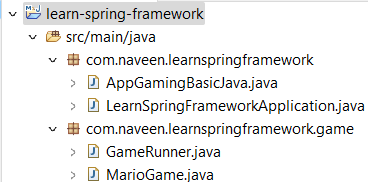
**GETTING STARTED WITH JAVA GAMING APPLICATION**

Let’s get started with:

1. **Iteration 1**: **Tightly Coupled Java Code**
   * GameRunner class
   * Game classes: Mario, SuperContra, Pacman etc…

Create java classes like the following **hierarchy**.



So, we will create a GameRunner class which runs the MarioGame.

**AppGamingBasicJava.java**

package com.naveen.learnspringframework;

import com.naveen.learnspringframework.game.GameRunner;

import com.naveen.learnspringframework.game.MarioGame;

public class AppGamingBasicJava {

    public static void main(String[] args) {

        var marioGame = new MarioGame();

        var gameRunner = new GameRunner(marioGame);

        gameRunner.run();

    }

}

Use **ctrl + 1** to show the hint provided by the Eclipse IDE. We can automatically generate code using this option.

To run the application, right click on the main class 🡪 Run as 🡪 Java Application.

**GameRunner.java**

package com.naveen.learnspringframework.game;

public class GameRunner {

    MarioGame game;

    public GameRunner(MarioGame game) {

        this.game = game;

    }

    public void run() {

        System.out.println("Running game: " + game);

        game.up();

        game.down();

        game.left();

        game.right();

    }

}

**MarioGame.java**

package com.naveen.learnspringframework.game;

public class MarioGame {

    public void up() {

        System.out.println("Jump");

    }

    public void down() {

        System.out.println("Go into a hole");

    }

    public void left() {

        System.out.println("Go back");

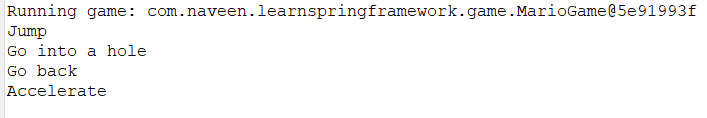
    }

    public void right() {

        System.out.println("Accelerate");

    }

}

**OUTPUT:**