



软件体系结构

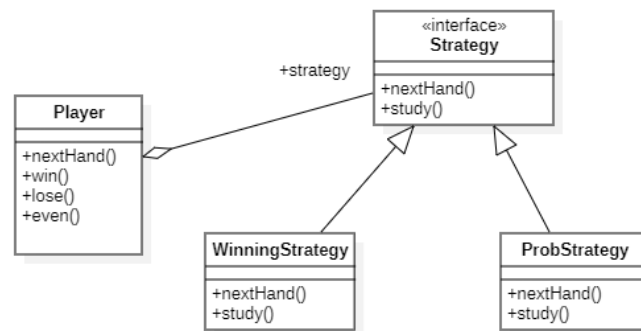
《软件体系结构作业十八》

学 号 22920212204396

姓 名 黄子安

2024 年 5 月 21 日

1、用 GUI 改写本例



实验的类图如上所示，核心思想就是把算法封装成一个个类，从而可以很方便的新增一个新的算法策略，同时对于其中一个具体算法的修改也不会封闭在自己类内部，不会扩散到其他的类中，也就是说符合开闭原则

示例代码中的两个策略：

- **WinningStrategy**：如果赢了就继续出一样，否则随机出
- **ProStrategy**：根据历史记录和概率进行计算

增加 GUI 部分的代码：

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<VBox fx:id="rootVBox" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="1000.0" prefWidth="600.0" alignment="CENTER" spacing="20"
xmlns="http://javafx.com/javafx/17.0.2-ea" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="cn.xmu.edu.strategy.StrategyController">
    <padding>
        <Insets top="20" right="20" bottom="20" left="20"/>
    </padding>
    <children>
        <HBox alignment="CENTER" spacing="20">
            <children>
                <VBox alignment="CENTER" spacing="10">
                    <children>
                        <Label text="Player 1" style="-fx-font-size: 20px; -fx-font-weight: bold;" />
                        <TextField fx:id="player1TextField" prefHeight="25.0" prefWidth="150.0" />
                        <ComboBox fx:id="player1ComboBox" prefHeight="25.0" prefWidth="150.0" />
                    </children>
                </VBox>
            </children>
        </HBox>
    </children>
</VBox>
  
```

```
<VBox alignment="CENTER" spacing="10">
  <children>
    <Label text="Player 2" style="-fx-font-size: 20px; -fx-font-weight: bold;" />
    <TextField fx:id="player2TextField" prefHeight="25.0" prefWidth="150.0" />
    <ComboBox fx:id="player2ComboBox" prefHeight="25.0" prefWidth="150.0" />
  </children>
</VBox>
</children>
</HBox>
<VBox alignment="CENTER" spacing="10">
  <children>
    <Label text="Process" style="-fx-font-size: 20px; -fx-font-weight: bold; -fx-font-family: Consolas " />
    <TextArea fx:id="textArea" prefHeight="400.0" prefWidth="100.0"
      style="-fx-border-color: gray;
        -fx-border-width: 1px;
        -fx-padding: 10px;
        -fx-font-size: 14px;
        -fx-font-family: Consolas" />
  </children>
</VBox>
<VBox alignment="CENTER" spacing="10">
  <children>
    <Label text="Result"
      style="-fx-font-size: 20px;
        -fx-font-weight: bold;
        -fx-font-family: Consolas " />
    <TextArea fx:id="resultArea" prefHeight="250.0" prefWidth="100.0"
      style="-fx-border-color: gray;
        -fx-border-width: 1px;
        -fx-padding: 10px;
        -fx-font-size: 14px;
        -fx-font-family: Consolas" />
  </children>
</VBox>
<HBox alignment="CENTER" spacing="20">
  <children>
    <Button fx:id="confirmButton" onAction="#confirm" mnemonicParsing="false"
text="Confirm"
      style="-fx-font-size: 14px;
        -fx-font-family: Consolas" />
    <Button fx:id="resetButton" onAction="#reset" mnemonicParsing="false" text="Reset"
      style="-fx-font-size: 14px;
        -fx-font-family: Consolas" />
  </children>
</HBox>
</children>
</VBox>
```

```
package cn.xmu.edu.strategy;

import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.scene.image.Image;
import javafx.scene.layout.*;

import java.lang.reflect.InvocationTargetException;

public class StrategyController {
    public TextArea textArea;
    public TextArea resultArea;
    public Button confirmButton;
    public Button resetButton;
    public TextField player1TextField;
    public TextField player2TextField;
    public ComboBox player1ComboBox;
    public ComboBox player2ComboBox;
    @FXML
    private VBox rootVBox;

    @FXML
    public void initialize() {
        Image backgroundImage = new Image(getClass().getResource("/img.png")
            .toExternalForm());
        BackgroundImage background = new BackgroundImage(backgroundImage,
            BackgroundRepeat.NO_REPEAT,
            BackgroundRepeat.NO_REPEAT,
            BackgroundPosition.CENTER,
            BackgroundSize.DEFAULT);
        rootVBox.setBackground(new Background(background));
        player1ComboBox.getItems().addAll("ProbStrategy", "WinningStrategy");
        player2ComboBox.getItems().addAll("ProbStrategy", "WinningStrategy");
    }

    @FXML
    protected void confirm() throws ClassNotFoundException, NoSuchMethodException,
        InvocationTargetException, InstantiationException, IllegalAccessException {
        if(player1TextField.getText().isEmpty() ||
            player2TextField.getText().isEmpty() ||
            player1ComboBox.getSelectionModel().isEmpty() ||
            player2ComboBox.getSelectionModel().isEmpty()) {
            Alert alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error");
            alert.setHeaderText(null);
            alert.setContentText("Please input username and strategy");
            alert.showAndWait();
        }
        else {
            long seed1 = System.currentTimeMillis();
            long seed2 = System.currentTimeMillis() + 271828;
            Strategy strategy1 = (Strategy) Class
                ..forName("cn.xmu.edu.strategy." + player1ComboBox.getValue())
                .getConstructor(Long.class)
                .newInstance(seed1);

            Strategy strategy2 = (Strategy) Class
                ..forName("cn.xmu.edu.strategy." + player2ComboBox.getValue())
                .getConstructor(Long.class)
                .newInstance(seed2);
        }
    }
}
```

```

        Player player1 = new Player(player1TextField.getText(), strategy1);
        Player player2 = new Player(player2TextField.getText(), strategy2);

        textArea.clear();
        resultArea.clear();
        for(int i = 0; i < 100; ++i) {
            Hand nextHand1 = player1.nextHand();
            Hand nextHand2 = player2.nextHand();
            if (nextHand1.isStrongerThan(nextHand2)) {
                textArea.appendText("Winner: " + player1 + "\n");
                player1.win();
                player2.lose();
            } else if (nextHand2.isStrongerThan(nextHand1)) {
                textArea.appendText("Winner: " + player2 + "\n");
                player1.lose();
                player2.win();
            } else {
                textArea.appendText("Even ... " + "\n");
                player1.even();
                player2.even();
            }
        }

        resultArea.appendText("Total result:\n");
        resultArea.appendText(player1 + "\n");
        resultArea.appendText(player2 + "\n");
    }
}

@FXML
protected void reset() {
    player1TextField.clear();
    player2TextField.clear();
    textArea.clear();
    resultArea.clear();
    player1ComboBox.getSelectionModel().clearSelection();
    player2ComboBox.getSelectionModel().clearSelection();
}
}

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.io.IOException;

public class StrategyApplication extends Application {
    @Override
    public void start(Stage stage) throws IOException {
        FXMLLoader fxmlLoader = new
            FXMLLoader(StrategyApplication.class.getResource("strategy-view.fxml"));
        Scene scene = new Scene(fxmlLoader.load(), 600, 400);
        stage.setTitle("Strategy Game!");
        stage.setScene(scene);
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}

```

最后运行结果如下所示：提供复选框选择策略算法，之后输出游戏过程和结果（等宽字体 yyds）

