

Midterm Review

October 17, 2022

Java Basics

Shadowing

Primitive types

Objects vs instances

Public and private information hiding

Physical vs structural equality

Interfaces and information hiding

Packages, the Java Standard Library

Inheritance

Delegation vs inheritance

Delegation is when we put an instance within the class we are trying to define and access that other class's features through that instances. With inheritance everything in the superclass is accessed directly through the subclass.

There is a performance overhead to delegation but it is often much more user-friendly than inheritance.

Exceptions (checked vs unchecked)

A checked exception *must* be caught using `catch`. If it is not caught, this must be declared in the method's signature.

For example,

```
public int read() throws IOException
```

Unchecked exceptions can be caught, but are not required to be caught. The program terminates if an unchecked exception is not caught in `main()`.

Overriding

Iterators

External: A class implements `.hasNext()` and `.next()`. The client decides when to take an iteration step.

Internal: A class has some `.iterate(fun)` method. The module carries out the iteration and calls back to some client code for each item.

UML diagrams

Coupling vs cohesion

Creational patterns: singleton, enumeration, factory

Structural patterns: adapter, proxy, decorator (“wrappers”)

Behavioral patterns: observer, visitor

Patterns in other languages: OO in C, match/case, monads

Reflection: classes (Class), constructors, methods, fields, hello world with reflection, reflection design patterns

Readings and projects