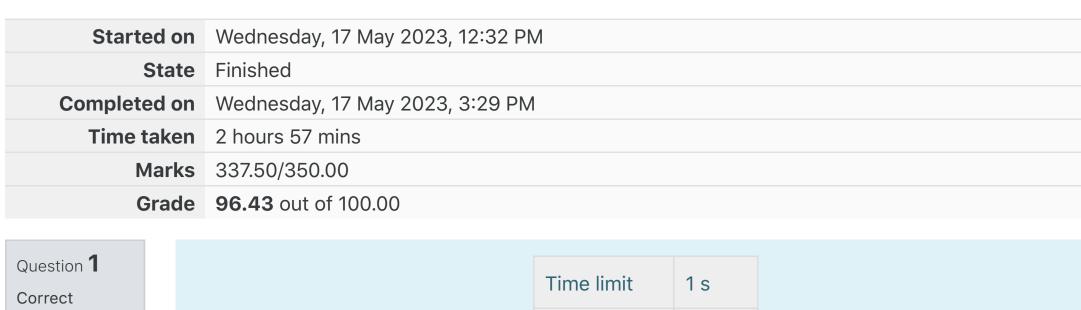
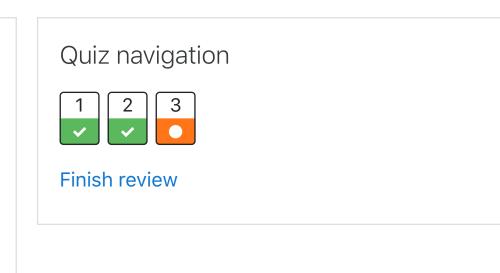
ITB_IF2210_2_2223 Pemrograman Beriorientasi Objek Dashboard / My courses / ITB_IF2210_2_2223 / Ujian Praktikum 2 / Bagian 1 - Ujian Praktikum 2





Mark 100.00 out of 100.00 ▼ Flag question

Memory limit | 64 MB BoxOperator.java merupakan kelas ajaib yang dapat melakukan filtering isi dari Box.java berdasarkan tipe kelas dari object-object yang berada di dalam Box.java. Silahkan implementasi method getContents yang berada di BoxOperator.java. Notes: Jangan ubah-ubah kelas Box.java, cukup implementasikan getContents yang ada di dalam BoxOperator.java dan kumpulkan BoxOperator.java. Contoh Input dan Output: Input (Pseudocode): Output (Pseudocode): BoxOperator.getContents([21, 3.4f, true, "TEST", 'c'], Integer.class) [21] BoxOperator.getContents([21, 3.4f, true, "TEST", 'c'], Object.class) [21, 3.4f, true, "TEST", 'c'] Hint: Gunakan fungsi getClass() dari kelas object untuk mendapat Class<?> dari suatu object. • Tipe Class<T> memiliki fungsi public boolean isAssignableFrom(Class<?> cls) yang dapat digunakan untuk memeriksa apakah sebuah Class<?> bisa di assign ke Class<T>. • Tipe Class<T> memiliki fungsi public T cast(Object obj) yang dapat digunakan untuk cast object ke T dengan kelas Class<T> Java 8 ♦ BoxOperator.java

Score: 60 Blackbox Score: 60 Verdict: Accepted **Evaluator: Exact Score Verdict Description** Accepted 0.07 sec, 29.28 MB 10 Accepted 0.07 sec, 26.19 MB 2 10 Accepted 0.07 sec, 28.75 MB 10 3 Accepted 0.07 sec, 29.07 MB 10 Accepted 0.07 sec, 28.92 MB 5 10 6 10 Accepted 0.07 sec, 29.00 MB

Question 2 Correct Mark 100.00 out of 100.00 Flag question

Buatlah sebuah program untuk sebuah sistem e-commerce untuk mengelola produk-produk yang dijual. Program sebagai berikut: • Kelas abstrak Product yang memiliki atribut id (integer), name (string), price (integer),

{price} - {warrantyPeriod} - {description}.

dan description (string). Kelas Product memiliki method abstrak dengan nama getInfo yang mengembalikan sebuah string. Untuk setiap atribut, terdapat method setter dan getter.

Time limit

1 s

Memory limit 64 MB

Catatan: Dipastikan setiap Product memiliki ID yang beda (Unique ID). • Kelas ElectronicProduct yang merupakan turunan dari Product yang memiliki atribut tambahan warrantyPeriod (string) yaitu periode garansi. Kelas ini memiliki method tambahan getWarrantyPeriod yang mengembalikan string berupa warrantyPeriod.

Method getInfo di-override untuk mengembalikan string dengan format {id}: {name} -

tambahan size (string) yaitu periode garansi. Kelas ini memiliki method tambahan getSize yang mengembalikan string berupa size. Method getInfo di-override untuk mengembalikan string dengan format {id}: {name} - {price} - {size} -{description}. • Kelas FoodProduct yang merupakan turunan dari Product yang memiliki atribut

• Kelas FashionProduct yang merupakan turunan dari Product yang memiliki atribut

- tambahan expiryDate (string) yaitu tanggal kadaluwarsa. Kelas ini memiliki method tambahan getExpiryDate yang mengembalikan string berupa expiryDate. Method getInfo di-override untuk mengembalikan string dengan format {id}: {name} -{price} - {expiryDate} - {description}. • Interface ProductManagement yang memiliki beberapa method sebagai berikut:
- 2. removeProduct bertipe Product menerima parameter bertipe id berupa integer. 3. updateProduct bertipe void menerima parameter
 - bertipe id berupa integer, name bertipe string, price bertipe integer,

roductManagement.java, ProductCatalog.java sebagai Product.zip.

kembalikan null.

Product.zip

Score: 100

Verdict: Accepted

beberapa aturan yaitu.

3. Semua methodnya:

1. addProduct bertipe void menerima parameter bertipe Product.

- dan description bertipe string. 4. getAllProduct bertipe List<Product> dan tidak menerima parameter apapun. 5. getProduct bertipe Product dan menerima parameter id berupa integer.
- Kelas ProductCatalog yang mengimplementasikan interface ProductManagement. Kelas ini memiliki member productList bertipe List<Product> untuk menyimpan produk-produk yang

ada. Ketentuan implementasi method dari interface ProductManagement sebagai berikut:

- 1. addProduct menambahkan Product ke productList. 2. removeProduct menghilangkan Product dari productList berdasarkan id dan
- mengembalikannya sebagai return value. Apabila tidak ditemukan, kembalikan null. 3. updateProduct mengubah atribut name, price, dan description dari sebuah Product berdasarkan id. Apabila tidak ditemukan, print pesan Produk dengan id
- {id} tidak ditemukan. 4. getAllProduct mengembalikan seluruh daftar Product.

5. getProduct mengembalikan sebuah Product berdasarkan id. Apabila tidak ditemukan,

Java 8 ♦

Kumpulkan Product.java, ElectronicProduct.java, FashionProduct.java, FoodProduct.java, P

Score: 100 Blackbox

Evaluator: Exact **Score Verdict Description** Accepted 0.63 sec, 30.60 MB 9 2 9 Accepted 0.43 sec, 28.41 MB 3 9 Accepted 0.37 sec, 28.61 MB 4 9 Accepted 0.39 sec, 26.85 MB Accepted 0.22 sec, 28.08 MB 5 9 6 9 Accepted 0.17 sec, 28.21 MB 9 Accepted 0.13 sec, 28.28 MB Accepted 0.16 sec, 26.29 MB 8 9 Accepted 0.17 sec, 29.18 MB 9 9 Accepted 0.55 sec, 28.55 MB 10 9 10 Accepted 0.40 sec, 27.86 MB Time limit 1 s Memory limit 64 MB

question

Question 3

Mark 137.50 out of 150.00

Partially correct

Flag

1. Merupakan kelas yang "concrete" artinya tidak akan meng-inherit maupun di-inherit (tidak merupakan kelas child maupun kelas parent). 2. Semua attribute-nya bertipe **String atau Integer**. 1. Memiliki return value String atau Integer

Detektif

Terdapat sebuah kelas rahasia (method dan attribute-nya tidak diketahui) yang akan kita

sebut sebagai kelas MysteriousCase. Meskipun rahasia, MysteriousCase ternyata memiliki

parameternya dipastikan consecutive Integer atau consecutive String, parameter Integer dan String tidak akan bercampur. Consecutive = 1..n 4. Access modifier pada class, attribute, maupun method tidak dibatasi (namun tetap

2. Blsa menerima parameter ataupun tidak dan apabila menerima parameter maka

- sesuai aturan), contohnya public, private, protected, static, final dst... Buatlah kelas **Detective** (yang pada dasarnya adalah **reflection class**) yang bertujuan untuk membongkar isi dari MysteriousCase. Detective mempunyai 8 method utama yaitu
 - 3. investigateSecretMethodWithParam() 4. **getNumberOfIntUsed()** <- Sudah diimplementasikan 5. getNumberOfStringUsed()

6. printAllMethodSignature()

1. investigateSecretAttribute() <- Sudah diimplementasikan

2. investigateSecretMethod() <- Sudah diimplementasikan

- 7. printAllAttributeSignature() <- Sudah diimplementasikan Detail parameter dan tujuan fungsi dapat dibaca pada file yang disediakan. Disediakan Detective.java yang sudah diimplementasikan beberapa (4) method utamanya, lengkapilah
- bahwa 1. Disediakan juga MainTest.java dan MysteriousCase1.java yang berguna untuk membantu melakukan pengujian

awal sudah ada dan tidak menambah import baru.

Java 8 ♦ Detective.java

implementasi method utama yang lainnya dan kumpulkan kembali Detective.java . Perhatikan

2. Anda boleh membuat method intermediate/helper selama tidak merubah kode yang dari

Score: 110

Description

0.12 sec, 29.03 MB

0.15 sec, 28.07 MB

0.18 sec, 26.76 MB

Verdict: Wrong answer **Evaluator: Exact Score Verdict**

Blackbox

Score: 110

2 10 Accepted

10

10

10 Accepted 4 5 10 Accepted

0.14 sec, 27.95 MB Accepted 0.15 sec, 28.58 MB 0.16 sec, 28.33 MB Wrong answer

Jump to...

Bagian 2 - Ujian Praktikum 2

3

Accepted

0.17 sec, 28.85 MB

0.27 sec, 28.46 MB 0.37 sec, 26.51 MB 0.31 sec, 28.01 MB 0.22 sec, 28.12 MB

0.24 sec, 27.97 MB

Accepted

10

11 12 0

Announcements

ITB IF2210 2 2223 Data retention summary

Finish review

Get the mobile app

You are logged in as 13521129 Chiquita Ahsanunnisa (Log out)