

## 12.2 Exceptions

In this section, we will learn how to throw exceptions to effectively handle errors that may arise in our code.

Part 3 - exceptions



Our `ArraySet` implementation from the previous section has a small error. When we add `null` to our `ArraySet`, we get a `NullPointerException`.

The problem lies in the `contains` method where we check `items[i].equals(x)`. If the value at `items[i]` is `null`, then we are calling `null.equals(x)` → `NullPointerException`.

Exceptions cause normal flow of control to stop. We can in fact choose to throw our own exceptions. In python you may have seen this with the `raise` keyword. In Java, Exceptions are objects and we throw exceptions using the following format:

```
throw new ExceptionObject(parameter1, ...)
```

Let's throw an exception when a user tries to add null to our `ArraySet`. We'll throw an `IllegalArgumentException` which takes in one parameter (a `String` message).

Our updated `add` method:

```
/* Associates the specified value with the specified key in this map.
   ... Throws an IllegalArgumentException if the key is null. */
public void add(T x) {
    if (x == null) {
        throw new IllegalArgumentException("can't add null");
    }
    if (contains(x)) {
        return;
    }
    items[size] = x;
    size += 1;
}
```

We get an Exception either way - why does this better?

1. We have control of our code: we consciously decide at what point to stop the flow of our program
2. More useful Exception type and helpful error message for those using our code

However, it would be better if the program doesn't crash at all. There are different things we could do in this case. Here are some below:

**Approach 1:** Don't add `null` to the array if it is passed into `add`

**Approach 2:** Change the `contains` method to account for the case if `items[i] == null`.

Whatever you decide, it is important that users know what to expect. That is why documentation (such as comments about your methods) is very important.

[Previous](#)  
12.1 Lists and Sets in Java

[Next](#)  
12.3 Iteration

Last updated 1 year ago

