The price of gold

source: nowiknow.com

The price of gold

Problem

Write an app that prints the price of one kilo of gold in US dollars.

Static attributes

```
public static type attributeName
```

- All attributes we will use in our apps are `public`.
- All attributes we will use in our apps are `static`.
- `type` is the type of the attribute.

Static attributes

```
public static final type attributeName
```

- All attributes we will use in our apps are `public`.
- All attributes we will use in our apps are `static`.
- The attribute is a constant.
- `type` is the type of the attribute.

Static attributes

```
public static final double GRAMS_PER_TROY_OUNCE
```

Question

What is the type of the attribute `GRAMS_PER_TROY_OUNCE`?

Static attributes

```
public static final double GRAMS_PER_TROY_OUNCE
```

Question

What is the type of the attribute `GRAMS_PER_TROY_OUNCE`?

Answer

double.

Static attributes

```
public static final double GRAMS_PER_TROY_OUNCE
```

Question

What is the type of the attribute `GRAMS_PER_TROY_OUNCE`?

Answer

double.

Question

Is the attribute `GRAMS_PER_TROY_OUNCE` a constant?

Static attributes

```
public static final double GRAMS_PER_TROY_OUNCE
```

Question

What is the type of the attribute `GRAMS_PER_TROY_OUNCE`?

Answer

double.

Question

Is the attribute `GRAMS_PER_TROY_OUNCE` a constant?

Answer

Yes.

Using a static attribute

Consider the attribute `public static type attributeName` in the class `className`.
The attribute is used as `className.attributeName`.

Using a static attribute

Question

How do you use the attribute `GRAMS_PER_TROY_OUNCE` of the class `Gold`?

Using a static attribute

Question

How do you use the attribute `GRAMS_PER_TROY_OUNCE` of the class `Gold`?

Answer

```
Gold.GRAMS_PER_TROY_OUNCE
```

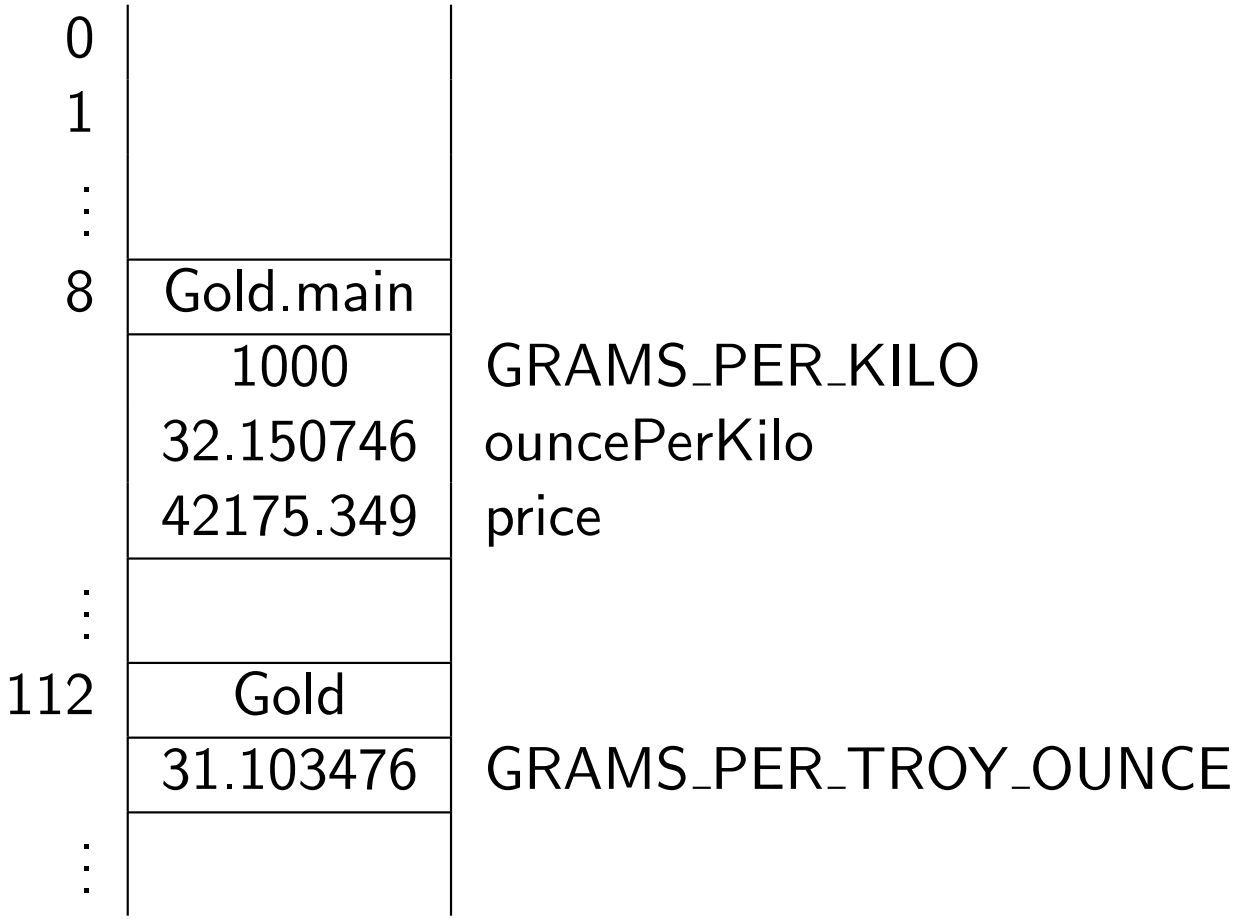
Code convention for names of constants

- Use uppercase characters.
- If the name is made up of more than one word, separate the words by an underscore.

In the lab, you can use the command checkstyle.sh to check if your app adheres to some of the code conventions. For example, typing checkstyle.sh GoldPrice.java may result in the following:

```
There were some style errors:
Starting audit ...
GoldPrice.java:7:19: Constant 'gramsPerKilo' should be in all
uppercase, start with a letter, and contain only letters, digits,
and '_'.
Audit done.
```

Memory model



The price of gold

Problem

Write an app that prints the price of one kilo of gold in Canadian dollars.

Static methods

```
public static double convert(double amount, String
from, String to)
```

Question

What is the return type of the method `convert`?

Static methods

```
public static double convert(double amount, String
from, String to)
```

Question

What is the return type of the method `convert`?

Answer

`double`.

Static methods

```
public static double convert(double amount, String
from, String to)
```

Question

What is the return type of the method `convert`?

Answer

`double`.

Question

How many parameters does the method `convert` have?

Static methods

```
public static double convert(double amount, String
from, String to)
```

Question

What is the return type of the method `convert`?

Answer

`double`.

Question

How many parameters does the method `convert` have?

Answer

Three.

Static methods

```
public static double convert(double amount, String
from, String to)
```

Question

What is the signature of the method `convert`?

Static methods

```
public static double convert(double amount, String
from, String to)
```

Question

What is the signature of the method `convert`?

Answer

`convert(double, String, String)`

Precondition

The precondition

```
amount >= 0.0,
from == Currency.CAD || from == Currency.USD || from == Currency.EUR,
to == Currency.CAD || to == Currency.USD || to == Currency.EUR
```

can be read as

amount $\geq$ 0.0 and

(from == Currency.CAD or from == Currency.USD or from == Currency.EUR) and

(to == Currency.CAD or to == Currency.USD or to == Currency.EUR)

Precondition

The precondition

```
amount >= 0.0,
from == Currency.CAD || from == Currency.USD || from == Currency.EUR,
to == Currency.CAD || to == Currency.USD || to == Currency.EUR
```

can be read as

amount $\geq$ 0.0 and

(from == Currency.CAD or from == Currency.USD or from == Currency.EUR) and

(to == Currency.CAD or to == Currency.USD or to == Currency.EUR)

Question

Who is responsible for the precondition, the client of the implementer?

Precondition

The precondition

```
amount >= 0.0,
from == Currency.CAD || from == Currency.USD || from == Currency.EUR,
to == Currency.CAD || to == Currency.USD || to == Currency.EUR
```

can be read as

amount $\geq$ 0.0 and

(from == Currency.CAD or from == Currency.USD or from == Currency.EUR) and

(to == Currency.CAD or to == Currency.USD or to == Currency.EUR)

Question

Who is responsible for the precondition, the client of the implementer?

Answer

The client.

Contract

Question

If `Currency.convert(1.0, Currency.USD, Currency.CAD)` returns -1.03, who is to blame?

Contract

Question

If `Currency.convert(1.0, Currency.USD, Currency.CAD)` returns -1.03, who is to blame?

Answer

The implementer, since the client has done its job by providing arguments that satisfy the precondition, whereas the implementer did not satisfy the postcondition.

Contract

Question

If `Currency.convert(1.0, Currency.USD, "YEN")` returns -99.13, who is to blame?

Contract

Question

If `Currency.convert(1.0, Currency.USD, "YEN")` returns -99.13, who is to blame?

Answer

The client, since the third argument "YEN" does not satisfy the precondition, which is the client's responsibility.

The price of gold

Problem

Write an app that pops up a dialog box with the title "The Price of Gold" and the message "Enter the amount of gold in kilos" and, after the user has entered the amount k and clicks the OK button, pops up another dialog box with the title "The Price of k kilos of Gold" and the current price of k kilos of Gold in Canadian dollars.

Static methods

`public static String showInputDialog(Component parent, String message, String title, int messageType)`[1] of class `JOptionPane` of package `javax.swing`.

Question

What is the return type of the method `showInputDialog`?

[1] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static String showInputDialog(Component parent, String message, String title, int messageType)`[1] of class `JOptionPane` of package `javax.swing`.

Question

What is the return type of the method `showInputDialog`?

Answer

`String`.

[1] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static String showInputDialog(Component parent, String message, String title, int messageType)`[1] of class `JOptionPane` of package `javax.swing`.

Question

What is the return type of the method `showInputDialog`?

Answer

`String`.

Question

How many parameters does the method `showInputDialog` have?

[1] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static String showInputDialog(Component parent, String message, String title, int messageType)`[1] of class `JOptionPane` of package `javax.swing`.

Question

What is the return type of the method `showInputDialog`?

Answer

`String`.

Question

How many parameters does the method `showInputDialog` have?

Answer

Four.

[1] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static String showInputDialog(Component parent, String message, String title, int messageType)` of class `JOptionPane` of package `javax.swing`.

Question

What is the signature of the method `showInputDialog`?

[2] We will come back to `null` later in the course.

Static methods

`public static String showInputDialog(Component parent, String message, String title, int messageType)` of class `JOptionPane` of package `javax.swing`.

Question

What is the signature of the method `showInputDialog`?

Answer

`showInputDialog(Component, String, String, int)`

[2] We will come back to `null` later in the course.

Static methods

`public static String showInputDialog(Component parent, String message, String title, int messageType)` of class `JOptionPane` of package `javax.swing`.

Question

What is the signature of the method `showInputDialog`?

Answer

`showInputDialog(Component, String, String, int)`

In our case, we do not need a parent component (whatever that may be), and therefore we use the default value `null` as the first argument.[2]

[2] We will come back to `null` later in the course.

Static methods

`public static double parseDouble(String s)` of class `Double` of package `java.lang`.

Question

What is the return type of the method `parseDouble`?

Static methods

`public static double parseDouble(String s)` of class `Double` of package `java.lang`.

Question

What is the return type of the method `parseDouble`?

Answer

`double`.

Static methods

`public static double parseDouble(String s)` of class `Double` of package `java.lang`.

Question

What is the return type of the method `parseDouble`?

Answer

`double`.

Question

How many parameters does the method `parseDouble` have?

Static methods

`public static double parseDouble(String s)` of class `Double` of package `java.lang`.

Question

What is the return type of the method `parseDouble`?

Answer

`double`.

Question

How many parameters does the method `parseDouble` have?

Answer

One, of type `String`.

Static methods

`public static void showMessageDialog(Component parent, String message, String title, int messageType)`[3] of class `JOptionPane` of package `javax.swing`.

Question

What is the return type of the method `showMessageDialog`?

[3] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static void showMessageDialog(Component parent, String message, String title, int messageType)`[3] of class `JOptionPane` of package `javax.swing`.

Question

What is the return type of the method `showMessageDialog`?

Answer

None.

[3] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static void showMessageDialog(Component parent, String message, String title, int messageType)`[3] of class `JOptionPane` of package `javax.swing`.

Question

What is the return type of the method `showMessageDialog`?

Answer

None.

Question

How many parameters does the method `showMessageDialog` have?

[3] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static void showMessageDialog(Component parent, String message, String title, int messageType)`[3] of class `JOptionPane` of package `javax.swing`.

Question

What is the return type of the method `showMessageDialog`?

Answer

None.

Question

How many parameters does the method `showMessageDialog` have?

Answer

Four.

[3] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static void showMessageDialog(Component parent, String message, String title, int messageType)` of class `JOptionPane` of package `javax.swing`.

Question

What is the signature of the method `showMessageDialog`?

Static methods

`public static void showMessageDialog(Component parent, String message, String title, int messageType)` of class `JOptionPane` of package `javax.swing`.

Question

What is the signature of the method `showMessageDialog`?

Answer

`showMessageDialog(Component, String, String, int)`

Static methods

`public static void showMessageDialog(Component parent, String message, String title, int messageType)` of class `JOptionPane` of package `javax.swing`.

Question

What is the signature of the method `showMessageDialog`?

Answer

`showMessageDialog(Component, String, String, int)`

In our case, we do not need a parent component (whatever that may be), and therefore we use the default value `null` as the first argument.

Static methods

`public static String format(String format, double value)`[4] of class `String` of package `java.lang`.

Question

What is the return type of the method `format`?

[4] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static String format(String format, double value)`[4] of class `String` of package `java.lang`.

Question

What is the return type of the method `format`?

Answer

`String`.

[4] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static String format(String format, double value)`[4] of class `String` of package `java.lang`.

Question

What is the return type of the method `format`?

Answer

`String`.

Question

How many parameters does the method `format` have?

[4] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

Static methods

`public static String format(String format, double value)`[4] of class `String` of package `java.lang`.

Question

What is the return type of the method `format`?

Answer

`String`.

Question

How many parameters does the method `format` have?

Answer

Two.[a]

[a] As we will see later, the answer is "at least one."

[4] The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

The price of gold

Write an app that pops up a dialog box with the title "The Price of Gold" and the message "Enter the amount of gold in kilos" and, after the user has entered the amount k and clicks the OK button, pops up another dialog box with the title "The Price of k kilos of Gold" and the current price of k kilos of Gold in Canadian dollars. If the users enters a negative amount, the app crashes with the message "The amount of gold cannot be negative."

To do

Study Section 2.3 of the textbook.

Test 1

- When: Friday January 17, during the lab (14:30–16:00)
- Where: Lassonde building, labs 1006, 1004, 1002
- Material: Chapter 1 of the textbook
- What: One programming question similar to Check01C and five multiple choice/short answer questions
- Advise: Do the five multiple choice/short answer questions first
- Note: Your code is not only marked for correctness but also style

How to submit your code during the test?

To submit your code, use something like
submit 1020 test1 Test1A.java

If successful, you see something like
submitted: Test1A.java (123 bytes)
All files successfully submitted.
If you are in the incorrect directory, you see something like
error: cannot find source Test1A.java

How to submit your code during the test?

For those using eclipse, first locate the file Test1A.java. Use the commands ls and cd. For example, if you named your project test then use something like
cd workspace/test/src

Advice: if you plan to use eclipse, get familiar with the commands ls and cd before the test.