

Started on	Monday, 28 April 2025, 8:10 AM
State	Finished
Completed on	Monday, 28 April 2025, 9:49 AM
Time taken	1 hour 39 mins
Grade	60.00 out of 300.00 (20%)

Question **1**  
Not answered  
Marked out of 100.00

Time limit	1 s
Memory limit	64 MB

Pasti kalian pernah mencoba membuat fungsi yang dapat menjumlahkan semua bilangan di dalam sebuah list.

```
public static int sum(List<Integer> list) {
    int result = list[0];
    for (int i=1; i<list.size(); i++) {
        result += list[i];
    }
    return result;
}
```

Atau mungkin sebuah fungsi yang dapat menggabungkan (konkatenasi) seluruh string didalam sebuah list.

```
public static String join(List<String> list) {
    String result = list[0];
    for (int i=1; i<list.size(); i++) {
        result += list[i];
    }
    return result;
}
```

Atau bahkan sebuah fungsi yang dapat mencari nilai terbesar dalam sebuah list integer.

```
public static int max(List<Integer> list) {
    int result = list[0];
    for (int i=1; i<list.size(); i++) {
        if (list[i] > result) {
            result = list[i];
        }
    }
    return result;
}
```

Jika dilihat, bentuk ketiga fungsi tersebut sangat serupa. Seketika, Tuan Bil mengingat fungsi-fungsi [Fold](#)), yang salah satu fungsinya adalah, **Reduce**.  
**Reduce** adalah fungsi yang dapat menggabungkan seluruh elemen dalam List menjadi satu nilai dengan menerapkan operasi fungsi secara berurutan. Karena Tuan Bil sudah lama tidak memegang bahasa ~~Java~~ Java, ia meminta kalian untuk membuat fungsi tersebut!

Untuk menggunakan binary operator, dapat menggunakan fungsi apply(x, y) dengan x dan y adalah dua data yang akan dioperasikan.  
Diberikan file [Fold.java](#) untuk dilengkapi dan disubmit kembali.

Java 8

Question **2**

Incorrect

Mark 0.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

# MiniMax PrioQueue

Setelah diminta membuat Deque, kali ini kalian diminta untuk membuat **MiniMax PrioQueue**. MiniMax PrioQueue terdiri dari 3 kelas yaitu:

- **AbstractQueue**: Kelas abstrak yang berisi method-method yang akan digunakan oleh kelas Queue lainnya.
- **MinQueue**: Kelas yang mengimplementasikan queue dengan prioritas minimum.
- **MaxQueue**: Kelas yang mengimplementasikan queue dengan prioritas maksimum.

Kelas-kelas ini akan menggunakan generic type **T** yang merupakan subclass dari **Comparable<T>** sehingga kalian dapat menggunakan fungsi yang disediakan oleh **ArrayList** atau **Collections** untuk mengurutkan elemen-elemen di dalam queue.

Untuk menggunakan method dalam Collections, kalian perlu mengimportnya terlebih dahulu:

```
import java.util.Collections;
```

Berikut method yang dapat digunakan oleh ArrayList:

- **add** : Menambahkan elemen ke dalam list.
- **remove** : Menghapus elemen dari list.
- **get** : Mengambil elemen dari list.

Kumpulkan file **MinQueue.java**, **MaxQueue.java**, dan **AbstractQueue.java** dalam format ZIP dengan nama bebas. Pastikan file ZIP tersebut tidak dalam folder apapun, hanya langsung berisi 3 file tersebut.

## Attachments

Attachments: [attachments.zip](#)



Score: 0

Blackbox

Score: 0

Verdict: Compile error

Evaluator: Exact

No	Score	Verdict	Description
	0	Compile error	<div>javac Main.java AbstractQueue.java MaxQueue.java MinQueue.java MaxQueue.java:1: error: cannot find symbol import java.util.collection; ^ symbol: class collection location: package java.util MinQueue.java:1: error: cannot find symbol import java.util.collection; ^ symbol: class collection location: package java.util MaxQueue.java:12: error: cannot find symbol collection.sort(element); ^ symbol: variable collection location: class MaxQueue&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MaxQueue MaxQueue.java:26: error: cannot find symbol collection.sort(elements); ^ symbol: variable collection location: class MaxQueue&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MaxQueue MaxQueue.java:27: error: cannot find symbol value = elements[0]; ^ symbol: variable value location: class MaxQueue&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MaxQueue MaxQueue.java:27: error: array required, but List&lt;T&gt; found value = elements[0]; ^ where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MaxQueue MaxQueue.java:28: error: cannot find symbol elements.remove(value); ^ symbol: variable value location: class MaxQueue&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MaxQueue MaxQueue.java:29: error: cannot find symbol return T; ^ symbol: variable T location: class MaxQueue&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MaxQueue MaxQueue.java:43: error: cannot find symbol collection.sort(elements); ^ symbol: variable collection location: class MaxQueue&lt;T&gt; where T is a type- variable: T extends Comparable&lt;T&gt; declared in class MaxQueue MaxQueue.java:44: error: cannot find symbol value = elements.get(0); ^ symbol: variable value location: class MaxQueue&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MaxQueue MaxQueue.java:45: error: cannot find symbol return value; ^ symbol: variable value location: class MaxQueue&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MaxQueue MinQueue.java:11: error: cannot find symbol collection.sort(element); ^ symbol: variable collection location: class MinQueue&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MinQueue MinQueue.java:24: error: cannot find symbol collection.sort(elements); ^ symbol: variable collection location: class MinQueue&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MinQueue MinQueue.java:25: error: cannot find symbol value = elements.size - 1; ^ symbol: variable value location: class MinQueue&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class MinQueue MinQueue.java:25: error: cannot find symbol value = elements.size - 1; ^ symbol: variable size location: variable elements of type List&lt;T&gt; where T is a type-variable: T extends Comparable&lt;T&gt; declared in class AbstractQueue</div>

Question **3**  
Partially correct  
Mark 60.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

# Keranjang

Anda diberikan kelas **Barang** yang memiliki subclass **Baju** dan **Handphone**. Anda diminta mengimplementasikan kelas **Keranjang**, sebuah kelas yang menyimpan tipe data generic untuk kelas-kelas tersebut.

Kelas keranjang akan menyimpan barang (t) yang bertipe T dan tipe dari barang tersebut (type) yang disimpan dalam bentuk String.

Lengkapi metode dan kelas yang ada pada **Keranjang.java**

Hint: gunakan metode getClass().getName() untuk mencari tipe data

attachment : [Keranjang.java](#)

Java 8

 [Keranjang.java](#)

Score: 60

## Blackbox

Score: 60

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	0	Wrong answer	0.12 sec, 29.16 MB
2	0	Wrong answer	0.13 sec, 28.33 MB
3	20	Accepted	0.23 sec, 28.83 MB
4	20	Accepted	0.11 sec, 28.07 MB
5	20	Accepted	0.10 sec, 28.21 MB

[◀ Pra-Praktikum 5](#)

Jump to...

[Feedback Form ▶](#)