1．Java中提供了名为（）的包装类来包装原始字符串类型。

A．Integer

B．Char

C．Double

D．String

答案：D

2．java.lang包的（）方法比较两个对象是否相等，相等返回true。

A．toString（）

B．equals（）

C．compare（）

D．以上所有选项都不正确

答案：B

3．使用（）方法可以获得Calendar类的实例。

A．get（）

B．equals（）

C．getTime（）

D．getInstance（）

答案：D

4．下面的集合中，（）不可以存储重复元素。

A．Set

B．Collection

C．Map

D．List

答案：C

5．关于Map和List，下面说法正确的是(  )。

A．Map继承List

B．List中可以保存Map或List

C．Map和List只能保存从数据库中取出的数据

D．Map的value可以是List或Map

答案：D

6．给定如下Java代码，编译运行的结果是（）。

import java.util.\*;

public class Test {

public static void main(String[] args) {

LinkedList list=new LinkedList();

list.add(“A”);

list.add(2,”B”);

String s=(String)list.get(1);

System.out.println(s);

}

}

A．编译时发生错误

B．运行时引发异常

C．正确运行，输出：A

D．正确运行，输出：B

答案：B

7．请写出下列语句的输出结果

char[] data={‘a’,'b’,'c’,'d’};

System.out.println(String.valueOf(10D));

System.out.println(String.valueOf(3>2));

System.out.println(String.valueOf(data,1,3));

参考答案：

10.0

true

bcd

8．写出下面代码运行后的输出结果是。

public class Arrtest {

public static void main(String kyckling[]){

int i[ ] = new int[5];

System.out.println(i[4]);

amethod();

Object obj[ ] = new Object[5];

System.out.println(obj[2]);

}

public static void amethod(){

int K[ ] = new int[4];

System.out.println(K[3]);

}

}

参考答案：

0

0

null

9．什么是封装？Java语言中的封装类有哪些？

参考答案：

封装是表示把数据项和方法隐藏在对象的内部，把方法的实现内容隐藏起来。Java中的封装类有Double、Integer、Float、Byte、Long、Character、Short和Boolean等类。

10．什么是泛型？使用泛型有什么优点？泛型List和普通List有什么区别？

参考答案：

泛型是对Java语言的数据类型系统的一种扩展，以支持创建可以按类型进行参数化的类。可以把类型参数看做是使用参数化类型时指定的类型的一个占位符。

优点：提高Java程序的类型安全；消除强制类型转换；提高代码的重用率。

泛型List可以实例化为只能存储某种特定类型的数据，普通List可以实例化为存储各种类型的数据。通过使用泛型List对象，可以规范集合对象中存储的数据类型，在获取集合元素时不用进行任何强制类型转换。

11．编写一个程序，实现下列功能：

? 测试两个字符串String str1=”It is”和String str2=”It is”；是否相等

? 将” a book.”与其中的str1字符串连接

? 用m替换新字符串中的i

参考代码：

public class Ex11 {

public static void main(String[] args) {

String str1=”It is”;

String str2=”It is”;

//比较字符串

System.out.println(“str1==str2的结果：“+(str1==str2));

System.out.println(“str1.equals(str2)的结果：“+str1.equals(str2));

//连接字符串

String str3=str1.concat(“a book”);

System.out.println(“连接后的字符串为：“+str3);

//替换字符

String str4=str3.replace(‘i’,'m’);

System.out.println(“替换后的字符串为：“+str4);

}

}

12．编程计算距当前时间10天后的日期和时间，并用“xxxx年xx月xx日”的格式输出新的日期和时间。

参考代码：

import java.util.Calendar;

public class Ex12 {

public static void main(String[] args) {

Calendar cal=Calendar.getInstance();

cal.add(Calendar.DAY\_OF\_YEAR,10);

String strDate=cal.get(Calendar.YEAR)+”年“

+(cal.get(Calendar.MONTH)+1)+”月“

+cal.get(Calendar.DATE)+”日“;

System.out.println(“10天后的日期为：“+strDate);

}

}

13．创建一个类Stack，代表堆栈（其特点为：后进先出），添加方法add(Object obj)、方法get（）和delete( )，并编写main方法进行验证。

参考代码：

import java.util.LinkedList;

import java.util.List;

class Stack{

LinkedList list;

public Stack() {

list = new LinkedList();

}

public void add(Object obj){

list.addFirst(obj);

}

public Object get(){

return list.getFirst();

}

public void delete(){

list.removeFirst();

}

}

public class Ex13 {

public static void main(String[] args) {

Stack stack=new Stack();

stack.add(“1″);

stack.add(“2″);

stack.add(“3″);

System.out.println(stack.get());

stack.delete();

System.out.println(stack.get());

}

}

14．编写程序，计算任意两个日期之间间隔的天数。

参考代码：

import java.util.Calendar;

public class Ex14 {

public static void main(String[] args) {

Calendar c1 = Calendar.getInstance();

c1.set(2010, 7, 1);

Calendar c2 = Calendar.getInstance();

long ca1 = c1.getTimeInMillis();

long ca2 = c2.getTimeInMillis();

// 计算天数

long days = (ca2 - ca1) / (24 \* 60 \* 60 \* 1000);

System.out.println(days);

}

}

15．创建一个HashMap对象，添加一些学生的姓名和成绩：张三：90分，李四，83分。接着从HashMap中获取并他们的姓名和成绩，然后把李四的成绩改为100分，再次输出他们的信息。

参考代码：

import java.util.HashMap;

public class Ex15 {

public static void main(String[] args) {

HashMap map=new HashMap();

map.put(“张三“,90);

map.put(“李四“,83);

System.out.println(“修改前的成绩：“);

System.out.println(map);

map.put(“李四“,100);

System.out.println(“修改后的成绩：“);

System.out.println(map);

}

}

16．编写一个程序，用parseInt方法将字符串200由十六进制转换为十进制的int型数据，用valueOf方法将字符串123456转换为float型数据。

参考代码：

public class Ex16 {

public static void main(String[] args) {

String str1=”200″;

System.out.println(Integer.parseInt(str1,16));

String str2=”123456″;

System.out.println(Float.parseFloat(str2));

}

}

17．编写程序，将long型数据987654转换为字符串，将十进制数365转换为十六进制数表示的字符串。

参考代码：

public class Ex17 {

public static void main(String[] args) {

long num=987654L;

int i=365;

System.out.println(“Long类型转换为String：“+String.valueOf(num));

String HexI=DtoX(i);

System.out.println(HexI);

}

//转换函数

public static String DtoX(int d)

{

String x=”";

if(d<16){

x=change(d);

}

else{

int c;

int s=0;

int n=d;

while(n>=16){

s++;

n=n/16;

}

String [] m=new String[s];

int i=0;

do{

c=d/16;

//判断是否大于10，如果大于10，则转换为A-F的格式

m[i++]=change(d%16);

d=c;

}while(c>=16);

x=change(d);

for(int j=m.length-1;j>=0;j–){

x+=m[j];

}

}

return x;

}

//判断是否为10-15之间的数，如果是则进行转换

public static String change(int d){

String x=”";

switch(d){

case 10:

x=”A”;

break;

case 11:

x=”B”;

break;

case 12:

x=”C”;

break;

case 13:

x=”D”;

break;

case 14:

x=”E”;

break;

case 15:

x=”F”;

break;

default:

x=String.valueOf(d);

}

return x;

}

}

18．编写一个程序，接收以克为单位的一包茶叶的单位重量、卖出的包数和每克的价格，计算并显示出销售的总额。其中三个数据一行输入，数据间用“-”分隔。比如：输入“3-100-2.1”，表示每包的重量为3克，共卖出100包，每克的价格为2.1元。此时的销售总额为630元。

参考代码：

import java.util.Scanner;

public class Ex18 {

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);

System.out.println(“请依次输入重量、包数、价格，并以-分隔：“);

String strIn=scan.nextLine();

Scanner sc=new Scanner(strIn);

sc.useDelimiter(“-”); //设置分隔符

int num=sc.nextInt();

int bag=sc.nextInt();

float price=sc.nextFloat();

double total=price\*num\*bag;

System.out.println(“销售总额为：“+total);

}

}

19．编写一个泛型方法，能够返回一个int类型数组的最大值和最小值、String类型数组的最大值和最小值（按字典排序）。

参考代码：

class Pair<T>{

private T min;

private T max;

public Pair() {  min = null; max = null;  }

public Pair(T min, T max) {  this.min = min;  this.max = max;  }

public T getMin() {  return min;  }

public T getMax() {  return max;  }

public void setMin(T newValue) {  min = newValue;  }

public void setMax(T newValue) {  max = newValue;  }

}

class ArrayAlg {

public static <T extends Comparable> Pair<T> minmax(T[ ] a) {

if (a == null || a.length == 0) {

return null;

}

T min = a[0];T max = a[0];

for (int i = 1; i < a.length; i++) {

if (min.compareTo(a[i]) > 0) {min = a[i];}

if (max.compareTo(a[i]) < 0) { max = a[i];}

}

return new Pair<T>(min, max);

}

}

public class Ex19 {

public static void main(String[] args) {

//测试整型数组

Integer[] arrI={1,2,3,4,5,6};

Pair<Integer> p1=ArrayAlg.minmax(arrI);

System.out.println(“整型数组的最小值：“+p1.getMin().intValue());

System.out.println(“整型数组的最大值：“+p1.getMax().intValue());

//测试字符串数组

String[ ] words ={“able”,”word”,”excel”,”course”,”java”,”c#”};

Pair<String> p2=ArrayAlg.minmax(words);

System.out.println(“字符串数组的最小值：“+p2.getMin());

System.out.println(“字符串数组的最大值：“+p2.getMax());

}

}

20．编写一个泛型方法，接受对象数组和集合作为参数，将数组中的对象加入集合中。并编写代码测试该方法。

参考代码：

import java.util.\*;

public class Ex20 {

static <T> void fromArrayToCollection(T[]a, Collection<T> c){

for (T o : a){

c.add(o);

}

}

public static void main(String[] args) {

Integer[] ia = new Integer[100];

Collection<Number> cn = new ArrayList<Number>();

fromArrayToCollection(ia, cn);// T是Number类型

System.out.println(cn);

}

}

21．试编写一个List类型的对象只能存储通讯录（存储同学的姓名和联系方式），并输出通讯录的列表到控制台。

参考代码：

import java.util.\*;

class Student{

private String name;

private String phone;

public Student(String name, String phone) {

this.name = name;

this.phone = phone;

}

public String toString() {

return name+”:”+phone;

}

}

public class Ex21 {

public static void main(String[] args) {

Student st1=new Student(“John”,”23214″);

Student st2=new Student(“Alice”,”4563″);

List<Student> list=new ArrayList<Student>();

list.add(st1);list.add(st2);

for(int i=0;i<list.size();i++)

System.out.println(list.get(i));

}

}

22．设计一个程序，基于泛型Map实现10个英文单词的汉语翻译，即通过单词得到它的中文含义。

参考代码：

import java.util.\*;

public class Ex22 {

public static void main(String[] args) {

String[] eng={“Apple”,”Orange”,”Green”};

String[] chs={“苹果“,”桔子“,”绿色“};

Map<String,String> map=new HashMap<String,String>();

for(int i=0;i<eng.length;i++)

map.put(eng[i],chs[i]);

String test=”Orange”;

System.out.println(test+”翻译：“+map.get(test));

}

}

1．什么是异常？什么是Java的异常处理机制？

参考答案：

异常是指程序运行过程中产生的错误，它出现在程序运行过程中。

异常处理机制是为程序提供错误处理的能力。根据这个机制，对程序运行时可能遇到的异常情况，预先提供一些处理的方法。在程序执行代码的时候，一旦发生异常，程序会根据预定的处理方法对异常进行处理，处理完成后，程序进行运行。

2．Java中的异常分为哪几类？

参考答案：

Java中的异常分为两种类型：

? 内部错误：又称为致命错误。比如：硬盘损坏、软驱中没有软盘

? 运行时异常：比如除数为0、数组下标越界

3．所有异常的父类是（）。

A．Error B．Throwable C．RuntimeException D．Exception

答案：B

4．下列（）操作不会抛出异常。

A．除数为零 B．用负数索引访问数组

C．打开不存在的文件 D．以上都会抛出异常

答案：D

5．能单独和finally语句一起使用的块是（）。

A．try B．throws

C．throw D．catch

答案：A

6．在多重catch块中同时使用下列类时，（）异常类应该最后列出。

A．Exception

B．ArrayIndexOutOfBoundsException

C．NumberFormatException

D．ArithmeticException

答案：A

7．执行下面的代码会引发（）异常。

String str=null;

String strTest=new String(str);

A．InvalidArgumentException

B．IllegalArgumentException

C．NullPointerException

D．ArithmeticException

答案：C

8．这段代码的输出结果是（）。

try{

System.out.print(“try，“);

return;

} catch(Exception e){

System.out.print(“catch，“);

} finally {

System.out.print(“finally”);

}

A．try，

B．try，catch，

C．try，finally

D．try， catch，finally

答案：C

9．这个方法的返回值是（）。

public int count() {

try{

return 5/0;

} catch(Exception e){

return 2\*3;

} finally {

return 3;

}

}

A．0

B．6

C．3

D．程序错误

答案：C

10．编写一个程序，产生ArrayIndexOutOfBoundsException异常，并捕获该异常，在控制台输出异常信息。

参考代码：

public class Ex10 {

public static void main(String[] args) {

int[] arr=new int[2];

try {

System.out.println(arr[2]);

} catch (ArrayIndexOutOfBoundsException e) {

e.printStackTrace();

}

}

}

11．设计一个Java程序，自定义异常类，从键盘输入一个字符串，如果该字符串值为“abc”，则抛出异常信息，如果从键盘输入的是其他字符串，则不抛出异常。

参考代码：

import java.util.Scanner;

class MyException extends Exception{

private String errorMsg;

//getter和setter方法

public MyException(String errorMsg){

this.errorMsg=errorMsg;

}

@Override

public String toString() {

return errorMsg;

}

}

public class Ex11 {

public static void main(String[] args) {

String strIn;

Scanner scan=new Scanner(System.in);

strIn=scan.nextLine();

try {

if(strIn.equals(“abc”))

throw new MyException(“输入的字符串不正确！“);

} catch (MyException e) {

System.out.println(e);

}

}

}

12．设计一个Java程序，从键盘输入两个数，进行减法运算。当输入串中含有非数字时，通过异常处理机制使程序正常运行。

参考代码：

import java.util.\*;

public class Ex12 {

public static void main(String[] args) {

int num1,num2;

Scanner in=new Scanner(System.in);

try {

num1=in.nextInt();

} catch (InputMismatchException e) {

System.out.println(“第一个数格式不对“);

num1=0;

}

try {

num2=in.nextInt();

} catch (InputMismatchException e) {

System.out.println(“第二个数格式不对“);

num2=0;

}

System.out.println(“num1-num2=”+(num1-num2));

}

}

13．自定义异常类，在进行减法运算时，当第一个数大于第二个数时，抛出“被减数不能小于减数”，并编写程序进行测试。

参考代码：

import java.util.Scanner;

//MyException类的定义（同第11题）

public class Ex13 {

public static void main(String[] args) {

int num1,num2;

Scanner scan=new Scanner(System.in);

num1=scan.nextInt();

num2=scan.nextInt();

try {

if(num1<num2)

throw new MyException(“被减数不能小于减数“);

} catch (MyException e) {

System.out.println(e);

}

}

}