Interfaces

- another important one is the `Cloneable` interface - declares whether an object can be cloned using the `clone()` method in `Object` class

```java
public interface Cloneable {
    // no methods in this interface, called a marker interface
}
```

- recall **shallow** copy = if the copied field is an object, only the reference is copied
  
  **deep** copy = copies the contents of an object into a new object with different reference

- if an object contains other objects that also need to be copied to be properly cloned, you have to write your own clone method that overrides the one in `Object`. To override, must have same signature

```java
@Override // also change accessibility from protected to public
public Object clone() throws CloneNotSupportedException{
    // 1. clone easy stuff using super.clone() and cast to class type
    // 2. next clone objects used in this class type
}
```

since original clone method throws this exception, this method must too; or else put in try catch block