Interfaces

CSCI 2300

Review

- Last time we discussed
  - Polymorphism
  - Abstract classes
- Can anyone summarize key points about abstract classes?

Suppose you are…

- Designing a game of PacMan
  - PacMan roams through a Maze
  - Goal: eat the dots and avoid ghosts
  - Advance to next level when all dots are eaten

Let's talk design

- What classes do we need?
  - Maze
  - PacMan
  - Ghost
  - Dot
- What problem do we see in the Maze class?
- Can we resolve this problem with abstract classes?

```
class Maze
{
    public void place(PacMan pacMan)
    {
        //select location on the Maze
        packMan.draw(location);
    }
    public void place(Dot dot)
    {
        //select location on the Maze
        dot.draw(location);
    }
    public void place(Ghost ghost);
    public void place(Wall wall);
}
```

We can do better, with using Interfaces

- Interface – a "contract" specifying methods that a class will implement
- A class can "implement" an interface

```
public interface IPlaceable
{
    public void draw(Location);
}
```

```
public class PacMan implements IPlaceable
{
    public void draw(Location location)
    {
        //draws itself at the given location
    }
}
```

```
class Maze
{
    public void place(IPlaceable p)
    {
        //select location on the Maze
        p.draw(location);
    }
}
```

PackMan is substitutable for IPlaceable because PacMan implements Placeable

```
Maze maze = new Maze();
PackMan packMan = new PackMan();
Maze.place(pacMan);
```

Interface versus Inheritance

Inheritance

- A class can have only one parent class

- A class can be instantiated (if it is not abstract)

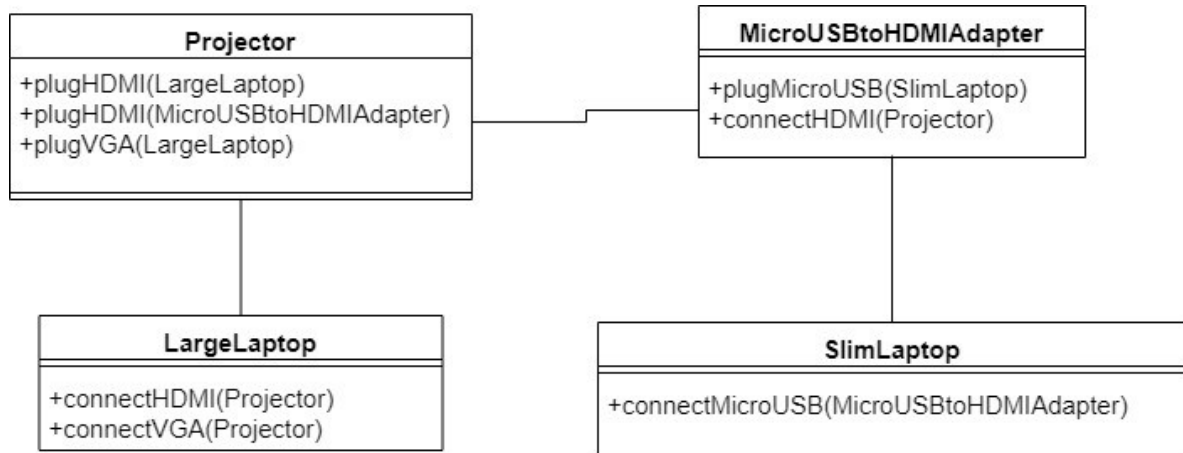- A class inherits parent class's methods and does not have to override them

Interface

- A class can implement multiple interfaces

- An interface cannot be instantiated

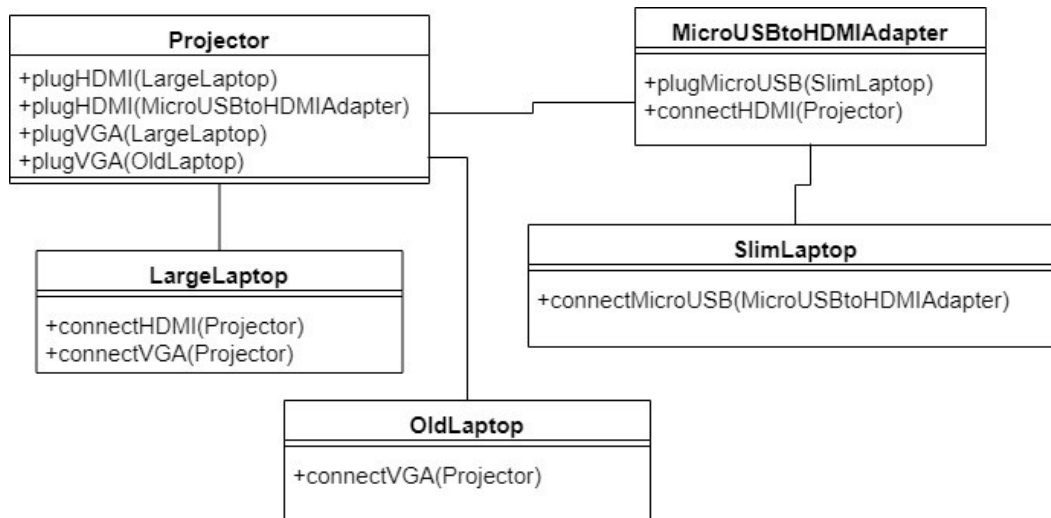- A class implementing an interface must implement all interface methods

Projector Example

- We are designing software that controls our projector
- We want to be able to plug in:
  - Small laptop (via MicroUSB to HDMI adapter)
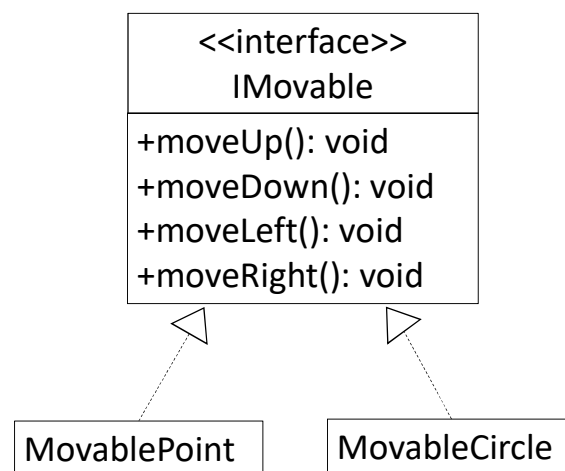 - LargeLaptop (via HDMI)
  - Desktop (via VGA)

2/7/2019

Initial Design

Projector

+plugHDMI(LargeLaptop)
+plugHDMI(MicroUSBtoHDMIAdapter)
+plugVGA(LargeLaptop)

MicroUSBtoHDMIAdapter

+plugMicroUSB(SlimLaptop)
+connectHDMI(Projector)

LargeLaptop

+connectHDMI(Projector)
+connectVGA(Projector)

SlimLaptop

+connectMicroUSB(MicroUSBtoHDMIAdapter)

Now we also need to plug in an "old laptop" with no HDMI port

Projector

+plugHDMI(LargeLaptop)
+plugHDMI(MicroUSBtoHDMIAdapter)
+plugVGA(LargeLaptop)
+plugVGA(OldLaptop)

MicroUSBtoHDMIAdapter

+plugMicroUSB(SlimLaptop)
+connectHDMI(Projector)

LargeLaptop

+connectHDMI(Projector)
+connectVGA(Projector)

SlimLaptop

+connectMicroUSB(MicroUSBtoHDMIAdapter)

OldLaptop

+connectVGA(Projector)

Projector example redesigned with interfaces
(handout)

Lab 7

<<interface>> IMovable
+moveUp(): void +moveDown(): void +moveLeft(): void +moveRight(): void

MovablePoint		MovableCircle

- Pull your git repos, you will find `movable` directory
- Complete code noted in TODO comments. Add a driver to test your code.