

Started on	Friday, 7 November 2025, 12:26 PM
State	Finished
Completed on	Friday, 7 November 2025, 1:01 PM
Time taken	35 mins 16 secs
Marks	300.00/300.00
Grade	10.00 out of 10.00 (100%)

Time limit	1 s
Memory limit	64 MB


**Nama File:** Char.zip


Implementasikan kelas dalam file [berikut](#).

Upload jawaban sebagai berkas **Char.zip** yang berisi tepat 6 file:  
Character.java, SpellCastFailedException.java, InsufficientManaException.java InvalidManaAmountException.java, SpellService.java, Spellcaster.java

**Catatan:** Pastikan setiap output diakhiri oleh newline ("\\n") atau menggunakan `println`.

Untuk membantu melakukan pengetesan, silahkan gunakan main file [berikut](#). Berikut [outputnya](#).

Java 8 

 [Char.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.06 sec, 29.09 MB
2	10	Accepted	0.06 sec, 28.21 MB
3	10	Accepted	0.06 sec, 30.50 MB
4	10	Accepted	0.06 sec, 28.21 MB
5	10	Accepted	0.06 sec, 28.33 MB
6	10	Accepted	0.06 sec, 28.48 MB
7	10	Accepted	0.06 sec, 28.37 MB
8	10	Accepted	0.06 sec, 28.13 MB
9	10	Accepted	0.06 sec, 28.57 MB
10	10	Accepted	0.06 sec, 28.34 MB

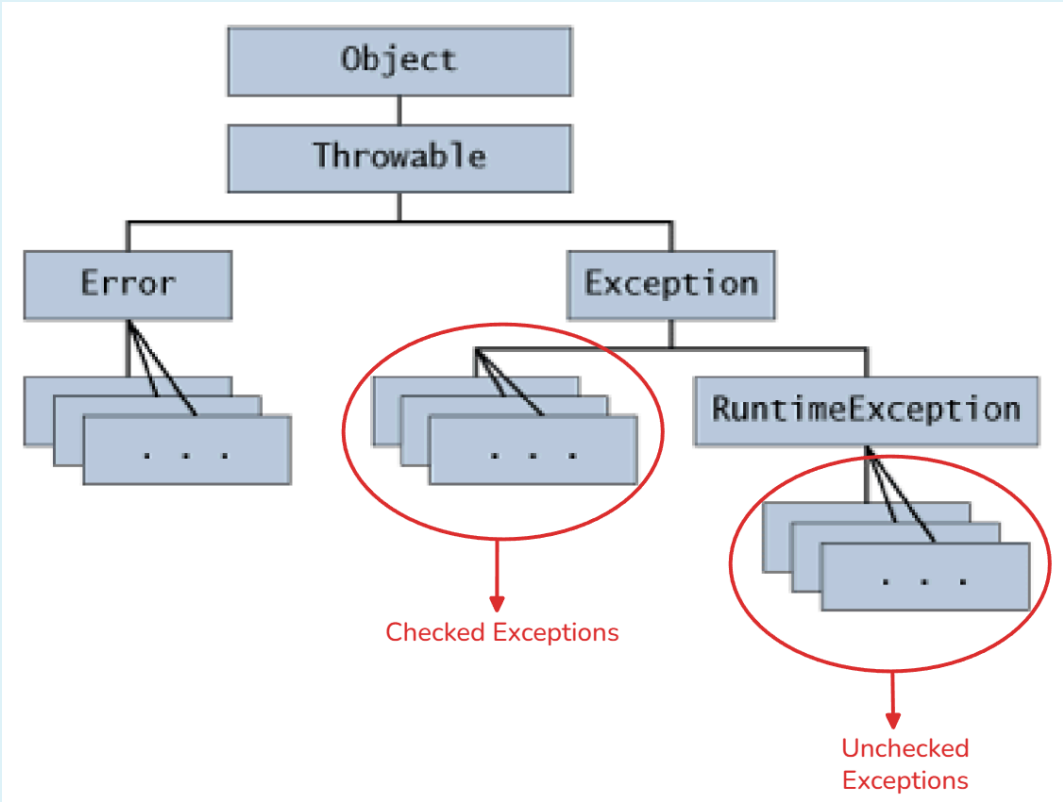
Time limit	1 s
Memory limit	64 MB

**Nama File:** CheckoutService.java

Dalam Java, Exceptions terbagi menjadi dua, yaitu Checked Exceptions dan Unchecked Exceptions.

- Checked Exceptions: kondisi yang masih bisa diprediksi dan ditangani oleh program, sehingga **wajib** ditangani oleh programmer dengan `try-catch` atau `throws`
- Unchecked Exceptions: kesalahan akibat logika pemrograman atau input tidak valid yang **tidak dipaksa** ditangani karena seharusnya **dihindari sejak awal** oleh programmer.

Struktur hierarki exceptions secara umum dapat digambarkan sebagai berikut:



Implementasikan kelas CheckoutService.java dalam zip [berikut](#).

Upload jawaban sebagai berkas **CheckoutService.java**

Gunakan main file [berikut](#) untuk melakukan pengujian hasil implementasi kalian.  
Berikut [output](#) yang diharapkan

**Catatan:** Pastikan setiap output diakhiri oleh newline ("`\n`") atau menggunakan `println`. Tambahkan `import java.io.IOException` pada file **CheckoutService.java**

Java 8

[CheckoutService.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.07 sec, 28.95 MB
2	10	Accepted	0.07 sec, 28.71 MB

No	Score	Verdict	Description
3	10	Accepted	0.07 sec, 28.01 MB
4	10	Accepted	0.06 sec, 29.04 MB
5	10	Accepted	0.06 sec, 29.22 MB
6	10	Accepted	0.07 sec, 28.34 MB
7	10	Accepted	0.06 sec, 28.09 MB
8	10	Accepted	0.06 sec, 28.07 MB
9	10	Accepted	0.06 sec, 27.92 MB
10	10	Accepted	0.06 sec, 29.00 MB

Time limit	1 s
Memory limit	64 MB

**Nama File:** Order.zip

Implementasikan kelas dalam file [berikut](#).

Upload jawaban sebagai berkas **Order.zip** yang berisi tepat 13 file

**Catatan:** Pastikan setiap output diakhiri oleh newline ("`\n`") atau menggunakan `println`.

Untuk membantu melakukan pengetesan, silahkan gunakan driver file [berikut](#) dengan expected output seperti [berikut](#).

Java 8 ▾

 [Order.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	7	Accepted	0.07 sec, 29.04 MB
2	7	Accepted	0.10 sec, 30.11 MB
3	7	Accepted	0.10 sec, 31.74 MB
4	7	Accepted	0.09 sec, 29.97 MB
5	7	Accepted	0.09 sec, 31.78 MB
6	7	Accepted	0.09 sec, 30.11 MB
7	7	Accepted	0.07 sec, 28.57 MB
8	7	Accepted	0.09 sec, 32.48 MB
9	7	Accepted	0.09 sec, 29.77 MB
10	7	Accepted	0.09 sec, 31.83 MB
11	7	Accepted	0.09 sec, 30.35 MB
12	7	Accepted	0.06 sec, 28.02 MB
13	7	Accepted	0.10 sec, 30.59 MB
14	9	Accepted	0.10 sec, 32.54 MB

