Parsons Problems — Week 13

Exceptions · Problem 1 of 2

Problem 1 — Wardrobe

Rearrange the slips to form a `Wardrobe` class that stores an array of items and whose `checkInventory` method reports what is in a given slot, catching the exception thrown when the slot number does not exist.

Example: for a `Wardrobe` holding `{"scarf", "jacket"}`, `checkInventory(0)` ⟶ `Slot contains: scarf`; `checkInventory(2)` ⟶ `Slot is empty!`

2 of the slips are distractors — they look plausible but do not belong in the solution.

```
private String[] items;
```

```
public String checkInventory(int slot) {
```

```
public Wardrobe(String[] items) { items = items; }
```

```
public Wardrobe(String[] items) { this.items = items; }
```

```
public class Wardrobe {
```

```
return "Slot contains: " + items[slot];
```

```
try {
```

```
}
```

```
}
```

```
} catch (ArrayIndexOutOfBoundsException e) { return "Slot is empty!"; }
```

```
} catch (NullPointerException e) { return "Slot is empty!"; }
```

Parsons Problems — Week 13

Exceptions · Problem 2 of 2

Problem 2 — Calculator

Rearrange the slips to form a `Calculator` class with a `divideEvenly` method that returns `a / b`, throwing a custom checked `DivisorException` if `b` is `0` or if `b` does not evenly divide `a`.

Example: `divideEvenly(100, 4)` ⟶ `25`; `divideEvenly(100, 0)` throws a `DivisorException` with message `Cannot divide by zero!`; `divideEvenly(100, 3)` throws a `DivisorException` with message `100 is not evenly divisible by 3!`

2 of the slips are distractors — they look plausible but do not belong in the solution.

```
DivisorException(String message) { super(message); }
```

```
DivisorException(String message) { }
```

```
class DivisorException extends Exception {
```

```
if (a % b != 0) throw new DivisorException(a + " is not evenly divisible by " + b + "!");
```

```
if (b == 0) throw new DivisorException("Cannot divide by zero!");
```

```
public class Calculator {
```

```
return a / b;
```

```
static int divideEvenly(int a, int b) throws ArithmeticException {
```

```
static int divideEvenly(int a, int b) throws DivisorException {
```

```
}
```

```
}
```

```
}
```