

```

public class C {
    public void m1(int p1, String s) {
        try {
            System.out.println("1");
            p1 = m2(p1 + 1, s);
            System.out.println("2");
        } catch (ArithmeticException e) {
            System.out.println("3");
        } catch (Exception e) {
            System.out.println("4");
        }
        System.out.println("5:" + p1);
    }
    public int m2(int q1, String s) {
        q1 = q1 + 1;
        try {
            System.out.println("6");
            if (q1 != 10)
                throw new IOException();
            if (s.length() == 5)
                q1 = q1 / 0;

            m3(s);
            System.out.println("7");
        } catch (ArithmeticException e) {
            System.out.println("8");
            String s2 = null;
            System.out.println(s2.length());
            System.out.println("9");
        } catch (IOException e) {
            System.out.println("10");
            int arr[] = new int[10];
            arr[10] = -5;
            System.out.println("11");
        } catch (NullPointerException e) {
            System.out.println("12");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("13");
        }
        return q1 + 2;
    }

    public void m3(String s) {
        try {
            if (s.length() % 2 == 0)
                throw new IllegalAccessException();
        } catch (Exception e) {
            System.out.println("14");
        } finally {
            System.out.println("15");
        }
    }
}

```

```

public class Main {
    public static
    void main(String[] args) {
        C obj = new C();
        obj.m1(8, "BOTOL"); //1
        obj.m1(8, "abcdef"); //2
        obj.m1(9, "abc"); //3
    }
}

```

Membuat kelas Exception sendiri

Soal 1

Buatlah sebuah kelas **Person**, yang memiliki dua buah instance variable **name** dan **age**. Berikan **konstruktor** serta **getter/setter** method yang sesuai (sederhana saja).

Buatlah sebuah kelas exception baru yang bernama **WrongAgeException**, yang meng-**extends** kelas exception yang sudah ada pada Java dan akan memberikan keterangan apabila **umur bernilai negatif**.

Sebelum menentukan **superclass** dari kelas **WrongAgeException**, Anda harus menentukan terlebih dahulu apakah **WrongAgeException** masuk kategori **checked exception** atau **unchecked exception**? Kali ini, kita tentukan saja bahwa **WrongAgeException** termasuk **unchecked exception**.

Kemudian, modifikasi **konstruktor** dan **method setAge()** pada kelas **Person** supaya dapat melempar **WrongAgeException** jika terjadi kesalahan.

Soal 2

Buatlah sebuah kelas exception, yang dinamakan **ExceptionLineTooLong** yang akan mencetak message, misal **"String terlalu panjang: 87 karakter"**, seandainya exception jenis ini terjadi dan tidak ditangani.

Buatlah **sebuah program yang membaca semua line pada file teks (program utama)**, dan melemparkan sebuah exception tipe **ExceptionLineTooLong** apabila terdapat baris yang panjangnya lebih dari **80 karakter**.

ExceptionLineTooLong perlu menyimpan informasi berapa panjang baris yang menyebabkan kesalahan (> 80 karakter) tersebut. Tampilkan juga informasi panjang baris ini pada pesan exception.

ExceptionLineTooLong termasuk kategori **checked exception** karena permasalahan ini termasuk ke dalam permasalahan I/O dan sulit dihindari oleh programmer.