Started on	Friday, 16 May 2025, 2:19 PM
State	Finished
Completed on	Friday, 16 May 2025, 4:46 PM
Time taken	2 hours 26 mins
Grade	**300.00** out of 300.00 (**100**%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Purry Levelling (Lagi ?!?!?)

Setelah berhasil menyelesaikan tugas besar pertama dengan membuat game *Purry Levelling* dalam bahasa C++, kini Anda akan beralih ke tugas berikutnya menggunakan bahasa pemrograman Java. Implementasikan kelas-kelas berikut dengan menerapkan konsep *inheritance* dan *interface* sederhana.

Spesifikasi

Interface `Damageable`
Memiliki satu method abstrak:

```
void takeDamage(int damage);
```

Interface `Interactable`
Memiliki satu method abstrak:

```
void interact();
```

Kelas abstrak `Unit`
- **Atribut**: `name` (*String*), `health` (*int*)
- **Constructor**: `Unit(String name, int health)`
- **Getter / Setter**: `getName()`, `setName(String)`, `getHealth()`, `setHealth(int)`
- **Method abstrak**: `String toString();`

Kelas `Character` (subclass `Unit`, mengimplementasikan `Damageable`)
- **Atribut tambahan**: `score` (*int*, default 0)
- **Constructors**:
  - *Default*: `Character()` → name="Player1", health=100, score=0
  - *User-defined*: `Character(String name, int health, int score)`
- **Method tambahan**:
  - Getter / Setter: `getScore()`, `setScore(int)`
  - `takeDamage(int)` — mengurangi `health`; jika ≤ 0, cetak `{name} has been defeated`
  - `increaseScore(int points)` — menambah `score`
- **Override** `toString()`: {name} [Health: {health}, Score: {score}]

Kelas `NPC` (subclass `Unit`, mengimplementasikan `Interactable`)
- **Atribut tambahan**: `dialogue` (*String*, default "*Hello there!*")
- **Constructors**:
  - *Default*: `NPC()` → name="NPC1", health=100, dialogue="Hello there!"
  - *User-defined*: `NPC(String name, int health, String dialogue)`
- **Method tambahan**:
  - Getter / Setter: `getDialogue()`, `setDialogue(String)`
  - `interact()` — mencetak `dialogue`
- **Override** `toString()`:
  {name} [Health: {health}, Dialogue: "{dialogue}"]

Contoh Penggunaan

```java
public class Main {
    public static void main(String[] args) {
        Character p1 = new Character();
        Character p2 = new Character("Alice", 50, 120);

        System.out.println(p1);
        System.out.println(p2);

        p2.takeDamage(30);
        p2.increaseScore(20);
        System.out.println(p2);

        p2.takeDamage(100);

        NPC npc = new NPC("Gandalf", 200, "You shall not pass!");
        System.out.println(npc);
        npc.interact();
    }
}
```

Output yang diharapkan

```
Player1 [Health: 100, Score: 0]
Alice [Health: 50, Score: 120]
Alice [Health: 20, Score: 140]
Alice has been defeated
Gandalf [Health: 200, Dialogue: You shall not pass!]
You shall not pass!
```

Pengumpulan

Kumpulkan berkas-berkas berikut dalam satu berkas .zip bernama `PurryLevellingV2.zip`:

- `Damageable.java`
- `Interactable.java`
- `Unit.java`
- `Character.java`
- `NPC.java`

Java 8 ⇕

📄 PurryLevellingV2.zip

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.10 sec, 28.39 MB
2	10	Accepted	0.07 sec, 29.06 MB
3	10	Accepted	0.08 sec, 28.74 MB
4	10	Accepted	0.09 sec, 28.31 MB
5	10	Accepted	0.09 sec, 28.00 MB
6	10	Accepted	0.10 sec, 27.84 MB
7	10	Accepted	0.07 sec, 27.88 MB
8	10	Accepted	0.07 sec, 28.85 MB

No	Score	Verdict	Description
9	10	Accepted	0.06 sec, 28.94 MB
10	10	Accepted	0.07 sec, 28.31 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Find My

Pak Asep kehilangan di kantor pusat (HQ) O.W.C.A, bantu Pak Asep untuk mengimplementasikan hape beliau dengan file bernama **IPhone.java,** diberikan juga interface **Trackable.java** dan **Device.java** yang masing-masing merupakan interface dan kelas abstrak dari **IPhone.java,** nama dari kelas iPhone tentunya "**IPhone"** tanpa tanda petik.
Berikut templatenya IPhone.java. Kumpulkan IPhone.javanya saja

Java 8 ⬍

⚙ IPhone.java

Score: 100

Blackbox
Score: 100
Verdict: Accepted
Evaluator: Exact

No	Score	Verdict	Description
1	12.5	Accepted	0.06 sec, 28.16 MB
2	12.5	Accepted	0.06 sec, 26.16 MB
3	12.5	Accepted	0.07 sec, 28.76 MB
4	12.5	Accepted	0.06 sec, 28.96 MB
5	12.5	Accepted	0.06 sec, 28.79 MB
6	12.5	Accepted	0.07 sec, 28.92 MB
7	12.5	Accepted	0.06 sec, 28.09 MB
8	12.5	Accepted	0.06 sec, 28.20 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Implementasikan kelas **SmartDoorLock** yang akan menjadi bagian dari sistem rumah pintar. Kelas ini harus menggunakan *inheritance* dan *interface* yang telah disediakan.

Sistem ini menggunakan beberapa komponen:

1. `Controllable.java` - Interface yang mendefinisikan kemampuan perangkat untuk dikendalikan (dihidupkan/dimatikan)
2. `EnergyMonitored.java` - Interface untuk memantau konsumsi energi perangkat
3. `SecureDevice.java` - Interface yang mendefinisikan fitur keamanan perangkat
4. `SmartDevice.java` - Kelas abstrak dasar untuk semua perangkat pintar
5. `SecurityDevice.java` - Kelas abstrak untuk perangkat keamanan

Anda harus mengimplementasikan kelas **SmartDoorLock** yang mewarisi **SecurityDevice** dan mengimplementasikan **EnergyMonitored**.

Kumpulkan **SmartDoorLock.java** (ini ada templatenya)

Java 8 ⬍

⚙ SmartDoorLock.java

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	30	Accepted	0.07 sec, 28.80 MB
2	30	Accepted	0.06 sec, 28.41 MB
3	10	Accepted	0.06 sec, 28.50 MB
4	10	Accepted	0.06 sec, 26.17 MB
5	10	Accepted	0.06 sec, 29.13 MB
6	10	Accepted	0.06 sec, 27.98 MB

◄ Praktikum 4

Jump to... ⬍

Tutorial 5 ►