吉软 java 基础第二周测试

【金城君组】【蔡俊豪】

说明：
1. 上方【组】填入所在的组，上方【姓名】填入自己的真实姓名。
2. 答题方式，基于 Word 文档基础上答题
3. 编程题可利用工具编程完以后，复制到该文档内。
4. 答完以后，导成 PDF。以姓名.PDF 命名。上传至老师指定邮箱。

一、 选择题（共 10 题，每题 2 分）

1、 以下对异常的描述不正确的有(C)

A) 异常分为 Error 和 Exception

B) Throwable 是所有异常类的直接父类

C) Exception 是所有异常类直接父类

D) Exception 包括 RuntimeException 和 RuntimeException 之外的异常

2、 在 try-catch-finally 语句块中，以下可以单独与 finally 一起使用的是(B)

A) catch

B) try

C) throws

D) throw

3、下面代码运行结果是(B)

```
1  public class Demo{
2      public int add(int a,int b){
3          try{
4              return a+b;
5          }catch(Exception e){
6              System.out.println("catch 语句块");
7          }finally{
8              System.out.println("finally 语句块");
9          }
10         return 0;
11     }
12
13     public static void main(String[] args){
14         Demo demo = new Demo();
15         System.out.println("和是："+demo.add(9,34));
16     }
17 }
```

A) 编译异常

B) finally 语句块 和是：43

C) 和是：43 finally 语句块
D) catch 语句块 和是：43

4. 以下描述不正确的有(D)
A) try 块不可以省略
B)可以使用多重 catch 块
C) finally 块可以省略
D) catch 块和 finally 块可以同时省略

5、以下对自定义异常描述正确的是(C)
A) 自定义异常必须继承 Exception
B) 自定义异常可以继承自 Error
C) 自定义异常可以更加明确定位异常出错的位置和给出详细出错信息
D) 程序中已经提供了丰富的异常类，使用自定义异常没有意义

6、在 Java 中，关于 HashMap 类的描述，以下选项错误的是(B)
A) HashMap 使用键/值得形式保存数据
B) HashMap 能够保证其中元素的顺序
C) HashMap 允许将 null 用作键
D) HashMap 允许将 null 用作值

7、Java 中的集合类包括 ArrayList、LinkedList、HashMap 等类，下列关于集合类描述错误的是(D)
A) ArrayList 和 LinkedList 均实现了 List 接口
B) ArrayList 的访问速度比 LinkedList 快
C) 添加和删除元素时，ArrayList 的表现更佳
D) HashMap 实现 Map 接口，它允许任何类型的键和值对象，并允许将 null 用作键或值

8、下列哪个不是线程生命周期的状态(C)
A)新建状态
B)阻塞状态
C)启动状态
D)就绪状态

9、下列哪个关键词可以解决线程同步问题(B)
A)finally
B)synchronized
C)return
D)static

10、 给定如下所示的 JAVA 代码，则运行时，会产生(B)类型的异常

```
String s = null;        s.concat("abc");
```

A) ArithmeticException

B)NullPointerException
C)IOException
D)ClassNotFoundException

二、 填空题（共 10 题，每题 2 分）

1、____Math__类提供了许多用于数学运算的静态方法。

2、手动抛出异常可以用____throw___关键字。

3、异常处理流程中，其中_____catch__代码块紧跟在 try 代码块后面，用来捕获并处理异常。

4、线程的 start 方法存在于__Thread___类中。

5、在 Java 中，只包含一个抽象方法的接口被称为_函数式接口__

6、向 HashSet 集合添加元素时，要保证元素的唯一性，需要重写__hashCode__、_equals 方法

7、Java 的多线程中，当线程进入阻塞状态后，会恢复成 __就绪__状态。

8、SimpleDateFormat 类中用___format____方法把 Date 类型转成 String 类型。

9、Java 中，存放最大单值的集合的最顶级接口是___Collection____

10、Java 中用____enum____关键字声明一个枚举类。

三、 简答题（共 5 题，每题 4 分）

1、分析 sleep()和 wait()方法的区别。
sleep()常用于一定时间内暂停线程执行。 wait()常用于线程间交互和通信。
相同点：这两个方法都可以让当前线程进入阻塞状态
不同点：方法声明的位置不同。调用的要求不同，sleep 可以任意位置调用，wait 必须在同步代码块或同步方法中。是否释放同步监视器：如果 2 个方法都是用在同步代码块或同步方法中，sleep 不会释放锁，wait 释放锁。

2、简述 List、Set、Map 集合的特点。
List：有顺序以线性方式存储,可以存放重复对象；

Set：无顺序，不包含重复的元素；
Map：键必须是唯一。

3、说明创建线程有几种方法？分别是什么？
4 种
继承 Thread 类，重写 run()方法。实现 Runnable 接口，重写 run()方法，用这个实例作为 Thread 类一个对象的 target。实现 Callable 接口，并结合 Future 实现，通过线程池创建线程。

4、简述 ArrayList 和 LinkedList 以及 Vector 的区别？
ArrayList：一个可改变大小的数组.当元素加入时,其大小将会动态地增长.内部的元素可以直接通过 get 与 set 方法进行访问.元素顺序存储 ,随机访问很快，删除非头尾元素慢，新增元素慢而且费资源，非线程安全.
LinkedList： 是一个双链表,在添加和删除元素时具有比 ArrayList 更好的性能.但在 get 与 set 方面弱于 ArrayList，非线程安全.
Vector：同步的，开销比 ArrayList 要大。Vector 和 ArrayList 在更多元素添加进来时会请求更大的空间。Vector 每次请求其大小的双倍空间。

5、说明 String、StringBuffer、StringBuilder 的区别？
都是 final 类，都不允许被继承；String 类长度是不可变的，StringBuffer 和 StringBuilder 类长度是可以改变的；StringBuffer 类是线程安全的，StringBuilder 不是线程安全的；

四、 编程题（共 3 题，第（1）(2）题 10 分，第（3）题 20 分）

1、模拟注册登录案例：

a) 创建 User 类，用来描述用户信息，分别有 username 和 password 属性。
b) 从控制台输入注册用户的信息保存在集合中。
c) 从控制台输入登录用户的信息，判断集合中是否有匹配的信息

```
4 个用法  新 *
public class User {
    3 个用法
    String username;
    2 个用法
    String password;

    1 个用法  新 *
    public User() {
    }

    1 个用法  新 *
    public String getUsername() {
        return username;
    }

    1 个用法  新 *
    public void setUsername(String username) {
        this.username = username;
    }

    1 个用法  新 *
    public String getPassword() {
        return password;
    }

    1 个用法  新 *
    public void setPassword(String password) {
        this.password = password;
    }
}
```

```
0 个用法  新 *
public class UserLauncher {
    新 *
    public static void main(String[] args) {
        Map<Integer,User> map = new HashMap<Integer,User>();
        System.out.println("请输入用户名: ");
        Scanner scanner = new Scanner(System.in);
        User user = new User();
        user.setUsername(scanner.nextLine());
        System.out.println("请输入密码: ");
        user.setPassword(scanner.nextLine());
        map.put(1, user);
        System.out.println(user.getUsername());
        System.out.println(user.getPassword());
        System.out.println(map.containsValue(user));
        System.out.println("请输入要查询的用户名: ");
        String name = scanner.nextLine();

        Set<Integer> keySet = map.keySet();

        for (Integer key : keySet) {
            if (map.get(key).username.equals(name)){
                System.out.println("存在");
                System.out.println(map.get(key));
            }
        }
    }
}
```

2、创建两条线程，线程 A 打印 1-100 的所有奇数，线程 B 打印 1-100 所有的偶数

```
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/15
 */
0 个用法  新 *
public class Demo02 {

    新 *
    public static void main(String[] args) {
        Thread1 t1 = new Thread1();
        Thread2 t2 = new Thread2();
        t1.start();
        t2.start();
    }
}

2 个用法  新 *
class Thread1 extends Thread{
    新 *
    public void run(){
        for(int i = 2;i <= 100;i += 2){
            System.out.println("i= " + i);
        }
    }
}

2 个用法  新 *
class Thread2 extends Thread{
    新 *
    public void run(){
        for(int i = 1;i <= 100;i += 2){
            System.out.println("i= " + i);
        }
    }
}
```

3、请用面向对象的思想，设计自定义类描述演员和运动员的信息

设定

1)演员类:

属性包括：姓名，年龄，性别，毕业院校，代表作

方法包括： 自我介绍

2)运动员类:

属性包括：姓名，年龄，性别，运动项目，历史最好成绩

方法包括：自我介绍

要求

a.分析演员和运动员的公共成员，提取出父类—人类

b.利用继承机制，实现演员类和运动员类

c.编写测试类，分别测试人类，演员类和运动员类对象及相关方法

```java
public class Athlete extends Person {

    2 个用法
    String sportItem;
    2 个用法
    String gradeOfHistory;

    1 个用法  新 *
    public Athlete(String name, Integer age, String gender, String sportItem, String gradeOfHistory) {
        super(name, age, gender);
        this.sportItem = sportItem;
        this.gradeOfHistory = gradeOfHistory;
    }



    2 个用法  新 *
    @Override
    public void introduceMyself() {
        System.out.println("我是运动员");
    }

    新 *
    @Override
    public String toString() {
        return "Athlete{" +
                "sportItem='" + sportItem + '\'' +
                ", gradeOfHistory='" + gradeOfHistory + '\'' +
                ", name='" + name + '\'' +
                ", age=" + age +
                ", gender='" + gender + '\'' +
                '}';
    }
}
```

```java
package Exams.Exam02;

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/15
 */
0 个用法  新 *
public class LauncherForActorAndAthlete {
    新 *
    public static void main(String[] args) {
        Person actor = new Actor( name: "小红",  age: 22,  gender: "女",  graduateSchool: "吉林艺术大学",  production: "阿里郎");
        Athlete athlete = new Athlete( name: "小刚",  age: 20,  gender: "男",  sportItem: "跳高",  gradeOfHistory: "230");
        actor.introduceMyself();
        athlete.introduceMyself();
        System.out.println(actor.toString());
        System.out.println(athlete.toString());
    }
}
```

```
1个用法  新 *
public class Actor extends Person {
    2个用法
    String graduateSchool;
    2个用法
    String production;

    1个用法  新 *
    public Actor(String name, Integer age, String gender, String graduateSchool, String production) {
        super(name, age, gender);
        this.graduateSchool = graduateSchool;
        this.production = production;
    }

    2个用法  新 *
    @Override
    public void introduceMyself() {
        System.out.println("我是演员");
    }

    新 *
    @Override
    public String toString() {
        return "Actor{" +
                "graduateSchool='" + graduateSchool + '\'' +
                ", production='" + production + '\'' +
                ", name='" + name + '\'' +
                ", age=" + age +
                ", gender='" + gender + '\'' +
                '}';
    }
}
```