Java Objects Part 2 Notes

Team Treehouse

May 22, 2020

1 constants

- Are named *IN_CAPITALIZED_SNAKE_CASE*
- Can be done using *static* keyword
- Allows variables and methods to be exposed without instantiation

```
1     public class PezDispenser {
2         public static final int MAX_PEZ = 12; // <- 1. static declared
      here :)
3         ...
4     }
5
```

Listing 1: lesson_1/PezDispenser.java

```
1     import java.io.Console;
2
3     public class Example {
4         public static void main(String[] args) {
5             ...
6             System.out.printf("FUN FACT: There are %d PEZ allowed in
      every dispenser\n", PezDispenser.MAX_PEZ); // 2. <- And is used
      here :)
7             ...
8         }
9     }
10
```

Listing 2: lesson_1/Example.java

Notes:

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

2 Exercise 1

- Solution included in *exercise_1.java*

3 Filling the Dispenser

- *void* keyword means nothing is returned at the end of a method

```
1    public class PezDispenser {
2        public void fill() { // <- This little guy here :)
3            this.pezCount = MAX_PEZ;
4            System.out.printf("The current count of delicious PEZ is %
   d\n", this.pezCount);
5        }
6    }
7
```

Listing 3: lesson_3/PezDispenser.java

```
1    import java.io.Console;
2
3    public class Example {
4        public static void main(String[] args) {
5            ...
6            dispenser.fill(); // <- 2. Is used like this
7
8        }
9    }
10
```

Listing 4: lesson_3/Example.java

Notes:

- Files can be compiled and displayed by typing *javac Example.java && java Example* in terminal
- Always start with private methods, and turn to public when needed.

4 Exercise 2

- Solution included in *exercise_2.java*

5 Abstraction at Play

- *Golden Rule* Don't make users understand object internally
  - Simple questions such as 'is it empty?' is sufficent

```
1   public class PezDispenser {
2       public boolean isEmpty() { // <- This little guy here :)
3           return this.pezCount == 0;
4       }
5
6       ...
7   }
8
```

Listing 5: lesson_5/PezDispenser.java

```
1   import java.io.Console;
2
3   public class Example {
4       public static void main(String[] args) {
5           ...
6           if (dispenser.isEmpty()) {
7               System.out.printf("Dispenser is empty"); // <- 2. with
    this little fellow here
8           }
9
10          ...
11          if (!dispenser.isEmpty()) {
12              System.out.printf("Dispenser is full\n"); // <- 3. and
    this guy as well
13          }
14
15      }
16  }
17
```

Listing 6: lesson_5/Example.java

Notes:

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

6 Exercise 3

- Solution included in *exercise_3.java*

7 Incrementing and Decrementing

- *INT_VARIABLE–:* Decrements the value in variable by 1
- *INT_VARIABLE++:* Increments the value in variable by 1

```
    public class PezDispenser {
        ...
        public boolean dispense() { // <- 1. This little guy here :)
            boolean wasDispensed = false;
            if (!this.isEmpty()) {
                this.pezCount--; // <- 2. With decrement count here
                wasDispensed = true;
            }

            return wasDispensed;

        }
    }

```

Listing 7: lesson_7/PezDispenser.java

```
    import java.io.Console;

    public class Example {
        public static void main(String[] args) {
            ...
            while (dispenser.dispense()) {
                System.out.println("Chomp!"); // <- 3. This will print
    as long as .dispensed() returns true
            }

            if (dispenser.isEmpty()) {
                System.out.println("Ate all the PEZ");
            }
        }
    }

```

Listing 8: lesson_7/Example.java

```
    >>> javac Example.java && java Example
    We are making a new PEZ dispenser

    FUN FACT: There are 12 PEZ allowed in every dispenser
    Dispenser is emptyThe dispenser is Yoda
    Filling the dispenser with delicious PEZ...
    The current count of delicious PEZ is 12
    Dispenser is full
    Chomp!
```

```
    Chomp!
    Chomp!
    Chomp!
    Chomp!
    Chomp!
    Chomp!
    Chomp!
    Chomp!
    Chomp!
    Chomp!
    Ate all the PEZ

```

Notes:

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