Java Objects Part 1 Notes

Team Treehouse

May 21, 2020

1 Welcome Back

- STRING.toLowerCase()
 - Turns string into lowercase letter
- STRING.contains(...)
 - checks if value ... is contained inside String

2 Quiz 1

1. Please fill in the correct answer in each blank provided below.

```
String someWords = "These are words";
someWords.____("words");
```

Answer:

```
String someWords = "These are words";
someWords.contains("words");
```

- 2. The boolean datatype is used to store:
 - A. numbers
 - B. text

C. true or false values

Answer: C

3. Please fill in the correct answer in each blank provided below.

What operator do we use to ensure that both of these conditions are met:

```
boolean isRefreshed = true;
boolean isReadyToGetStarted = true;
boolean shouldContinue = isRefreshed____isReadyToGetStarted;
4
```

Answer:

```
boolean isRefreshed = true;
boolean isReadyToGetStarted = true;
boolean shouldContinue = isRefreshed && isReadyToGetStarted;
```

4. Please fill in the correct answer in each blank provided below.

```
int weightOfCraigsKid = 50;
int weightMonty = 130;
if (weightMonty _____ weightOfCraigsKid) {
    console.printf("Whoa that's a huge dog!");
}
```

Answer:

```
int weightOfCraigsKid = 50;
int weightMonty = 130;
if (weightMonty > weightOfCraigsKid) {
    console.printf("Whoa that's a huge dog!");
}
```

- 5. To define a new variable to store a name it would look something like this:
 - A. String firstName = "Bob";
 - B. "Bob" = first.name
 - C. firstName = "Bob";
 - D. first_name = 'Bob'

Answer: A

- 6. The boolean datatype is used to store:
 - A. numbers
 - B. text
 - C. true or false values

Answer: C

3 Creating Classes

```
class PezDispenser { // <- 1. Class is created in a separate and
matching file :)

String characterName = "Yoda";
}</pre>
```

Listing 1: lesson_3/PezDispenser.java

```
import java.io.Console;

public class Example {
    public static void main(String[] args) {

        System.out.println("We are making a new PEZ dispenser");

        PezDispenser dispenser = new PezDispenser(); // <- 2. And is used here :)

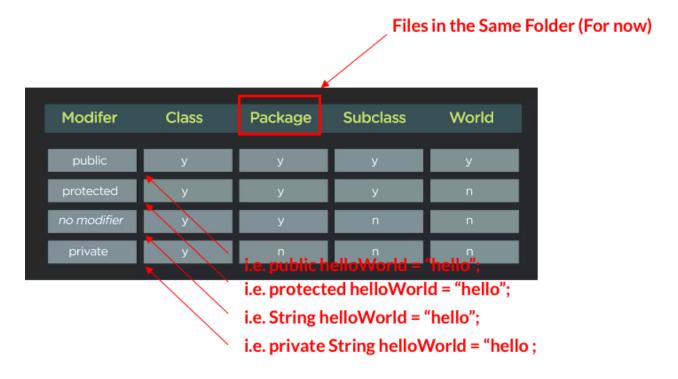
        System.out.printf("The dispenser is %s", dispenser. characterName);
        ...
     }
}</pre>
```

Listing 2: lesson_3/Example.java

4 Exercise 1

• Solution included in exercise_1.java

5 Access Modifiers



- Determines who is intended to access the information
- Adding access modifiers to attributes and methods in class is called **Encapsulation**

```
class PezDispenser {
    private String characterName = "Yoda"; // <- 1. attribute is turned private
}
```

Listing 3: lesson_5/PezDispenser.java

```
import java.io.Console;

public class Example {
    public static void main(String[] args) {

        System.out.println("We are making a new PEZ dispenser");

        PezDispenser dispenser = new PezDispenser();

        System.out.printf("The dispenser is %s", dispenser.
        characterName); // <- 2. and it can't be accessed outside of class
        }
}
</pre>
```

Listing 4: lesson_5/Example.java

Notes:

- Files can be compiled and displayed by typing javac Example.java && java Example in terminal

6 Exercise 2

• Solution included in exercise_2.java

7 Methods

- is a collection of statements that are grouped together to perform an operation
- is like *verb*
- ACCESS_MODIFIER DATA_TYPE get<ATTRIBUTE_NAME > is called **getter**

```
public class PezDispenser {
    private String characterName = "Yoda";

public String getCharacterName() { // <- 1. Getter method added here
    return characterName;
}

}
</pre>
```

Listing 5: lesson_7/PezDispenser.java

```
import java.io.Console;

public class Example {
    public static void main(String[] args) {

        System.out.println("We are making a new PEZ dispenser");

        PezDispenser dispenser = new PezDispenser();

        System.out.printf("The dispenser is %s", dispenser.
        getCharacterName()); // <- 2. And is used here
        }
}</pre>
```

Listing 6: lesson_7/Example.java