Started on	Thursday, 17 March 2022, 11:30 AM
State	Finished
Completed on	Thursday, 17 March 2022, 1:31 PM
Time taken	2 hours
Marks	300.00/300.00
Grade	**100.00** out of 100.00

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Pada soal ini, Anda akan mengimplementasikan kelas Time dalam bahasa Java. Lengkapi berkas Time.java berikut.

Java 8

Time.java

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.31 sec, 27.91 MB
2	10	Accepted	0.57 sec, 30.65 MB
3	10	Accepted	0.71 sec, 28.05 MB
4	10	Accepted	0.66 sec, 27.90 MB
5	10	Accepted	0.70 sec, 30.05 MB
6	10	Accepted	0.27 sec, 28.37 MB
7	10	Accepted	0.30 sec, 28.80 MB
8	10	Accepted	0.28 sec, 28.98 MB
9	10	Accepted	0.32 sec, 28.93 MB
10	10	Accepted	0.33 sec, 28.45 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Struktur data *Map* (atau yang biasa disebut *dictionary*) adalah sebuah struktur data Collection yang menyimpan elemen-elemennya dalam *tuple* **(key, value)**.

Salah satu contoh aplikasi dari struktur data *map* adalah untuk menyimpan **Phonebook.**

```
map["Anies"] = "940329502345"
map["Basuki"]  = "382901849102"
map["Jokowi"]  = "481904189208"
```

Selanjutnya apabila kita mengeksekusi `map.get("Jokowi")` akan mengembalikan `"481904189208"`, tetapi `map.get("AAA")` akan mengembalikan `NULL` (karena tidak terdapat pada *dictionary*).

Diberikan *interface* **Map** dan kelas **MapEntry** sebagai berikut.

```
// Map.java

public interface Map {
    /**
     * Menambahkan (key, value) ke dalam map
     * Melakukan overwrite jika sudah terdapat elemen dengan key yang sama.
     */
    public void set(String key, String value);

    /**
     * Mengembalikan value yang tersimpan untuk key tertentu pada map
     * Mengembalikan NULL apabila map tidak mengandung key masukan.
     */
    public String get(String key);

    /**
     * Menghitung jumlah elemen yang ada pada map
     */
    public int size();
}
```

```
// MapEntry.java

public class MapEntry {
    private final String key;
    private String value;

    public MapEntry (String key, String value) {
        this.key = key; this.value = value;
    }

    public String getKey() { return key; }
    public String getValue() { return value; }
    public void setValue(String x) { value = x; }
}
```

Tugas Anda adalah mengimplementasikan kelas **ListMap** yang mengimplementasikan *interface* **Map**. Kelas **ListMap** memiliki sebuah atribut ***first*** bertipe **ListMapEntry** yang memiliki kerangka program sebagai berikut.

```
// ListMapEntry.java

public class ListMapEntry extends MapEntry {
    private ListMapEntry next;

    public ListMapEntry(String key, String value) { } // next = null

    public ListMapEntry(String key, String value, ListMapEntry next) { }

    public ListMapEntry getNext() { }
    public void setNext(ListMapEntry next) { }
}
```

Pengujian pada *autograder* dilakukan hanya untuk *method-method* yang didefinisikan pada *interface* Map. Artinya, Anda bebas menambahkan *fungsi antara* yang mungkin dapat mempermudah pekerjaan kalian. Anda juga dibebaskan untuk menggunakan pendekatan *iteratif* maupun *rekursif* untuk pemrosesan *list* nya.

Kumpulkan [**ListMapEntry.java, ListMap.java**] yang disatukan menjadi satu file *zip*!

Note : gunakan Object.equals() untuk membandingkan dua buah object String

Java 8

Soal2.zip

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	12	Accepted	0.11 sec, 28.01 MB
2	12	Accepted	0.10 sec, 26.93 MB
3	12	Accepted	0.09 sec, 29.71 MB
4	12	Accepted	0.08 sec, 27.83 MB
5	13	Accepted	0.09 sec, 27.86 MB
6	13	Accepted	0.09 sec, 28.96 MB
7	13	Accepted	0.09 sec, 28.82 MB
8	13	Accepted	0.08 sec, 30.16 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Java sudah menyediakan beberapa interface untuk dapat diimplementasikan. Salah satunya adalah interface Comparable. Dengan mengimplementasikan interface Comparable, Anda dapat menggunakan library Java untuk sort, dan lain-lainnya.

Untuk mengimplemen interface Comparable, sebuah kelas harus mendefinisikan method `compareTo()`. Sebagai contoh, ini adalah kelas Mahasiswa yang mengimplementasikan Comparable.

```
import java.lang.Comparable;

class Mahasiswa implements Comparable<Mahasiswa> {
  private float ipk;

  public Mahasiswa(float ipk) {
    this.ipk = ipk;
  }

  public int compareTo(Mahasiswa m) {
    // compareTo mengembalikan:
    // 0 bila this sama dengan m
    // 1 bila this lebih dari m
    // -1 bila this kurang dari m
    if (this.ipk == m.ipk) {
      return 0;
    } else if (this.ipk > m.ipk) {
      return 1;
    } else {
      return -1;
    }
  }
}
```

Di soal ini, Anda diminta mengubah kelas mahasiswa di atas. Anda perlu menambahkan atribut `kodeJurusan` (int) dan `angkatan` (int). Lalu, buatlah juga method berikut:

- constructor yang menerima 3 parameter dengan urutan `(int kodeJurusan, int angkatan, float ipk)`
- getter untuk tiap atribut, dengan nama `getKodeJurusan`, `getAngkatan`, dan `getIpk`
- method `compareTo(Mahasiswa m)` yang membandingkan dua mahasiswa.
  - Mahasiswa A disebut kurang dari mahasiswa B bila:
    - kode jurusan mahasiswa A lebih kecil dari mahasiswa B
    - kode jurusan mahasiswa A sama dengan mahasiswa B, tapi angkatannya lebih muda (bilangannya lebih besar)
    - kode jurusan dan angkatan mahasiswa A sama dengan mahasiswa B, tapi ipk nya lebih besar
  - Mahasiswa A sama dengan mahasiswa B bila kode jurusan, angkatan, dan ipk kedua mahasiswa bernilai sama
  - Selain itu, mahasiswa A disebut lebih dari mahasiswa B

Kumpulkan **Mahasiswa.java**

Java 8

Mahasiswa.java

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.07 sec, 28.34 MB
2	10	Accepted	0.07 sec, 28.01 MB

No	Score	Verdict	Description
3	10	Accepted	0.07 sec, 28.48 MB
4	10	Accepted	0.07 sec, 28.22 MB
5	10	Accepted	0.07 sec, 27.82 MB
6	10	Accepted	0.09 sec, 28.34 MB
7	10	Accepted	0.15 sec, 28.40 MB
8	10	Accepted	0.09 sec, 27.99 MB
9	10	Accepted	0.08 sec, 29.04 MB
10	10	Accepted	0.08 sec, 28.01 MB

◄ Slide Minggu 9

Jump to...