
Started on Thursday, 14 April 2022, 11:00 AM

State Finished

Completed on Thursday, 14 April 2022, 12:08 PM

Time taken 1 hour 8 mins

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

Lengkapilah kelas Car yang telah diberikan berikut ini dengan Exception yang sesuai.

Silakan download kelas Car dan exception yang telah disediakan di [sini](#).

Kumpulkan file **Car.java**!

```
public class Car {

    private int fuel;
    private int fuelCapacity;
    private boolean isRunning;

    public Car() {
        this.fuel = 0;
        this.fuelCapacity = 100;
        this.isRunning = false;
    }

    // Diasumsikan fuel > 0 dan fuelCapacity > 0
    public Car(int fuel, int fuelCapacity) {
        this.fuel = fuel;
        this.fuelCapacity = fuelCapacity;
        this.isRunning = false;
    }

    // TODO
    // Mengurangi fuel sebanyak (km) dengan km > 0
    // Jika km tidak valid, lemparkan InvalidInputException
    // Jika fuel kurang dari km, lemparkan NotEnoughFuelException dengan jumlah fuel yang diperlukan sebagai keterangan
    exception
    // Jika mobil belum dinyalakan, lemparkan CarNotStartedException
    public void drive(int km) {

        this.fuel -= km;
    }

    // TODO
    // Mengisi fuel sebanyak (fillFuel) dengan fuel > 0
    // Jika fuel tidak valid, lemparkan InvalidInputException
    // Jika fuel yang diisi melebihi kapasitas, lemparkan FuelOverflowException dengan jumlah fuel yang kelebihan sebagai
    keterangan exception
    public void fillCar(int fillFuel) {

        this.fuel += fillFuel;
    }

    // Memutar starter mobil
    public void turnIgnition() {
        this.isRunning = !this.isRunning;
        if (this.isRunning) {
            system.out.println("Car is now started.");
        } else {
            system.out.println("Car is now stopped.");
        }
    }

    @Override
    public String toString() {
        String run = (this.isRunning) ? "is running" : "is stopped";
        return "Fuel: " + this.fuel + " | FuelCapacity: " + this.fuelCapacity + " | " + run;
    }
}
```

Java 8

 [Car.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.17 sec, 28.14 MB
2	10	Accepted	0.47 sec, 28.83 MB
3	10	Accepted	0.86 sec, 29.44 MB
4	10	Accepted	0.67 sec, 28.57 MB
5	10	Accepted	0.61 sec, 28.41 MB
6	10	Accepted	0.52 sec, 28.36 MB
7	10	Accepted	0.20 sec, 29.01 MB
8	10	Accepted	0.41 sec, 27.88 MB
9	10	Accepted	0.59 sec, 28.06 MB
10	10	Accepted	0.54 sec, 28.19 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit

1 s

Memory limit

64 MB

Di game Minecraft Aether Realms, ada sebuah jenis kartu Character. Kartu ini menggambarkan karakter yang dapat menyerang lawan. Berikut kelas [Character](#).

Kelas ini memiliki *responsibility* yakni untuk merepresentasikan kartu yang ada. Padahal, kartu karakter dapat di-summon ke arena. Pada saat di arena, karakter memiliki level dan exp. Karakter juga dapat ditingkatkan levelnya. Selain itu, karakter pada arena juga mungkin memiliki spell yang sedang aktif, berikut adalah kelas [Spell](#). Ketika sudah di-summon, karakter memiliki interface lain lagi, yakni [ISummoned](#):

```
import java.util.List;

// ISummoned.java
interface ISummoned {
    // getter
    int getLevel();
    int getExp();

    // menambahkan spell ke dalam daftar spell aktif
    void addSpell(Spell s);

    // mengembalikan daftar spell aktif
    List getActiveSpells();

    // meningkatkan level karakter sebanyak 1
    // mereset xp ke 0
    // meningkatkan baseAtk dan baseHp sebanyak attackUp dan healthUp
    void levelUp();

    // menggambar kartu ke layar
    void render();
}
```

Untuk tidak merusak Single Responsibility Principle, Anda ingin membuat kelas adapter yang menyambungkan interface Summoned ke karakter. Buatlah kelas SummonedCharacter yang:

- Memiliki konstruktor yang menerima argumen berturut-turut:
 - Sebuah objek karakter yang disummon
 - integer yang merupakan level dari karakter saat ini
 - integer yang merupakan exp dari karakter saat ini
- Mencatat level dan exp karakter di arena sekarang
- Meningkatkan level
- Mencatat spell yang sedang aktif
- Menambahkan spell ke karakter
- Mengimplemen interface ISummoned
- Method render akan menuliskan pesan ke layar, diikuti baris baru.

```
Nama: {{nama}}
Level: {{level}}
Exp: {{exp}}
Atk: {{baseAtk}}
Hp: {{baseHp}}
Memiliki {{jumlah spell aktif}} spell aktif:
- {{nama spell 1}}
- {{nama spell 2}}
- ...
- {{nama spell n}}
```

Kumpulkan **SummonedCharacter.java**

Notes: untuk mempermudah pengerjaan, Spell hanya bertipe permanen dan hanya memiliki atribut nama. Mungkin Spell yang sebenarnya bisa memiliki durasi dan atribut lain? Duh.

Java 8

 [SummonedCharacter.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.22 sec, 28.51 MB
2	20	Accepted	0.30 sec, 28.87 MB
3	20	Accepted	0.50 sec, 30.55 MB
4	20	Accepted	0.72 sec, 27.92 MB
5	20	Accepted	0.59 sec, 28.02 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit

1 s

Memory limit

64 MB

Kalian pasti tahu tentang Gofud, kan?

Ketika ada pesanan yang masuk, tentunya restoran akan mengabari *driver* di sekitar restoran untuk mengambil dan mengantarkan pesanan seseorang. Oleh karena itu, dapat diterapkan sebuah sistem *publish-subscribe* sederhana untuk mensimulasikan pengiriman pesan kepada seluruh *driver* yang dekat dengan lokasi!

Implementasikan dan kumpulkan **GofudServer.java**! Baca juga kode lainnya untuk mengetahui konteks dan cara kerja dari *design pattern publish-subscribe*.

Berikut ada beberapa file yang diperlukan yang menjadi dasar untuk mengimplementasikan pola *publish-subscribe*. Tentunya, file-file ini bisa di-*extend* sesuai kebutuhan.

1. [Event.java](#)
2. [EventBroker.java](#)
3. [Publisher.java](#)
4. [Subscriber.java](#)

Untuk mengimplementasikan **GofudServer.java**, telah diimplementasikan beberapa kelas tambahan:

1. [OrderEvent.java](#), diturunkan dari interface **Event**
2. [GofudServer.java](#), diturunkan dari interface **EventBroker**
3. [GofudDriver.java](#), diturunkan dari interface **Subscriber**
4. [GofudRestaurant.java](#), diturunkan dari interface **Publisher**

Java 8

 [GofudServer.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	14	Accepted	0.15 sec, 27.88 MB
2	14	Accepted	0.13 sec, 30.75 MB
3	14	Accepted	0.12 sec, 28.63 MB
4	14	Accepted	0.22 sec, 28.59 MB
5	14	Accepted	0.11 sec, 27.89 MB
6	14	Accepted	0.12 sec, 28.45 MB
7	16	Accepted	0.12 sec, 27.95 MB

◀ Slide Minggu 13

Jump to...

