CSE1720

Week 05, **Class Meeting 13** (Lecture 09)

Winter 2013 ◆ Tuesday, Feb 05, 2013

YORK UNIVERSITÉ UNIVERSITY

This lecture will be using code from the following package to illustrate concepts:

`game_Lect07Version`

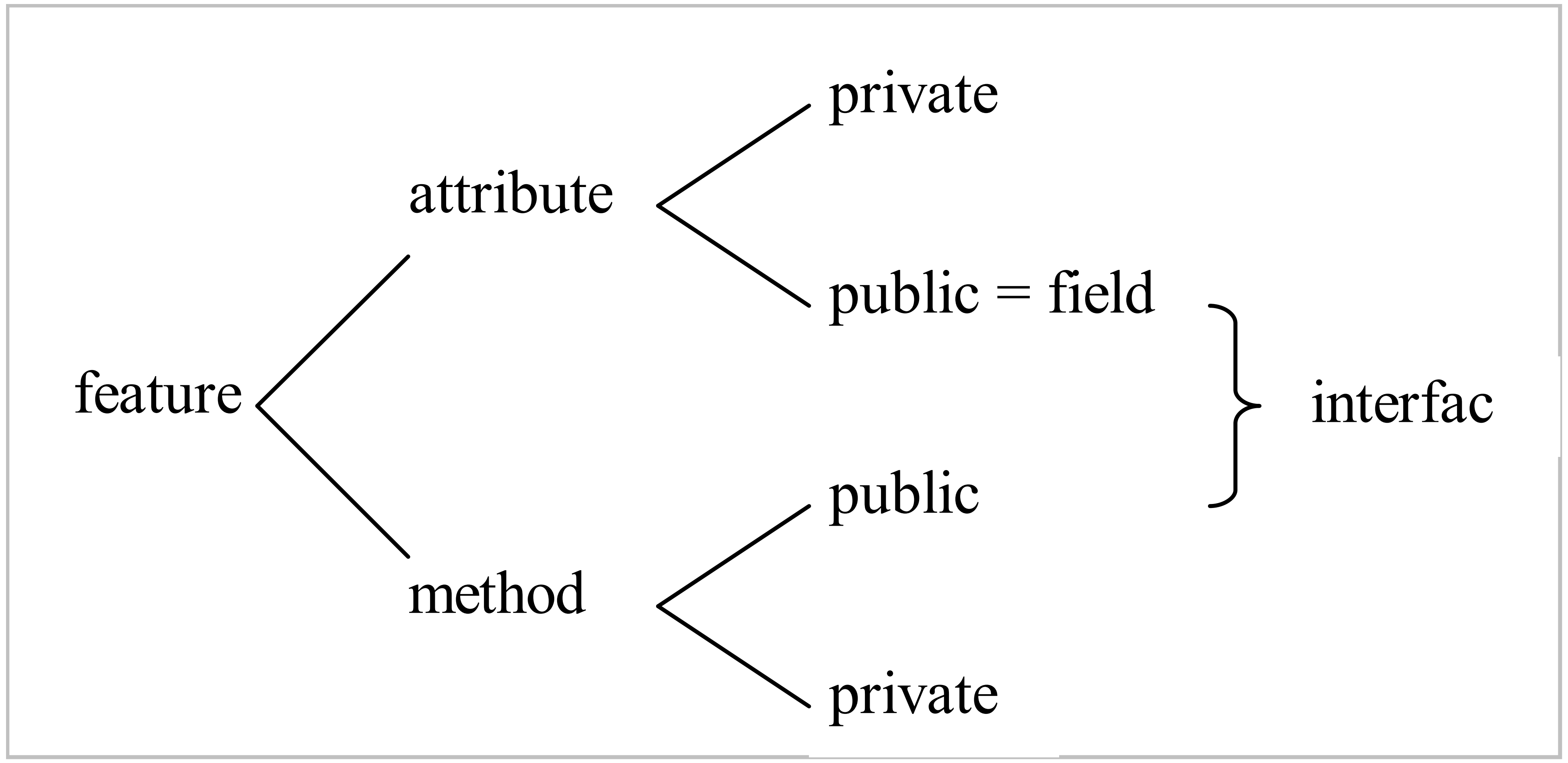
Objectives for this class meeting

- Understand the application architecture, through:
    - UML class diagrams
    - UML object diagrams

Big picture recap…

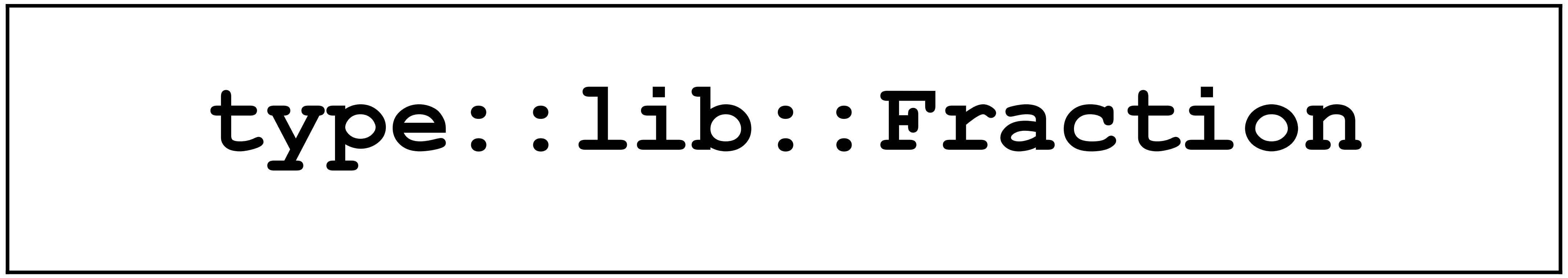
- So far:
  - we have an app that is interactive (user can shoot)
  - needs functionality! We need to add:
    - user should be able to move shooter
    - game should deploy targets
    - game should implement scoring!
  - before we can do this, we need to understand the architecture of the current app

Review of Terminology…



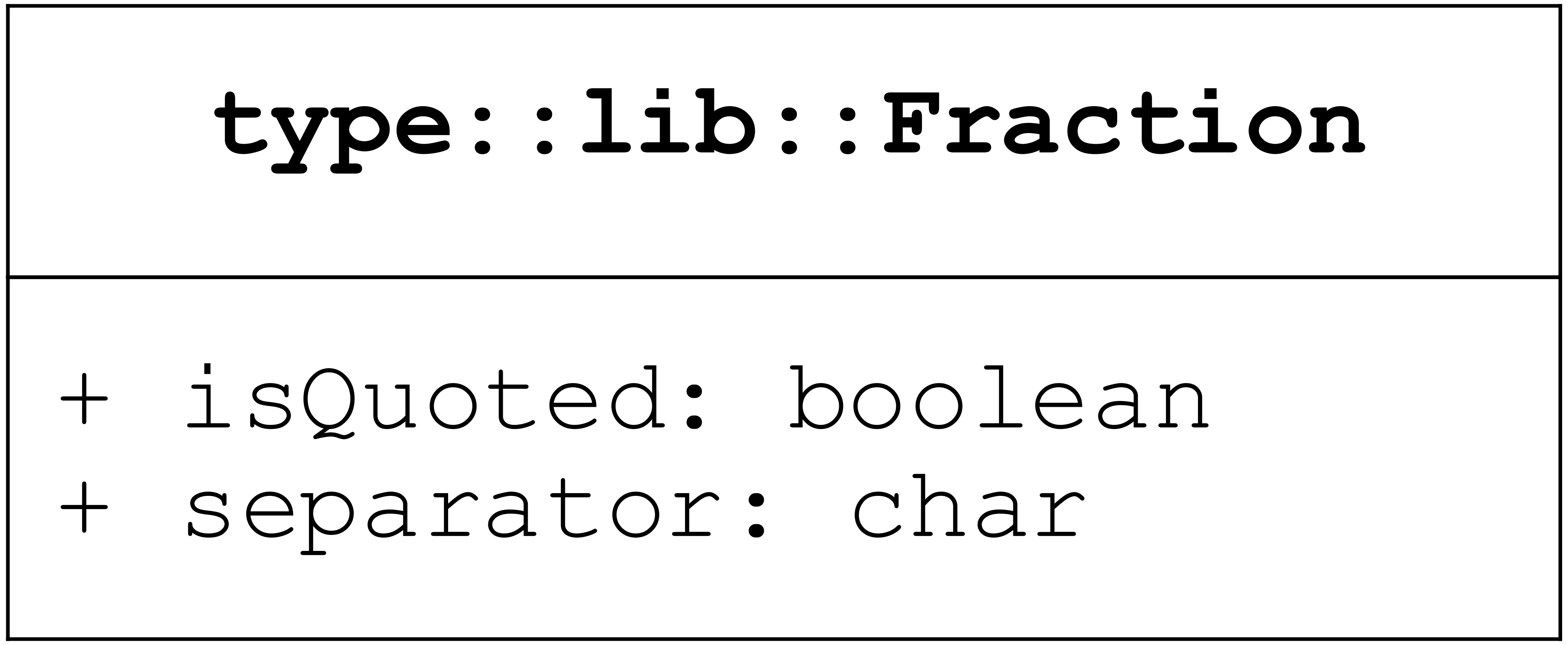
About UML Diagrams…

- see JBA: 7.1.3 Elements of UML
- This is a simple **Class Diagram**. It is appropriate for early iterations



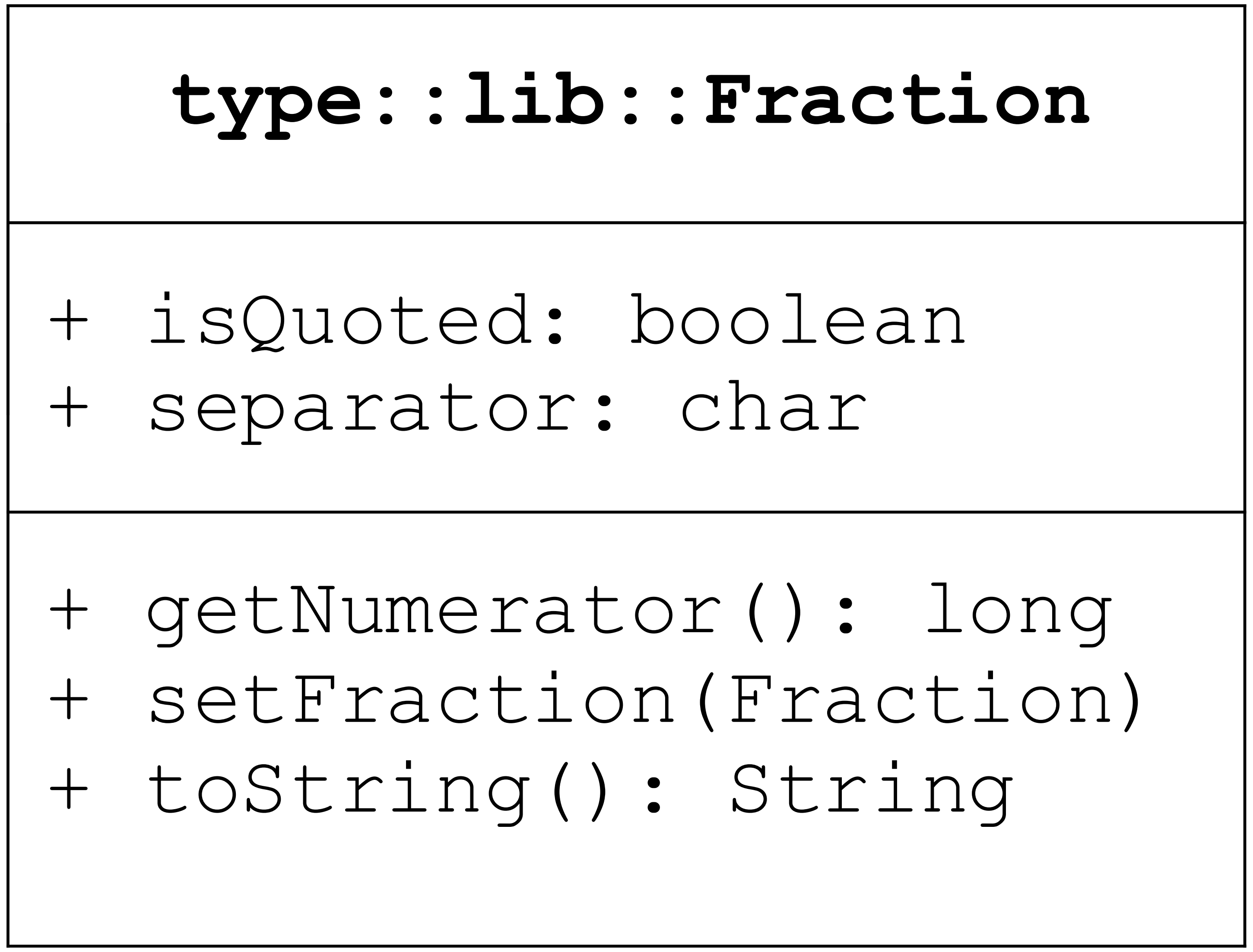
UML Class Diagrams

- A more elaborated Class Diagram that contains details about some class **features**
- in this example, the features are the **fields**



UML Class Diagrams

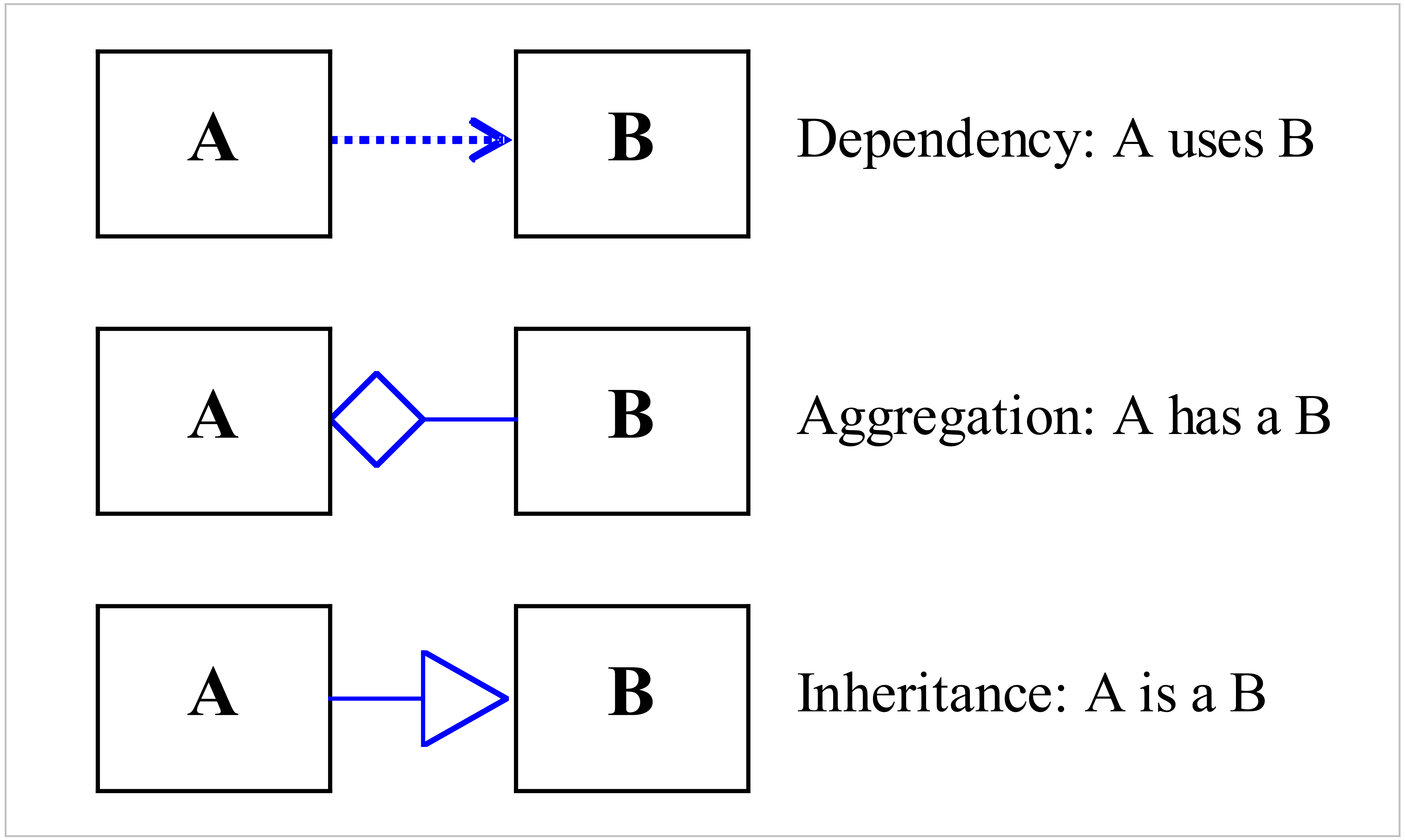
- A yet more elaborated Class Diagram that contains details about more class **features**
- in this example, the features are the **fields**and **public methods**



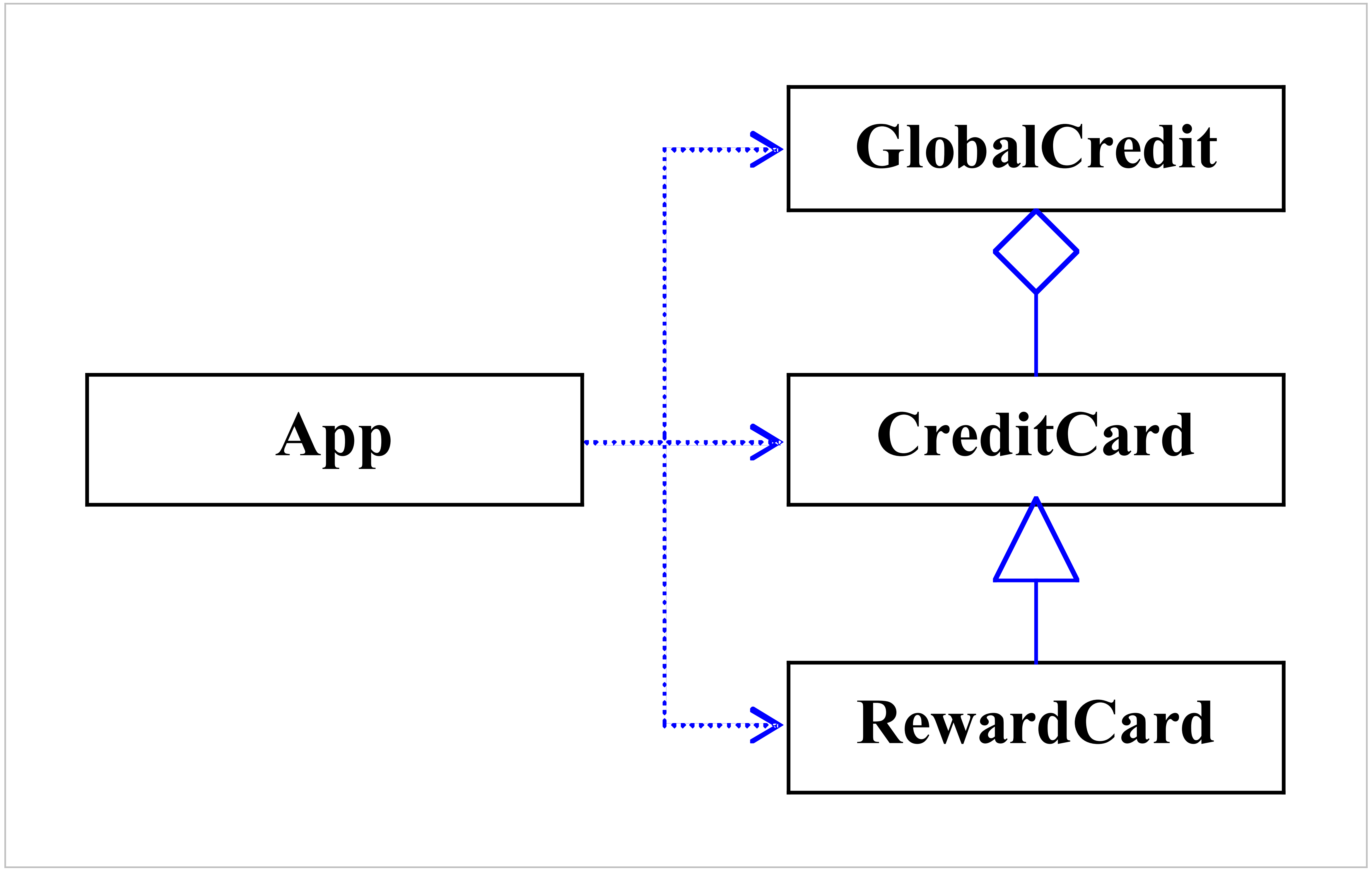
```
type::lib::Fraction
---------------------------
+ isQuoted: boolean
+ separator: char
---------------------------
+ getNumerator(): long
+ setFraction(Fraction)
+ toString(): String
```

UML Class Diagrams

- Notation for relationships



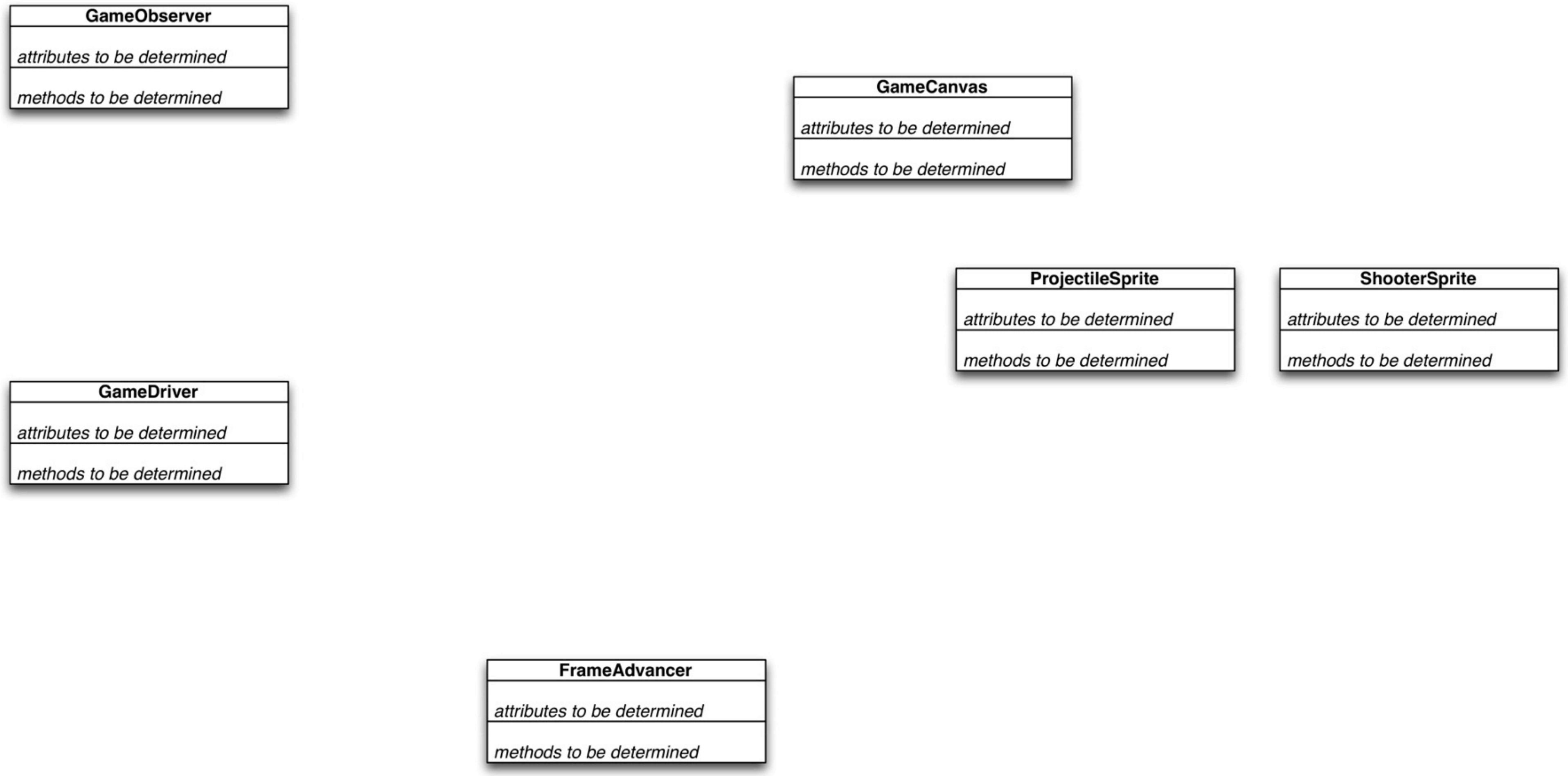
An Example of a multi-class app



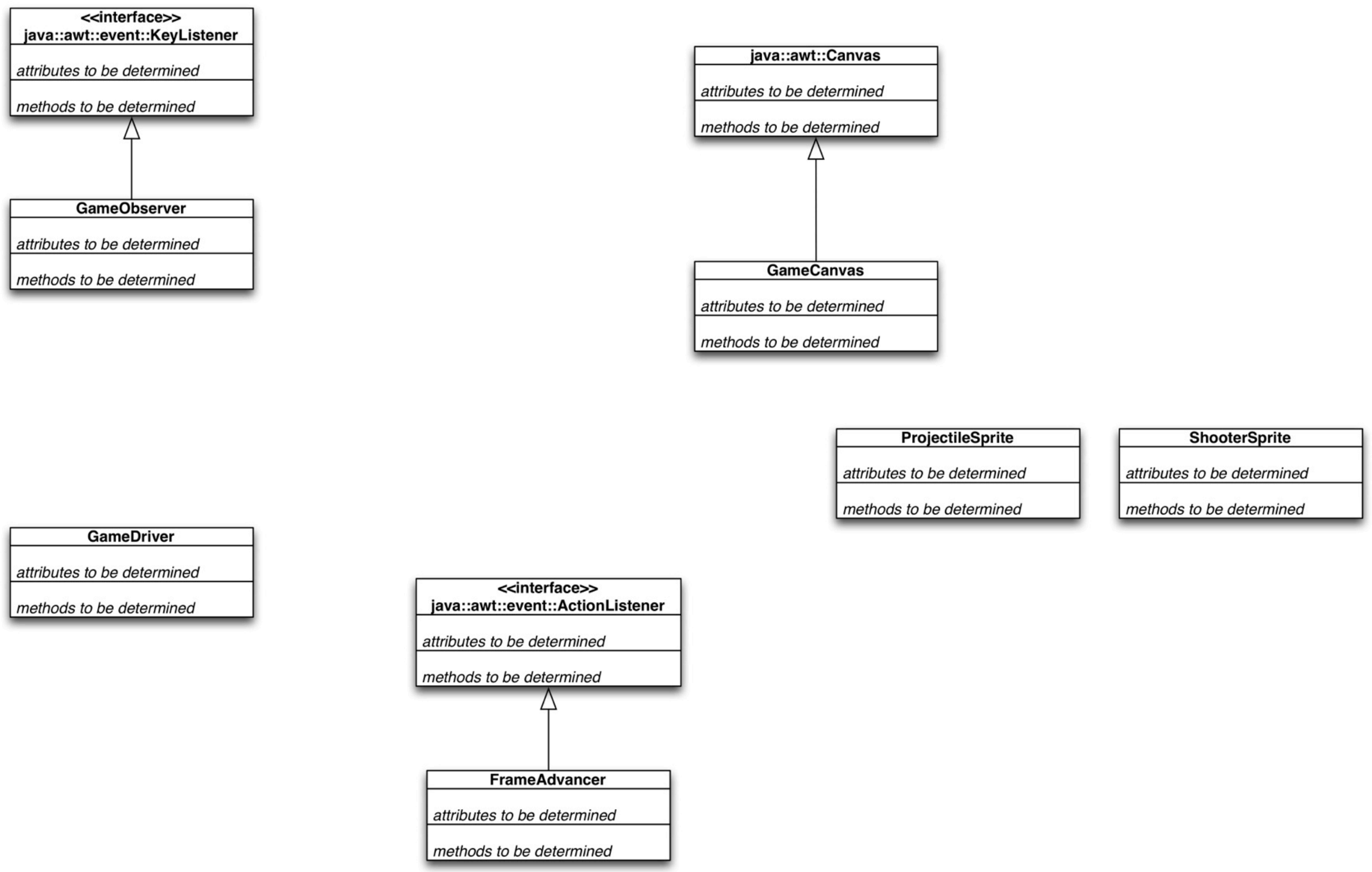
Now Let’s Turn to the Codebase

- What follows is an example of a workflow
- but…you can use whichever approach works for you

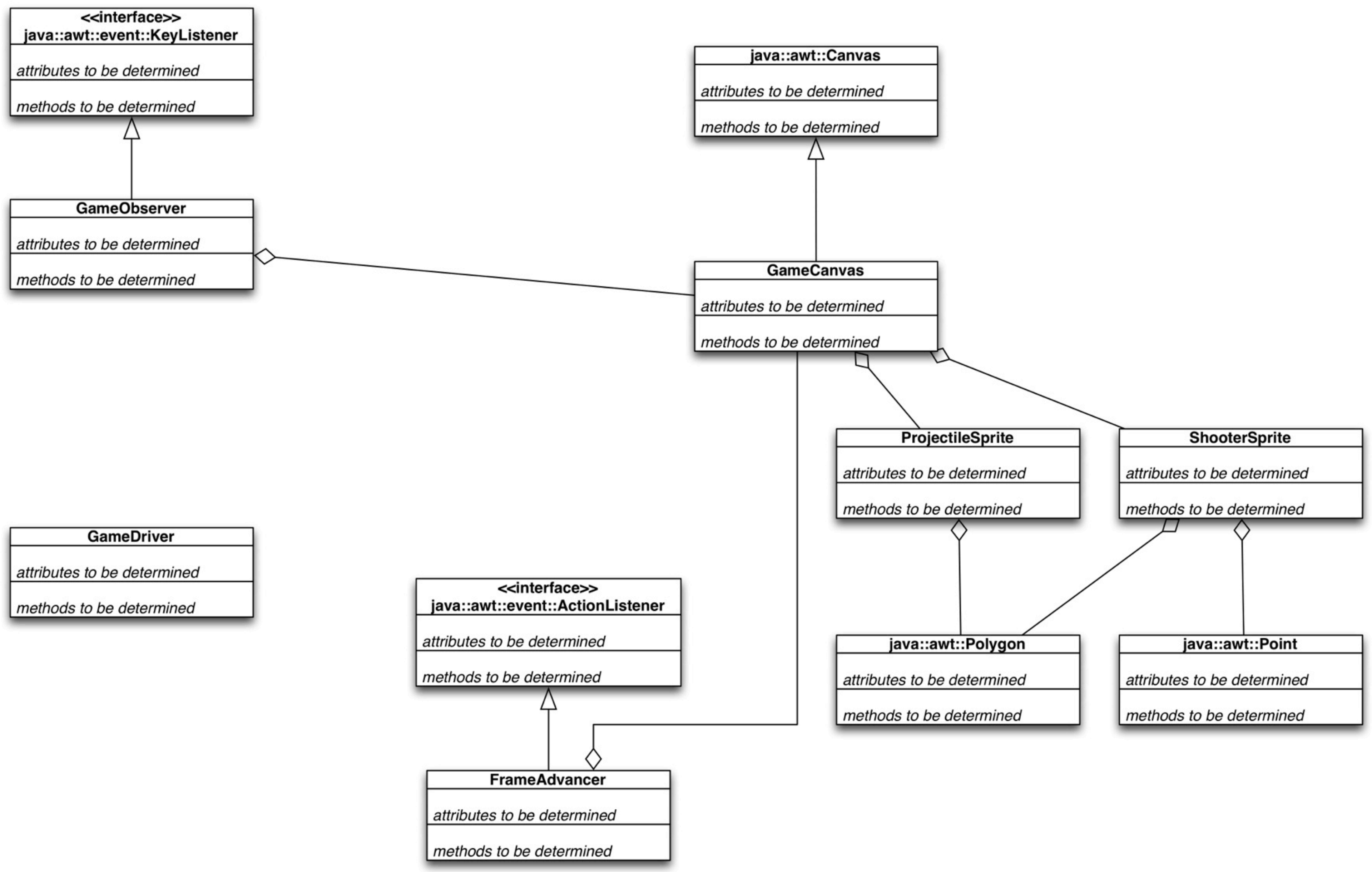
Step 1 – Diagram the classes



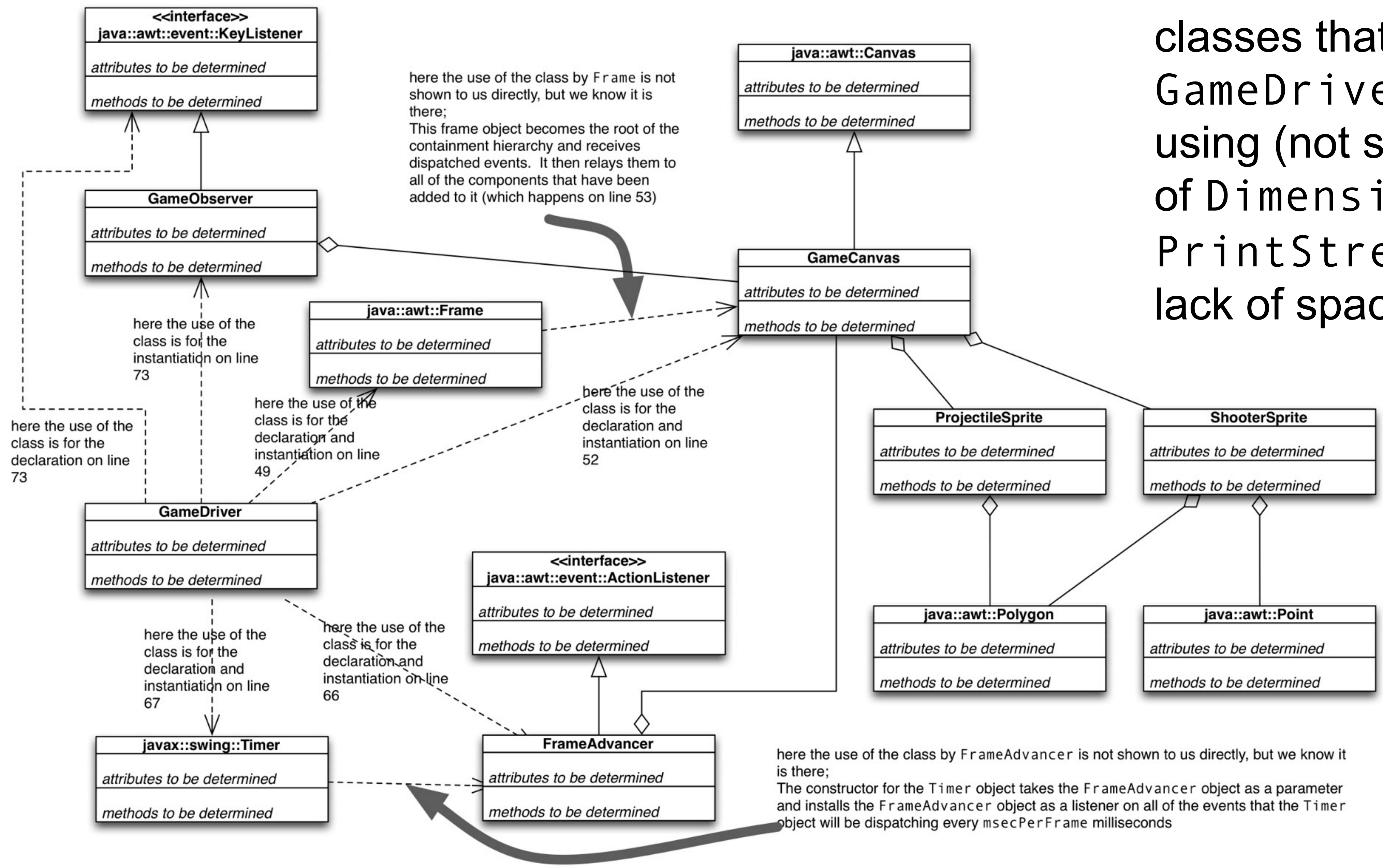
Step 2 – Add the IS-A relationships



Step 3 – Add the HAS-A relationships



Step 4a – Add the USES relationships

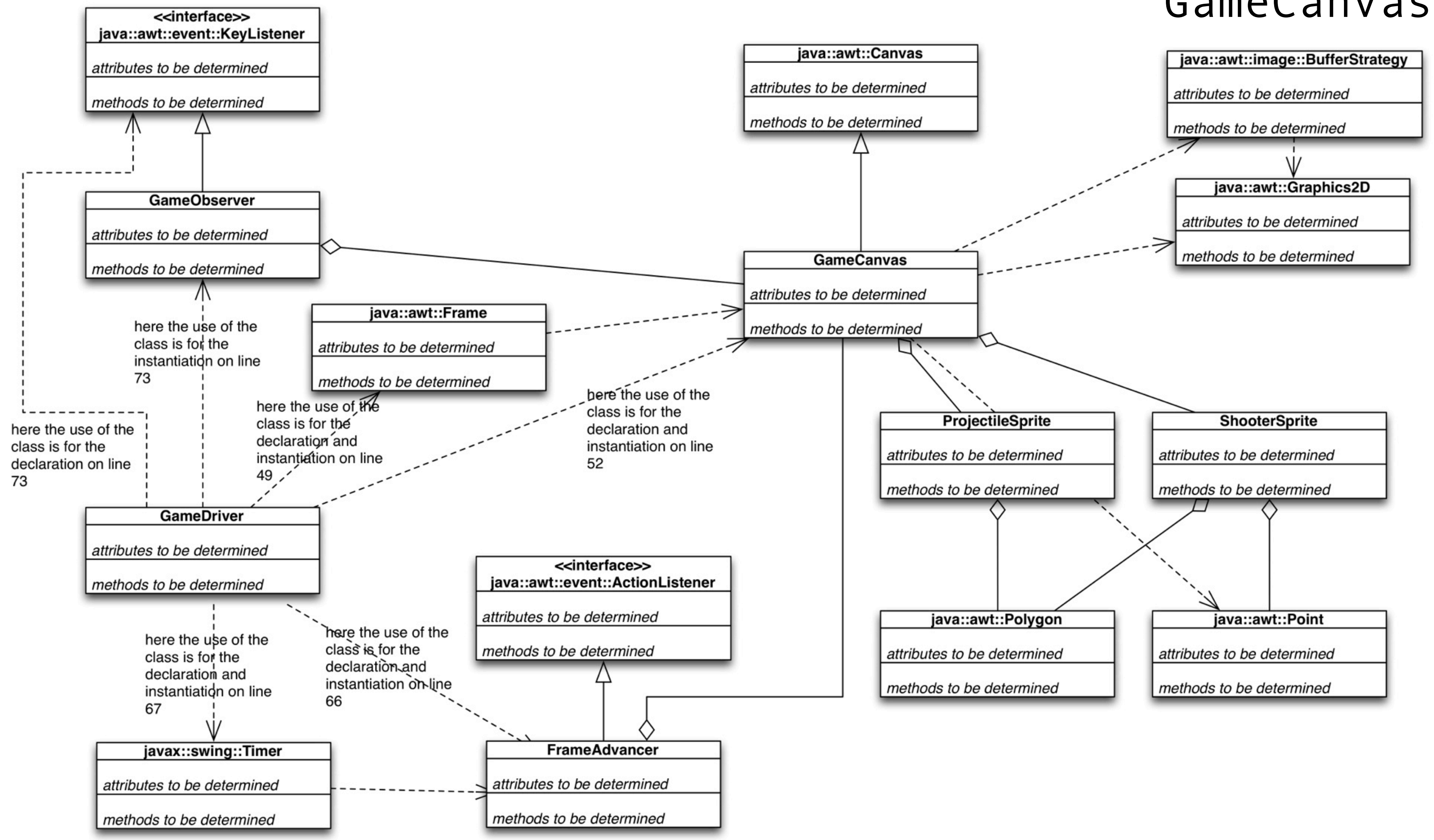


<<interface>> java::awt::event::KeyListener
attributes to be determined
methods to be determined
GameObserver
java::awt::Canvas
GameCanvas
java::awt::Frame
here the use of the class by Frame is not shown to us directly, but we know it is there;
This frame object becomes the root of the containment hierarchy and receives dispatched events. It then relays them to all of the components that have been added to it (which happens on line 53)
here the use of the class is for the instantiation on line 73
here the use of the class is for the declaration on line 73
here the use of the class is for the declaration and instantiation on line 49
here the use of the class is for the declaration and instantiation on line 52
ProjectileSprite
ShooterSprite
GameDriver
<<interface>> java::awt::event::ActionListener
java::awt::Polygon
java::awt::Point
here the use of the class is for the declaration and instantiation on line 67
here the use of the class is for the declaration and instantiation on line 66
javax::swing::Timer
FrameAdvancer
here the use of the class by FrameAdvancer is not shown to us directly, but we know it is there;
The constructor for the Timer object takes the FrameAdvancer object as a parameter and installs the FrameAdvancer object as a listener on all of the events that the Timer object will be dispatching every msecPerFrame milliseconds

first let's start with the classes that the `GameDriver` class is using (not showing use of `Dimension` and `PrintStream` due to lack of space)

Step 4b – Add the USES relationships

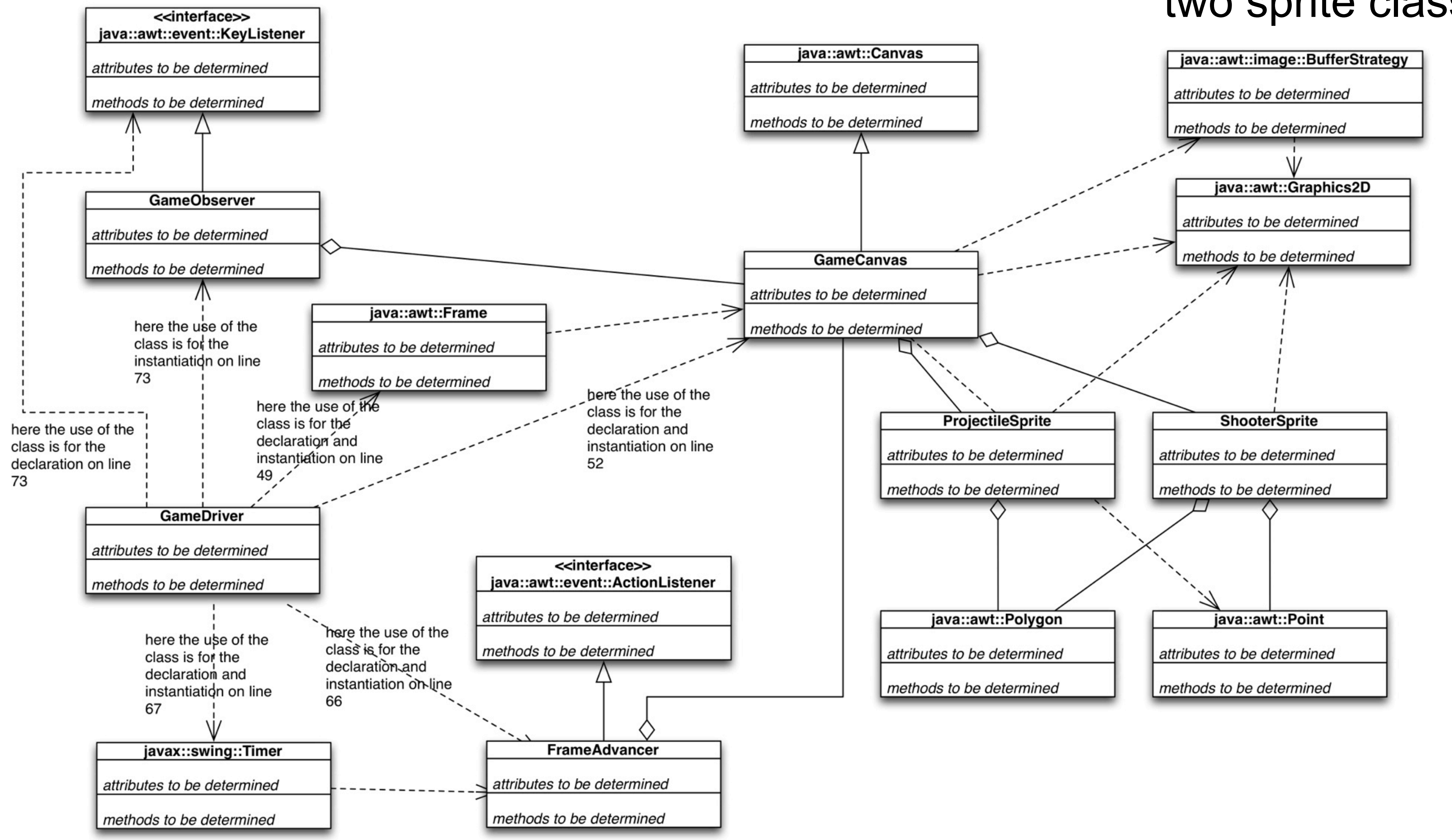
now we repeat for the `GameCanvas` class



<<interface>>
java::awt::event::KeyListener
attributes to be determined
methods to be determined
java::awt::Canvas
attributes to be determined
methods to be determined
java::awt::image::BufferStrategy
attributes to be determined
methods to be determined
java::awt::Graphics2D
attributes to be determined
methods to be determined
GameObserver
attributes to be determined
methods to be determined
GameCanvas
attributes to be determined
methods to be determined
java::awt::Frame
attributes to be determined
methods to be determined
here the use of the class is for the instantiation on line 73
here the use of the class is for the declaration on line 73
here the use of the class is for the declaration and instantiation on line 49
here the use of the class is for the declaration and instantiation on line 52
ProjectileSprite
attributes to be determined
methods to be determined
ShooterSprite
attributes to be determined
methods to be determined
GameDriver
attributes to be determined
methods to be determined
<<interface>>
java::awt::event::ActionListener
attributes to be determined
methods to be determined
java::awt::Polygon
attributes to be determined
methods to be determined
java::awt::Point
attributes to be determined
methods to be determined
here the use of the class is for the declaration and instantiation on line 67
here the use of the class is for the declaration and instantiation on line 66
javax::swing::Timer
attributes to be determined
methods to be determined
FrameAdvancer
attributes to be determined
methods to be determined

Step 4c – Add the USES relationships

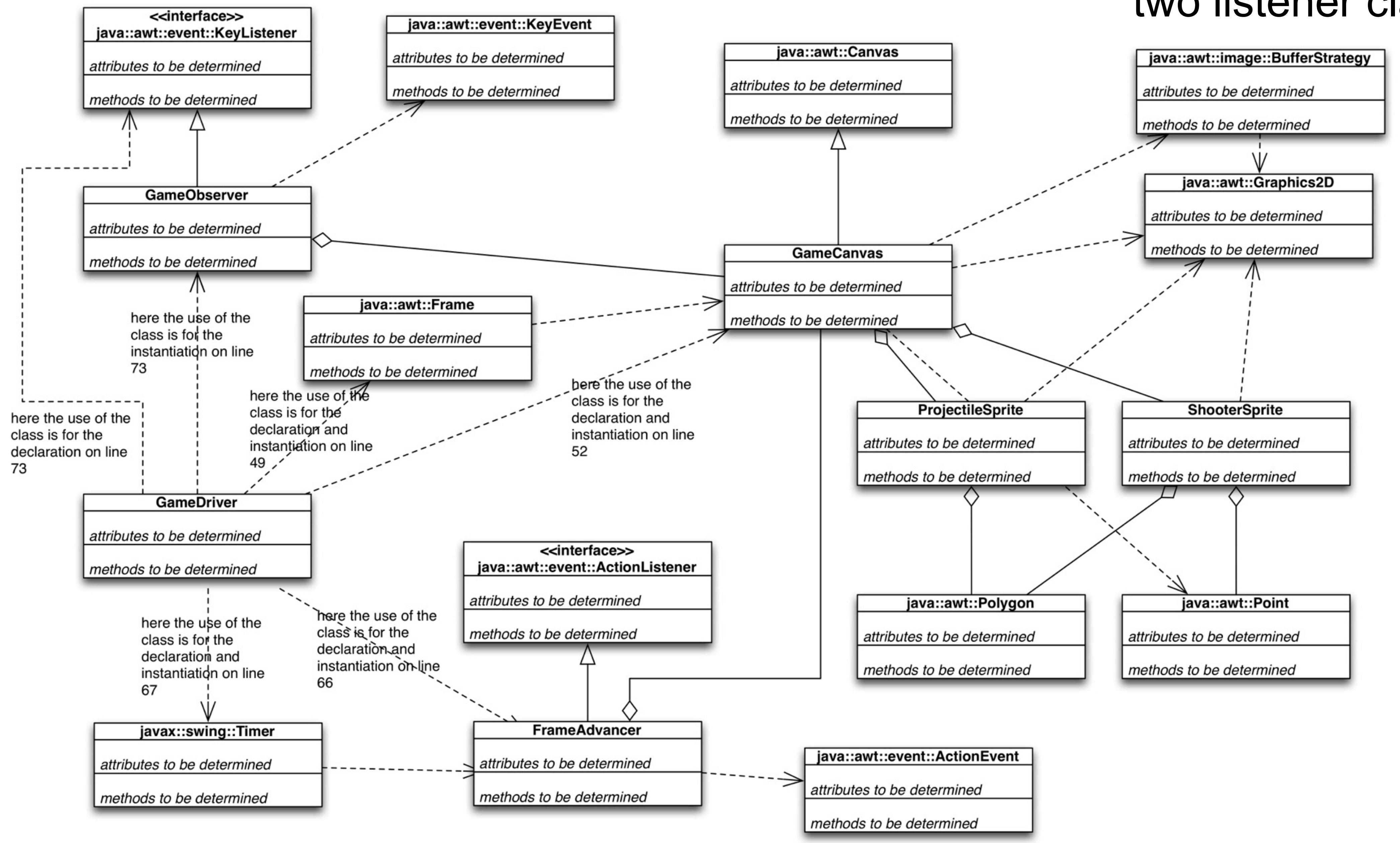
now we repeat for the two sprite classes



<<interface>>
java::awt::event::KeyListener
attributes to be determined
methods to be determined
GameObserver
attributes to be determined
methods to be determined
java::awt::Canvas
attributes to be determined
methods to be determined
java::awt::image::BufferStrategy
attributes to be determined
methods to be determined
java::awt::Graphics2D
attributes to be determined
methods to be determined
GameCanvas
attributes to be determined
methods to be determined
java::awt::Frame
attributes to be determined
methods to be determined
here the use of the class is for the instantiation on line 73
here the use of the class is for the declaration on line 73
here the use of the class is for the declaration and instantiation on line 49
here the use of the class is for the declaration and instantiation on line 52
ProjectileSprite
attributes to be determined
methods to be determined
ShooterSprite
attributes to be determined
methods to be determined
GameDriver
attributes to be determined
methods to be determined
<<interface>>
java::awt::event::ActionListener
attributes to be determined
methods to be determined
java::awt::Polygon
attributes to be determined
methods to be determined
java::awt::Point
attributes to be determined
methods to be determined
here the use of the class is for the declaration and instantiation on line 67
here the use of the class is for the declaration and instantiation on line 66
javax::swing::Timer
attributes to be determined
methods to be determined
FrameAdvancer
attributes to be determined
methods to be determined

Step 4d – Add the USES relationships

now we repeat for the two listener classes



<<interface>> java::awt::event::KeyListener
java::awt::event::KeyEvent
java::awt::Canvas
java::awt::image::BufferStrategy
java::awt::Graphics2D
GameObserver
GameCanvas
java::awt::Frame
ProjectileSprite
ShooterSprite
GameDriver
<<interface>> java::awt::event::ActionListener
java::awt::Polygon
java::awt::Point
javax::swing::Timer
FrameAdvancer
java::awt::event::ActionEvent
attributes to be determined
methods to be determined
here the use of the class is for the instantiation on line 73
here the use of the class is for the declaration on line 73
here the use of the class is for the declaration and instantiation on line 49
here the use of the class is for the declaration and instantiation on line 52
here the use of the class is for the declaration and instantiation on line 67
here the use of the class is for the declaration and instantiation on line 66