

Dasar-Dasar Pemrograman 2

Lab 07

Abstract Class



FAKULTAS
ILMU
KOMPUTER

Pada tutorial/lab sebelumnya, kalian telah mempelajari salah satu konsep pemrograman dimana satu objek memperoleh semua properti dan perilaku objek induk yang dinamakan Inheritance. Kali ini, kita akan mempelajari salah satu *class* pada java yang fungsinya sedikit berbeda dengan *class* Java pada umumnya yaitu *abstract class*.

Abstract Classes

Abstract class merupakan jenis class yang berperan sebagai ‘kerangka dasar’ bagi class turunannya. Sebuah abstract class tidak bisa diinstansiasi (tidak bisa dibuat menjadi objek).

Abstract method adalah method yang belum diimplementasikan di abstract class, tetapi **wajib di-override di class yang meng-extend abstract class tersebut**. Sebuah abstract class boleh memiliki abstract method dan concrete method. Method-method di abstract class dapat berupa public, protected, dan private method. Dalam mendeklarasi sebuah abstract method, method dibuat tanpa implementasi (tanpa kurung kurawal, dan diakhiri dengan titik koma) seperti contoh dibawah ini.

```
public abstract int getPrice();
```

Sebuah abstract class digunakan saat *superclass* dan *subclass* memiliki method yang harus berbeda tiap subclass (abstract) tetapi ada method yang sama untuk tiap subclass (concrete). Perlu diingat bahwa jika subclass dari abstract class tidak mengimplementasikan semua abstract method yang ada di abstract class, maka subclass tersebut harus dideklarasikan abstract juga. Abstract class juga dapat memiliki variabel non-static dan non-final.

Example of Abstract Classes

Mari kita coba mengimplementasikan abstract class pada program sederhana berikut ini.

```
abstract class KopiPacil {  
  
    public abstract String getRecipe();  
  
    public abstract int getPrice();  
  
    public void makeDrink(){  
        System.out.println("~ brew brew brew ~");  
    }  
}  
  
class CafeLatte extends KopiPacil{  
    private int price;  
    private String recipe = "Kopi sachet dan susu";  
  
    CafeLatte(int price){  
        this.price = price;  
    }  
  
    public String getRecipe(){  
        return this.recipe;  
    }  
  
    public int getPrice(){  
        return this.price;  
    }  
}
```

Jika kita perhatikan contoh implementasi kode diatas, abstract class KopiPacil memiliki dua abstract method yaitu getRecipe() dan getPrice(). Kedua method tersebut diimplementasikan di class CafeLatte, yang mana class tersebut

meng-extend abstract class KopiPacil. Jika ternyata CafeLatte tidak mengimplementasikan semua abstract method milik KopiPacil, maka class CafeLatte harus dideklarasikan sebagai abstract juga.

Soal Lab 07

Lost in Pacilgeon



Source: <https://soundcloud.com/vinter-grotte-1/the-evil-dark-dungeon>

Setelah berusaha membuat program kunjungan virtual ke Fasilkom UI, kamu pun kelelahan dan mengambil waktu untuk beristirahat. Tanpa kamu sadari, kamu telah masuk ke alam mimpi dan bertemu dengan seseorang yang sangat kamu kenal, Angewomon. “Mengapa kamu ada di sini?” Tanya Angewomon keheranan. Situasi di sekitar sangat gelap, hanya terdapat obor yang menggantung di dinding, menyinari sedikit dari jalan yang dapat kamu lihat.

“Tempat apa ini?” Kamu bertanya kepada Angewomon.

“Ini adalah Pacilgeon, sebuah dungeon yang dibuat oleh programmer dunia lain untuk menjebak orang-orang yang memasukinya. Aku sendiri sudah terjebak beberapa hari di sini dan sangat kesulitan untuk mencari jalan keluar karena ada banyak monster yang menyerang jika aku bergerak.”

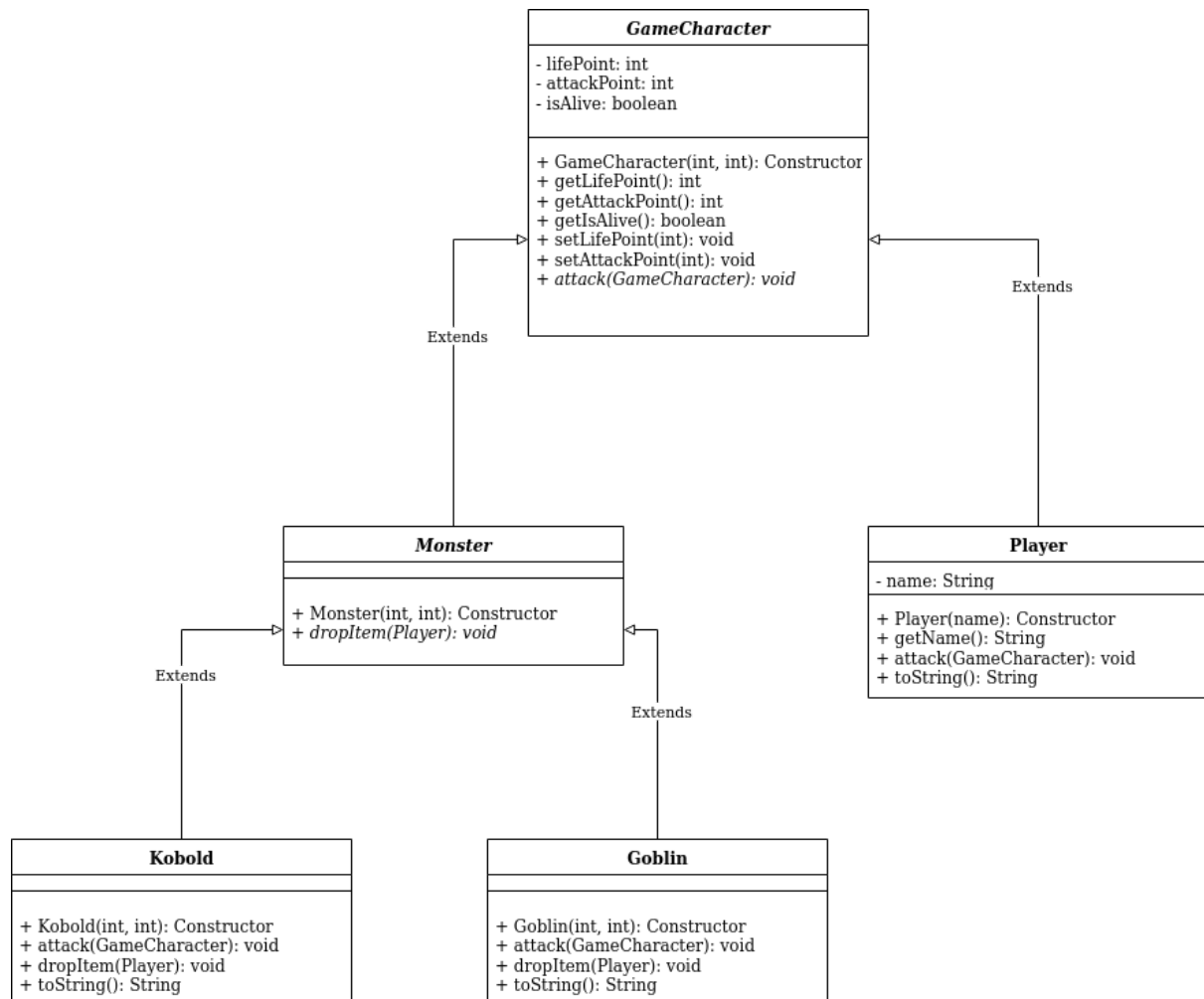
“Bagaimana cara untuk menemukan jalan keluar?” Kamu ingin cepat-cepat keluar dari tempat ini.

Angewomon menjawab, “Aku telah berkeliling dan mendapatkan sebuah peta ajaib yang menunjukkan lokasi kita sekarang, monster di sekitar kita, dan juga letak pintu keluar. Tetapi, bagian yang sulitnya adalah menyerang monster-monster tersebut. Jika kita berhasil mengalahkan monster, monster itu akan menjatuhkan item berupa *potion*, tetapi *potion* ini dapat memberi kita keuntungan atau kerugian terhadap lifePoint dan attackPoint kita. Jadi, kita harus berhati-hati dalam melang-.”

Tiba-tiba saja, seekor monster menyerang! Dan... kamu terbangun dengan perasaan kaget. Kamu menyadari bahwa itu semua adalah mimpi. Tetapi, kamu masih bisa mengingat mimpi itu dan memutuskan untuk membuat simulasi Pacilgeon dalam sebuah program.

Spesifikasi Program

Class Diagram:



Komponen-komponen yang harus dibuat:

- **GameCharacter (Abstract)**

- **Constructor:**

- Parameter dari constructor adalah `lifePoint` dan `attackPoint`. Constructor akan meng-assign nilai atribut `lifePoint` dan `attackPoint` sesuai parameter dan nilai `isAlive` sama dengan `true` (asumsikan `GameCharacter` selalu hidup saat di-construct).

- Implementasikan method-method lainnya sesuai UML diagram (kecuali method-method abstract).
 - Catatan untuk atribut lifePoint:
 - Atribut lifePoint memiliki nilai maksimal 100. Jika bernilai kurang dari 1, maka isAlive menjadi false.
- **Monster (Abstract)**
 - **Constructor:**
 - Parameter dari constructor adalah lifePoint dan attackPoint. Constructor akan meng-assign nilai atribut lifePoint dan attackPoint sesuai parameter dan nilai isAlive sama dengan *true* (asumsikan Monster selalu hidup saat di-construct).
 - Implementasikan sesuai konsep OOP.
 - Implementasikan method-method lainnya sesuai UML diagram (kecuali method-method abstract).
- **Player**
 - **Constructor:**
 - Parameter dari constructor adalah name. Constructor akan meng-assign nilai atribut name sesuai parameter tersebut dan nilai awal dari lifePoint sebesar 100, attackPoint sebesar 15, dan nilai isAlive sama dengan *true*.
 - Implementasikan sesuai konsep OOP.
 - Method **attack(GameCharacter monster):**
 - Serang monster dengan mengurangi lifePoint sebanyak attackPoint dari player, dan keluarkan pesan seperti berikut yang menunjukkan banyak lifePoint yang dikurangi. Contoh:


```
Monster received 15 point(s) damage from Dekdepe.
```
 - Method **toString():**
 - Keluarkan status dari Player saat method ini dipanggil. Contoh:


```
Player | name: Dekdepe, lifePoint: 46, attackPoint: 2
```
- **Kobold**
 - **Constructor:**
 - Parameter dari constructor adalah lifePoint dan attackPoint. Constructor akan meng-assign nilai atribut lifePoint dan attackPoint sesuai parameter dan nilai isAlive sama dengan *true* (asumsikan Kobold selalu hidup saat di-construct).
 - Implementasikan sesuai konsep OOP.
 - Method **attack(GameCharacter player):**
 - Serang player dengan mengurangi lifePoint sebanyak attackPoint dari Kobold, dan keluarkan pesan seperti berikut yang menunjukkan banyak lifePoint yang dikurangi. Contoh:

```
Dekdepe received 1 point(s) damage from Kobold.
```

- Method **dropItem(Player player)**:

- Ubah status lifePoint dari player, dan keluarkan pesan seperti berikut yang menunjukkan banyak lifePoint yang ditambahkan. Nilai yang ditambahkan tersebut harus acak dengan nilai dari -10 s/d 10 (inklusif). Contoh:

```
Kobold has been defeated. It dropped a life potion.  
The potion added 9 point(s) to your lifePoint
```

- Method **toString()**:

- Keluarkan status dari Kobold saat method ini dipanggil. Contoh:

```
Kobold | lifePoint: 12, attackPoint: 10
```

- **Goblin**

- **Constructor:**

- Parameter dari constructor adalah lifePoint dan attackPoint. Constructor akan meng-assign nilai atribut lifePoint dan attackPoint sesuai parameter dan nilai isAlive sama dengan *true* (asumsikan Goblin selalu hidup saat di-construct).
- Implementasikan sesuai konsep OOP.

- Method **attack(GameCharacter player)**:

- Serang player dengan mengurangi lifePoint sebanyak attackPoint dari Goblin ditambah dengan suatu angka acak dari 0 s/d 4 (inklusif) karena goblin punya faktor keberuntungan yang lebih tinggi, dan keluarkan pesan seperti berikut yang menunjukkan banyak lifePoint yang dikurangi. Contoh:

```
Dekdepe received 1 point(s) damage from Goblin.
```

- Method **dropItem(Player player)**:

- Ubah status attackPoint dari player, dan keluarkan pesan seperti berikut yang menunjukkan banyak attackPoint yang ditambahkan. Nilai yang ditambahkan tersebut harus acak dengan nilai dari -10 s/d 10 (inklusif). Contoh:

```
Goblin has been defeated. It dropped an attack potion.  
The potion added 8 point(s) to your attackPoint.
```

- Method **toString()**:

- Keluarkan status dari Goblin saat method ini dipanggil. Contoh:

```
Goblin | lifePoint: 12, attackPoint: 10
```

Simulasi telah disediakan di file **Pacilgeon.java**.

Contoh Masukan dan Keluaran 1

(Tulisan yang di-highlight biru adalah user input)

```
Row size: 3
Column size: 3
Input map:
K: Kobold, G: Goblin
...
KKK
GGG
Player name: Dekdepe

Player stats:
Player | name: Dekdepe, lifePoint: 100, attackPoint: 15
Map: (K: Kobold, G: Goblin, o: player is here)
o..
KKK
GGG
Current position | row: 0, col: 0
Move (W / A / S / D): D
=====

Nothing to do here.

=====

Player stats:
Player | name: Dekdepe, lifePoint: 100, attackPoint: 15
Map: (K: Kobold, G: Goblin, o: player is here)
.o.
KKK
GGG
Current position | row: 0, col: 1
Move (W / A / S / D): D
=====

Nothing to do here.

=====

Player stats:
Player | name: Dekdepe, lifePoint: 100, attackPoint: 15
Map: (K: Kobold, G: Goblin, o: player is here)
..o
KKK
GGG
Current position | row: 0, col: 2
Move (W / A / S / D): S
=====
```


Battle with a monster!

Kobold | lifePoint: 22, attackPoint: 0

Monster received 15 point(s) damage from Dekdepe.

Dekdepe received 0 point(s) damage from Kobold.

Monster received 15 point(s) damage from Dekdepe.

Kobold has been defeated. It dropped a life potion.

The potion added -7 point(s) to your lifePoint.

=====

Player stats:

Player | name: Dekdepe, lifePoint: 93, attackPoint: 15

Map: (K: Kobold, G: Goblin, o: player is here)

...

KKo

GGG

Current position | row: 1, col: 2

Move (W / A / S / D): S

=====

Battle with a monster!

Goblin | lifePoint: 18, attackPoint: 5

Monster received 15 point(s) damage from Dekdepe.

Dekdepe received 5 point(s) damage from Goblin.

Monster received 15 point(s) damage from Dekdepe.

Goblin has been defeated. It dropped an attack potion.

The potion added -6 point(s) to your attackPoint.

=====

Congratulations! You have finished the Pacilgeon :D

Contoh Masukan dan Keluaran 2

Row size: 3

Column size: 3

Input map:

K: Kobold, G: Goblin

.KG

GKG

G.K

Player name: Dekdepe

Player stats:

Player | name: Dekdepe, lifePoint: 100, attackPoint: 15

Map: (K: Kobold, G: Goblin, o: player is here)

```

oKG
GKG
G.K
Current position | row: 0, col: 0
Move (W / A / S / D): S
=====

Battle with a monster!
Goblin | lifePoint: 74, attackPoint: 2

Monster received 15 point(s) damage from Dekdepe.
Dekdepe received 5 point(s) damage from Goblin.
Monster received 15 point(s) damage from Dekdepe.
Dekdepe received 3 point(s) damage from Goblin.
Monster received 15 point(s) damage from Dekdepe.
Dekdepe received 5 point(s) damage from Goblin.
Monster received 15 point(s) damage from Dekdepe.
Dekdepe received 2 point(s) damage from Goblin.
Monster received 15 point(s) damage from Dekdepe.
Goblin has been defeated. It dropped an attack potion.
The potion added -9 point(s) to your attackPoint.

=====

Player stats:
Player | name: Dekdepe, lifePoint: 85, attackPoint: 6
Map: (K: Kobold, G: Goblin, o: player is here)
.KG
oKG
G.K
Current position | row: 1, col: 0
Move (W / A / S / D): S
=====

Battle with a monster!
Goblin | lifePoint: 87, attackPoint: 1

Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 1 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 4 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 4 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 5 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 5 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 4 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.

```

Dekdepe received 3 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 2 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 1 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 4 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 3 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 5 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 2 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Dekdepe received 5 point(s) damage from Goblin.
Monster received 6 point(s) damage from Dekdepe.
Goblin has been defeated. It dropped an attack potion.
The potion added -9 point(s) to your attackPoint.

=====
Player stats:
Player | name: Dekdepe, lifePoint: 37, attackPoint: -3
Map: (K: Kobold, G: Goblin, o: player is here)
.KG
.KG
o.K
Current position | row: 2, col: 0
Move (W / A / S / D): D

=====
Nothing to do here.

=====
Player stats:
Player | name: Dekdepe, lifePoint: 37, attackPoint: -3
Map: (K: Kobold, G: Goblin, o: player is here)
.KG
.KG
.oK
Current position | row: 2, col: 1
Move (W / A / S / D): D

=====
Battle with a monster!
Kobold | lifePoint: 52, attackPoint: 14

Monster received -3 point(s) damage from Dekdepe.
Dekdepe received 14 point(s) damage from Kobold.

```
Monster received -3 point(s) damage from Dekdepe.
Dekdepe received 14 point(s) damage from Kobold.
Monster received -3 point(s) damage from Dekdepe.
Dekdepe received 14 point(s) damage from Kobold.
```

```
=====
```

```
You have been defeated. Try again next time.
```

Catatan Mengenai Input

- Pintu masuk ada di (baris, kolom) = (0, 0).
- Jika N adalah ukuran baris, dan M adalah ukuran kolom, maka pintu keluar ada di (baris, kolom) = (N - 1, M - 1).
- Ukuran Pacilgeon lebih dari 1 satuan luas.
- Asumsikan di pintu masuk tidak ada monster.
- Dalam input map, selain 'G' dan 'K' akan tidak dianggap.

Komponen Penilaian

- 25% Implementasi class GameCharacter
- 25% Implementasi class Player
- 20% Implementasi abstract class Monster
- 20% Implementasi class Kobold, dan Goblin
- 10% Dokumentasi dan kerapian kode

Revisi

- **Revisi 1: (Highlight berwarna kuning)**
Perubahan pada contoh masukan dan keluaran 2. Hal utama yang berubah adalah *damage* dari goblin dapat berubah-ubah.
- **Revisi 2: (Highlight berwarna hijau)**
Perubahan pada class diagram, method `setLifePoint()` pada GameCharacter memiliki parameter sehingga diubah menjadi `setLifePoint(int)`.

Kumpulkan berkas .java yang telah di-zip dengan format penamaan seperti berikut.

Lab07_[Kelas]_[KodeAsdos]_[NPM]_[NamaLengkap].zip

Contoh:

Lab07_A_LN_1234567890_DekDepe.zip