**广州大学学生实验报告**

**开课实验室：**电子楼416B **2019年9月9日**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | 计算机科学与网络工程学院 | | **年级、专业、班** | 软件171 | **姓名** | 谢金宏 | **学号** | 1706300001 |
| **实验课程名称** | | Java语言实验 | | | | | **成绩** |  |
| **实验项目名称** | | 实验1 | | | | | **指导老师** | 王宇 |
| **一、实验目的**  **1. 掌握开发Java应用程序的步骤，掌握Java应用程序的基本结构。**  **2. 掌握Java基本数据类型在命令行的输入和输出的方法。**  **3. 熟悉如何使用Java分支和循环语句解决问题。**  **4. 熟悉类的基本设计方法，根据Java类的继承机制有效解决问题。**  **二、实验任务**  **1. 编写一个Java程序，在主类的main方法中实现下列功能：**  **·程序随机分配给用户一个1至100之间的整数。**  **·用户通过键盘输入自己的猜测。**  **·程序返回提示信息，提示信息分别是：“猜大了”、“猜小了”和“猜对了”。**  **·用户可根据提示信息再次输入猜测，直到提示信息是“猜对了”。**  **·用户猜对以后，显示猜测次数，并提供“重新开始”和“退出”功能。**  **2. 假定要为某个公司编写雇员工资支付程序，这个公司有各种类型的雇员（Employee），不同类型的雇员按不同的方式支付工资：**  **·经理（Manager）：每月获得一份固定工资。**  **·销售人员（Salesman）：在基本工资的基础上每月还有销售提成。**  **·工人（Worker）：按照每月工作的天数计算工资。**  **根据上述要求使用类的继承和相关机制描述这些功能，并编写一个Java应用程序，演示这些类的用法。**  **三、实验内容**  **1. 猜数字**  /\*\*   \* GuessNumber.java   \*/  import java.util.Scanner;  import java.util.Random;  public class GuessNumber {      private static Scanner in = new Scanner(System.in);        public static void main(String args[]) {          System.out.println("猜数字游戏\n");            do {              play();          } while (resume());          System.out.println("再见！");      }      // 与用户进行猜数字游戏。      private static void play() {          int guessCount = 0;  // 储存用户猜测的次数          // 生成[1, 100]的随机数。          Random rand = new Random();          int randNum = 1 + rand.nextInt(100);            System.out.println("我已经想好一个1~100之间的数字，请输入一个数字：");          while (in.hasNext()) {              try {                  // 获取用户的输入。                  String line = in.nextLine();                  int guessNum = Integer.parseInt(line);                  guessCount++;                    // 根据用户的猜测给出提示。                  if (guessNum < randNum) {                      System.out.println("猜的数字太小，请再猜一次。");                  } else if (guessNum > randNum) {                      System.out.println("猜的数字太大，请再猜一次。");                  } else {                      System.out.println("恭喜你猜对啦！");                      System.out.println("你已经猜了" + guessCount + "次。");                      return;                  }              } catch (Exception e) {                  e.printStackTrace();                  System.out.println("请不要输入一些我不懂的东西哦。");              }          }      }      // 询问用户是否再来一局。      private static boolean resume() {          System.out.println("再来一局吗？[y/n]");            while (in.hasNext()) {              try {                  String line = in.nextLine().toLowerCase();                  if (line.equals("y")) {                      return true;                  } else if (line.equals("n")) {                      return false;                  }                  throw new Exception("用户输入不合法。");              }              catch (Exception e) {                  e.printStackTrace();                  System.out.println("请不要输入一些我不懂的东西哦。");              }          }          return false;      }  }  **2. 雇员工资**  /\*\*   \* SalaryCalculator.java   \*/  public class SalaryCalculator {      public static void main(String args[]) {          Employee[] employees = {              new Manager("经理", 9800),              new Salesman("销售A", 5000, 1200), new Salesman("销售B", 4800, 1400),              new Worker("工人A", 25), new Worker("工人B", 27), new Worker("工人C", 28)          };          for (Employee employee : employees) {              System.out.println(employee.name + "的工资为：" + employee.ComputeSalary());          }      }  }  abstract class Employee {      String name;      abstract double ComputeSalary();      Employee(String name) {          this.name = name;      }  }  class Manager extends Employee {      double salary;        Manager(String name, double salary) {          super(name);          this.salary = salary;      }      @Override      double ComputeSalary() {          return salary;      }  }  class Salesman extends Employee {      double basicSalary;      double bonus;      Salesman(String name, double basicSalary, double bonus) {          super(name);          this.basicSalary = basicSalary;          this.bonus = bonus;      }      @Override      double ComputeSalary() {          return basicSalary + bonus;      }  }  class Worker extends Employee {      final static double SALARY\_PER\_DAY = 180;        int workdays;      Worker(String name, int workdays) {          super(name);          this.workdays = workdays;      }      @Override      double ComputeSalary() {          return workdays \* SALARY\_PER\_DAY;      }  }  **四、实验结果记录（程序运行结果截图）**    Figure 1 猜数字游戏a      Figure 2 猜数字游戏b    Figure 3 雇员工资 | | | | | | | | |
|  | | | | | | | | |