Java Programming Foundations 1

Week 13: Review + Polymorphism

Overview

- Abstract classes review
- Inheritance review
- Polymorphism
- Practice time

Abstract classes

Abstract classes

An abstract class is just like a regular class but cannot be instantiated.

Abstract classes

Abstract classed are meant to be inherited.

Inheritance

Inheritance

Classes are able to extend each other with inheritance.

Inheritance

A "child" class inherits from a "parent" class.

Inherited example

```
public class Parent
{
    // ...
}

public class Child extends Parent
{
    // ...
}
```

Inheritance

A "child" class inherits all behaviour from its "parent" class and can be used in place of its parent.

... can be used in place of its parent.

This is a form of polymorphism. More on polymorphism in a minute.

Abstract class inheritance example (1)

```
public abstract class Pokemon
{
    protected String name;
    protected int attack;
    protected int health;

    public String getName() { return this.name; }
    public int getAttack() { return this.attack; }
    public int getHealth() { return this.health; }
}

public class Pikachu extends Pokemon
{
    public Pikachu()
    {
        this.name = "Pikachu";
        this.attack = 6;
        this.health = 110;
    }
}
```

Abstract class inheritance example (2)

```
public class App
{
    public static void main(String[] args)
    {
        Pokemon pokemon1 = new Pikachu();

        System.out.println(pokemon1.getName() + " has " +
            pokemon1.getHealth() + " health and " +
            pokemon1.getAttack() + " attack");
    }
}
```

Polymorphism

Polymorphism

There are different forms of polymorphism, but the one we will focus on is polymorphism via inheritance.

Polymorphism

Polymorphism allows you to work with objects of different types using a single interface.

Practice time