Comp 401
Interfaces

Instructor: Prasun Dewan

Prerequisite

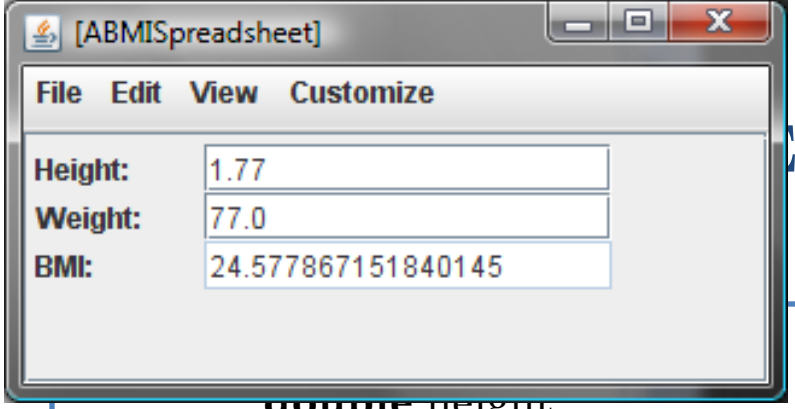
- State Properties

Interfaces

- Define contracts between our users and implementers
- Optional – they may not be used
- Good style to use them

Motivation: Two Ways of Doing the BMI Spreadsheet

- ABMISpreadsheet is one way to implement the spreadsheet user-interface
- Let us create AnotherBMISpreadsheet to illustrate another way
- Difference is in number of variables used



IMPLEMENTATION

Height

Weight

BMI

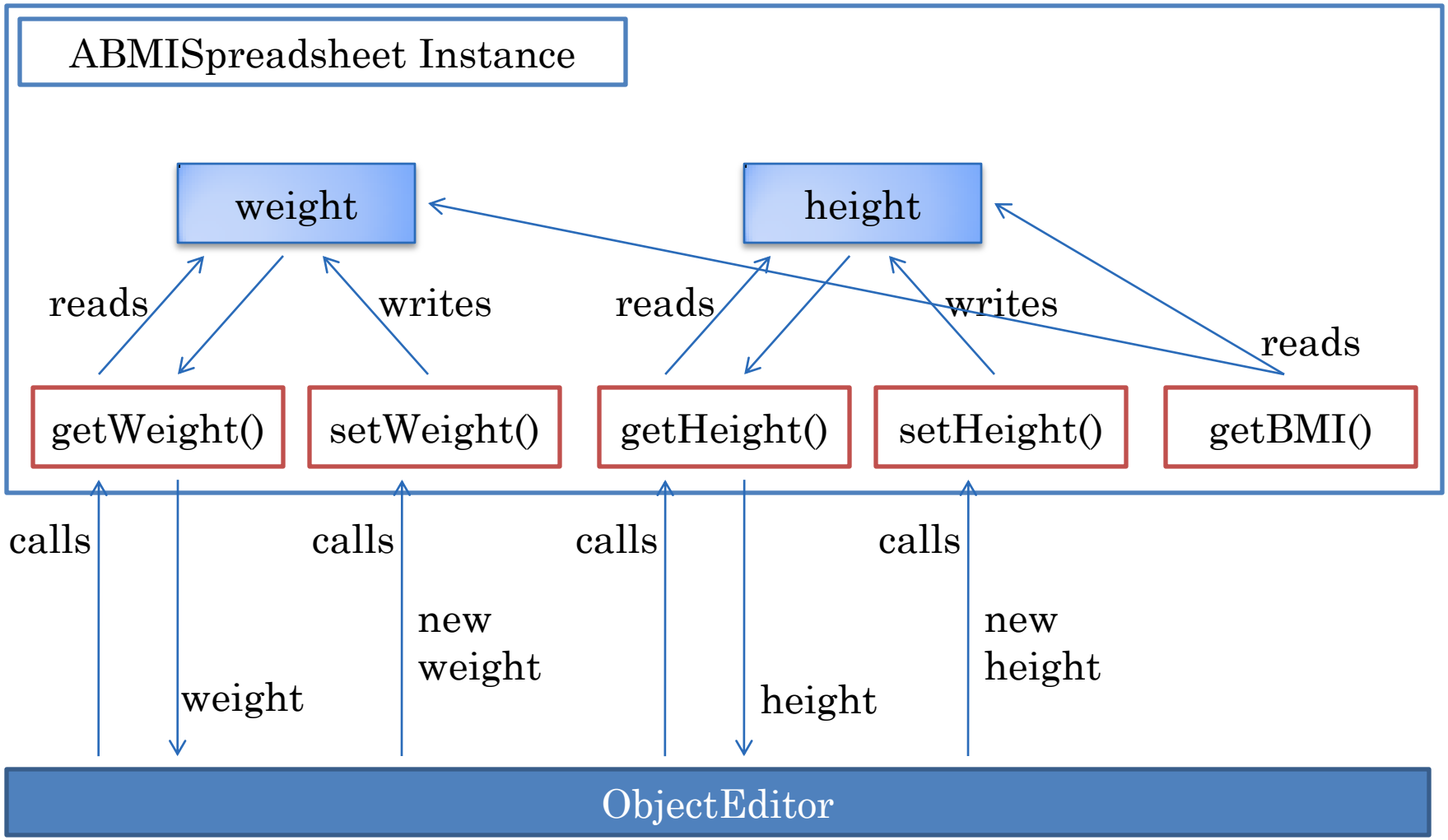
```
double height;
public double getHeight() {
        return height;
}
public void setHeight(double newHeight) {
        height = newHeight;
}
double weight;
public double getWeight() {
        return weight;
}
public void setWeight(double newWeight) {
        weight = newWeight;
}
public double getBMI() {
        return weight/(height*height);
}
}
```

Editable
Independent
Stored

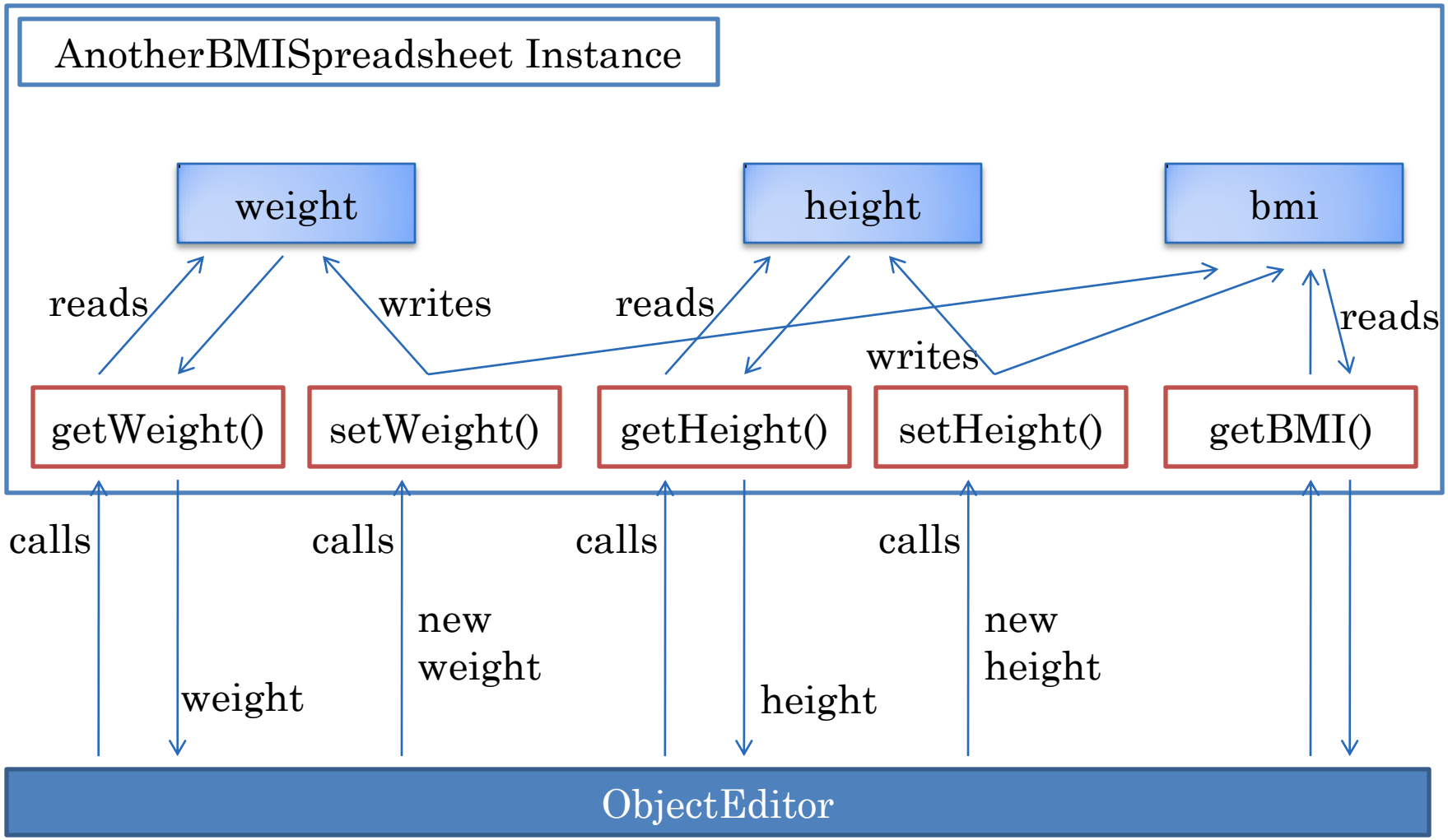
Editable
Independent
Stored

Read-only
Dependent
Computed

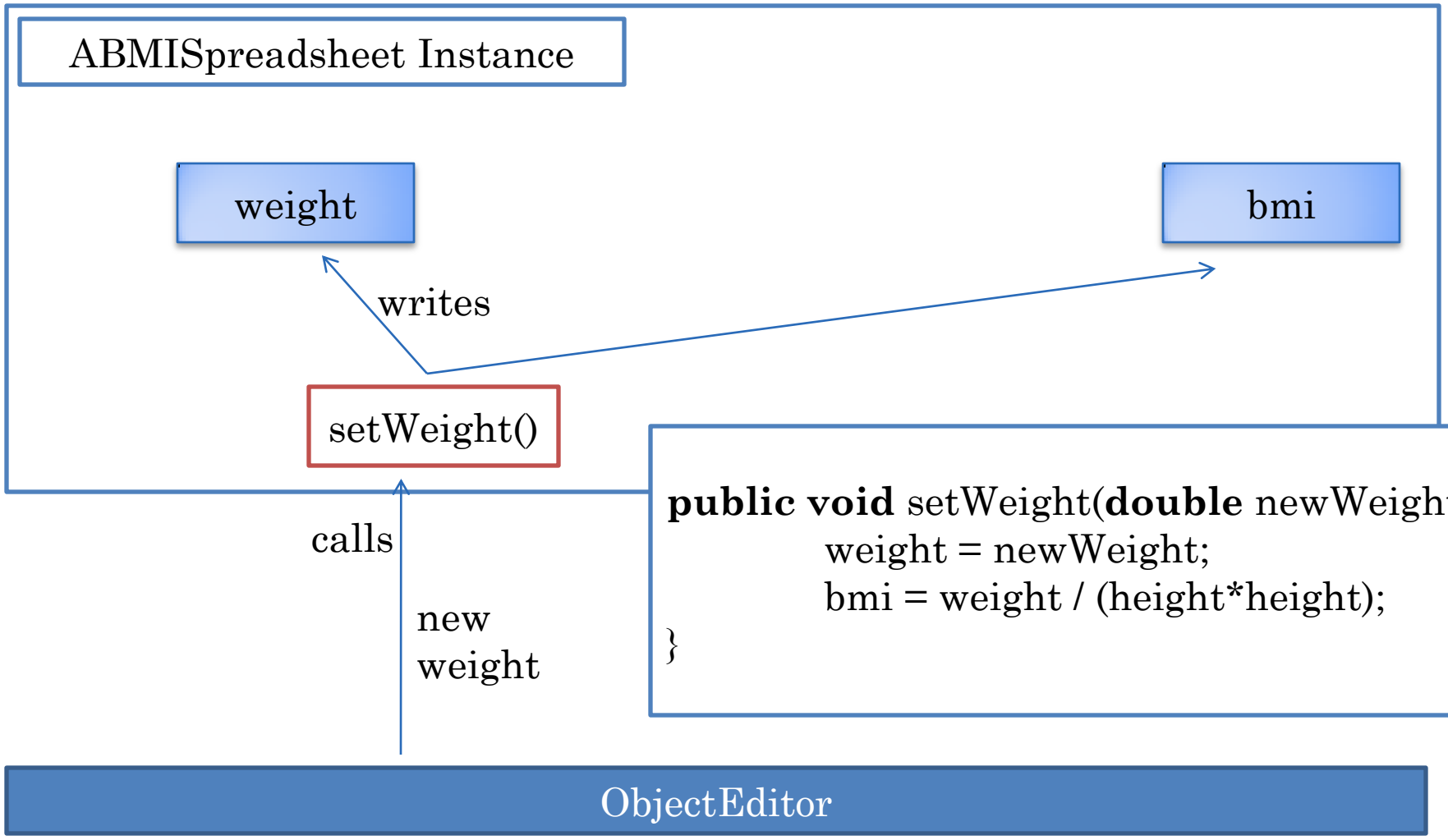
ABMISpreadsheet



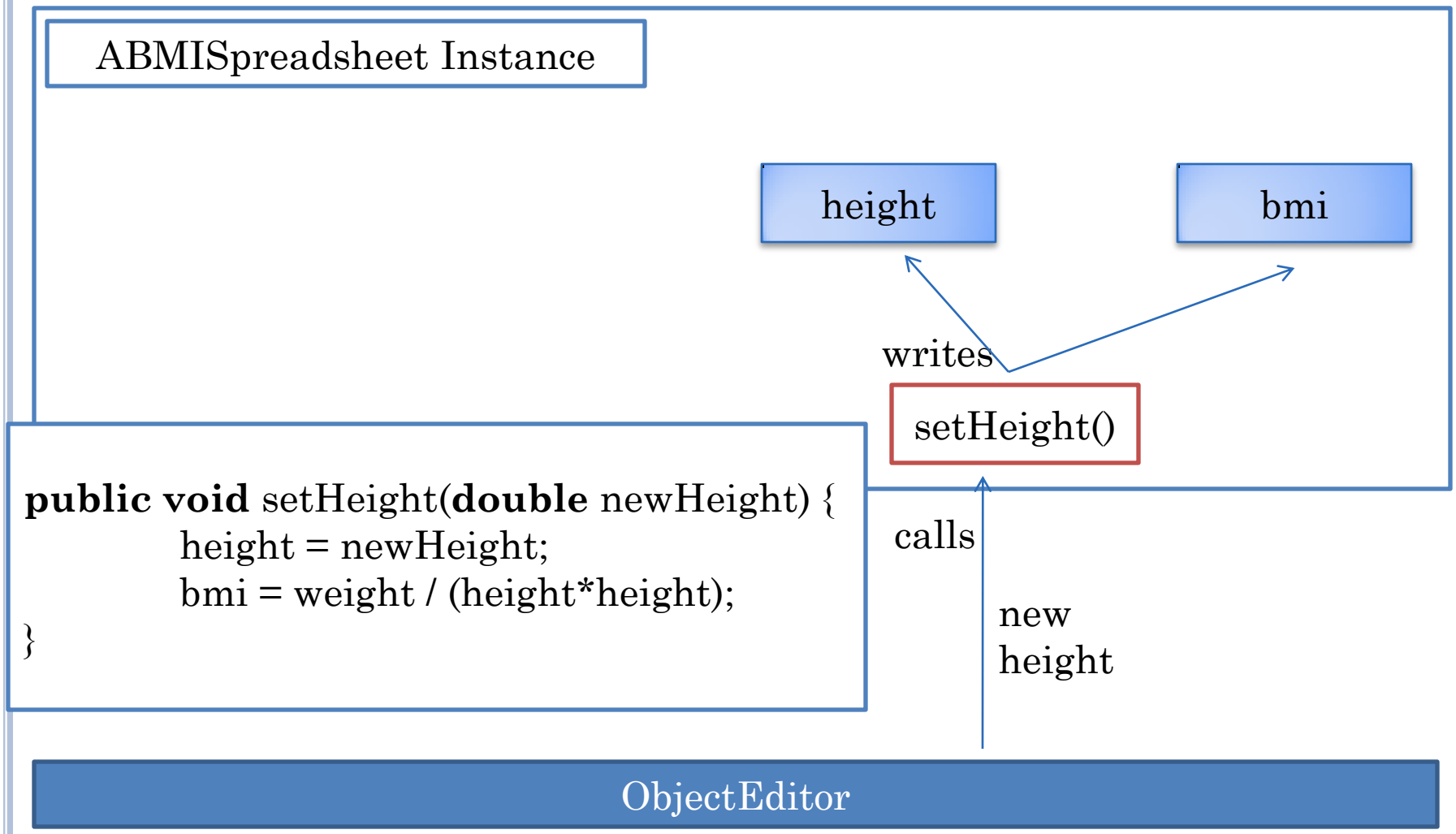
AnotherBMISpreadsheet



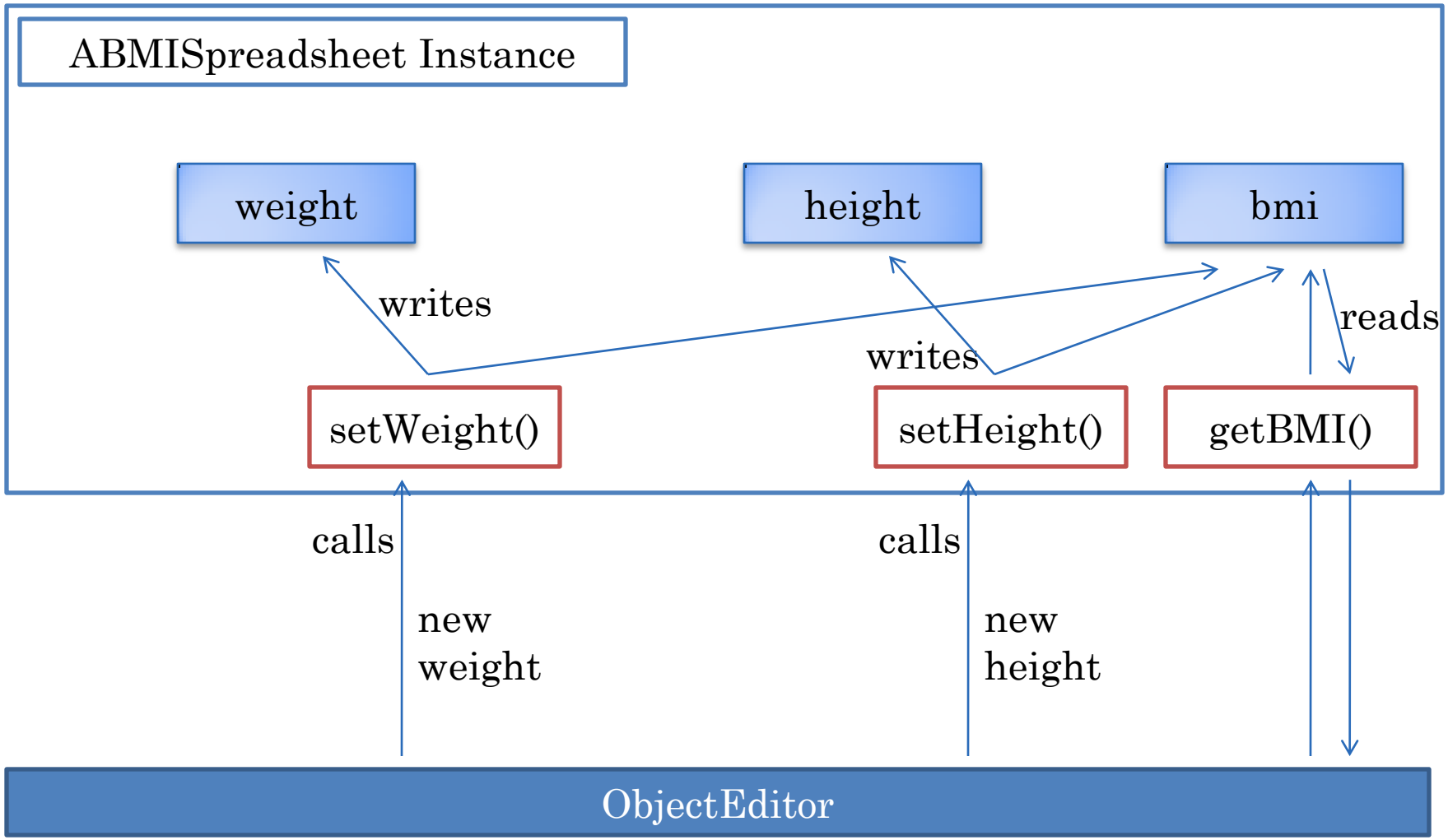
setWeight()


ABMISpreadsheet Instance
weight
bmi
writes
setWeight()
calls
new
weight
public void setWeight(double newWeight) {
weight = newWeight;
bmi = weight / (height*height);
}
ObjectEditor

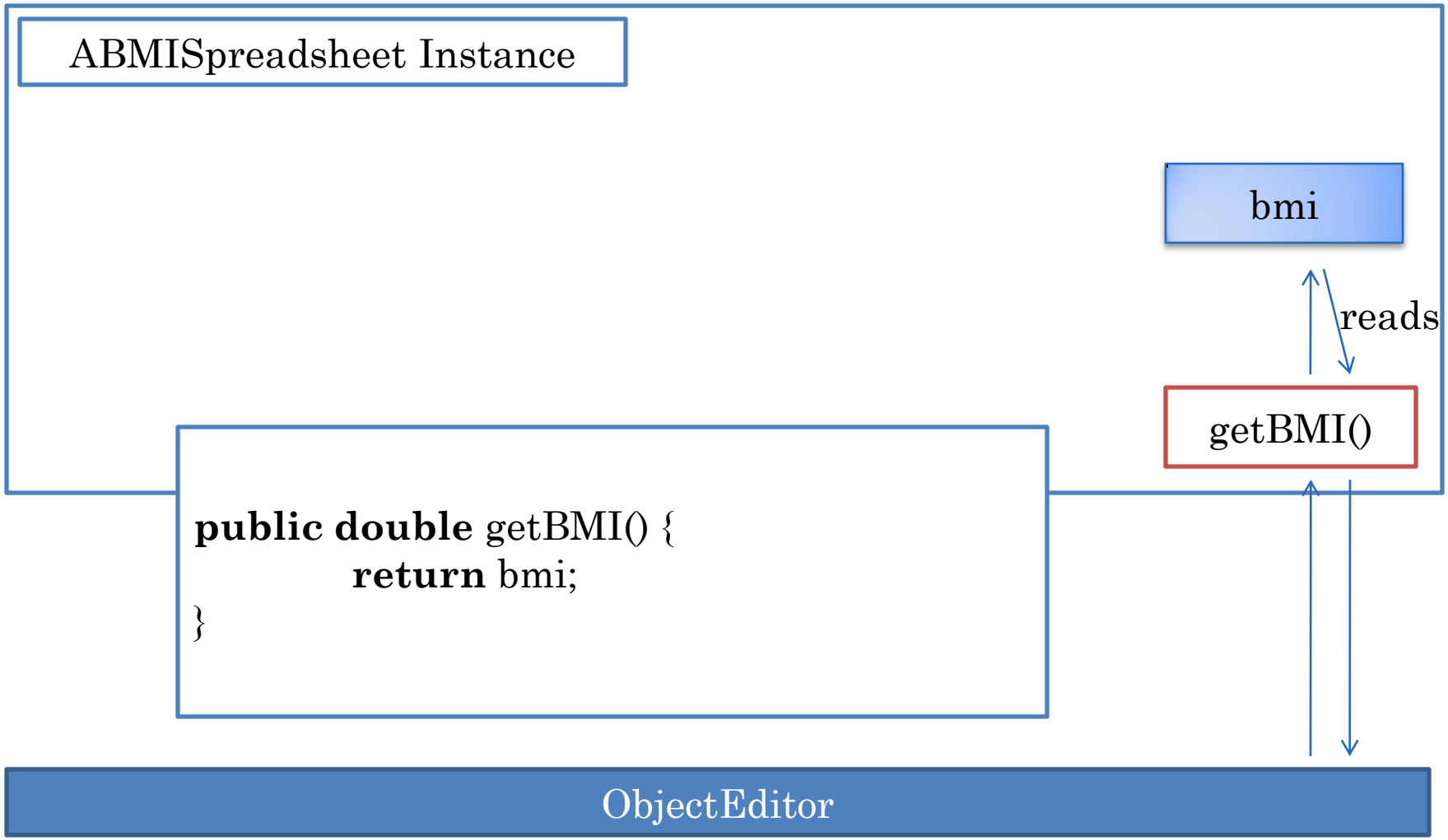
setHeight()


ABMISpreadsheet Instance
height
bmi
writes
setHeight()
public void setHeight(double newHeight) {
height = newHeight;
bmi = weight / (height*height);
}
calls
new
height
ObjectEditor

Methods that Change



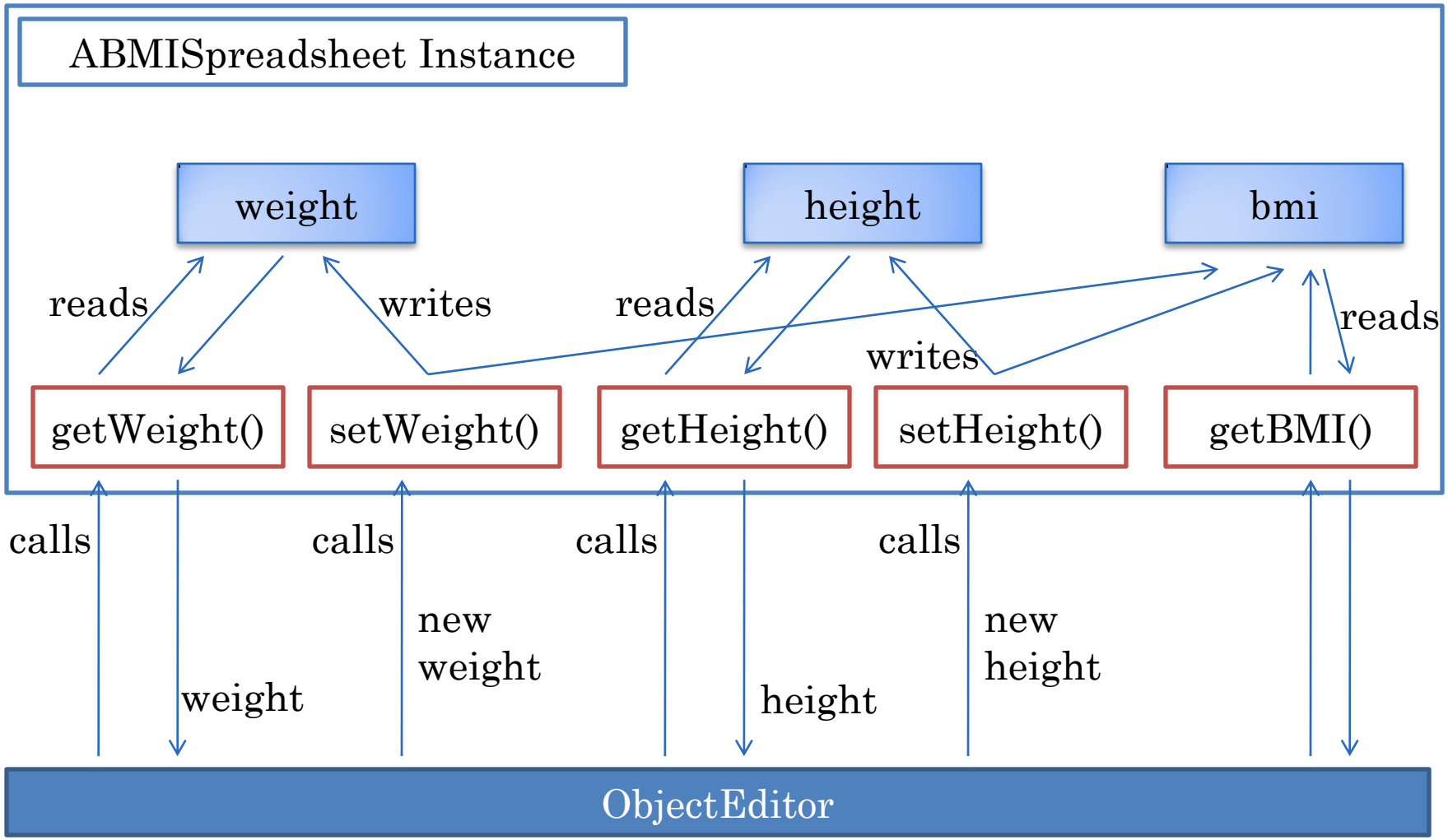
getBMI()


ABMISpreadsheet Instance
bmi
reads
getBMI()
public double getBMI() {
return bmi;
}
ObjectEditor

ANOTHERBMISPPREADSHEET

```
public class AnotherBMISpreadsheet {
    double height, weight, bmi;
    public double getHeight() {
        return height;
    }
    public void setHeight(double newHeight) {
        height = newHeight;
        bmi = weight/(height*height);
    }
    public double getWeight() {
        return weight;
    }
    public void setWeight(double newWeight) {
        weight = newWeight;
        bmi = weight/(height*height);
    }
    public double getBMI() {
        return bmi;
    }
}
```

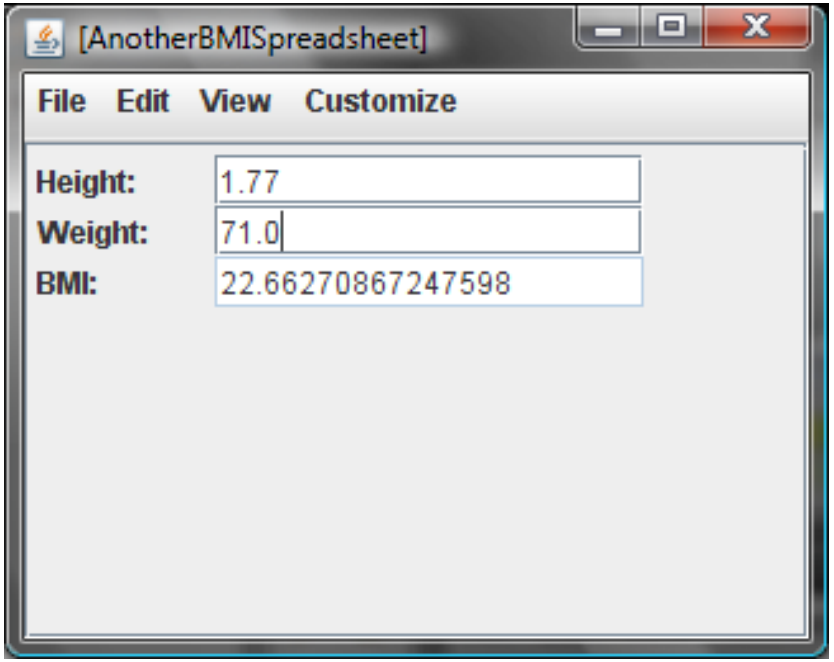
Graphical Algorithm

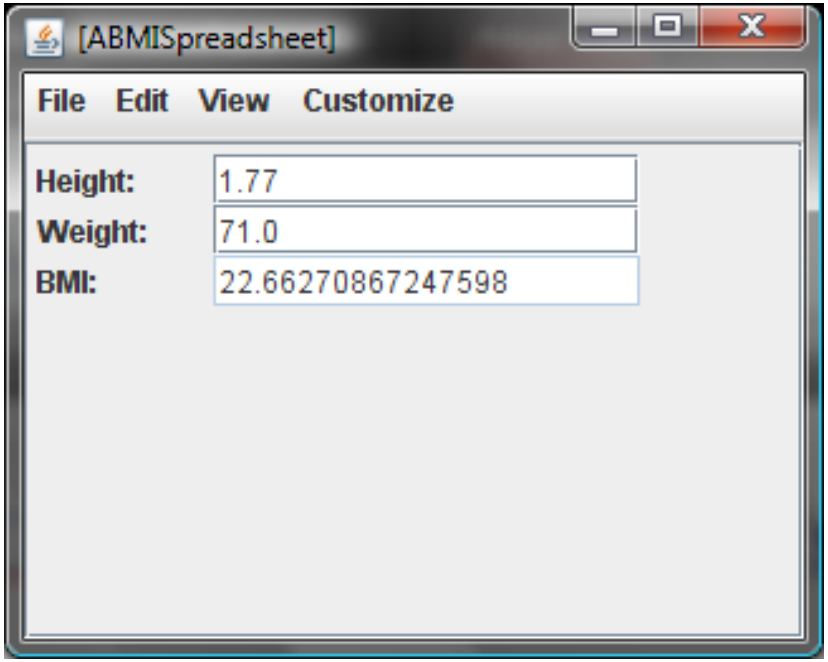


ObjectEditor User Interface?

```
public class AnotherBMISpreadsheet {
    double height, weight, bmi;
    public double getHeight() {
        return height;
    }
    public void setHeight(double newHeight) {
        height = newHeight;
        bmi = weight/(height*height);
    }
    public double getWeight() {
        return weight;
    }
    public void setWeight(double newWeight) {
        weight = newWeight;
        bmi = weight/(height*height);
    }
    public double getBMI() {
        return bmi;
    }
}
```

ObjectEditor User Interfaces





Similarities in the Two Classes

```
public class AnotherBMISpreadsheet {
    double height, weight, bmi;
    public double getHeig
        return height;
    }
    public void setHeight(double
            newHeight) {
        height = newHeight;
        bmi = weight/(heig
    }
    public double getWeight() {
        return weight;
    }
    public void setWeight(double
            newWeight) {
        weight = newWeight
        bmi = weight/(height*height);
    }
    p
    }
}
```

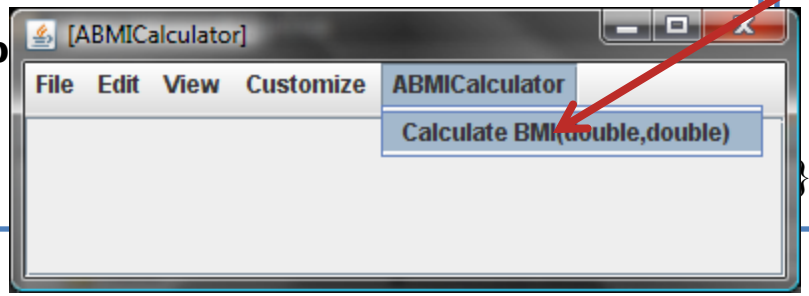
```
public class ABMISpreadsheet {
    double height;
                Height() {
    }
    public void setHeight(double
            newHeight) {
        height = newHeight;
                ght() {
                uble
    }
}
```

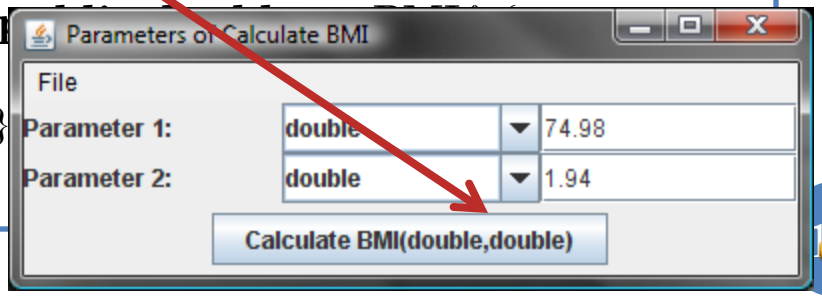
Same headers, signatures

Signature = Header – parameter names

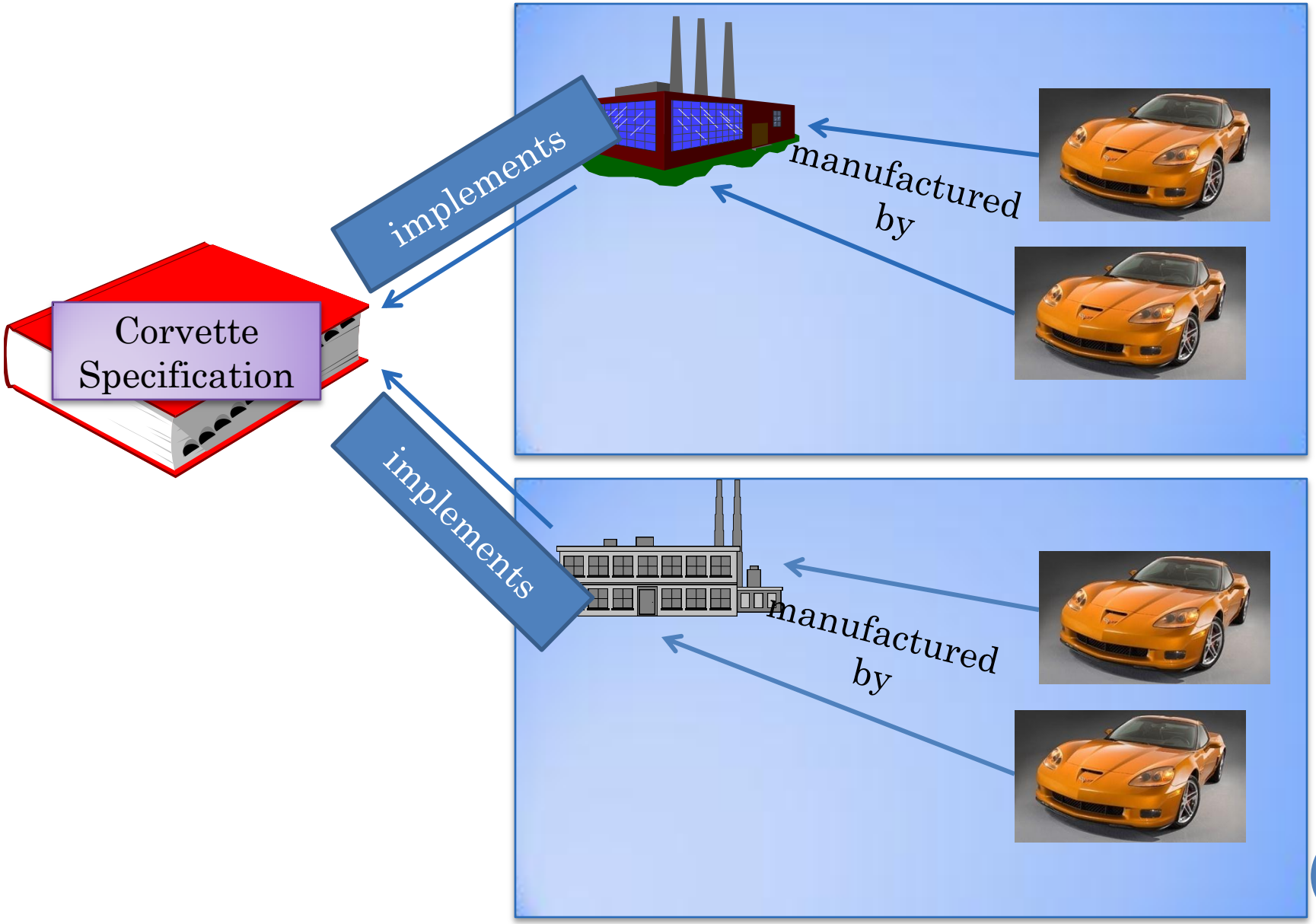
public void setHeight(**double**)

OE Beautified Signature

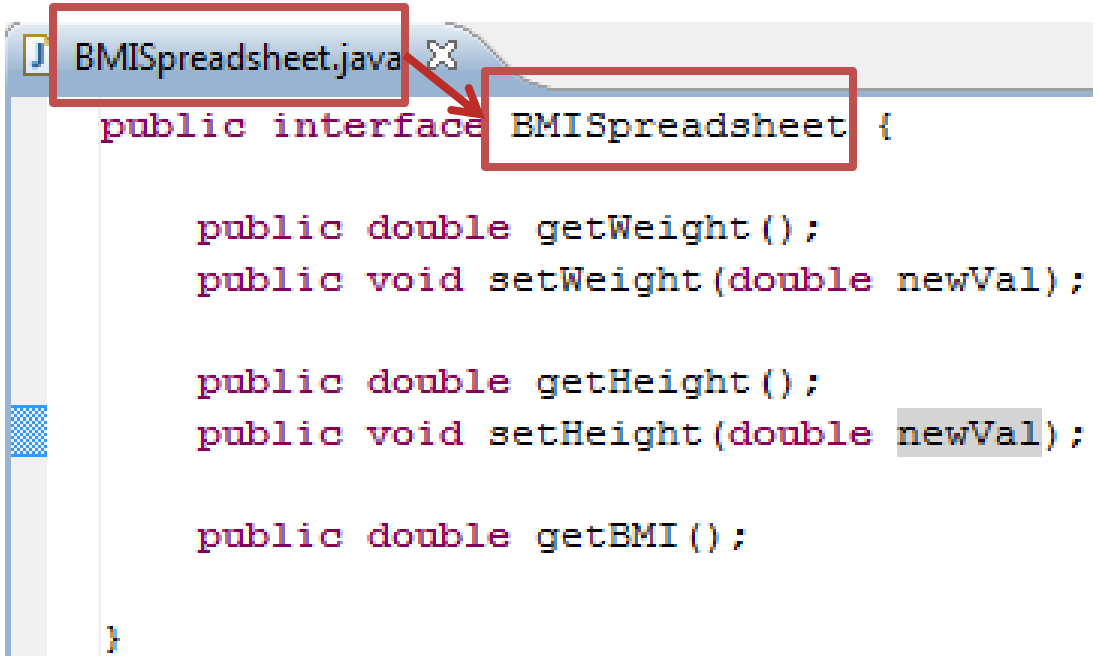




Real-World Analogy



Interface


BMISpreadsheet.java
public interface BMISpreadsheet {
public double getWeight();
public void setWeight(double newVal);
public double getHeight();
public void setHeight(double newVal);
public double getBMI();
}

Implementing an Interface

Contract

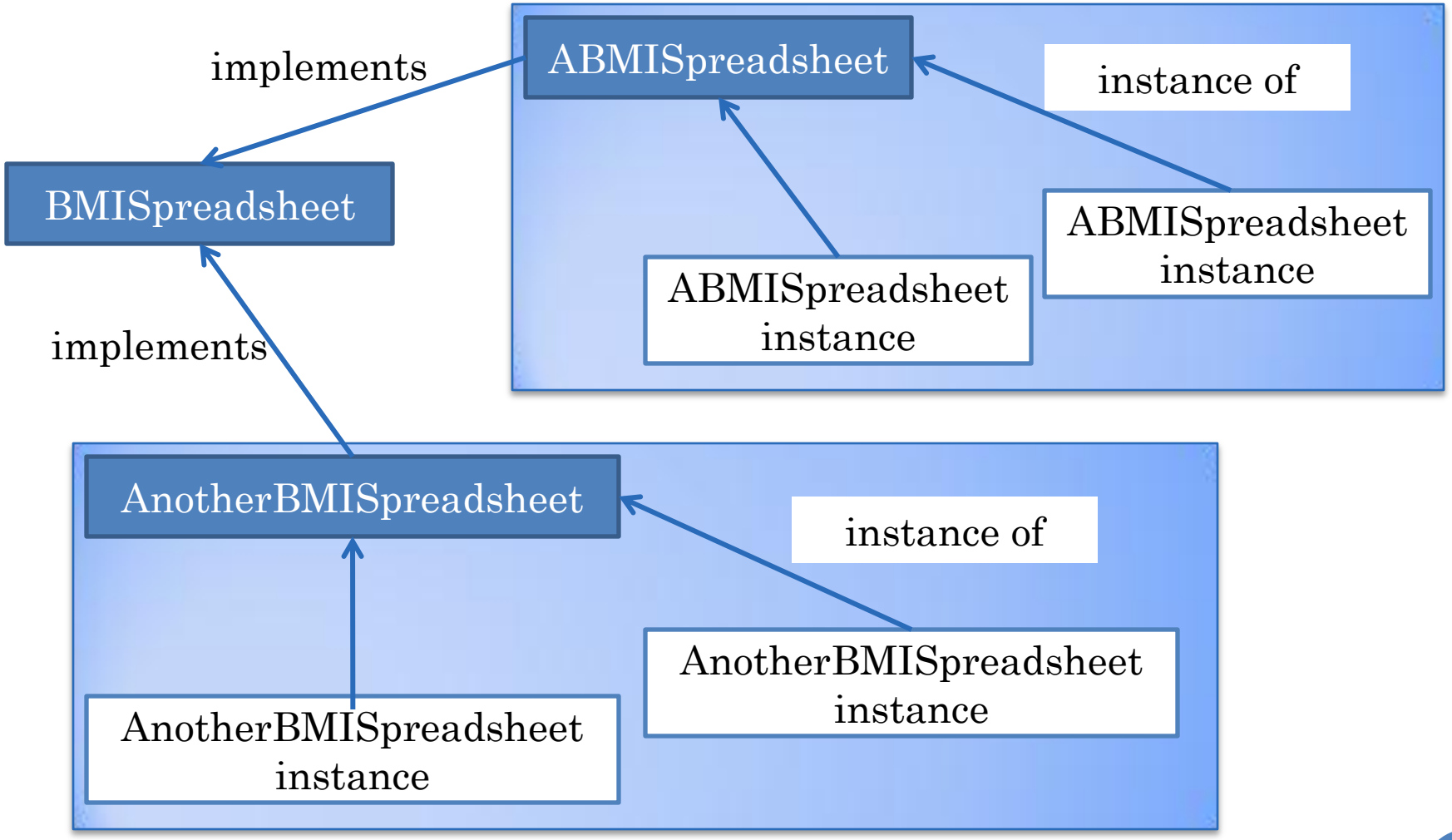
```
public class AnotherBMISpreadsheet implements BMISpreadsheet {
    double height, weight, bmi;
    public double getHeight() {
        return height;
    }
    public void setHeight(double newHeight) {
        height = newHeight;
        bmi = weight/(height*height);
    }
    …
```

Parameter names never matter to Java

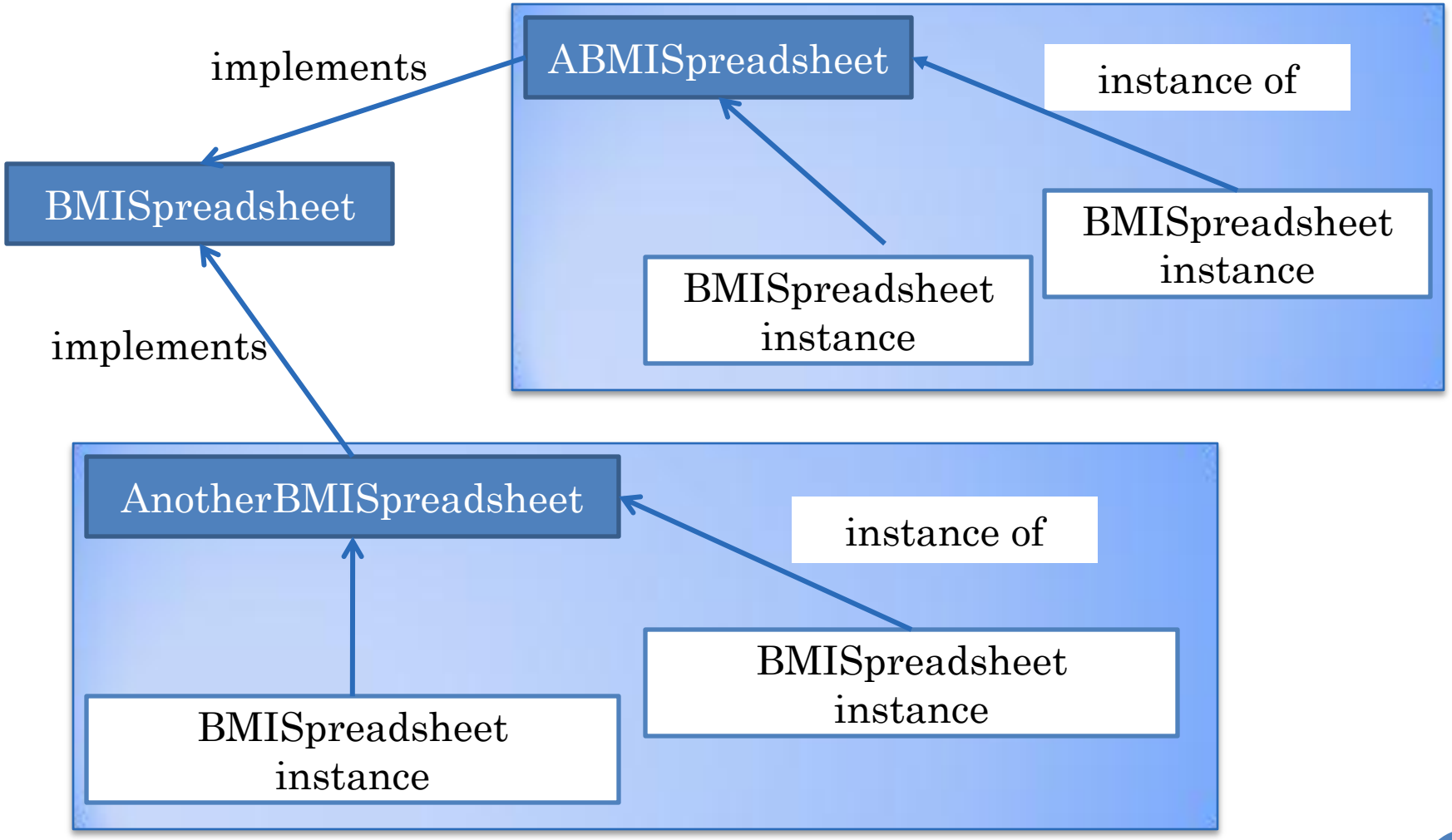
BMISpreadsheet.java

```
public interface BMISpreadsheet {

    public double getWeight();
    public void setWeight(double newVal);

    public double getHeight();
    public void setHeight(double newVal);

    public double getBMI();

}
```

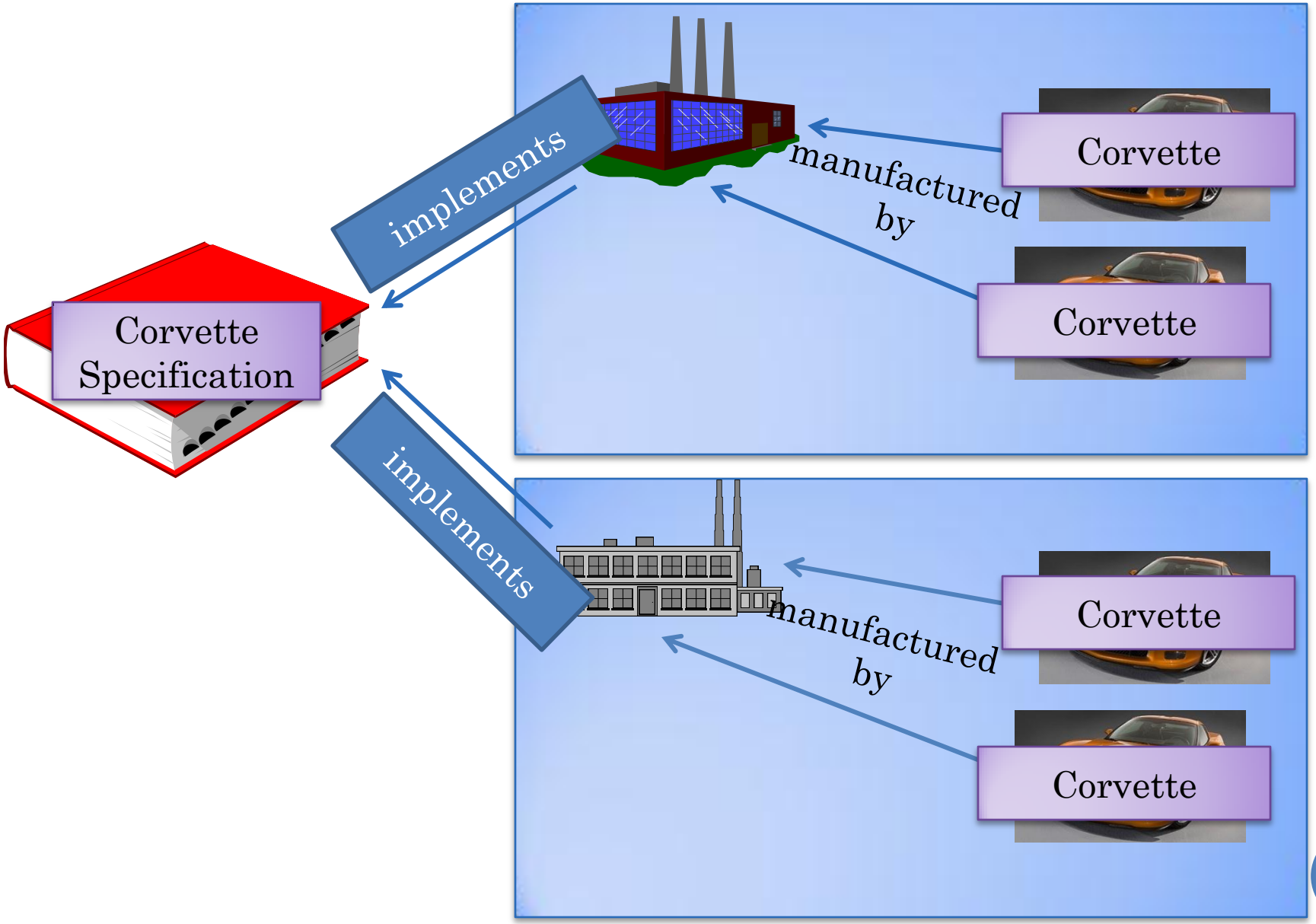
Interface


implements
ABMISpreadsheet
instance of
BMISpreadsheet
ABMISpreadsheet instance
ABMISpreadsheet instance
implements
AnotherBMISpreadsheet
instance of
AnotherBMISpreadsheet instance
AnotherBMISpreadsheet instance

Using Interfaces to Classify



Using Car Specifications to Classify



Using Interfaces to Type

```
public class BMISpreadsheetUser {
    public static void main(String[] args) {
        BMISpreadsheet bmiSpreadsheet = new
ABMISpreadsheet(1.77, 75);
        System.out.println(bmiSpreadsheet.getBMI());
        bmiSpreadsheet = new AnotherBMISpreadsheet();
        bmiSpreadsheet.setHeight(1.77);
        bmiSpreadsheet.setWeight(75);
        System.out.println(bmiSpreadsheet.getBMI());
    }
}
```

Interface as type

Same variable assigned instances of two different classes

Type Checking

```
public class BMISpreadsheetUser {
    public static void main(String[] args) {
        BMISpreadsheet bmiSpreadsheet = new
ABMISpreadsheet(1.77, 75);
        System.out.println(bmiSpreadsheet.obtainBMI());
        bmiSpreadsheet = new AnotherBMISpreadsheet();
        bmiSpreadsheet.setHeight(1.77);
        bmiSpreadsheet.setWeight(75);
        System.out.println(bmiSpreadsheet.getBMI());
    }
}
```

Interface Methods Considered in Type Checking

```
public class BMISpreadsheetUser {
    public static void main(String[] args) {
        BMISpreadsheet bmiSpreadsheet = new
ABMISpreadsheet(1.77, 75);
        System.out.println(bmiSpreadsheet.getBMI());
        bmiSpreadsheet = new AnotherBMISpreadsheet();
        bmiSpre
        bmiSpre
        System.o
    }
}
```

BMISpreadsheet.java

```
public interface BMISpreadsheet {

    le getWeight();
    setWeight(double newVal);

    le getHeight();
    public void setHeight(double newVal);

    public double getBMI();

}
```

Not defined in interface so illegal, even though defined in class

Yet Another Spradsheet Class

```
public class ABMISpreadsheetAndCalculator implements ??? {
    double height, weight, bmi;
    public double getHeight() {
        return height;
    }
    public void setHeight(double newHeight) {
        height = newHeight;
        bmi = calculateBMI(height, weight);
    }
    public double getWeight() {
        return weight;
    }
    public void setWeight(double newWeight) {
        weight = newWeight;
        bmi = calculateBMI(height, weight);
    }
    public double getBMI() {
        return bmi;
    }
    public double calculateBMI(double height, double weight) {
        return weight/(height*height);
}
```

Implementing Multiple Interfaces

```
public class ABMISpreadsheetAndCalculator implements BMISpreadsheet, BMICalculator{
    double height, weight, bmi;
    public double getHeight() {
        return height;
    }
    public void setHeight(double newHeight) {
        height = newHeight;
        bmi = calculateBMI(height, weight);
    }
    public double getWeight() {
        return weight;
    }
    public void setWeight(double newWeight) {
        weight = newWeight;
        bmi = calculateBMI(height, weight);
    }
    public double getBMI() {
        return bmi;
    }
    public double calculateBMI(double height, double weight) {
        return weight/(height*height);
}
```

BMICalculator Interface

```
public interface BMICalculator {
   public double calculateBMI(double height, double weight);
}
```

ABMICalculatorWithInterface

```
public class ABMICalculatorWithInterface implements BMICalculator {
  public double calculateBMI(double height, double weight) {
    return weight/(height*height);
  }
}
```

Typing

```
 public static void main (String[] args) {
    BMICalculator bmiCalculator = new ABMISpreadsheetAndCalculator();
    BMISpreadsheet bmiSpreadsheet = new ABMISpreadsheetAndCalculator();
    double bmi = bmiCalculator.calculateBMI(1.77, 75);
    bmi = bmiSpreadsheet.getBMI();
//     bmi = bmiCalculator.getBMI();
//     bmi = bmiSpreadsheet.calculateBMI(1.77, 75);

    }
```

```
public static void main (String[] args) {
  BMICalculator[] bmiCalculators = {new ABMISpreadsheetAndCalculator(),
                                    new ABMICalculatorWithInterface()};
  }
```

Car Analogy

- A car is characterized by
  - Its make
  - License plate
  - Registration
- Licensing authority groups car by the registration or license plate

Cannot Instantiate Specification

- Cannot order a car from a specification
  - Must order from factory
  - A car defined by Corvette specification ordered from factory implementing the specification

- Cannot instantiate interface
  - Must instantiate class
  - BMISpreadsheet instance created by instantiating class implementing interface

Interface as a Syntactic Specification

```
public class ABMISpreadsheet implements BMISpreadsheet{
    double height;
    public double getHeight() {
        return height;
    }
    public void setHeight(double newHeight) {
        height = newHeight;
    }
    double weight;
    public double getWeight() {
        return weight;
    }
    public void setWeight(double newWeight) {
        weight = newWeight;
    }
    public double getBMI() {
        return weight/(height*height);
    }
}
```

Interface as a Syntactic Specification

```
public class ABMISpreadsheet implements BMISpreadsheet{
    double height;
    public double getHeight() {
        return height;
    }
    public void setHeight(double newHeight) {
        height = newHeight;
    }
    double weight;
    public double getWeight() {
        return weight;
    }
    public void setWeight(double newWeight) {
        weight = newWeight;
    }
    public double getBMI() {
        return 13450;
    }
}
```

Syntactic Contract

Bombay Market Index

Interface Required

- Define interfaces for
  - All classes (that are instantiated)
  - Some are not
  - Each public method of a class should be in some interface it implements

Impact of Differences in the Two Classes

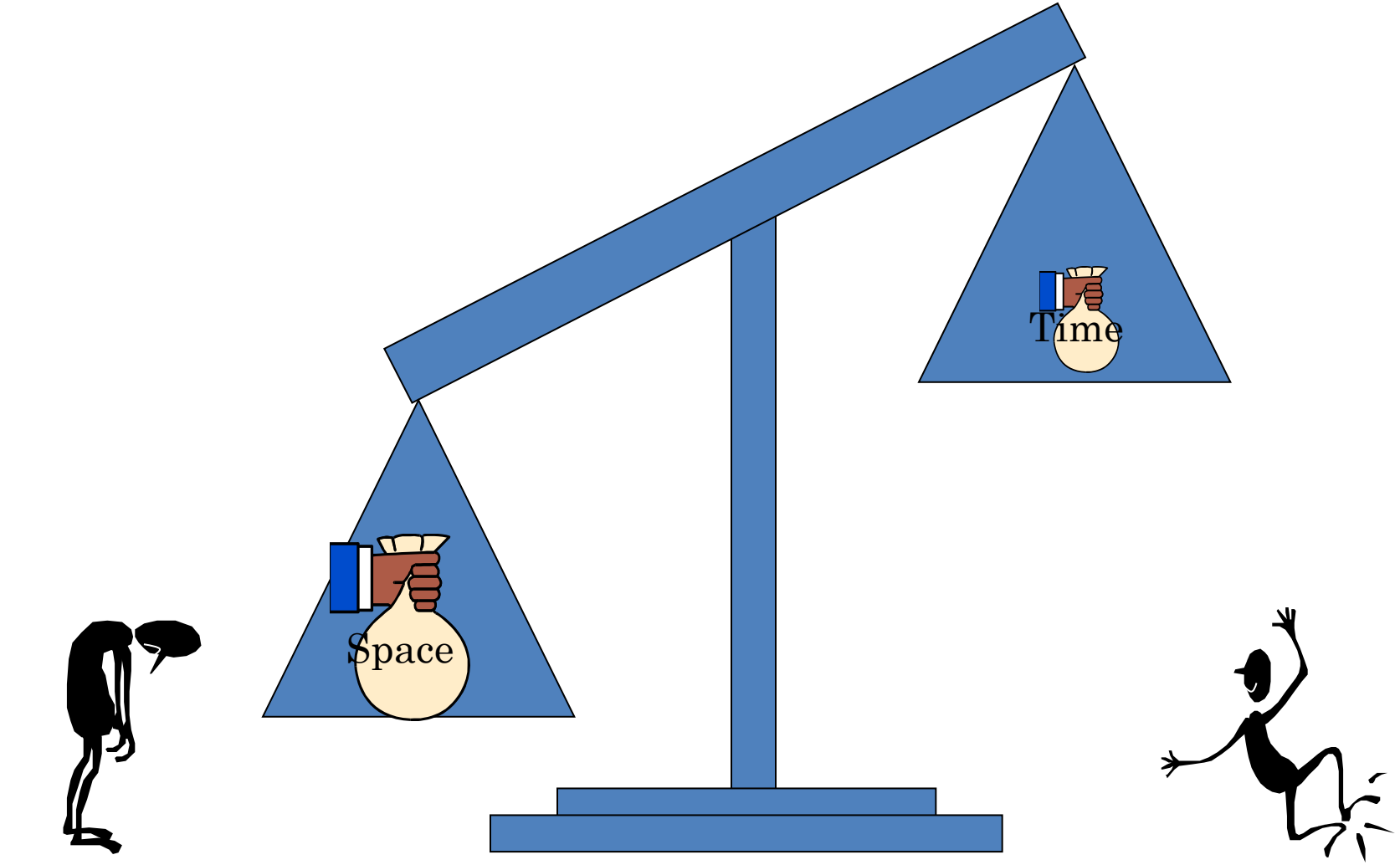
```
public class AnotherBMISpreadsheet
implements BMISpreadsheet {
    double height, weight, bmi;
    public double getHeight() {
        return height;
    }
    public void setHeight(double
                newHeight) {
        height = newHeight;
        bmi = weight/(height*height);
    }
    public double getWeight() {
        return weight;
    }
    public void setWeight(double
                newWeight) {
        weight = newWeight;
        bmi = weight/(height*height);
    }
    public double getBMI() {
        return bmi;
    }
}
```

```
public class ABMISpreadsheet
implements BMISpreadsheet {
    double height;
    public double getHeight() {
        return height;
    }
    public void setHeight(double
                newHeight) {
        height = newHeight;
    }
    double weight;
    public double getWeight() {
        return weight;
    }
    public void setWeight(double
                newWeight) {
        weight = newWeight;
    }
    public double getBMI() {
        return weight/(height*height);
    }
}
```

ABMISpreadsheet vs. AnotherBMISpreadsheet

- AnotherBMISpreadsheet harder to program
  - Computation of bmi done in multiple setters.
  - These setters have "side effects"
- AnotherBMISpreadsheet does eager evaluation
  - Value computed may never be used.
- ABMISpreadsheet uses less space (variables)
- Getter methods of AnotherBMISpreadhseet are faster
- Setter methods of ABMISpreadsheet are faster
- Usually getter methods are called more often that setter methods
  - e.g. when ObjectEditor refresh command is executed
- Typically AnotherBMISpreadsheet will be faster, overall

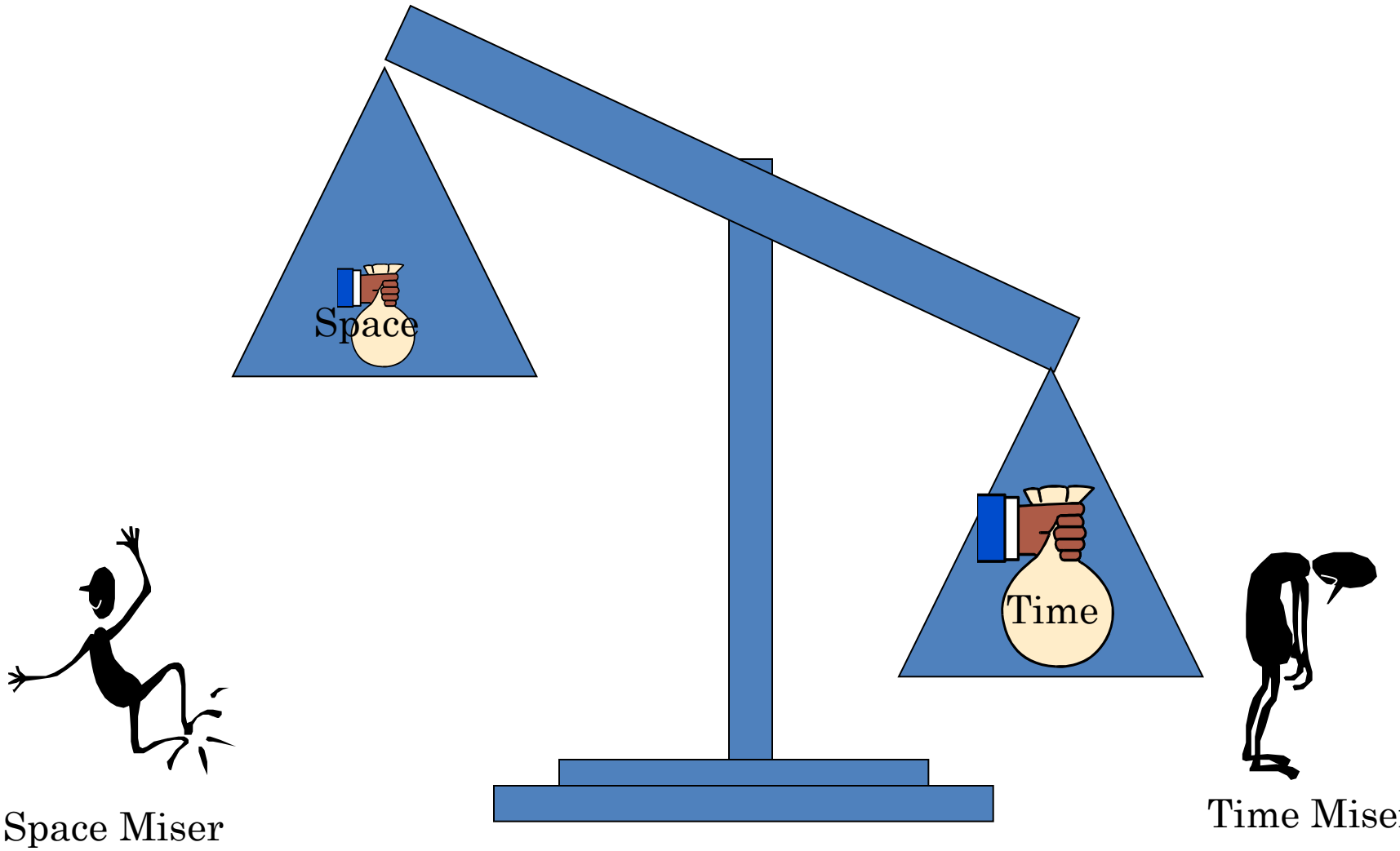
Time-Space Tradeoff



Space Miser

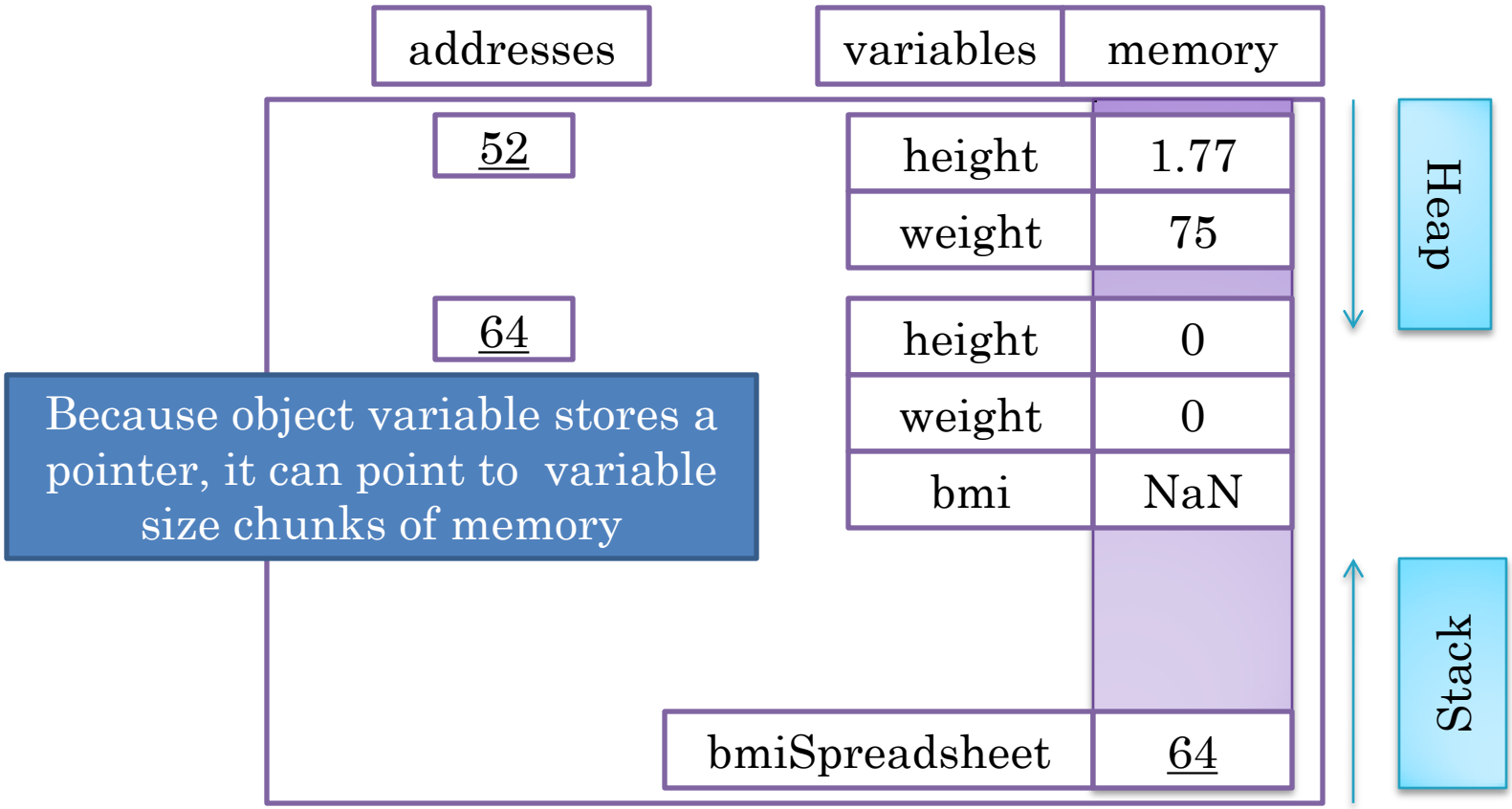
Time Miser

Time-Space Tradeoff



Variable Pointing to AnotherBMISpreadhseet Instance

```
BMISpreadsheet bmiSpreadsheet = new ABMISpreadsheet(1.77, 75);
System.out.println(bmiSpreadsheet.getBMI());
bmiSpreadsheet = new AnotherBMISpreadsheet();
```



Relating Interface and Class Names

Class Name:

<Qualifier><Interface>
- ABMISpreadsheet
- ASpaceEfficientBMISpreadsheet
- SpaceEfficientBMISpreadsheet

<Interface><Qualifier> Impl
- BMISpreadsheetImpl
- BMISpreadsheetSpaceEfficientImpl

Interface Name:

<ClassName>Interface
- ABMISpreadsheetInterface

Assumes only one implementation of interface will be created