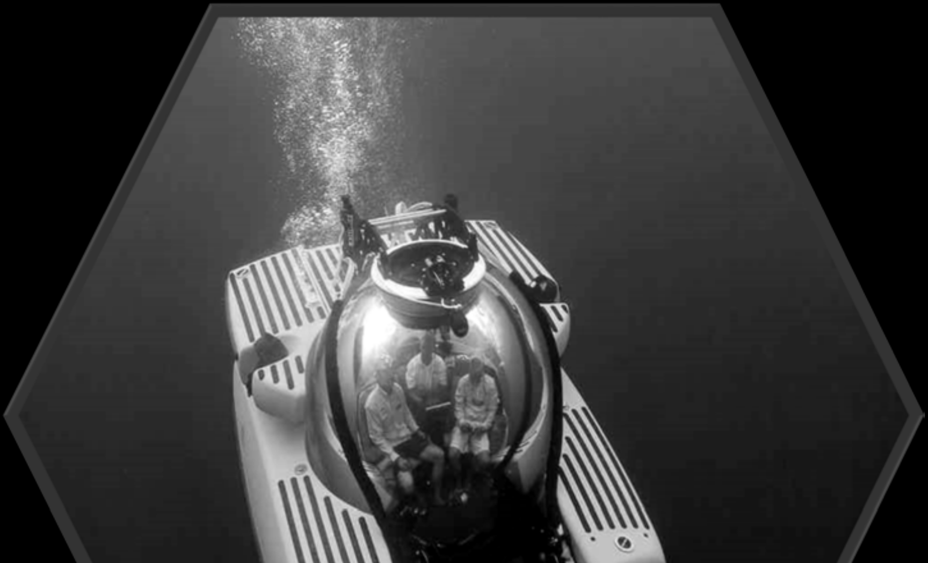
Inheritance / Sub-classes (RECAP)



Inheritance Recap

- Problem: you have written a class (e.g. **Robot**), which almost does what you want, but requires some extensions
- Idea: **extend** features from the existing class by creating a **child class** that automatically receives all features of the parent class (e.g. **name**, **talk()**,…) without writing code again
- Implementation: you define a new class (e.g. **TranslationRobot**) inheriting **all** features from the existing parent class, but add or adapt features so that the new class does exactly what you want
- Result: leads to **DRY** (do-not-repeat-yourself) code where each feature has a **single code source**

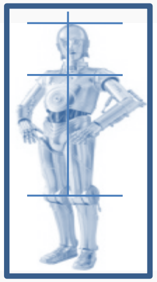
parent class Robot provides all its features to the child class

```
class Robot {
  String name;
  int numLegs;
  float powerLevel;

 Robot(String productName) {
    name = productName;
    numLegs = 2;
    powerLevel = 2.0f;
  }

  void talk(String phrase) {
    if (powerLevel >= 1.0f) {
      System.out.println(name+" says "+
                         phrase);
      powerLevel -= 1.0f;
    } else {
      System.out.println(name +
      " is too weak to talk.");
  } }

  void charge(float amount) {
    System.out.println(name+" charges.");
    powerLevel += amount;
} }
```



'extends' signals inheritance from Robot class

```
public class TranslationRobot extends Robot {
  // class has everything that Robot has implicitly
  String substitute; //and more features

  TranslationRobot(String substitute) {
    this.substitute = substitute;
  }

  void translate(String phrase) {
    this.talk(phrase.replaceAll("a", substitute));
  }

  @Override
  void charge(float amount) { //overriding
    System.out.println(name + " charges double.");
    powerLevel = powerLevel + 2 * amount;
} }
```

added method here

parent method is replaced here