CS62: Spring 2025 | Lecture #5 (Interfaces & Generics) worksheet | Jingyi Li

- Create an interface called `Adoptable` that contains four abstract methods: a `void requestAdoption()`, `boolean isAdopted()`, `void completeAdoption()`, and `String makeHappyNoise()`.
- Have the class `Animal implement` the interface. You can provide some very minimal implementation of the methods so that you don't receive a compile-time error.
- Override the `makeHappyNoise()` in the `Cat` and `Dog` subclasses.

(There's no starter code for this problem: practice remembering the syntax by yourself!)

```java
import java.util.ArrayList;

class Box<T> {
    private ArrayList<T> items = new ArrayList<T>();

    public void addItem(________) {
        items.add(item);
    }

    ________ getItem(int index) {
        return items.get(index);
    }

    public void removeItem(int index) {
        items.remove(index);
    }

    public int getSize() {
        return items.size();
    }
}

public class WithGenerics {
    public static void main(String[] args) {
        Box<String> fruitBox = _____________
        fruitBox.addItem("Grape");
        fruitBox.addItem("Banana");

        __________ weightBox = new Box<Double>();
        weightBox.addItem(10.0);
        weightBox.addItem(12.5);
        weightBox.addItem(25.0);
        weightBox.removeItem(0);

        System.out.println(_________________); //# of fruits in fruitbox
        System.out.println(weightBox.getItem(1));
    }
}
```

You are making a new data structure, a box, to store similar kinds of items. Box is implemented using an ArrayList.

1. Fill in the 5 blanks.

2. What gets printed?

3. If you called `fruitBox.addItem(47)`, what would happen and why?

4. How do generics ensure type safety in this example?