ORACLE Academy

Understanding Abstraction

What Will I Learn?

Objectives

In this lesson you will learn to:

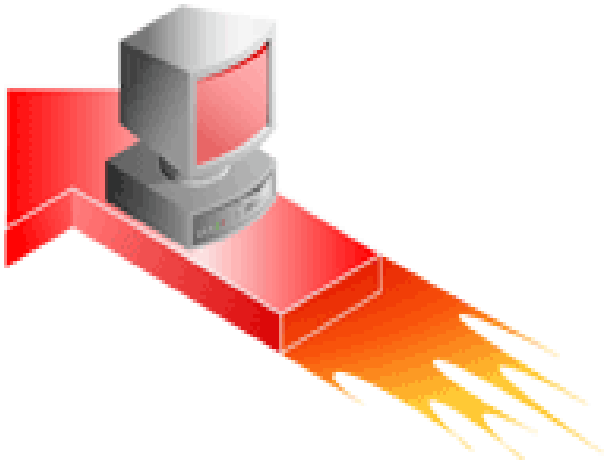
- Define abstraction and provide an example of when it is used

Why Learn It?

Purpose

Take your game to the next level by automating the creation of your world and its objects.

This will enable you to create games faster, so you can spend more time perfecting the movement of objects and events that happen in your game.



Abstraction

You may hard code a new instance to perform a single, specific task, such as play a sound when a specific keyboard key is pressed, or display a range of questions and answers.

To create programs of a larger scale, say with 10 objects that each perform different actions, you need to write code that can allow each new instance to perform a different type of task.

There is a technique you can use to command newly-created instances to perform different actions. This technique is called abstraction.

Abstraction with Variables and Parameters

Abstraction occurs many ways in programming. This lesson uses an abstraction technique with variables and parameters to pass different types of information, such as keyboard key names, words, or sounds, to each instance created.

This technique includes:

- Variables, such as:
  - Keyboard key the instance reacts to
  - Sound file to play when key is pressed
  - String with words
- Constructor with parameters that initialize the variables
  - Constructor to pass a keyboard key and sound file to an instance

Create Constructor to Store Variables

First, create a constructor to store the key and sound variables.

```
import greenfoot.*;  // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * Write a description of class Duke here.
 *
 * @author Oracle Academy
 * @version JF_S03_L05
 */
public class Duke extends Animal
{
    private GreenfootImage image1;
    private GreenfootImage image2;
    private boolean isKeyDown;
    private String key;
    private String sound;
    /**
     * Create a Duke and initialize his two images. Link Duke to a specific keyboard
     * key and sound.
     */
    public Duke(String keyName, String soundFile)
    {
        key = keyName;
        sound = soundFile;
        image1 = new GreenfootImage("Duke.PNG");
        image2 = new GreenfootImage("duke2.png");
        setImage(image1);
    }
```

Create and Place Instances

Once the sound and key variables are assigned to the class, create code to automatically add instances of the class to the world. The following programming statement, added to the Duke constructor, achieves this.

```
addObject (new Duke (“k”, “test.wav”), 150, 150);
```

This statement:

- Creates a new instance of Duke each time the world is re-initialized.
- Passes a specific key and sound file to the instance.
- Places the instance in the world at specific x and y coordinates.

Constructor Example

This constructor adds a new Duke to the world and assigns this instance a specific keyboard key and sound file.

```
import greenfoot.*;  // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * Write a description of class DukeWorld here.
 *
 * @author Oracle Academy
 * @version JF_S03_L05
 */
public class DukeWorld extends World
{
    /**
     * This constructor creates a new world and the objects that start the
     * game.
     */
    public DukeWorld()
    {
        super(600, 400, 1);
        addObject (new Duke ("k", "test.wav"), 150, 100);
    }
}
```

Terminology

Key terms used in this lesson included:

Abstraction

Summary

In this lesson, you learned how to:

- Define abstraction and provide an example of when it is used

Practice

The exercises for this lesson cover the following topics:

- Actor abstraction
- Journaling abstraction