山东大学 软件 学院

面向对象 课程实验报告

|  |  |
| --- | --- |
| 实验题目：猜数字游戏 | |
| 实验学时：16 | 实验日期： 2024.5.18 |
| 实验目的：  《面向对象技术》是计算机科学与技术、软件工程专业学生的一门必修课程。面向对象技术是当前最重要的程序设计思想与方法，利用抽象性、封装性、继承性、多态性等特性帮助人们开发出模块化、复用性强、易修改、可扩展的软件系统和程序。本实验通过若干个备选的相对复杂、独立的题目，使学生在实验过程中全面、系统的熟悉面向对象技术的基本概念和设计技术，分析、设计一个相对完整的应用，并使用面向对象编程语言进行实现，通过实验掌握使用面向对象技术进行程序设计、开发的思想和技能 | |
| 硬件环境：  CPU 2GHz、内存4GB、存储空间5GB。 | |
| 软件环境：  IntelliJ IDEA、OpenJDK-17 | |
| 实验步骤与内容：  1.实验要求：  完成一个猜数字的GUI游戏。  开始游戏后，产生一个没有重复数字的4位随机数，用户每猜一个数字，显示出“完全猜中的数字个数”和“猜中数字但位置错误的数字个数”，比如nAmB，数字n表示猜中的位置正确的数字个数，数字m表示数字正确而位置不对的数字个数。例如，正确答案为5234，如果用户猜5346，则显示：1A2B，数字1表示数字5及其位置猜对了，数字3和4这两个数字猜对了，但是位置没对，记为2B。然后，用户根据游戏提示的信息继续猜，直到猜中为止。同时设计规则，根据猜中的次数计算积分，并可以显示不同用户的排行榜。  **2.** **功能与目标分析**  本游戏的核心功能是提供一个图形用户界面（GUI）来让玩家猜测一个四位的随机数。游戏需要实现以下几个主要目标：  随机数生成：游戏开始时，生成一个不重复的四位随机数作为答案。  用户输入：玩家通过GUI输入一个四位的猜测数字。  比较逻辑：将玩家的猜测与正确答案进行比较，并计算出“完全猜中的数字个数”（A）和“猜中数字但位置错误的数字个数”（B）。  反馈显示：将比较结果以nAmB的格式显示在GUI上，同时提供积分和排行榜信息。  积分与排行榜：根据玩家猜中的次数计算积分，并提供排行榜来显示不同用户的积分情况。  **3.** **设计思路**  使用面向对象的思想和方法，我们可以将猜数字游戏拆分为以下几个主要的类和对象：  User（用户类）：  属性：用户名（username）、当前积分（score）等。  方法：获取和设置用户名、获取和设置当前积分等。  Scoreboard（积分榜类）：  属性：用户列表（userList，包含User对象），可能还有如总用户数（totalUsers）、最高分（highScore）等统计信息。  方法：添加用户到排行榜、更新用户积分、获取排行榜列表、获取最高分等。  注意：Scoreboard类可以继承一个更通用的“积分管理类”或“排行榜管理类”，但为了简化设计，这里直接定义Scoreboard类。  GuessingGame（猜数字游戏类）：  属性：正确答案（answer）、当前猜测（currentGuess）、当前玩家（currentPlayer）等。  方法：生成随机数作为正确答案、接收用户的猜测、比较猜测与答案并返回nAmB结果、更新玩家积分等。  注 意：虽然名为Game，但此类的职责主要是处理游戏的逻辑，而非控制整个游戏流程。  GameController（游戏控制器类）：  属性：Game实例（game）、GUI实例（gui）等。  方 法：初始化游戏（包括生成随机数和初始化GUI）、控制游戏流程（如开始、结束游戏，处理用户输入等）。  注意：此类作为游戏的“大脑”，负责协调Game和GUI之间的交互。  GUI（图形用户界面类）：  属性：游戏控制器实例（gameController）等。  方法：显示游戏界面、接收用户输入（如用户名、猜测数字等）、调用游戏控制器的相应  方法来处理用户输入、显示游戏结果和排行榜等。  **UML图**    游戏流程：  初始化：GameController类初始化时创建Game实例和GUI实例，Game实例生成随机数作为正确答案，GUI实例初始化游戏界面。  用户输入：GUI类接收用户输入的用户名、猜测数字等，并传递给GameController类。  比较与反馈：GameController类调用Game实例的方法将用户的猜测与正确答案进行比较，计算出nAmB结果，并通过Game实例的方法更新玩家的积分。  显示结果：GUI类根据GameController类返回的结果显示nAmB结果和积分在界面上。  排行榜：GameController类调用Scoreboard实例的方法来管理排行榜，GUI类负责显示排行榜。排行榜可以在用户每次猜测后更新，也可以在游戏结束时统一显示。  **4.实验步骤：**  （1）写出核心猜数字算法。  （2）建立对象类如用户类等，添加属性方法等。  （3）将控制台输出文字逐渐修改为由图形界面展示。  （4）搭建GUI界面，创建各种面板及按钮等，包括开始面板、游戏界面、排行界面和选项列表。  （5）规定各个按钮的响应事件。  （6）进行测试，调整代码顺序以使游戏正常进行。  （7）生成可执行文件。  （8）完成实验报告及设计文档。   1. **实现界面：**   如下图所示：  （1）    （2）    （3）    （4） | |
| 结论分析与体会：  这次面向对象实验，我选择的是猜数字题目，题目比较简单，猜数字的核心思想也比较简单。我主要是在GUI界面上花的功夫比较多，从这次实验中，我了解到了很多java里面的控件，并且初步地去尝试了怎么使用他们，对于界面布局、控件使用、监听器、容器等等也有了更深入的理解。并且通过这次实验，加深了在课堂上学习的继承、实现、关联、依赖等面向对象知识，对于大一所学的java又有了更加深入的了解。在这次实验中，我也初步尝试画UML图，可能画得还不够完善，但我也从中切切实实地感受到了用UML图，能够更加简明地表达自己的代码思想和算法。总而言之，在这次实验中，我学到了很多，也希望自己接下去的学习也能够收获颇丰！ | |

**实验代码：**

package com.example.guessnum;  
  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
public class HelloApplication extends Application {  
 @Override  
 public void start(Stage stage) throws IOException {  
 FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));  
 Scene scene = new Scene(fxmlLoader.load(), 500, 400);  
 stage.setTitle("猜数字!");  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 public static void main(String[] args) {  
 *launch*();  
 }  
}

package com.example.guessnum;  
  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Scene;  
import javafx.scene.control.Alert;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
import javafx.scene.control.TextInputDialog;  
import javafx.scene.layout.Pane;  
import javafx.scene.text.Font;  
import javafx.stage.Modality;  
import javafx.stage.Stage;  
  
import java.io.File;  
import java.io.IOException;  
import java.util.\*;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class HelloController {  
 @FXML  
 protected TextField shurukuang;  
 private int aright;  
 private int sright;  
 private String name="Player1";  
  
 private int resNum;  
 private int res[]=new int[4];  
 @FXML  
 private Label resLabel;  
 @FXML  
 private Label resLabel0;  
 public Scoreboard scoreboard=new Scoreboard();  
 @FXML  
 private void initialize() {  
 // 设置字体大小，例如16px  
 resLabel.setFont(new Font(24));  
 resNum=*generateUniqueRandomFourDigitNumber*();  
 int number=resNum;  
 for (int i = 3; i >= 0; i--) {  
 res[i] = number % 10; // 取最后一位数字  
 number /= 10; // 去掉最后一位数字  
 }  
 }  
  
  
 @FXML  
 protected void onuserClick()  
 {  
 TextInputDialog dialog = new TextInputDialog(name);  
 dialog.setTitle("猜数字游戏");  
 dialog.setHeaderText("请输入用户名");  
 dialog.setContentText("在这里输入内容:");  
 // 显示对话框并等待用户输入  
 Optional<String> result = dialog.showAndWait();  
 result.ifPresent(namekuang -> {  
 if(namekuang!=null)  
 name = namekuang; // 赋值给String变量  
 // 这里可以使用userName变量进行后续操作  
 });  
 }  
 @FXML  
 protected void onlistClick() throws IOException {  
 FXMLLoader fxmlLoader=new FXMLLoader(HelloController.class.getResource("score-list.fxml"));  
 Scene scene = new Scene(fxmlLoader.load());  
 Stage stage1 = new Stage();  
 stage1.setTitle("积分榜");  
 stage1.setScene(scene);  
 stage1.showAndWait();  
  
 }  
  
  
 @FXML  
 protected void onGuessButtonClick() {  
 System.*out*.println(resNum);  
 aright=0;  
 sright=0;  
 String Guess=shurukuang.getText();  
 if (!(*isNumeric*(Guess)))  
 {  
 if(Guess=="")  
 {  
 Alert alert = new Alert(Alert.AlertType.*ERROR*);  
 alert.setTitle("错误");  
 alert.setContentText("输入框不能为空");  
 alert.show();  
 resLabel0.setText("请输入四位数字");  
 }  
 else {  
 Alert alert = new Alert(Alert.AlertType.*ERROR*);  
 alert.setTitle("错误");  
 alert.setContentText("不要输入字母");  
 alert.show();  
 resLabel0.setText("请输入四位数字");  
 }  
 }  
 else if(Guess.length()!=4)  
 {  
 Alert alert = new Alert(Alert.AlertType.*ERROR*);  
 alert.setTitle("错误");  
 alert.setContentText("四位数字哦");  
 alert.show();  
 resLabel0.setText("请输入四位数字");  
 }  
 else {  
 if (Integer.*parseInt*(Guess) == resNum) {  
 resLabel0.setFont(new Font(20));  
 resLabel.setText(" ");  
 resLabel0.setText("恭喜你答对了,积分榜+1分 可以猜下一次了！");  
 if(scoreboard.cunzai(name)){  
 scoreboard.updateScore(name,scoreboard.getScore(name)+1);}  
 else {  
 scoreboard.addScore(name,1);  
 }  
 initialize();  
 } else {  
 int number=Integer.*parseInt*(Guess);  
 int gue[]=new int[4];  
 for (int i = 3; i >= 0; i--) {  
 gue[i] = number % 10; // 取最后一位数字  
 number /= 10; // 去掉最后一位数字  
 }  
 for(int i=0;i<4;i++)  
 {  
 for(int j=0;j<4;j++)  
 {  
 if(res[i]==gue[j])  
 {  
 if(i==j)  
 {  
 aright++;  
 }  
 else  
 {  
 sright++;  
 }  
 }  
 }  
  
 }  
 resLabel0.setText("猜错了，提示：");  
 resLabel.setText(aright+"A"+sright+"B");  
 }  
 }  
  
  
  
  
 }  
 @FXML  
 protected void ruleClick()  
 {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle("游戏规则");  
 alert.setHeaderText("猜数字的游戏规则如下");  
 alert.setContentText("开始游戏后，产生一个没有重复数" +  
 "字的 4 位随机数，用户每猜一个数字，显示出“完全猜中的数字个数”和“猜" +  
 "中数字但位置错误的数字个数”，比如 nAmB，数字 n 表示猜中的位置正确的数" +  
 "字个数，数字 m 表示数字正确而位置不对的数字个数。例如，正确答案为" +  
 "5234，如果用户猜 5346，则显示：1A2B，数字 1 表示数字 5 及其位置猜对了，" +  
 "数字 3 和 4 这两个数字猜对了，但是位置没对，记为 2B。然后，用户根据游戏提示的信息继续猜，直到猜中为止。");  
 alert.showAndWait();  
 }  
 public static boolean isNumeric(String str) {  
 if (str == null) {  
 return false; // 或者你可以根据需要抛出异常  
 }  
 Pattern pattern = Pattern.*compile*("\\d+");  
 Matcher matcher = pattern.matcher(str);  
 return matcher.matches();  
 }  
 public static int generateUniqueRandomFourDigitNumber() {  
 List<Integer> digits = new ArrayList<>();  
 for (int i = 0; i < 10; i++) {  
 digits.add(i);  
 }  
  
 Collections.*shuffle*(digits); // 打乱数字顺序  
  
 // 取前四位数字  
 int num = digits.get(0) \* 1000 + digits.get(1) \* 100 + digits.get(2) \* 10 + digits.get(3);  
  
 return num;  
 }  
}

package com.example.guessnum;  
  
import java.io.Serializable;  
  
public class Score implements Serializable {  
 private String playerName;  
 private int score;  
  
 public Score(String playerName, int score) {  
 this.playerName = playerName;  
 this.score = score;  
 }  
  
 public String getPlayerName() {  
 return playerName;  
 }  
  
 public int getScore() {  
 return score;  
 }  
  
 public void setScore(int score) {  
 this.score = score;  
 }  
  
 @Override  
 public String toString() {  
 return playerName + ": " + score;  
 }  
}

package com.example.guessnum;  
  
import javafx.scene.control.Alert;  
  
import java.io.\*;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Iterator;  
import java.util.List;  
  
public class Scoreboard {  
 private static final String *SCORES\_FILE* = "scores.txt";  
 private List<Score> scores;  
  
 public Scoreboard() {  
 scores = new ArrayList<>();  
 loadScores();  
 }  
  
 public void addScore(String playerName, int score) {  
 Score scoreObj = new Score(playerName, score);  
 scores.add(scoreObj);  
 saveScores();  
 }  
  
 public int getScore(String playername) {  
 for (Score score : scores) {  
 if (score.getPlayerName().equals(playername)) {  
 return score.getScore();  
 }  
 }  
 return 0;  
 }  
 public boolean cunzai(String playerName)  
 {  
 for (Score score : scores) {  
 if(score==null)break;  
 if (score.getPlayerName().equals(playerName)) {  
 return true;  
 }  
 }  
 return false;  
 }  
  
 public void updateScore(String playerName, int newScore) {  
 for (Score score : scores) {  
 if (score.getPlayerName().equals(playerName)) {  
 score.setScore(newScore);  
 saveScores();  
 return;  
 }  
 }  
 // 如果玩家不存在，可以选择添加新分数或不做任何操作  
 }  
  
 public List<Score> getScores() {  
 return Collections.*unmodifiableList*(scores);  
 }  
 public void deleteScore(String playerName) {  
 Iterator<Score> iterator = scores.iterator();  
 while (iterator.hasNext()) {  
 Score score = iterator.next();  
 if (score.getPlayerName().equals(playerName)) {  
 iterator.remove();  
 saveScores();  
 return;  
 }  
 }  
 // 如果玩家不存在，可以选择打印消息或不做任何操作  
 Alert alert = new Alert(Alert.AlertType.*ERROR*);  
 alert.setTitle("错误");  
 alert.setContentText("用户不存在");  
 alert.show();  
 }  
  
  
 private void saveScores() {  
 try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(*SCORES\_FILE*))) {  
 oos.writeObject(scores);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private void loadScores() {  
 File scoresFile = new File(*SCORES\_FILE*);  
 if (scoresFile.exists()) {  
 try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(scoresFile))) {  
 scores = (List<Score>) ois.readObject();  
 } catch (IOException | ClassNotFoundException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

package com.example.guessnum;  
  
import javafx.application.Platform;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.TableColumn;  
import javafx.scene.control.TableView;  
import javafx.scene.control.TextInputDialog;  
import javafx.scene.control.cell.PropertyValueFactory;  
  
import java.util.List;  
import java.util.Optional;  
  
public class ScoreController {  
 @FXML  
 private TableColumn<Score,String> userN;  
 @FXML  
 private TableColumn<Score,Integer> point;  
 @FXML  
 private TableView<Score> myTable;  
 @FXML  
 private Button deletename;  
 public Scoreboard scoreboard=new Scoreboard();  
 private String dname;  
 @FXML  
 private void initialize() {  
 Scoreboard scoreboard=new Scoreboard();  
 userN.setCellValueFactory(new PropertyValueFactory<>("playerName"));  
 point.setCellValueFactory(new PropertyValueFactory<>("score"));  
 ObservableList<Score>data=FXCollections.*observableArrayList*(scoreboard.getScores());  
 myTable.setItems(data);  
  
 }  
 @FXML  
 public void deleteName()  
 {  
  
 TextInputDialog dialog = new TextInputDialog();  
  
 dialog.setHeaderText("要删除的用户名是");  
 dialog.setContentText("在这里输入内容:");  
 // 显示对话框并等待用户输入  
 Optional<String> result = dialog.showAndWait();  
 result.ifPresent(name -> {  
 dname=name;// 赋值给String变量  
 // 这里可以使用userName变量进行后续操作  
 scoreboard.deleteScore(dname);  
 });  
 ObservableList<Score>data=FXCollections.*observableArrayList*(scoreboard.getScores());  
 myTable.setItems(data);  
  
 }  
  
}

**附上修改器可以快速修改排行榜**

package com.example.guessnum;  
  
import java.util.List;  
  
public class 修改器 {  
 public static void main(String[] args) {  
 Scoreboard scoreboard = new Scoreboard();  
  
 // 添加一些分数  
 scoreboard.addScore("hua",123);  
 scoreboard.addScore("小米",264);  
 scoreboard.addScore("小米2",64);  
  
 // 获取并打印积分榜  
 List<Score> scores = scoreboard.getScores();  
 for (Score score : scores) {  
 System.*out*.println(score);  
 }  
  
 // 退出程序时，分数已经保存在文件中了，下次程序启动时会自动加载  
 }  
}

**界面代码**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.geometry.Insets?>  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.control.ToolBar?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.layout.Pane?>  
<?import javafx.scene.layout.VBox?>  
  
<VBox alignment="CENTER" prefHeight="361.0" prefWidth="443.0" spacing="20.0" stylesheets="@css/ElementFX.css" xmlns="http://javafx.com/javafx/19" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.guessnum.HelloController">  
 <padding>  
 <Insets bottom="20.0" left="20.0" right="20.0" top="20.0" />  
 </padding>  
 <ToolBar prefHeight="114.0" prefWidth="403.0">  
 <items>  
 <Button mnemonicParsing="false" onAction="#ruleClick" text="游戏规则" />  
 <Button mnemonicParsing="false" onAction="#onuserClick" text="用户名" />  
 <Button mnemonicParsing="false" onAction="#onlistClick" text="积分榜" />  
 </items>  
 </ToolBar>  
 <Pane prefHeight="151.0" prefWidth="403.0" style="-fx-border-color: black;">  
 <children>  
 <Label fx:id="resLabel0" layoutX="22.0" layoutY="24.0" prefHeight="41.0" prefWidth="256.0" text="还没有开始猜哦" />  
 <Label fx:id="resLabel" layoutX="110.0" layoutY="37.0" text=" " />  
 </children>  
 </Pane>  
 <AnchorPane prefHeight="200.0" prefWidth="200.0">  
 <children>  
 <TextField fx:id="shurukuang" layoutX="152.0" layoutY="40.0" />  
 <Button layoutX="246.0" layoutY="77.0" onAction="#onGuessButtonClick" text="就是你了!" />  
 <Label layoutX="29.0" layoutY="44.0" text="请输入你的猜测" />  
 </children>  
 </AnchorPane>  
</VBox>

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.TableColumn?>  
<?import javafx.scene.control.TableView?>  
<?import javafx.scene.layout.AnchorPane?>  
  
<AnchorPane prefHeight="424.0" prefWidth="353.0" stylesheets="@css/ElementFX.css" xmlns="http://javafx.com/javafx/19" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.guessnum.ScoreController">  
 <children>  
 <TableView fx:id="myTable" layoutX="-2.0" layoutY="-2.0" prefHeight="367.0" prefWidth="355.0">  
 <columns>  
 <TableColumn fx:id="userN" prefWidth="112.00000154972076" text="用户名" />  
 <TableColumn fx:id="point" prefWidth="242.39995574951172" text="积分" />  
 </columns>  
 </TableView>  
 <Button fx:id="deletename" layoutX="14.0" layoutY="383.0" mnemonicParsing="false" onAction="#deleteName" text="删除" />  
 </children>  
</AnchorPane>