Peer Assisted Study Session

FIT2099 - Week 4

Monash University

Objectives

- Be able to draw a model from a given set of requirements
- Explain abstraction and its benefits
- Identify code smells in a given code

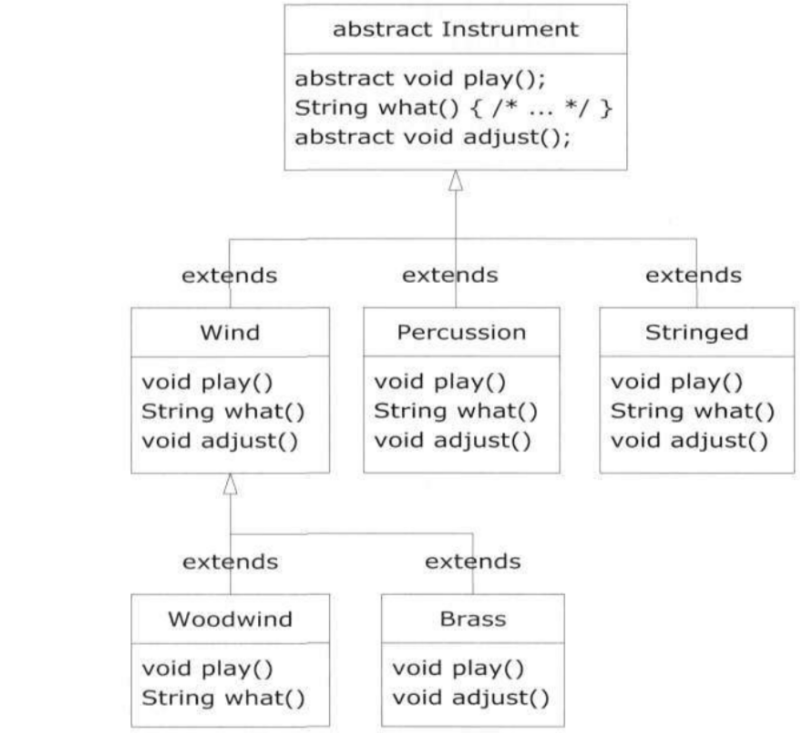
Estimated Time

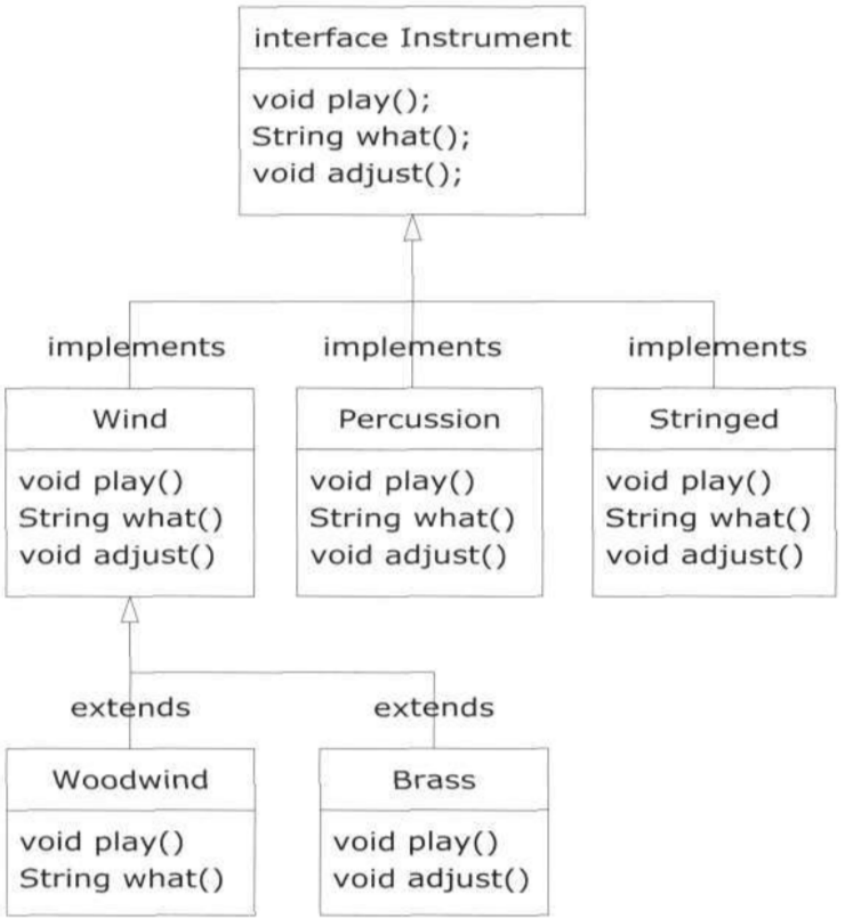
Question 1 - Question 3 (20 Minutes)
Question 4 (5 Minutes)
Question 5 (5 Minutes)
Question 6 (10 Minutes)

Questions

[Adapted from Bruce Eckel's book "Thinking in Java"]
Consider a part of a simulation where you are tasked to model different musical Instruments. The three types we're interested in modeling are Wind instruments, Percussion and Stringed instruments. Wind instruments can be subdivided into variants we need to model: Woodwind and Brass instruments. In the simulation, all instruments can be played, they can be adjusted, and they can be queried to obtain a string representing what instrument they are. You can assume play and adjust are void methods.

1. Given this information, your job is to draw a class diagram to model the types and use either interfaces or abstract classes in your model.


abstract Instrument
abstract void play();
String what() { /* ... */ }
abstract void adjust();
extends
extends
extends
Wind
void play()
String what()
void adjust()
Percussion
void play()
String what()
void adjust()
Stringed
void play()
String what()
void adjust()
extends
extends
Woodwind
void play()
String what()
Brass
void play()
void adjust()


interface Instrument
void play();
String what();
void adjust();
implements
implements
implements
Wind
void play()
String what()
void adjust()
Percussion
void play()
String what()
void adjust()
Stringed
void play()
String what()
void adjust()
extends
extends
Woodwind
void play()
String what()
Brass
void play()
void adjust()

2. Elaborate why you chose to model them the way you did?

3. What are the disadvantages of your proposed model?

4. What are packages? State three benefits of using packages?

A package is a group of related classes.
Its benefits are as follows:

It makes it easier to find related classes
It can prevent name clashes
Eliminates dependencies

5. Explain the benefits of using **abstraction** in Object Oriented Design?

Accrue less technical debt

Make iterative development easier

Respond more readily to changes in requirements or environment

Reduce cognitive load on a developer

6. Read the following pseudo-java and find the code smell(s) it is exhibiting and identify the locations of the code smell(s):

```
public class SmellyProgram {
 private int x;
 File f;

 public SmellyProgram(File f) {
   x = 0;
   this.f = f;
 }
```

```
 public void run() {
   for (Line line in f) {
     if (line.isEmpty())
       x++;
   }
 }
}
```

Two possible code smells:
(i) The field x would be better named “emptyLinesCount”

(ii) Code smell (i) also applies to field f but more importantly it should be made private unless there is a need for it to be package-private.
This reduces coupling between this class and potential clients which may rely on accessing this field directly.