

LAPORAN PRAKTIKUM

Pemrograman Berorientasi Objek

Diajukan untuk memenuhi salah satu tugas praktikum mata kuliah Pemrograman berorientasi objek



Disusun Oleh:

Hasbi Andi Muttaqin (231511049)

**Jurusan Teknik Komputer dan Informatika
Program Studi D-3 Teknik Informatika
Politeknik Negeri Bandung 2024**

DAFTAR ISI

TASK 1	3
TASK 2	4
TASK 3	Error! Bookmark not defined.
LINK GITHUB.....	8

Tanggal praktikum : 24 Oktober 2024

CASE 1 Exceptions Aren't Always Errors

Nama Program: CountLetters.java

Deskripsi

berisi program yang membaca kata dari pengguna dan mencetak jumlah kemunculan setiap huruf dalam kata tersebut.

Tugas:

1. Mengetes program awal dari CountLetters dengan memasukan sebuah frasa yang terdiri lebih dari 1 kata dengan spasi

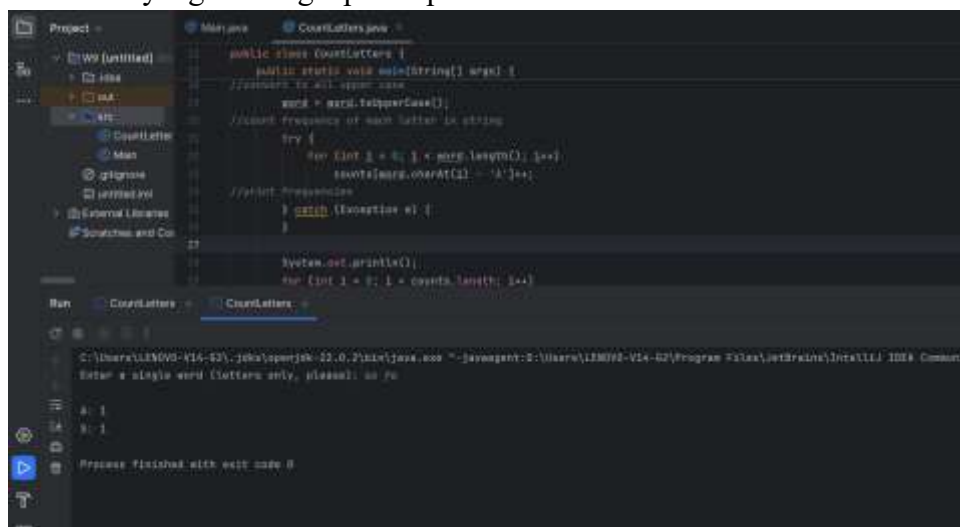
```
C:\Users\LENOVO-V14-G2\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:D:\Users\LENOVO-V14-G2\Program Files\JetBrains\IntelliJ IDE
Enter a single word (letters only, please): as as
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index -33 out of bounds for length 26
    at CountLetters.main(CountLetters.java:13)

Process finished with exit code 1
```

2. Meletakkan syntax try di body pada for pertama

```
try {
    int i;
    char current;
    for (i = 0; i < word.length(); i++) {
        current = word.charAt(i);
        if (current >= 'A' && current <= 'Z') {
            counts[current - 'A']++;
        } else {
            throw new Exception(current + " Not a letter");
        }
    }
}
```

3. Tambahkan catch yang menangkap exception.



Program sudah bisa berjalan namun belum ada pesan yang menunjukkan mengapa “fa” tidak terprint out

4. Memodifikasi body dari catch sehingga mencetak pesan yang berguna

```
C:\Users\LENOVO-V14-G2\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:D:\
Enter a single word (letters only, please): hasbi*andi
' * ' Not a letter

A: 2
B: 1
D: 1
H: 1
I: 2
N: 1
S: 1
```

Permasalahan

-

Solusi

-

Nama teman yang membantu

CASE 2 : Placing Exception Handlers

Nama Program : ParseInts.java

Deskripsi

berisi program yang melakukan hal berikut:

- Meminta dan membaca baris masukan
- Menggunakan Pemindai kedua untuk mengambil baris input satu token dalam satu waktu dan menguraikan bilangan bulat dari masing-masing token saat diekstraksi.
- Menjumlahkan bilangan bulat.
- Mencetak hasil penjumlahan
- Jika Anda memberikan input
“ 10 20 30 40 “

maka ia akan mencetak Jumlah dari bilangan-bilangan bulat pada baris tersebut adalah 100.

Tugas

Anda akan mendapatkan `NumberFormatException` ketika program mencoba memanggil `Integer.parseInt` pada "We", yang bukan merupakan bilangan bulat. Salah satu cara untuk mengatasinya adalah dengan meletakkan loop yang membaca data di dalam blok try dan menangkap `NumberFormatException`, tetapi tidak melakukan apa pun dengannya. Dengan cara ini, jika datanya bukan bilangan bulat, tidak akan menyebabkan error; program akan masuk ke penanganan pengecualian yang tidak melakukan apa pun. Lakukan ini dengan cara berikut:

1. Modifikasi program untuk menambahkan pernyataan try yang mencakup seluruh loop while. Letakkan try dan pembuka kurung { sebelum while, dan catch setelah tubuh loop. Tangkap NumberFormatException dan biarkan tubuh catch kosong.

```
try {
    while (scanLine.hasNext()) {
        val = Integer.parseInt(scanLine.next());
        sum += val;
    }
} catch (NumberFormatException e) {
}
```

2. Kompilasi dan jalankan program, lalu masukkan satu baris dengan campuran bilangan bulat dan nilai lainnya. Anda akan melihat bahwa program berhenti menjumlahkan pada nilai non-bilangan bulat pertama, sehingga baris di atas akan menghasilkan jumlah 0, dan baris "1 fish 2 fish" akan menghasilkan jumlah 1. Ini terjadi karena seluruh loop berada di dalam blok try, jadi ketika pengecualian terjadi, loop akan dihentikan.



- 3.
4. Pindahkan blok try dan catch di dalam loop. Sekarang, ketika pengecualian terjadi, pernyataan berikutnya adalah iterasi berikutnya dari loop, sehingga seluruh baris akan diproses. Input dogs-and-cats sekarang akan menghasilkan jumlah 3, demikian pula input fish.

```
while (scanLine.hasNext()) {
    try {
        val = Integer.parseInt(scanLine.next());
        sum += val;
    } catch (NumberFormatException e) {
    }
}
```



Pengamatan

NumberFormatException terjadi saat ada upaya untuk mengubah string dengan format yang tidak tepat menjadi nilai numerik. Artinya, saat tidak mungkin mengubah string dalam tipe numerik apa pun (float, int, dll.), Exception ini akan muncul. Ini adalah Runtime Exception (Unchecked Exception) di Java. Ini adalah subkelas dari kelas IllegalArgumentException. Untuk menangani pengecualian ini, blok try – catch dapat digunakan.

Nama teman yang membantu

-

Case 3 : Throwing Exceptions

Nama Program :

Deskripsi :

1. Menangani Faktorial Bilangan Negatif:

Identifikasi Kesalahan:

Saat ini, metode factorial mengembalikan 1 untuk bilangan negatif, yang secara matematis tidak benar karena faktorial tidak terdefinisi untuk bilangan negatif.

Solusi:

Modifikasi metode factorial di kelas MathUtils agar melempar exception IllegalArgumentException jika argumennya negatif. Gunakan pesan yang spesifik untuk menjelaskan masalahnya.

Penyesuaian pada Header dan Body Metode:

a. Header Metode:

Ubah header metode factorial untuk menunjukkan bahwa metode ini dapat melempar IllegalArgumentException.

b. Body Metode:

Tambahkan pemeriksaan pada nilai argumen, dan jika bernilai negatif, lempar IllegalArgumentException dengan pesan yang relevan.

Menangkap dan Menangani Exception di Kelas Factorials:

```
public static int factorial(int n){
    if (n < 1){
        throw new IllegalArgumentException("Bukan bilangan asli");
    }else if (n > 16){
        throw new IllegalArgumentException("Melebihi batas");
    }
    int fac = 1;
    for (int i=n; i>0; i--){
        fac *= i;
    }
    return fac;
}
```

Modifikasi di Kelas Factorials:

Tambahkan blok try-catch di dalam metode utama (main) pada kelas Factorials untuk menangkap `IllegalArgumentException`.

Penanganan Exception: Jika exception terdeteksi, cetak pesan error yang sesuai, namun lanjutkan program agar pengguna dapat memasukkan angka lain.

```
try {
    System.out.println("Factorial(" + val + ") = " + MathUtils.factorial(val));
} catch (IllegalArgumentException e) {
    System.out.println(e.getMessage());
}
```

2. Menangani Overflow pada Faktorial Bilangan Besar:

Identifikasi Masalah Overflow:

Program mengembalikan nilai negatif untuk faktorial lebih besar dari 16 karena overflow pada tipe data `int` (nilai faktorial terlalu besar untuk `int`).

Solusi Overflow:

Modifikasi metode `factorial` untuk melempar `IllegalArgumentException` jika argumen lebih besar dari 16, karena metode ini hanya dapat mengkomputasi faktorial hingga 16.

Pesan Exception yang Berbeda:

Pastikan untuk memberikan pesan berbeda yang menjelaskan alasan kesalahan jika argumen lebih besar dari 16, misalnya, "Faktorial hanya terdefinisi untuk nilai kurang dari atau sama dengan 16."

```
17         System.out.println("Factorial( " + val + ") = " + MathUtils.factorial(val));
18     } catch (IllegalArgumentException e) {
19         System.out.println(e.getMessage());
20     }
21     System.out.print("Another factorial? (y/n) ");
```

Run Factorials

C:\Users\LENOVO-V14-G2\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:D:\Users\LENOVO-V14-G2\jre\lib\dtplugin\javafx.jar" -Djava.awt.headless=true -Djava.class.path=D:\Users\LENOVO-V14-G2\jre\lib\dtplugin\javafx.jar -Djava.class.path=D:\Users\LENOVO-V14-G2\jre\lib\dtplugin\javafx.jar

Enter an integer: -1
Bukan bilangan asli
Another factorial? (y/n)

```
19         System.out.println(e.getMessage());
20     }
21     System.out.print("Another factorial? (y/n) ");
22     keepGoing = scan.next();
23 }
24 }
25 }
```

Run Factorials

C:\Users\LENOVO-V14-G2\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:D:\Users\LENOVO-V14-G2\jre\lib\dtplugin\javafx.jar" -Djava.awt.headless=true -Djava.class.path=D:\Users\LENOVO-V14-G2\jre\lib\dtplugin\javafx.jar -Djava.class.path=D:\Users\LENOVO-V14-G2\jre\lib\dtplugin\javafx.jar

Enter an integer: 17
Melebihi batas
Another factorial? (y/n)

LINK GITHUB

<https://github.com/Yorubreak/Pemrograman-Berorientasi-Objek.git>

