args[]){
        Dokument obj = new Dokument();
        obj.color();
    }
}

```

Ushtrimi 3

```

import java.io.*;
class Parent{
    void msg(){System.out.println("parent");}
}

class Dokument extends Parent{
    void msg()throws ArithmeticException{
        System.out.println("child");
    }
    public static void main(String args[]){
        Parent p=new Dokument();
        p.msg();
    }
}

```

Ushtrimi 4

```

import java.io.BufferedReader;

import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class MyClass{

    public static void main(String[] args){
        // open the file
        String fileName = "c:\\file.txt";
        String line = null;
        try{
            // read a text file
            FileReader fileReader = new FileReader(fileName);
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            while((line = bufferedReader.readLine()) != null){

```

```

        System.out.println(line);
    }
    bufferedReader.close();
}
catch(FileNotFoundException e){
    System.out.println("Error file " + fileName + " was not fount!");
}
catch(IOException e){
    e.printStackTrace();
}
}
}

```

Ushtrimi 6

```

import java.util.Scanner;
class Alfa{
    public static void main(String args[]) {
        Scanner vlera=new Scanner(System.in);
        System.out.println("Jep numrin e cunave ");
        int djem=vlera.nextInt();
        System.out.println("Jep numrin e gocave ");
        int goca=vlera.nextInt();

        if (djem==0 && goca==0) {
            System.out.println("Mesimi nuk zhvillohet se ska studente");
            System.exit(0);
        }
        else if (djem==0) {
            System.out.println("Mesimi nuk zhvillohet se ska djem");
            System.exit(0);
        }
        else if (goca==0) {
            System.out.println("Mesimi nuk zhvillohet se ska goca");
            System.exit(0);
        }
        if (goca>=djem ) {
            System.out.println("Djem qe do te kercejne jane " +
goca/(double)djem +"goca");
        }
        else
            System.out.println("Goca qe do te kercejne jane " +
djem/(double)goca +"djem");

        System.out.println("Fillo kusrin e kercimit");}

```

Ushtrimi 7

```
import java.io.*;

public class CheckingAccount {
    private double balance;
    private int number;

    public CheckingAccount(int number) {
        this.number = number;
    }

    public void deposit(double amount) {
        balance += amount;
    }

    public void withdraw(double amount) throws InsufficientFundsException {
        if(amount <= balance) {
            balance -= amount;
        }else {
            double needs = amount - balance;
            throw new InsufficientFundsException(needs);
        }
    }

    public double getBalance() {
        return balance;
    }

    public int getNumber() {
        return number;
    }
}
```

Ushtrim 8

```
class JavaException {
    public static void main(String args[]) {
        try {
            int d = 1;
            int n = 20;
            int fraction = n / d;
            int g[] = {
                1
            };
            g[20] = 100;
        }
    }
}
```

```

/*catch(Exception e){
    System.out.println("In the catch block due to Exception = "+e);
}*/
catch (ArithmeticException e) {
    System.out.println("In the catch block due to Exception = " + e);
} catch (ArrayIndexOutOfBoundsException e) {
    System.out.println("In the catch block due to Exception = " + e);
}
System.out.println("End Of Main");
}
}

```

Ushtrimi 9

```
import java.util.*;
```

```
public class ExceptionDemo {
```

```
    static void func(int a,int b) throws ArithmeticException,
    ArrayIndexOutOfBoundsException{
```

```
        System.out.println(10/a);
```

```
        int[] arr={ 1,2,3};
```

```
        System.out.println(arr[b]);
```

```
    }
```

```
    public static void main (String[] args) {
```

```
        Scanner in=new Scanner(System.in);
```

```
        for(int i=0;i<3;i++){
```

```
            try{
```

```
                func(in.nextInt(),in.nextInt());
```

```
        }catch(ArithmeticException e){  
  
            System.out.println("can't divide by zero");  
  
        }catch(ArrayIndexOutOfBoundsException e){  
  
            System.out.println("Out of bounds!");  
  
        }  
    }  
}
```

Ushtimi 10

Detyre:

- Le të supozojme se, ju jeni duke punuar me një kompani ajrore
- Ju jeni në departamentin e check-in të bagazhit dhe sipas rregullave, ju mund të lejoni 15 kg për klient.
- Deri më tani më shumë se 15 kg peshë është një gjendje anormale për ne ose me fjalë të tjera është një përjashtim
- Ky është përjashtimi ynë i bazuar në logjikë, kështu që ne do të krijojmë përjashtimin tonë të personalizuar `WeightLimitExcended`
- Sipas sintaksës, ajo do të zgjasë `Përjashtimin`.
- Ne përcaktojmë konstruktorin i cili do të thërritet sa më shpejt që do të ndodhi një përjashtim
- Ne duhet të krijojmë në mënyrë të qartë përjashtimin dhe kështu do të përdorim fjalën `throw` për këtë.