
Started on Thursday, 30 March 2023, 11:05 AM

State Finished

Completed on Thursday, 30 March 2023, 12:15 PM

Time taken 1 hour 10 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

Anda diminta untuk membuat kelas-kelas animal untuk memodelkan hewan di suatu kerajaan hewan. Hewan dimodelkan sebagai kelas **Animal** dengan berbagai jenis hewan merupakan turunan dari kelas **Animal**. Saat ini, kerajaan hewan hanya memiliki 1 jenis hewan yang merupakan raja para hewan yaitu **Gajah** yang dimodelkan sebagai kelas turunan **Gajah** dengan nama yang sama.

Pada soal ini, anda diminta untuk mengimplementasi kelas **Animal** yang merupakan sebuah *abstract class* yang memiliki 2 buah atribut:

1. **numberOfLegs**, menyimpan banyak kaki dari hewan dalam tipe data **int**
2. **children**, menyimpan banyak anak yang dimiliki hewan tersebut dengan tipe data **int** dengan default nilai 0.

Kelas **Animal** menerapkan interface **Comparable** dan juga memiliki beberapa metode yang memiliki peran berikut:

1. **getNumberOfLegs**, getter dari atribut **numberOfLegs**
2. **getNumberOfChildren**, getter dari atribut **children**
3. **increaseChild**, method **void** yang meningkatkan nilai atribut **children** dengan nilai argumen **inc** bertipe **int**
4. **getAnimalPower**, method abstrak dari kelas **Animal** yang mengembalikan kekuatan dari hewan tersebut bertipe **int**.
5. **compareTo**, method untuk membandingkan animal power antara dua **animal** dengan menerima 1 buah parameter animal bertipe **Animal**. Hasil method ini bertipe **int**

Kelas **Animal** memiliki 2 kelas turunan dengan ketentuan sebagai berikut:

1. **Gajah**

- Merupakan turunan dari kelas **Animal**
- Spesifikasi Atribut:
 - Memiliki atribut **basePower**, yang menyimpan kekuatan dasar dari hewan tersebut dengan tipe **int**.
 - Memiliki atribut **age**, yang menyimpan umur dari gajah dalam tipe **int**
- Spesifikasi Konstruktor:
 - Konstruktor menerima **basePower** dan **age** secara berturut-turut
 - Gajah memiliki 4 buah kaki (**numberOfLegs=4**)
- Spesifikasi Metode:
 - **getAge**, *getter* dari atribut **age**
 - **getAnimalPower**, implementasi metode abstrak **getAnimalPower** dari kelas *parent* **Animal** yang mengembalikan kekuatan dari hewan ini. Formulasnya adalah: $3 * (basePower + age + children)$

2. **Elang**

- Merupakan turunan dari kelas **Animal**
- Spesifikasi Atribut:
 - Memiliki atribut **basePower**, yang menyimpan kekuatan dasar dari hewan tersebut dalam tipe **int**.
 - Memiliki atribut **jumlahTelur**, yang menyimpan berapa banyak telur yang dimiliki oleh Elang tersebut dalam tipe **int**
- Spesifikasi Konstruktor:
 - Konstruktor menerima **basePower**.
 - Elang memiliki 2 buah kaki (**numberOfLegs=2**)
 - Pada awalnya, elang tidak memiliki telur sama sekali.
- Spesifikasi Metode:
 - **getJumlahTelur**, *getter* dari atribut **jumlahTelur**
 - **bertelur**, method **void** untuk menambahkan jumlah telur Elang dengan 1 telur lagi.
 - **getAnimalPower**, implementasi metode abstrak **getAnimalPower** dari kelas *parent* **Animal** yang mengembalikan kekuatan dari hewan ini. Elang harus melindungi telurnya namun dapat dibantu oleh anaknya. Formula kekuatannya adalah: $basePower * children - jumlahTelur$

Lengkapi [Animal.java](#), [Elang.java](#), dan [Gajah.java](#) dan kumpulkan sebagai **Animal.zip**.

Java 8

 [Animal.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	12	Accepted	0.17 sec, 29.84 MB
2	12	Accepted	0.26 sec, 29.00 MB
3	12	Accepted	0.11 sec, 29.87 MB
4	12	Accepted	0.13 sec, 26.79 MB
5	12	Accepted	0.21 sec, 27.12 MB
6	12	Accepted	0.13 sec, 29.83 MB
7	12	Accepted	0.11 sec, 30.05 MB
8	16	Accepted	0.12 sec, 28.13 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Diberikan tiga file, [IPaymentMethod.java](#), [DebitCard.java](#), dan [CreditCard.java](#), yang mendefinisikan suatu interface metode pembayaran IPaymentMethod, dan dua kelas yang mengimplementasikan interface tersebut.

Lengkapilah [DebitCard.java](#) dan [CreditCard.java](#) sehingga memiliki implementasi method yang tepat.

Kumpulkan [DebitCard.java](#) dan [CreditCard.java](#) dalam suatu file zip.

Java 8

 [PaymentCards.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.38 sec, 27.87 MB
2	10	Accepted	0.17 sec, 28.41 MB
3	10	Accepted	0.12 sec, 28.50 MB
4	10	Accepted	0.11 sec, 28.90 MB
5	10	Accepted	0.09 sec, 26.16 MB
6	10	Accepted	0.16 sec, 27.97 MB
7	10	Accepted	0.10 sec, 28.85 MB
8	10	Accepted	0.14 sec, 27.89 MB
9	10	Accepted	0.09 sec, 27.91 MB
10	10	Accepted	0.09 sec, 27.86 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit

1 s

Memory limit

64 MB

Belanja

Margaretta, pacar Albert, sering kali meminta bantuan pacarnya untuk pergi membelikan belanja agar ia dapat memasak. Namun, permasalahannya, kedua pasangan ini sering lupa. Dari Margaretta, ia sering sekali lupa merk/nama barang yang telah dibelinya. Dari Albert, ia tidak bisa mengingat belanjaan yang diminta untuk dibeli.

Oleh karena itu, sebagai **bantulah mereka untuk membuat program daftar belanja**.

Yang harus kalian buat:

Barang.java

- memiliki **constructor** dengan parameter **Nama (String)** dan **Amount (Integer)**.
- melakukan **override** terhadap metode toString(), hasil yang dikeluarkan adalah **x y** dengan x adalah amount dan y adalah nama barang. Contoh: 100 apel.

DaftarBelanja.java

- memiliki field bernama **listBelanja** yang bertipe Barang[]
- memiliki **constructor** dengan **parameter Max (integer)** yang akan digunakan untuk membuat array Barang[] sebesar Max.
- memiliki fungsi belanja(int lorong, String keterangan) yang akan menambah barang dengan nama: **keterangan (lorong x)** sejumlah 1. Contoh: kotak merah (lorong 10) jika dipanggil belanja(10, "kotak merah").
- memiliki fungsi belanja(int lorong, int qty, String keterangan) yang akan berlaku sama seperti nomor 3 hanya saja jumlahnya menjadi sebesar **qty**.
- memiliki fungsi belanja(String barang) yang akan menambahkan barang dengan nama: **barang** sejumlah 1.
- memiliki fungsi belanja(String barang, int qty) yang berlaku seperti nomor 5 hanya saja barang sejumlah **qty**.
- memiliki fungsi print() yang akan melakukan print barang berurut dari nomor 1 dan memanggil fungsi toString() dari barang. Format: **%d. %s** dimana **%d** adalah nomor urut dari 1 sampai N dan **%s** adalah hasil barang toString(). Contoh:
 - 100 shampo heri
 - 5 kotak merah tulisannya tailor kalau ga salah (lorong 25)
- apabila daftar belanja sudah penuh, barang belanja yang ditambahkan tidak dianggap/dibuang.

Kumpulkan **Barang.java** dan **DaftarBelanja.java** dalam suatu zip!

Java 8

 [Belanja.zip](#)

Score: 60

Blackbox

Score: 60

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.42 sec, 27.88 MB
2	10	Accepted	0.82 sec, 29.12 MB
3	10	Accepted	0.83 sec, 28.00 MB
4	10	Accepted	0.67 sec, 28.38 MB
5	10	Accepted	0.51 sec, 27.85 MB
6	10	Accepted	0.33 sec, 29.18 MB

