Started on	Monday, 21 April 2025, 8:12 AM
State	Finished
Completed on	Monday, 21 April 2025, 9:50 AM
Time taken	1 hour 37 mins
Grade	**194.00** out of 300.00 (**65**%)

Question **1**

Partially correct

Mark 84.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Toko Sepeda Esempe

Anda adalah seorang pedagang sepeda grosir bernama Toko Sepeda Esempe yang berlokasi di Magelang. Awalnya, bisnis berjalan dengan lancar dengan pencatatan berbasis kertas. Namun, belakangan ini, Anda mendapat klien besar: seorang politisi yang suka membagi-bagikan sepeda untuk keperluan pencitraan. Anda mulai merasa kewalahan dalam mengelola bisnis. Sebagai seorang ahli dalam pemrograman OOP, Anda memutuskan membuat program dasar untuk memudahkan pengelolaan toko.

Setiap Bike memiliki kriteria diskonnya masing-masing. Perhatikan logika implementasi untuk masing-masing Bike. Sejumlah Bike juga memiliki fitur khusus. Berhati-hatilah dalam mengoperasikan angka.

Implementasikan berkas Bike.java, Bim.java, Ngeng.java, Ultra.java dan Wush.java yang telah diberikan.
Submit jawaban Anda dalam berkas .zip, nama berkas dibebaskan. Pastikan Anda melakukan zip ke berkas (*file*) jawaban, bukan *folder*.

Lampiran: *attachment.zip*

Hint:

Mencetak desimal dapat dilakukan dengan String.format().
Contoh:

```
String spec = String.format("%s $%.2f, %.1fWatt", nama, variabel1, variabel2);
System.out.println(spec);
```

Java 8

no1.zip

Score: 84

Blackbox

Score: 84

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	0	Wrong answer	0.35 sec, 28.88 MB
2	16	Accepted	0.35 sec, 28.80 MB
3	16	Accepted	0.34 sec, 29.14 MB
4	16	Accepted	0.26 sec, 27.90 MB
5	16	Accepted	0.33 sec, 28.46 MB
6	20	Accepted	0.49 sec, 28.55 MB

Question **2**

Partially correct

Mark 40.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Dungeon, Dungeon and More Dungeon

Kamu diberikan tugas untuk memodelkan suatu permainan RPG bernama Dungeon, Dungeon and More Dungeon. Di dalam permainan ini player dapat memiliki role Mage dan Knight. Di dalam permainan ini, kamu dapat menggunakan basic attack dan ultimate attackmu untuk menghabisi enemy!

Lengkapilah metode-metode pada file sesuai dengan perintah pada file:

1. `Enemy.java`
2. `Knight.java`
3. `Mage.java`
4. `Player.java`

Attachment : attachment.zip

Submit dalam format ZIP dengan penamaan bebas.

Note : Untuk inisiasi player itu ada kesalahan komen ya, bukan inisiasi enemy itu inisiasi player :D
Note 2 : untuk yang tc4 error, ganti mage.java yang ultimate attack yang "Lightning" jadi "Lighting" ya! Ingat untuk ubah command dan output lightningnya!

Java 8

no2.zip

Score: 40

Blackbox

Score: 40

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	0	Wrong answer	0.11 sec, 28.99 MB
2	0	Wrong answer	0.13 sec, 28.02 MB
3	0	Wrong answer	0.09 sec, 27.98 MB
4	0	Wrong answer	0.08 sec, 27.98 MB
5	0	Wrong answer	0.08 sec, 30.89 MB
6	10	Accepted	0.07 sec, 27.95 MB
7	10	Accepted	0.09 sec, 28.85 MB
8	10	Accepted	0.07 sec, 28.82 MB
9	10	Accepted	0.07 sec, 27.86 MB
10	0	Wrong answer	0.12 sec, 28.40 MB

Question **3**

Partially correct

Mark 70.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Number Sum III

Deskripsi

Anda diberikan suatu array statik bertipe `Number`, untuk setiap elemen di dalamnya Anda ditugaskan untuk menghitung hasil dari elemen-elemen dengan tipe data yang sama dengan melakukan penjumlahan, pengurangan, dan perkalian secara berurutan dan berulang.

Tipe data yang mungkin terdapat pada array tersebut: `Byte, Integer, Double, Float, Short, Long`

Untuk setiap data yang sama, lakukan penjumlahan, pengurangan, dan perkalian secara berurutan dan berulang hingga tipe data yang sama telah berakhir dalam array tersebut. Jika tipe data tersebut telah berakhir, lanjutkan ke tipe data berikutnya dengan mengulang dari penjumlahan kembali. Jika tipe data berikutnya tidak ada, isi dengan `null`. Jika suatu `number` dengan tipe data yang sama terpisah dari kawanannya, operasi tidak akan diulang.

Urutan operasi: `Penjumlahan, Pengurangan, Perkalian`.

Tidak perlu melakukan print sama sekali.

L, f, b, dan S pada angka menunjukkan tipe data dari angka tersebut. Contoh: 2L adalah Long, 4.5f adalah Float, 5b adalah Byte, dan 2S adalah Short. Tidak akan muncul jika di print

Tidak perlu memerdulikan floating point precision issue

Contoh

Contoh 1:

Array masukan:
[2L, 4.5f, 3.5, 45L, 100S, 5b, 2b] // [Long, Float, Double, Long, Short, Byte, Byte]

Array keluaran:
[47L, 4.5f, 3.5, 100S, 7b, null, null] // [Long, Float, Double, Short, Byte, null, null]

Penjelasan:
[(2L + 45L), (4.5f), (3.5), (100), (5+2)] [47L, 4.5f, 3.5, 100, 7, null, null] // [Long, Float, Double, Short, Byte, null, null]

Karena tersisa dua tempat kosong pada array keluaran, maka keduanya diisi dengan null

Contoh 2:

Array masukan:
[2, 4, 3, 45, 100, 5, 2] // [Integer, Integer, Integer, Integer, Integer, Integer, Integer]

Array keluaran:
[460, null, null, null, null, null, null] // [Integer, null, null, null, null, null, null]

Penjelasan:
[((2 + 4 - 3) *45 + 100 - 5)* 2, null, null, null, null, null, null] Operasi dilakukan secara berurutan dan berulang untuk tipe data yang sama hingga tipe data tersebut habis. Karena tipe data habis, maka tempat kosong diisi dengan null

Contoh 3:

Array masukan:
[1, 2, 2, 3, 4, 3.5, 4.5, 5.5, 5L, 6L, 7L, 7S, 8S, 8b, 9b, 10.5f, 11.5f, 12.5f, 5] // [Integer, Integer, Integer, Integer, Integer, Double, Double, Double, Long, Long, Long, Short, Short, Byte, Byte, Float, Float, Float, Integer]

Array keluaran:
[2, 2.5, 4L, 15S, 17b, 9.5f, null, null, null, null, null, null, null, null, null, null, null, null, null] // [Integer, Double, Long, Short, Byte, Float, null, null, null, null, null, null, null, null, null, null, null, null, null]

Penjelasan:
[((1 + 2 - 2) * 3 + 4 - 5), (3.5 + 4.5 - 5.5), (5 + 6 - 7), (7 + 8), (8 + 9), (10.5 + 11.5 - 12.5), null, null, null, null, null, null, null, null, null, null, null, null, null, null, null] -> [2, 2.5, 4L, 15S, 17b, 9.5f, null, null, null, null, null, null, null, null, null, null, null, null, null]

Perhatikan bahwa urutan tipe data array keluaran haruslah sama dengan array masukan. Perhatikan juga bahwa panjang array masukan haruslah sama dengan array keluaran. Jika terdapat tempat kosong setelah dilakukan operasi, tempat tersebut diisi dengan null

Hint:

- Untuk mengecek apakah dua objek merupakan kelas yang sama bisa dengan: `obj1.getClass().equal(obj2.getClass())`

- Untuk mengecek apakah suatu kelas merupakan subclass dari kelas lain bisa dengan keyword: `instanceof`

Lengkapilah metode-metode pada file `NumberSumIII.java`. Kemudian, Submit file `NumberSumIII.java`

Java 8

no3.zip

Score: 70

Blackbox

Score: 70

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	0	Wrong answer	0.07 sec, 28.00 MB
2	10	Accepted	0.16 sec, 28.04 MB
3	10	Accepted	0.13 sec, 28.98 MB
4	10	Accepted	0.13 sec, 26.22 MB
5	0	Wrong answer	0.14 sec, 28.95 MB
6	10	Accepted	0.20 sec, 29.00 MB
7	10	Accepted	0.21 sec, 28.03 MB
8	10	Accepted	0.09 sec, 28.78 MB
9	0	Wrong answer	0.13 sec, 28.34 MB
10	10	Accepted	0.09 sec, 28.18 MB

◄ Pra-kuis 1

Jump to...

Feedback Form ►