1.简述对JVM,JRE,JDK的理解。

JVM 是 Java 虚拟机英文的缩写，JVM 是Java实现跨平台的基础，不同的操作系统具有不同的JVM，系统中有JVM Java程序才能够在系统中运行。

JRE 是 Java 运行环境的英文缩写，JRE 包含了 JVM 及 Java 运行时的核心类库，Java 程序如果需要能够在系统中运行必需要在系统中安装JRE。

JDK 是 Java 开发工具集的英文缩写，JDK 包含了 JRE 及 Java 开发工具。

它们之间的关系是 JDK > JRE >JVM

2.简述环境变量path和classpath的作用？

配置环境变量的目的是为了让 Java 开发工具在任何路径下都能够被系统找到。

classpath 的作用是指定了 class 文件的路径信息。

3.独立编写hello world程序，编译并运行。

public class HelloWorld{

public static void main(String[] args){

System.out.pringln("Hello World!");

}

}

4.java语言是否区分大小写?class关键字的作用,javac,java 的作用？

Java 语言区分大小写

class 文件是 Java 文件通过 Java 编译器 javac 编译之后的文件。

class 关键字的作用是指明一个类的关键字，用于标注 class 文件的开头。

javac : 启动 Java 编译器，将 Java 源文件编译成 class 文件。

java : 启动 JVM 解析并执行 class 文件。

5.简述main方法(是什么，什么特点，怎么用，何时用？)？

main 方法是程序的主入口，一个 Java 工程有且仅有一个 main 方法。

main 方法的格式是固定 public static void mian(String[] args){}

public 是四种访问权限修饰中权限最大的一个关键字，它表明 main 方法可以被任何类进行调用，如果 main 不用 public 进行修饰那么程序可能无法启动。

static 是静态的意思，它表明 main 方法是在对象出现之前就已经创建了，在类加载的时候就可以调用 main 方法。

void 表明 main 方法没有返回值，因为 main 方法一旦停止表明程序就停止了。

当一个程序需要一个人口时需要 main 方法。

1.如果定义一个变量，如何使用变量?

对于成员变量来说，成员变量是有初始值的，byte short int long 的初始值为 0 ，float double 初始值为 0.0 ，boolean 数据类型初始值为 false ，char 数据类型初始值为 '0' ,引用数据类型的初始值为 null

对于局部变量来说，局部变量是没有初始值的必须要初始化才能够进行运算。

变量的初始化方式有两种：

静态初始化： int a = 10;

动态初始化： int a; a = 10;

2.基本数据类型有哪些？

8 种基本数据类型，byte short int long float doule char boolean

3.'a'+1，结果是什么？为什么？都做了什么事情呢？

结果是 98

char 数据类型在与 int 数据类型进行运算时会进行自动数据类型转换，'a' 对应的 ascll 码的数值为 97，自动数据类型转换后就是 97，97 + 1 = 98；

4.++在前，在后的区别？

++ 在前，先自加再运算

++ 在后，先运算再自加

5.short s = 3; s = s + 2; s+=2,有什么区别，为什么？

short s = 3; s = s + 2; 会报错，因为存在强制数据类型转换而没有标出。

short s = 3； s+= 2; 不会报错，因为编译器会进行优化，预处理使其并不会报错。

6. &和&&的区别?

相同点: 最后的运算结果都是相同的

不同点: && 的运算效率更高，一旦左边的为 false，右边不会在进行运算直接输出 false。

7.请写1个程序,这个程序要求用户从控制台输入两个数, 然后输出两个数的和.

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习请写1个程序,这个程序要求用户从控制台输入两个数, 然后输出两个数的和.

\*

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class GetSum {

public static void main(String[] args) {

// 创建键盘输入对象

Scanner scanner = new Scanner(System.in);

System.out.println("请输入数字1");

int num1 = scanner.nextInt();

System.out.println("请输入数字1");

int num2 = scanner.nextInt();

int sum = num1 + num2;

System.out.println("数字1与数字2的和为" + sum);

}

}

8.求用户输入矩形的长和宽,求出矩形的面积和周长,并将结果显示在屏幕上.

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习 求用户输入矩形的长和宽,求出矩形的面积和周长,并将结果显示在屏幕上.

\*

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class RectangularCalculation {

public static void main(String[] args) {

// 创建键盘输入对象

Scanner scanner = new Scanner(System.in);

System.out.println("请输入矩形从长");

int length = scanner.nextInt();

System.out.println("输入的长度为" + length);

System.out.println("请输入矩形的宽");

int width = scanner.nextInt();

System.out.println("输入的宽为" + width);

int perimeter = 2 \* length + 2 \* width;

System.out.println("矩形的周长为" + perimeter);

int area = length \* width;

System.out.println("矩形的面积为" + area);

}

}

9.要求用户输入圆形的半径,求出圆形的面积和周长,并将结果显示在屏幕上.

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习 要求用户输入圆形的半径,求出圆形的面积和周长,并将结果显示在屏幕上.

\*

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class CircularCalculation {

public static void main(String[] args) {

// 创建键盘输入对象

Scanner scanner = new Scanner(System.in);

System.out.println("请输入圆的半径");

int radius = scanner.nextInt();

System.out.println("输入的半径为" + radius);

double pi = 3.14;

double perimeter = 2 \* radius \* pi;

System.out.println("圆的周长为" + perimeter);

double area = pi \* Math.pow(radius, 2);

System.out.println("圆的面积为" + area);

}

}

10.判断一个整数是偶数还是奇数

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 联系 判断一个整数是偶数还是奇数

\*

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class JudgingNumbers {

public static void main(String[] args) {

// 创建键盘输入对象

Scanner scanner = new Scanner(System.in);

System.out.println("请输入要判断的整数");

int num = scanner.nextInt();

if (num % 2 == 0) {

System.out.println("输入的" + num + "是偶数");

} else {

System.out.println("输入的" + num + "是奇数");

}

}

}

11.使用程序判断假设今天是星期4，那么问10天后的今天是星期几？

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习 使用程序判断假设今天是星期4，那么问10天后的今天是星期几？

\*

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class JudgingWeek {

public static void main(String[] args) {

// 创建键盘输入函数

Scanner scanner = new Scanner(System.in);

int today = 4;

System.out.println("星期" + (((today + 10) % 7) + 1));

}

}

12.读取文件

package cn.hfming.baiscjava;

import java.io.\*;

/\*\*

\* 练习 读取文件

\*

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class FileRead {

public static void main(String[] args) {

// 创建文件对象

File file = new File("D:\\intelliJIDEAWorkplace\\core-java\\homework\\read.txt");

// 建立数据的输入通道

FileInputStream fileInputStream = null;

try {

fileInputStream = new FileInputStream(file);

// 创建缓存数组

byte[] content = new byte[1024];

// 循环读取数据

while ((fileInputStream.read(content)) != -1) {

// 输出读到的数据

System.out.println(new String(content, 0, 1024));

}

} catch (FileNotFoundException e) {

throw new RuntimeException(e);

} catch (IOException e) {

throw new RuntimeException(e);

} finally {

try {

fileInputStream.close();

} catch (IOException e) {

throw new RuntimeException(e);

}

}

}

}

13.给定3个数a，b和c,输出最大的数。

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习 给定3个数a，b和c,输出最大的数。

\*

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class GetBiggest {

public static void main(String[] args) {

// 创建键盘输入对象

Scanner scanner = new Scanner(System.in);

System.out.println("请依次输入三个数并用“，”号隔开");

String str = scanner.next();

// 将输入的字符串切割开

String[] arr = str.split(",");

int[] arr2 = new int[arr.length];

// 将字符串转发为数字

for (int i = 0; i < arr2.length; i++) {

arr2[i] = Integer.parseInt(arr[i]);

}

// 选出最大的数字

int max = arr2[0];

for (int i = 0; i < arr2.length; i++) {

if (arr2[i] > max) {

max = arr2[i];

}

}

System.out.println("输入的三个数字中最大的为" + max);

}

}

14.某电信公司的市内通话费计算标准如下：三分钟内0.2元，三分钟后每增加一分钟增加0.1元，不足一分钟的按一分钟计算。要求编写程序，给定一个通话时间（单位：秒），计算出应收费金额。

package cn.hfming.baiscjava;

import java.util.Scanner;

import java.util.ServiceConfigurationError;

/\*\*

\* 练习 某电信公司的市内通话费计算标准如下：三分钟内0.2元，三分钟后每增加一分钟增加0.1元，不足一分钟的按一分钟计算。要求编写程序，给定一个通话时间（单位：秒），计算出应收费金额。

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class Amount {

public static void main(String[] args) {

// 创建键盘输入对象

Scanner scanner = new Scanner(System.in);

System.out.println("请输入通话时间");

double time = scanner.nextDouble();

if(time <= 3.0){

System.out.println("收费0.2元");

}else if(time > 3){

System.out.println("收费" + (0.2 + Math.floor(time - 3) \* 0.1));

}

}

}

15.某市的出租车计费标准为：3公里以内10元，3公里以后每加0.5公里加收1元；每等待2.5分钟加收1元；超过15公里的加收原价的50%为空驶费。要求编写程序，对于任意给定的里程数（单位：公里）和等待时间（单位：秒）计算出应付车费，车费直接截去小数位，只取整数。

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习 某市的出租车计费标准为：3公里以内10元，3公里以后每加0.5公里加收1元；每等待2.5分钟加收1元；超过15公里的加收原价的50%为空驶费。要求编写程序，对于任意给定的里程数（单位：公里）和等待时间（单位：秒）计算出应付车费，车费直接截去小数位，只取整数。

\*

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class Amount2 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("请输入里程数");

double mileage = scanner.nextDouble();

System.out.println("请输入等待时间");

double time = scanner.nextDouble();

if (mileage <= 3) {

System.out.println("收费" + Math.floor(time / 2.5 + 10));

} else if (mileage < 15 && mileage > 3) {

System.out.println("收费" + Math.floor(10 + (mileage - 3) / 0.5 + time / 2.5));

} else {

System.out.println("收费" + Math.floor((10 + (mileage - 3) / 0.5 + time / 2.5)) \* 1.5);

}

}

}

16.编写程序，判断给定的某个年份是否是闰年。闰年的判断规则如下：（1）若某个年份能被4整除但不能被100整除，则是闰年。（2）若某个年份能被400整除，则也是闰年。

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习 编写程序，判断给定的某个年份是否是闰年。闰年的判断规则如下：（1）若某个年份能被4整除但不能被100整除，则是闰年。（2）若某个年份能被400整除，则也是闰年。

\*

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class LeapYear {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("请输入年份");

int year = scanner.nextInt();

if ((year % 100 != 0 && year % 4 == 0) || year % 400 == 0) {

System.out.println(year + "是闰年");

} else {

System.out.println(year + "不是闰年");

}

}

}

17.给定一个百分制的分数，输出相应的等级。

90分以上 A级

80~89 B级

70~79 C级

60~69 D级

60分以下 E级

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习 .给定一个百分制的分数，输出相应的等级。

\* 90分以上 A级

\* 80~89 B级

\* 70~79 C级

\* 60~69 D级

\* 60分以下 E级

\*

\* @author hfm

\* @version 1.01 2019-10-14

\*/

public class GetGrade {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("请输入成绩");

double grade = scanner.nextDouble();

if (grade < 60 && grade >= 0) {

System.out.println("成绩为E");

}

if (grade >= 60 && grade < 70) {

System.out.println("成绩为D");

}

if (grade >= 70 && grade < 80) {

System.out.println("成绩为C");

}

if (grade >= 80 && grade < 90) {

System.out.println("成绩为B");

}

if (grade >= 90 && grade <= 100) {

System.out.println("成绩为A");

}

if (grade < 0 || grade > 100) {

System.out.println("输入错误");

}

}

}

18.编写程序求 1+3+5+7+……+99 的和值。

package cn.hfming.baiscjava;

/\*\*

\* 练习 编写程序求 1+3+5+7+……+99 的和值。

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class GetSum2 {

public static void main(String[] args) {

int sum = 0;

for (int i = 1; i <= 99; i++) {

sum += i;

}

System.out.println("1到99的和为" + sum);

}

}

19.编写程序输出1-100之间所有能被7整除的偶数。

package cn.hfming.baiscjava;

/\*\*

\* 练习 编写程序输出1-100之间所有能被7整除的偶数。

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class Get7Divisible {

public static void main(String[] args) {

for (int i = 0; i <= 99 ; i++) {

if (i % 7 == 0){

System.out.println(i + "能够被7整除！");

}

}

}

}

20.求所有满足如下条件的四位数：千位上的数字大于百位数字，百位数字大于十位数字，十位数字大于个位数字，并且千位数字是其他三位数字的和。

package cn.hfming.baiscjava;

/\*\*

\* 练习 求所有满足如下条件的四位数：千位上的数字大于百位数字，百位数字大于十位数字，十位数字大于个位数字，并且千位数字是其他三位数字的和。

\*

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class GetNum3 {

public static void main(String[] args) {

for (int i = 1000; i <= 9999; i++) {

// 千位

int a1 = i / 1000;

// 百位

int b1 = (i / 100) % 10;

// 十位

int c1 = (i / 10) % 100 % 10;

// 个位

int d1 = i % 10;

if (a1 > b1 && b1 > c1 && c1 > d1 && (a1 == b1 + c1 + d1)) {

System.out.print(i + " ");

} else {

continue;

}

}

}

}

21.给定两个任意的正整数,求它们之间的所有完全平方数，完全平方数是指这个数是 某个整数的平方，例如 16，25，36等都是完全平方数。

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习 给定两个任意的正整数,求它们之间的所有完全平方数，完全平方数是指这个数是 某个整数的平方，例如 16，25，36等都是完全平方数。

\*

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class GetNum4 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("请输入两个正整数");

System.out.print("第一个正整数为：");

int a = scanner.nextInt();

System.out.print("第二个正整数数为：");

int b = scanner.nextInt();

//求最小值

int max;

int min;

if (a <= b) {

min = a;

max = b;

} else {

min = b;

max = a;

}

System.out.println(min);

System.out.println(max);

// 所有完全平方数

for (int i = min; i <= max; i++) {

if ((int) Math.pow((int) Math.sqrt(i), 2) == i) {

System.out.println(i);

}

}

}

}

22.编写程序求下列多项式的前50项的和：1-1/3+1/5-1/7+1/9-……

package cn.hfming.baiscjava;

/\*\*

\* 练习 编写程序求下列多项式的前50项的和：1-1/3+1/5-1/7+1/9-……

\*

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class GetSum3 {

public static void main(String[] args) {

double result = 0;

for (int i = 1; i <= 50; i++) {

result += Math.pow(-1, i + 1) / (2 \* i - 1);

System.out.println(result);

}

System.out.println(result);

}

}

23..请编写程序输出九九乘法表。

package cn.hfming.baiscjava;

/\*\*

\* 练习 请编写程序输出九九乘法表

\*

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class MultiplicationTable {

public static void main(String[] args) {

for (int i = 1; i <= 9; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(i + "\*" + j + "=" + (i \* j) + "\t");

}

System.out.println();

}

}

}

24.请编写程序输出九九乘法表。

/\*\*

\* 练习 请编写程序输出九九乘法表。

\*

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class PrimeNumber {

public static void main(String[] args) {

Outer:

for (int i1 = 2; i1 <= 100; i1++) {

Inner:

for (int j1 = 2; j1 <= i1 / 2; j1++) {

if (i1 % j1 == 0) {

continue Outer;

}

}

System.out.println(i1);

}

}

}

25.在屏幕上输出一个n行的金字塔图案，例如，若n=4,则图案如下：

package cn.hfming.baiscjava;

/\*\*

\* 练习 在屏幕上输出一个n行的金字塔图案，例如，若n=4,则图案如下

\* \*

\* \*\*\*

\* \*\*\*\*\*

\* \*\*\*\*\*\*\*

\*/

public class PrintImage {

public static void main(String[] args) {

String str1 = " ";

String str2 = "\*";

for (int i1 = 0; i1 < 10; i1++) {

for (int j1 = 0; j1 < 10 - i1; j1++) {

System.out.print(str1);

}

for (int k1 = 0; k1 < i1 \* 2 - 1; k1++) {

System.out.print(str2);

}

System.out.println();

}

}

}

26.求斐波那契数列前n项的和值,斐波那契数列如下： 1，1，2，3，5，8，13，21……前两位数是1，从第三位开始每位数都是前两位数之和。

package cn.hfming.baiscjava;

/\*\*

\* 练习 求斐波那契数列前n项的和值,斐波那契数列如下： 1，1，2，3，5，8，13，21……前两位数是1，从第三位开始每位数都是前两位数之和。

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class FbnqList {

public static void main(String[] args) {

//定义第一个加数a，初始值为1；定义第二个加数b，初始值为1；定义两个加数之和为c，初始值为0

int a = 1;

int b = 1;

int c = 0;

//首先在控制台打印出数列中第一个数和第二个数的值

System.out.print(a + "\t" + b + "\t");

//建立一个for循环，用于循环输出数列中第三位至第十位的数字

for (int i = 3; i <= 10; i++) {

//第三个数即为c，a+b等于c的值

c = a + b;

//将第一个加数a赋值为数列中的第二个数b的值

a = b;

//将第二个加数b赋值为数列中的第三个数c的值

b = c;

//在第二次循环打印时，将打印数列中的第四个数为：b + c = b + (a + b)

System.out.print(c + "\t");

}

}

}

27.给定一个整数，把它的各位数字倒排过来形成一个新的整数例如：给定12345 输出54321 ；给定 9870 输出789

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习 给定一个整数，把它的各位数字倒排过来形成一个新的整数例如：给定12345 输出54321 ；给定 9870 输出789

\*

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class GetNum5 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("请输入一个整数");

int input = scanner.nextInt();

int m = 0;

while (input > 0) {

m \*= 10;

m += input % 10;

input /= 10;

}

System.out.println(m);

}

}

1.如何理解函数（方法）？

函数是对实现某一功能代码的封装，函数的实现有利于功能代码的重复使用。

2.怎么定义一个函数？

public(访问权限修饰符) static void/返回值类型 函数名(型参列表){

函数体；

return 返回值；//用于停止函数

}

3.函数重载什么时候使用？

当前的代码不能满足需求时需要对函数进行重载。

4.编写一个方法接受一个用户名与密码，如果输入的用户名不是admin密码不是888，那么继续提示用户“用户名或者密码错误”，否则提示用户输入正确。

package cn.hfming.baiscjava;

import java.util.Scanner;

/\*\*

\* 练习 编写一个方法接受一个用户名与密码，如果输入的用户名不是admin密码不是888，那么继续提示用户“用户名或者密码错误”，否则提示用户输入正确。

\* @author hfmn

\* @version 1.01 2019-10-15

\*/

public class Admin {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("请输入用户名");

String name = scanner.next();

System.out.println("请输入密码");

String scrit = scanner.next();

if (("admin").equals(name) && ("888").equals(scrit)){

System.out.println("登陆成功");

}else{

System.out.println("用户名或密码错误");

}

}

}

5.数组是什么和数组的特点是什么？

数组是存储一组相同数据的容器。

数组一旦第定义它的长度不会发生改变的。

数组中的元素都有下角标值，可以通过下角标值进行访问。

数组中元素与元素之间的内存地址是连续的。

6.对于一个int数组，把其中的所有元素都向后移动一个位置定义一个长度为10的int数组,统计数组的最大值、最小值、奇数和偶数的个数

比如： int[] arr = { 1, 3, 5, 7, 9 } --> { 9, 1, 3, 5, 7 }

package cn.hfming.baiscjava;

/\*\*

\* 练习 对于一个int数组，把其中的所有元素都向后移动一个位置定义一个长度为10的int数组,统计数组的最大值、最小值、奇数和偶数的个数

\* 比如： int[] arr = { 1, 3, 5, 7, 9 } --> { 9, 1, 3, 5, 7 }

\*

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class GetArray {

public static void main(String[] args) {

int[] arr = new int[]{1,23,4,34356,5675};

int max = arr[0];

int min = arr[0];

int oddNum = 0;

int evenNum = 0;

for (int i = 0; i < arr.length; i++) {

if (max < arr[i]) {

max = arr[i];

}

}

System.out.println("最大值为：" + max);

for (int i = 0; i < arr.length; i++) {

if (min > arr[i]) {

min = arr[i];

}

}

System.out.println("最小值为：" + min);

for (int i = 0; i < arr.length; i++) {

if (arr[i] % 2 == 1) {

oddNum++;

}

}

System.out.println("奇数个数为:" + oddNum);

for (int i = 0; i < arr.length; i++) {

if (arr[i] % 2 == 0) {

evenNum++;

}

}

System.out.println("偶数个数为:" + evenNum);

}

}

7.请编写1个程序 求出1个整型数组中的最大值和次大值。

package cn.hfming.baiscjava;

/\*\*

\* 练习 请编写1个程序 求出1个整型数组中的最大值和次大值。

\*

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class GetMax {

public static void main(String[] args) {

int[] arr = new int[]{1,4345,56,56,43,454};

int max = 0;

int nex = 0;

for (int i = 0; i < arr.length; i++) {

if (max < arr[i]) {

max = arr[i];

}

}

System.out.println("最大值为：" + max);

for (int i = 0; i < arr.length; i++) {

if (nex < arr[i] && max > arr[i]) {

nex = arr[i];

}

}

System.out.println("次大值为：" + nex);

}

}

8.用户输入班级的人数,然后依次输入每个同学的成绩. 输入完毕之后,如果及格率没有达到60%， 就为每1个没有及格的成绩加2分,直到及格率达到60%为止。

package cn.hfming.baiscjava;

import java.util.Arrays;

import java.util.Scanner;

/\*\*

\* 练习 用户输入班级的人数,然后依次输入每个同学的成绩. 输入完毕之后,如果及格率没有达到60%， 就为每1个没有及格的成绩加2分,直到及格率达到60%为止。

\*

\* @author hfm

\* @version 1.01 2019.10.15

\*/

public class InproveGread {

public static void main(String[] args) {

System.out.println("请输入班级人数");

Scanner scanner = new Scanner(System.in);

int num = scanner.nextInt();

System.out.println("输入的班级人数为：" + num);

int[] arr = new int[num];

System.out.println("请依次输入班级同学的成绩");

for (int i = 0; i < arr.length; i++) {

arr[i] = scanner.nextInt();

}

System.out.println("正在计算平均分！");

int sum;

int average;

OUT:

for (; ; ) {

sum = 0;

average = 0;

for (int i = 0; i < arr.length; i++) {

sum += arr[i];

}

//System.out.println(sum);

average = sum / arr.length;

System.out.println(average);

if (average >= 60) {

System.out.println(Arrays.toString(arr));

break OUT;

} else {

for (int i = 0; i < arr.length; i++) {

if (arr[i] < 60) {

arr[i] += 2;

} else {

continue;

}

}

}

}

}

}

9.现在有如下的一个数组：int oldArr[]={1,3,4,5,0,0,6,6,0,5,4,7,6,7,0,5} ，要求清除数组中为0的元素，然后存储非零的数据存储到一个新数组中。

package cn.hfming.baiscjava;

import cn.hfming.array.ArrayRemoveZero;

import java.util.Arrays;

import java.util.Scanner;

/\*\*

\* 练习 现在有如下的一个数组：int oldArr[]={1,3,4,5,0,0,6,6,0,5,4,7,6,7,0,5} ，要求清除数组中为0的元素，然后存储非零的数据存储到一个新数组中。

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class RemoveZero {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("请输入数组，数值之间用“，”号隔开");

String arrString = scanner.next();

// 切割字符串

String[] arr = arrString.split(",");

int[] finalArr = new int[arr.length];

for (int i = 0; i < arr.length; i++) {

finalArr[i] = Integer.parseInt(arr[i]);

}

System.out.println(Arrays.toString(ArrayRemoveZero.arrayRemoveZreo(finalArr)));

}

}

package cn.hfming.array;

/\*\*

\* 数组工具类

\* 功能：除去数组中为0的元素，返回新的数组

\*

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class ArrayRemoveZero {

public static int[] arrayRemoveZreo(int[] arr) {

int zeroNumber = 0;

for (int i = 0; i < arr.length; i++) {

if (arr[i] == 0) {

zeroNumber++;

}

}

int index = 0;

int[] newArray = new int[arr.length - zeroNumber];

for (int i = 0; i < arr.length; i++) {

if (arr[i] != 0) {

newArray[index] = arr[i];

index++;

}

}

return newArray;

}

}

10.清除重复元素。 int[] arr = {1,4，9,4,1,1,7}；

package cn.hfming.baiscjava;

import java.util.Arrays;

import java.util.Scanner;

import cn.hfming.array.\*;

/\*\*

\* 练习 清除重复元素。 int[] arr = {1,4，9,4,1,1,7}；

\*

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class CleanRepeat {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("请输入数组，数值之间用“，”号隔开");

String arrString = scanner.next();

// 切割字符串

String[] arr = arrString.split(",");

int[] finalArr = new int[arr.length];

for (int i = 0; i < arr.length; i++) {

finalArr[i] = Integer.parseInt(arr[i]);

}

System.out.println(Arrays.toString(ArrayCleanRepeat.arrayCleanRepeat(finalArr)));

}

}

package cn.hfming.array;

/\*\*

\* 数组工具类

\* 功能：去除重复元素，返回新的数组

\* @author hfm

\* @version 1.01 2019-10-15

\*/

public class ArrayCleanRepeat {

public static int[] arrayCleanRepeat(int[] arr) {

// 统计重复元素的个数

int repeatNumber = 0;

for (int i = 0; i < arr.length; i++) {

for (int j = i + 1; j < arr.length; j++) {

if (arr[i] == arr[j]) {

repeatNumber++;

break;

}

}

}

//创建新的数组

int[] arr2 = new int[arr.length - repeatNumber];

int index = 0;

OUT:

for (int i = 0; i < arr.length; i++) {

for (int j = 0; j < arr2.length; j++) {

if (arr2[j] == arr[i]) {

continue OUT;

}

}

arr2[index] = arr[i];

index++;

}

return arr2;

}

}

1. 一台完整的计算机包括了（ ）

A. 计算机及外部设备 B. 主机、显示器 C. 硬件系统与系统软件 D. windows操作系统

C 计算机: 硬件 + 软件 = 计算机

2. 计算机存储数据的最小单位是二进制的( )。

A.位（比特） B.字节 C.字长 D.千字节KB

B

位：最小的存储单位称为位（bit）。只能容纳两个值（0或1）之一，不能在一个位中存储更多的信息。位是计算机存储的基本单位。位表示的是二进制位，一般称为比特，是计算机存储的最小单位，类似于（两，公斤，克）的概念，比如：11010100是一个8位二进制数。一个二进制位只可以表示0和1两种状态；两个二进制位可以表示00、01、10、11四种（22）状态；三位二进制数可表示八种状态。

字节：字节(byte)是常用的计算机存储单位。字节的标准定义：一个字节均为8位。由于上述所讲每个位或者是0或者是1，所以一个8位的字节包含256种可能的0，1组合。字节来自英文Byte，音译为“拜特”，习惯上用大写的“B”表示。字节是计算机中数据处理的基本单位。计算机中以字节为单位存储和解释信息，规定一个字节由八个二进制位构成，即1个字节等于8个比特（1Byte=8bit）。八位二进制数最小为00000000，最大为11111111；通常1个字节可以存入一个ASCII码，2个字节可以存放一个汉字国标码

字符：对于一种给定的计算机设计，字(word)是自然的存储单位。一个英文字符和英文标点占用一个字节，一个中文字符和中文标点占用2个字节。计算机进行数据处理时，一次存取、加工和传送的数据长度称为字（word）。一个字通常由一个或多个（一般是字节的整数位）字节构成。例如286微机的字由2个字节组成，它的字长为16；486微机的字由4个字节组成，它的字长为32位机。计算机的字长决定了其CPU一次操作处理实际位数的多少，由此可见计算机的字长越大，其性能越优越。

字长：计算机的每个字所包含的位数称为字长。根据计算机的不同，字长有固定的和可变的两种。固定字长，即字长度不论什么情况都是固定不变的；可变字长，则在一定范围内，其长度是可变的。计算的字长是指它一次可处理的二进创数字的数目。计算机处理数据的速率，自然和它一次能加工的位数以及进行运算的快慢有关。如果一台计算机的字长是另一台计算机的两倍，即使两台计算机的速度相同，在相同的时间内，前者能做的工作是后者的两倍。一般地，大型计算机的字长为32―64位，小型计算机为12―32位，而微型计算机为4一16位。字长是衡量计算机性能的一个重要因素。

bps：bits per second 的简称。一般数据机及网络通讯的传输速率都是以「bps」为单位。如56Kbps、100.0Mbps 等等。

Bps：Byte per second 的简称。而电脑一般都以Bps 显示速度，如1Mbps 大约等同 128 KBps。

bit：电脑记忆体中最小的单位，在二进位电脑系统中，每一bit 可以代表0 或 1 的数位讯号。

Byte：一个Byte由8 bits 所组成，可代表一个字元(A~Z)、数字(0~9)、或符号(,.?!%&+-\*/)，是记忆体储存资料的基本单位。

至於每个中文字则须要两Bytes。当记忆体容量过大时，位元组这个单位就不够用，因此就有千位元组的单位KB出现，以下乃个记忆体计算单位之间的相关性：

1 Byte = 8 Bits

1 KB = 1024 Bytes

1 MB = 1024 KB

1 GB = 1024 MB

usb2.0标准接口传输速率。许多人都将“480mbps”误解为480兆/秒。其实，这是错误的，事实上“480mbps”应为“480兆比特/秒”或“480兆位/秒”，它等于“60兆字节/秒”，

3.以下不是操作系统的是（ C ）

A. Windows B. Linux C. Java D. Android

4. 操作系统中的文件名中不允许使用( C )

A. 数字（0～9） B. 美元符号（$） C. 反斜线（\） D. 小写英文字母（a～z）

DOS操作系统规定文件名由文件主名和扩展名组成，文件主名由1~8个字符组成，扩展名由1~3个字符组成，主名和扩展名之间由一个小圆点隔开，一般称为8.3规则。其格式如下：

例如： G9401.DBF，这里G9401是主名，DBF是扩展名。文件主名和扩展名可以使用的字符是：

（1）英文字母：A—z（大小写等价）

（2）数 字：0—9

（3）汉 字

（4）特殊符号：$#&@（）-[]^～等。

（5）各种控制字符和下列字符不能用在文件名中：< > / \ | : " \* ?

因为这些字符已做它用。

windows突破了DOS对文件命名规则的限制，允许使用长文件名，其主要命名规则如下：

（1）文件名最长可以使用255个字符。

（2）可以使用扩展名，扩展名用来表示文件类型，也可以使用多间隔符的扩展名。如win.ini.txt是一个合法的文件名，但其文件类型由最后一个扩展名决定。

（3）文件名中允许使用空格，但不允许使用下列字符（英文输入法状态）：< > / \ | : " \* ?

（4）windows系统对文件名中字母的大小写在显示时有不同，但在使用时不区分大小写。

文件名是文件存在的标识,操作系统根据文件名来对其进行控制和管理.不同的操作系统对文件命名的规则略有不同,即文件名的格式和长度因系统而异.例如:UNIX/LINUX系统区分文件名的大小写,而有些系统则不区分如:MS-DOS和WINDOWS.

对于文件扩展名是用来区分文件的属性的.在有的系统中并不为系统所感知如:UNIX/LINUX系统中,hello.c文件与helloc文件对于系统来说没有什么区别,加上了扩展名.c与不加都一样,系统都不知道那是个什么文件.只是用户加上.c后,方便用户明白那是c语言源码文件.但是在MS-DOS与WINDOWS系统中,文件扩展名是为系统感知的.在WINDOWS中,对于不同的文件扩展名有其相关联的程序进行打开,这些扩展名除了如txt等是原来就有的以外其余一般是安装软件时被注册到系统中的.在注册表中的HKEY-CLASSES-ROOT根键下可以看到其相关信息.若将文件扩展名改为系统不认知的扩展名,则在打开时提示选择其打开程序.如将hello.txt改为hello.123,同时可以看到图标也发生了变化,变为不可知.但是对于如.doc格式的WORD文件,由于是MICROSOFT开发的,保持了很好的兼容性,系统在感知时并不只检查扩展名,还检查其文件内部格式标识,所以即使将.doc改为其它字符,在装了WORD的系统中,系统仍能够使其关联到WORD进行打开.

5. Java是什么( A )

A.计算机语言 B. 操作系统 C. 计算机单位 D. 操作系统的工具

6. 运行java程序的时候是至少要安装什么（ C ）

A. JDK B. Office办公软件 C. JRE D. C语言的程序

JDK是整个Java的核心，包括了Java运行环境、Java工具和Java基础类库。

JDK: java开发工具集

JDK = java开发工具 + jre

JRE: java运行环境。

JRE = java虚拟机 + 核心类库(辅助java虚拟机运行的文件)

JRE(Java Runtime Environment)Java运行环境。包括Java虚拟机(JVM Java Virtual Machine)和Java程序所需的核心类库等，如果想要运行一个开发好的Java程序，计算机中只需要安装JRE即可。

7. Java程序的执行过程中用到一套JDK工具，其中javac.exe指（ A ）

A. Java编译器 B. Java字节码解释器 C. Java文档生成器 D. Java类分解器

javac.exe: 启动java编译器对java源文件进行编译。

java.exe: 启动java虚拟机解释并执行指定的class文件。

8. Java源文件和编译后的文件扩展名分别为（ B ）

A .class和.java B .java和.class C .class和.class D.java和.java

9. 阅读以下程序判断会输出几次”hello world”( B )

int num = 0;

while(num < 5){

System.out.println("Hello World!");

}

A. 1次 B. 5次 C. 0次 D. 无数次

10. 15二进制数是多少（ C ）

A. 00000111 B. 00000011 C. 00001111 D. 00011101

15/2 ... 1

7/2 ... 1

3/2 ... 1

1/2 ... 1

1111

11．请你仔细观察数列的排列规律，选出括号中应写的数字：（ C ）

18，-27，36，( )，54

A.44 B．45 C．-45 D．-44

12.5000秒是几小时几分钟几秒？（需要写出计算过程）

5000/60/60=1小时

5000-3600=1400s/60=23分钟

1400-1380=20秒

13.根据下面的条件算出这个人生于何年，死于何年？（需要写出计算过程）

1.他生于19世纪，也同样死于19世纪；

2.他诞生的年份和逝世的年份都是由4个相同的数字组成，但排列的位置不同；

3.他诞生的那一年，4个数字之和是14；

4.他逝世那一年的数字的十位数是个位数的4倍。

18 小于100

18xy 18yx

1+8+x+y=14

Y=4x

X=1,y=4

生于1814，死于1841

14.计算兔子的繁殖数量（需要写出计算过程）

兔子的繁殖能力很强，一对兔子每个月可以生一对小兔子，而一对小兔子生下后第二个月也开始生小兔子。如果从刚生的一对兔子算起，那么一年可以繁殖出多少对兔子。(提示：斐波那契数列)

public static void pratice1() {

int result = 1;

// 记录上一个result

int x = 0;

// 记录上上个result

int y = 0;

// 保存上一个x的值

int temp = 0;

for (int i = 1; i <= 12; i++) {

y = temp;

if (i == 1 || i == 2) {

result = 1;

} else {

result = x + y;

}

temp = x;

// 将result赋值给x,让x记录上一个result

x = result;

}

System.out.println(result);

}

//定义第一个加数a，初始值为1；定义第二个加数b，初始值为1；定义两个加数之和为c，初始值为0

int a = 1;

int b = 1;

int c = 0;

//首先在控制台打印出数列中第一个数和第二个数的值

System.out.print(a + "\t" + b + "\t");

//建立一个for循环，用于循环输出数列中第三位至第十位的数字

for (int i = 3; i <= 10; i++) {

//第三个数即为c，a+b等于c的值

c = a + b;

//将第一个加数a赋值为数列中的第二个数b的值

a = b;

//将第二个加数b赋值为数列中的第三个数c的值

b = c;

//在第二次循环打印时，将打印数列中的第四个数为：b + c = b + (a + b)

System.out.print(c + "\t");

}

15.在控制台输出“Hello world.” 请写出相应的Java代码（10分）

class HelloWorld {

public static void main(String[] args) {

System.out.println("HelloWorld");

}

}

16.下列哪个声明是错误的？（b）

A. int i=10;

B. float f=1.1; //值需要加f

C. double d=34.4;

D. byte b=127;

17.8种基本数据类型byte 、 short 、 int 、 long 、 float 、 double 、 char 、 Boolean

注意: 一个小数如果没有加上任何的标识的时候，默认都double类型。如果是float类型的数据那么该数据必须要以f结尾。不区分大小写。

注意: 一个整数没有加上任何的标识的时候，那么该整数默认是int类型 的数据。

如果一个数据需要表示成一个长整型，那么该数据要以L结尾。L是不区分大小写的，但是建议使用大写。

18.下面哪个不是java中的关键字？（c）

A. public

B. true

C. main //main是标识符

D. class

19.下面哪个语句不会产生编译错误？（C）

A. float a =2.0; //值需要加f

B. char c =”a”; //字符用''

C. byte b =25; //先判断25是否超出-128~127，超出才报错

D. boolean d=0; //只有两个值turn falst

byte(字节) 8bit

short(短整型) 16bit

int(整型) 32bit

long (长整型) 64bit

小数：

float(单精度浮点型) 32bit

double(双精度浮点型) 64bit

布尔：

Boolean(不利恩) 一个字节或者是 4 个字节。 只有两个值true ，false。

字符：

char 2 个字节

20.下面程序哪个语句是正确的（C）

A. byte a=0,b=3; byte c =a+b; //a+b是int类型

B. short s =23; s=s+12; //s+12是Int类型

C. short s=23; s+=12; //s+=12本身就是short类型

D. float f = 23+23.23; //23+23.23是Int类型

考察表达式的数据类型自动提升

所有的 byte 型、short 型和 char 的值将被提升到 int 型。

如果一个操作数是 long 型，计算结果就是 long 型。

如果一个操作数是 float 型，计算结果就是 float 型。

如果一个操作数是 double 型，计算结果就是 double 型。

对于 short s1 = 1; s1 = s1 + 1; 由于 s1+1 运算时会自动提升表达式的类型，所以结果是 int 型，再赋值给short 类型 s1 时，编译器将报告需要强制转换类型的错误。对于 short s1 = 1; s1 += 1;由于 += 是 java 语言规定的运算符，java 编译器会对它进行特殊处理，因此可以正确编译。

21.byte类型的大小是（A）

A. -128 ~ 127

B. -228 ~128

C. -255 ~ 256

D. -255 ~ 255

考察基本数据类型使用的数据范围

整数：

byte(字节) 8bit

short(短整型) 16bit

int(整型) 32bit

long (长整型) 64bit

小数：

float(单精度浮点型) 32bit

double(双精度浮点型) 64bit

布尔：

Boolean(不利恩) 一个字节或者是 4 个字节。 只有两个值true ，false。

字符：

char 2 个字节

22.下面程序执行的结果是？（B）

public class Test{

public static void main(String[] args){

System.out.println(“”+’a’+1);

}

}

A. 98

B. a1

C. 971

D. 197

考察表达式的数据类型自动提升

所有数值运算符都可以用在 char 型数据上，如果另一个操作数是一个数字或者字符，那么 char 会自动提升为 int 型，如果另一个操作数是字符串，那么字符就会和字符串相连。

23.下面程序执行的结果是？（A）

int i = 100;

while(true){

if ( i++ > 100 ) //先判断，后++

break;

System.out.println(i);

}

A. 100

B. 101

C. 102

D. 103

考察while循环的使用 0到100共101个

24.while的结构

while( 条件表达式 ){

执行语句；

}

25.while循环语句要注意的细节

while 循环语句的循环次数一般是通过变量去控制的。

while 循环语句后面如果加上了分号，那么会影响循环的效果。

如果循环语句只有一个语句的时候，可以省略大括号，但是不建议省略。

26.下面程序执行的结果是？（C）

int a=2;

switch(a){

case 1:a+=1;break;

case 2:a+=2;

case 3:a+=3;break;

case 4:a+=4;break;

default:a=0;

}

System.out.println(a);

A. 5

B. 6

C. 7 //a=2+2+3

D. 8

考察switch选择判断语句的使用

1.switch 语句选择的类型只有四种：byte，short，int ， char。JDK1.7添加了String数据类型

2.case 之间与 default 没有顺序。先判断所有的 case，没有匹配的 case 执行 default。

3.switch 语句停止的条件是遇到了 break 关键字或者结束 switch 语句的大括号。

4.如果匹配的 case 或者 default 没有对应的 break，那么程序会继续向下执行，运行可以执行的语句，直到遇到 break 或者 switch 结尾结束。

5.switch case 中的值必须要与 switch 表达式的值具有相同的数据类型。而且 case 后跟的值必须是常量，不能跟变量。case就像选择题的答案之一。 break就是如果该答案正确那么就可以跳出switch了,意思就是说 已经找出了正确的答案了。那么这道题也就做完了。如果 case没有匹配接着进行下一个 case 匹配,直到匹配为止。 最后如果都没有匹配上,那么 switch 给提供了一个默认的答案,就是 default。

注意:case后跟的是冒号:每个case中的执行语句一定要加 break；

27.下面程序的运行结果是（d）

int a=3, b=1;

if(a==b)

System.out.println("a="+a);

A. a=1

B. a=3

C. 编译错误

D. 正常运行但没有输出

考察if条件语句的运用

忘记必要的括号：如果代码块中只有一条语句的时候，可以省略花括号，但是当花括号将多条语句扩在一起时，花括号就不能在省略。

double radius = 4;

double area;

if (radius >= 0)

area = radius \* radius \* 3.14;

System.out.println("The area " + " is " + area);

double radius = 4;

double area;

if (radius >= 0) {

area = radius \* radius \* 3.14;

System.out.println("The area " + " is " + area);

}

虽然代码一样多，但是第一个会编译报错（area 没有出初始化），第二个正常运行。就是因为少了花括号。所以一定要仔细

if 语句后出现分号

double radius = 0;

double area;

if (radius > 0); {

area = radius \* radius \* 3.14;

System.out.println("The area " + " is " + area);

}

注意：这是一个逻辑错误，编译和运行都不会报错，只是不会出现想要的结果。

相当于判断符合条件后，执行一个空语句。

double radius = 0;

double area;

if (radius > 0){}{

area = radius \* radius \* 3.14;

System.out.println("The area " + " is " + area);

}

28.下面程序的运行后,a,b,c的值正确的是：（b）

int a=1,b=2;

int c=(a+b>3?a++:++b);

A. a=2,b=3

B. a=1,b=3

C. a=1,b=2

D. c=2

考察三目运算符的使用

格式

(条件表达式)?表达式 1：表达式 2；

如果条件为 true，运算后的结果是表达式 1；

如果条件为 false，运算后的结果是表达式 2；

29.下面程序的运行结果（b）

public class Demo{

public static int fun(int c){

return c += 2;

}

public static void main(String[] args) {

int temp=fun(2);

System.out.println(temp);

}

}

A. 2

B. 4

C. 6

D. 8

运算符 运算 范例 结果

= 赋值 a=3,b=2 a=3,b=2

+= 加等于 a=3,b=3;a+=b; a=5,b=2;

-= 减等于 a=3,b=2,a-=b; a=1,b=2;

\*= 乘等于 a=3,b=2,a\*=b; a=6,b=2

/= 除等于 a=3,b=2,a/=b; a=1,b=2;

%= 模等于 a=3,b=2,a%=b; a=1,b=2

30.下面程序的运行结果，哪个是正确的（b）

int b=1;

while(++b<3)

System.out.println("LOOP");

A. 程序将会进入死循环导致无输出

B. 输出一次LOOP

C. 会输出多次LOOP

D. 程序中含有编译错误

考察while语句的使用及++自增运算符的使用

6．++(自增): 自增就是操作数+1

前自增: ++位于操作数的前面。 先自增，后使用。

后自增: ++位于操作数的后面。 先使用，后自增。

1先自增到2，输出一次，2先自增到3，不输出，总共输出一次

while 循环语句的循环次数一般是通过变量去控制的。

while 循环语句后面如果加上了分号，那么会影响循环的效果。

如果循环语句只有一个语句的时候，可以省略大括号，但是不建议省略。

31.下面数组定义错误的是（D）

A. int [] arr ={23,45,65,78,89};

B. int [] arr=new int[10] ;

C. int [] arr=new int[4]{3,4,5,6};

D. int [] arr={‘a’, 23 , 45 , 6}； //不同类型，不可以放在同一个数组

考察数组的定义

数组的定义格式

格式一：

数据类型[] 变量名 = new 数据类型[容量大小];

int [] arr = new int[8];

int: 表示该容器只能存储 int 类型的数据。

[] : 表示是一个数组类型。容器使用数组，那么如何来标识数组？.那么用一个特殊的符号[]中括号来表示。

arr： 变量名。

右边： new int[125]; 创建了一个 int 类型的数组对象，该对象的容量是 125 个数据。

new ： new 是用于创建一个对象的关键字。

int : 表示该数组对象只能存储 int 类型的数据

[]： 表示是一个数组类型。

8: 表示该数组容器的容量大小。

格式二：

元素类型[] 数组名 = new 元素类型[]{元素，元素，……};

int[] arr = new int[]{3,5,1,7};

int[] arr = {3,5,1,7};

32.下面程序执行的结果是？（D）

int x =1,y=1;

if(x++==2 & ++y==2){

x=7;

}

System.out.println("x="+x+" , y="+y);

A. x=1 y=2

B. x=7 y=1

C. x=7 y=2

D. x=2 y=2

考察自增运算符

++(自增): 自增就是操作数+1

前自增: ++位于操作数的前面。 先自增，后使用。

后自增: ++位于操作数的后面。 先使用，后自增。

--（自减）:自减就是操作数-1

前自减：--位于操作数 的前面，先自减，后使用。

后自减: --位于操作数 的后面，先使用，后自减

如果运算符在变量的前面，则该变量自增 1 或者自减 1，然后返回的是变量的新值，如果运算符在变量的后面，则变量也会自增或者自减 1，但是返回的是变量原来的值。

++在前就是先运算，再取值，++在后就是先取值，再运算。

33.下面不属于基本数据类型的是（D）

A. int

B. double

C. long

D. int[] //数组是引用数据类型

考察基本数据类型

8种基本数据类型byte 、 short 、 int 、 long 、 float 、 double 、 char 、 Boolean

34.给出下列代码:

boolean a = false;

boolean b = true;

boolean c= ( a&&b)&&(!b);

int result = (c = = false)?1:2;

执行完后,c 与 result 的值是( A).

A. false和1

B. true和2

C. true和1

D. false和2

考察操作符的使用

“&”和“&&”的区别：单与时，左边无论真假，右边都进行运算；双与时，如果左边为真，右边参与运算，如果左边为假，那么右边不参与运算。

格式

(条件表达式)?表达式 1：表达式 2；

如果条件为 true，运算后的结果是表达式 1；

如果条件为 false，运算后的结果是表达式 2；

35.阅读下列代码:

public class Test{

public static void main ( String []args ){

　　 int f=12;

　　 int i = 3;

　　 System.out.println( f / i );

　　}

}

}

程序运行结果为( C).

A. 3

B. 3.0

C. 4

D. 4.0

36.下面程序执行的结果是？（ C）

boolean b=true;

if(b=false){

System.out.println("a");

}else if(b){

System.out.println(b);

}else if(!b){

System.out.println("c");

}else

System.out.println("d");

A. a

B. true

C. c

D. d

考察if条件语句的使用与赋值语句的使用

运算符 运算 范例 结果

= 赋值 a=3,b=2 a=3,b=2

+= 加等于 a=3,b=3;a+=b; a=5,b=2;

-= 减等于 a=3,b=2,a-=b; a=1,b=2;

\*= 乘等于 a=3,b=2,a\*=b; a=6,b=2

/= 除等于 a=3,b=2,a/=b; a=1,b=2;

%= 模等于 a=3,b=2,a%=b; a=1,b=2

37.下面程序执行的结果是？（D）

int x=2,y=3;

switch(x){

default:y++;

case 3:y++;

case 4:y++;

}

System.out.println("y="+y);

A. 3

B. 4

C. 5

D. 6

考察switch选择判断语句的使用

1.switch 语句选择的类型只有四种：byte，short，int ， char。JDK1.7添加了String数据类型

2.case 之间与 default 没有顺序。先判断所有的 case，没有匹配的 case 执行 default。

3.switch 语句停止的条件是遇到了 break 关键字或者结束 switch 语句的大括号。

4.如果匹配的 case 或者 default 没有对应的 break，那么程序会继续向下执行，运行可以执行的语句，直到遇到 break 或者 switch 结尾结束。

5.switch case 中的值必须要与 switch 表达式的值具有相同的数据类型。而且 case 后跟的值必须是常量，不能跟变量。case就像选择题的答案之一。 break就是如果该答案正确那么就可以跳出switch了,意思就是说 已经找出了正确的答案了。那么这道题也就做完了。如果 case没有匹配接着进行下一个 case 匹配,直到匹配为止。 最后如果都没有匹配上,那么 switch 给提供了一个默认的答案,就是 default。

注意:case后跟的是冒号:每个case中的执行语句一定要加 break；

38.下面程序执行的结果

for(int i=1;i<=10;i++){

if (i<5)

continue;

System.out.println(“java基础班”);

}

在屏幕上打印几次java基础班?(B )

A. 5

B. 6

C. 7

D. 8

考察循环语句中的continue关键字的作用

continue 的适用范围是循环语句中。

continue 关键字的作用： 跳过本次的循环体语句，继续下一次循环。

continue 关键字要注意的细节：

continue 关键字也可以配合标签使用。

continue 关键字后面也是不能直接跟语句的。

continue 关键字：语句将控制权传递给它所在的封闭迭代语句的下一次迭代。(跳出本循环，执行下一次循环)。

39.阅读下面代码段：

public class Demo{

　　public static void main (String[] args){

int[] arr = new int [10];

　　 System.out.println(arr[1]);

　　}

}

执行结果正确的说法是( C )

A. 编译时将产生错误

B. 编译时正确，运行时将产生错误

C. 输出0

D. 输出空

考察数组元素的初始值

40.和下面代码能完成相同的选项是（B）

int i=1;

int sum=0;

while(i<=100){

if(i%2==0){

sum=sum+i;

}

i++;

}

A. for (int x =1; x<=100;x++){ sum=sum+x;}

B. for (int x =0; x<=100;x+=2){ sum=sum+x;}

C. for (int x =1; x<=100;x+=2){ sum=sum+x;}

D.上述全对

考察取模运算

5.%（取模，取余数） 取模运算的结果正负是取决于被除数的正负。

41.阅读下面代码段：

public class Test{

　　public static void main (String[ ] args){

　　 char ch=’c’;

　　 switch (ch){

　　 case ‘a’:System.out.print(“a”);break;

　　 case ’b’: System.out.print(“ab”);

　　 case ’c’:System.out.print(“c”);

　　 default:System.out.print(“d”);

　　 }

　　}

}

输出的结果为(D )

A. a

B. b

C. c

D. cd

考察switch选择判断语句的使用

1.switch 语句选择的类型只有四种：byte，short，int ， char。JDK1.7添加了String数据类型

2.case 之间与 default 没有顺序。先判断所有的 case，没有匹配的 case 执行 default。

3.switch 语句停止的条件是遇到了 break 关键字或者结束 switch 语句的大括号。

4.如果匹配的 case 或者 default 没有对应的 break，那么程序会继续向下执行，运行可以执行的语句，直到遇到 break 或者 switch 结尾结束。

5.switch case 中的值必须要与 switch 表达式的值具有相同的数据类型。而且 case 后跟的值必须是常量，不能跟变量。case就像选择题的答案之一。 break就是如果该答案正确那么就可以跳出switch了,意思就是说 已经找出了正确的答案了。那么这道题也就做完了。如果 case没有匹配接着进行下一个 case 匹配,直到匹配为止。 最后如果都没有匹配上,那么 switch 给提供了一个默认的答案,就是 default。

注意:case后跟的是冒号:每个case中的执行语句一定要加 break；

42.以下代码输出是（D ）

int i=0;

int sum=0;

while(i<=10){

i++;

if( i%2 != 0 )

continue;

sum+=i;//2+4+6+8+10

}

System.out.println(sum);

A. 55

B. 45

C. 35

D. 30

考察取模运算

5.%（取模，取余数） 取模运算的结果正负是取决于被除数的正负。

43.下列语句中执行跳转功能的语句是（C ）

A. for语句

B. while语句

C. continue语句

D. switch语句

考察循环语句中的continue关键字的作用

continue 的适用范围是循环语句中。

continue 关键字的作用： 跳过本次的循环体语句，继续下一次循环。

continue 关键字要注意的细节：

continue 关键字也可以配合标签使用。

continue 关键字后面也是不能直接跟语句的。

continue 关键字：语句将控制权传递给它所在的封闭迭代语句的下一次迭代。(跳出本循环，执行下一次循环)。

44.在switch(expression)语句中，expression的数据类型不能是（A ）

A. double

B. char

C. byte

D. short

考察switch选择判断语句的使用

1.switch 语句选择的类型只有四种：byte，short，int ， char。JDK1.7添加了String数据类型

2.case 之间与 default 没有顺序。先判断所有的 case，没有匹配的 case 执行 default。

3.switch 语句停止的条件是遇到了 break 关键字或者结束 switch 语句的大括号。

4.如果匹配的 case 或者 default 没有对应的 break，那么程序会继续向下执行，运行可以执行的语句，直到遇到 break 或者 switch 结尾结束。

5.switch case 中的值必须要与 switch 表达式的值具有相同的数据类型。而且 case 后跟的值必须是常量，不能跟变量。case就像选择题的答案之一。 break就是如果该答案正确那么就可以跳出switch了,意思就是说 已经找出了正确的答案了。那么这道题也就做完了。如果 case没有匹配接着进行下一个 case 匹配,直到匹配为止。 最后如果都没有匹配上,那么 switch 给提供了一个默认的答案,就是 default。

注意:case后跟的是冒号:每个case中的执行语句一定要加 break；

45.给出下列代码片段:

if ( x > 0 ) {

System.out.println(“Hello.”);

}else if ( x >-3 ) {

System.out.pirntln ( “ I am Tom. ”);

}else {

System.out.println (“How are you?”);

}

请问将打印字符串 “How are you ?”的x的范围是（C ）

A. x>0

B. x > -3

C. x <= -3

D. x <=0 & x >-3

46.下列代码中，将引起编译错误的行是（ B）

1）public class Exercise{

2） public static void main(String []args){

3） float f=0.0; //float值需要加f

4） f+=1.0;

5） }

6） }

A. 第2行

B. 第3行

C. 第4行

D. 第6行

考察float数据类型的定义

注意: 一个小数如果没有加上任何的标识的时候，默认都double类型。如果是float类型的数据那么该数据必须要以f结尾。不区分大小写。

47.下列代码执行的结果为(A )

class Demo{

public static void main(String[] args){

int num = max(43,34);

System.out.println(num);

}

public static int max(int a,int b){

return a>b?a:b;

}

}

A. 43

B. 23

C. 77

D. 9

考察函数的调用及函数的形参与实参的运算值传递运算

基本数据类型交换：发现交换值前后没有变量的值发生变化

原因：只是形参的值方式交换了

数组类型交换：交换值成功

操作的是同一个数组对象

对象的值交换：交换值成功

字符串的值交换:交换值失败

48.下面程序执行的结果是( A)

Class Demo{

public static void main(String [] args){

int a=10;

// 先判断，后自加

if(a++>10) {

a=20;

}

System.out.println(a);

}

}

A. 11

B. 12

C. 20

D. 21

考察自增运算符

6．++(自增): 自增就是操作数+1

前自增: ++位于操作数的前面。 先自增，后使用。

后自增: ++位于操作数的后面。 先使用，后自增。

7．--（自减）:自减就是操作数-1

前自减：--位于操作数 的前面，先自减，后使用。

后自减: --位于操作数 的后面，先使用，后自减

如果运算符在变量的前面，则该变量自增 1 或者自减 1，然后返回的是变量的新值，如果运算符在变量的后面，则变量也会自增或者自减 1，但是返回的是变量原来的值。

++在前就是先运算，再取值，++在后就是先取值，再运算。

49.下面哪些是合法的标志符（BCD）

A. 2variable

B. variable2

C. what$

D. \_3\_

考察标识符的名命规则

标识符的要注意的细节

1.标识符的组成元素是字母（a-z A-Z）,数字（0~9） 、下划线（\_）、 美元符号（$）

2.标识符不能以数字开头。

3.java 的标示符是严格区分大小写的。

4.标识符的长度可以是任意。

5.自定义的标识符名称一般要有意义。

6.关键字、保留字(goto)不能用于自定义的标识符。

Java 中的标识符命名规范

1.包名: 多个单词组成时所有字母小写（例:package com.itcast）

2.类名和接口: 多个单词组成时所有单词的首字母大写(例:HelloWorld)

3.变量名和函数名: 多个单词组成时第一个单词首字母小写，其他单词首字母大写（例:lastAccessTime、getTime）。

4.常量名: 多个单词组成时，字母全部大写，多个单词之间使用\_分隔（例:INTEGER\_CACHE）

注意: 只是为了增加规范性、可读性而做的一种约定，标识符在定义的时候最好见名知意，提高代码阅读性

50.当整数变量值为多少时，只会输出" message 2" （BC）

switch(i){

case 1: System.out.println("message1");

case 2:

case 3: System.out.println("message2");

break;

}

A. 1

B. 2

C. 3

D. 4

考察switch选择判断语句的使用

1.switch 语句选择的类型只有四种：byte，short，int ， char。JDK1.7添加了String数据类型

2.case 之间与 default 没有顺序。先判断所有的 case，没有匹配的 case 执行 default。

3.switch 语句停止的条件是遇到了 break 关键字或者结束 switch 语句的大括号。

4.如果匹配的 case 或者 default 没有对应的 break，那么程序会继续向下执行，运行可以执行的语句，直到遇到 break 或者 switch 结尾结束。

5.switch case 中的值必须要与 switch 表达式的值具有相同的数据类型。而且 case 后跟的值必须是常量，不能跟变量。case就像选择题的答案之一。 break就是如果该答案正确那么就可以跳出switch了,意思就是说 已经找出了正确的答案了。那么这道题也就做完了。如果 case没有匹配接着进行下一个 case 匹配,直到匹配为止。 最后如果都没有匹配上,那么 switch 给提供了一个默认的答案,就是 default。

注意:case后跟的是冒号:每个case中的执行语句一定要加 break；

51.下面哪些语句可以通过编译( ACD )

A. float a= 1.34f;

B. float b=1.0;

C. float c=2f;

D. float d=20f;

考察float数据类型的定义

注意: 一个小数如果没有加上任何的标识的时候，默认都double类型。如果是float类型的数据那么该数据必须要以f结尾。不区分大小写。

52.下列函数中能和函数int max(int a, int b, double c)构成重载关系的是(BC)

A. double max(int a, int b, double c)

B. void max(int a, double c, int b)

C. int max(double a, int b)

D. int max(int x, int y, double z)

考察函数重载的定义

函数重载的定义：在同一个类中，有一个以上的同名函数，只要函数的参数列表或参数类型不一样即可，与返回值无关， 这些统称为方法的重载。

函数的重载：函数的重载就是在一个类中出现了两个或者两个以上同名的函数。

函数的重载的作用： 一个函数名可以定义多个函数以应对各种不同类型的参数。

函数重载的要求：

1. 函数名必须一致。

2. 形参列表不一致（形式参数的个数不一致或者是对应的数据类型不一致）

3. 函数的重载与返回值类型无关及是否静态无关。

53.代码阅读，下列代码运行输出的结果为（AC）

int [] arr={23,54,76,87 };

for(int i=0;i<arr.length;i+=2){

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

}

A. 23

B. 54

C. 76

D. 87

54.下面数组定义正确的是？（D）

A. int arr[] = new int[3]; //Int不是关键字。

B. int arr[] = new int[3]{1,2,3}; //Int不是关键字。有元素参数就不可以有值。

C. int [][]x = new int[][]; //缺少数组维

D. int[][] x = new int[2][]; //创建两个NULL的空间，并且放在栈上。

55.下面的说法正确的是(AC )

A. byte ,short,char 数据类型可以自动转化为int

B. float 数据类型可以自动转换为long。

C. 在java语言中默认小数为double

D. byte a=23; byte b=12; a+b的结果为byte类型　//结果为int类型

考察表达式的数据类型自动提升

算术表达式，逻辑表达式

所有的 byte 型、short 型和 char 的值将被提升到 int 型。

如果一个操作数是 long 型，计算结果就是 long 型。

如果一个操作数是 float 型，计算结果就是 float 型。

如果一个操作数是 double 型，计算结果就是 double 型。

56.下面哪几段代码不会出现编译错误(AC)

A. boolean b=true;

boolean b2=true;

if(b==b2) { System.out.println("So true"); }

B. int i=0; if (i) { System.out.println("Hi"); }//if比较表达式结果必须为boolean

C. int i=1; int j=2; if(i==1|| j==2) System.out.println("OK");

D. int i=1; int j=2; if (i==1 &| j==2) System.out.println("OK");

考察逻辑运算符与if语句

if比较表达式结果必须为boolean

57.下面关于函数描述正确的是（AD）

A. 函数是对功能代码块的封装

B. 函数没有返回值的时候什么都不用写

C. 没有返回值的函数，不能有return 语句

D. 函数是可以没有形参的

考察函数的定义

1、定义函数可以将功能代码进行封装，函数的作用就是把一个功能代码封装起来，已达到功能代码可以被复用的效果。

2、便于对该功能进行复用

3、函数只有被调用才会被执行

4、函数的出现提高了代码的复用性

5、对于函数没有具体返回值的情况，返回值类型用关键字void表示，那么该函数中的return语句如果在最后一行可以省略不写。函数中只能调用函数，不可以在函数内部定义函数。

58.下面关于循环描述正确的是（AD）

A. while循环先判断循环条件，后执行循环操作

B. while 至少会执行一次

C. do-while先进行循环条件判断，后执行循环操作

D. do-while循环至少执行一次，后进行循环判断

do while 特点是条件无论是否满足，循环体至少被执行一次。

while:先判断条件，只有条件满足才执行循环体。

do while: 先执行循环体，再判断条件，条件满足，再继续执行循环体。

简单一句话：do while：无论条件是否满足，循环体至少执行一次。

注意一个细节 do while 后面的分号;

59. Java的基本数据类型分为四型八种，分别指那些？我们选择数据类型的时候按照哪种原则选用？

整数类型：syte,shot、int、long

浮点数类型：float、double

布尔类型：boolean

字符类型：char

选择数据类型的时候尽量取占用字节少的，因为少的比较省内存空间。

60.编程实现时间转换: 5000秒是几小时几分钟几秒？请把代码粘贴到下面。

class Test{

public static void main(String[] args) {

int time=500;

int hour,minute,second;

hour=time/3600;

minute=(time-hour\*3600)/60;

second=time-hour\*3600-minute\*60;

System.out.println(time+"秒是:"+hour+"小时,"+minute+",:分钟,"+second+":秒");

}

}

61.下列哪项不是JDK所包含的内容？（A）

A．Java编程语言

B．工具及工具的API

C．Java EE扩展API

D．Java平台虚拟机

62.下列关于JDK、JRE和JVM的描述。哪项正确？D

A．JDK中包含了JRE，JVM中包含了JRE

B．JRE中包含了JDK，JDK中包含了JVM

C．JRE中包含了JDK，JVM中包含了JRE

D．JDK中包含了JRE，JRE中包含了JVM

63.下列哪个工具可以编译java源文件？ A

A．javac

B．jdb

C．javadoc

D．junit

64.JDK工具javadoc的作用是哪项？ A

A．生成Java文档

B．编译Java源文件

C．执行Java类文件

D．测试Java代码

65.以下哪些包是Java标准库中常用的包？（A,C,B）

A．java．lang

B．javax．servlet .http

C．j ava. io

D．java．sql

66.使用JDK工具生成的Java文档的文件格式是？ D

A．XML格式

B．自定义格式

c．二进制格式

D．HTML格式

67.以下关于JVM的叙述，哪项正确？（选两项） A C

A．JVM运行于操作系统之上，它依赖于操作系统

B．JVM运行于操作系统之上，它与操作系统无关

C．JVM支持Java程序运行，它能够直接运行Java字节码文件

D．JVM支持Java程序运行，它能够直接运行Java源代码文件

68.以下关于支持Java运行平台的叙述，哪项错误？ D

A．Java可在Solaris平台上运行

B．Java可在Windows平台上运行

C．Java语言与平台无关。Java程序的运行结果依赖于操作系统

D．Java语言与平台无关。Java程序的运行结果与操作系统无关

69.以下关于Applet和Java程序之间关系的叙述，哪项错误？ B

A． -个Applet就是一段Java程序

B．Applet是一种特殊的Java程序，它需要运行在Web服务器上

C．Applet是一种特殊的Java程序，它需要运行在Web浏览器上

D．Applet是一种Java SE平台的应用程序

70.以下关于Java HotSpot的描述，哪两项错误？（选两项） DA

A．JavaHotSpot是一种热编译技术，在编译Java源程序时会被使用

B．Java HotSpot是一种热编译技术，在运行Java代码时会被使用

C．Java HotSpot是一种热编译技术，它只对程序的部分字节码进行优化

D．Java HotSpot是一种热编译技术，它会对程序的全部字节码进行优化

71.环境变量PATH中含有多个路径时，路径和路径之间可以用哪项来隔开？A

A.： //用；分开

B. ，

C.\*

D:|

72.CLASSPATH中的“．”的含义是哪项？B

A．省略号

B．当前目录

C．所有目录

D．上级目录

73.JVM在执行一个Java类时，大致采用以下过程？ B

A．执行类中的代码一>装载类一>校验类

B．校验类一>装载类一>执行类中的代码

C．装载类一>执行类中的代码一>校验类

D．装载类一>校验类一>执行类中的代码

74.当运行—个Java程序时，传递参数的格式是哪项？ C

A． java类名参数1，参数2

B． javac类名参数1参数2

C． java类名参数1参数2

D． java类名参数1+参数2

75.以下关于Java文件名的叙述，正确的有？（选两项） A

A．Java源文件的扩展名应为.java

B．Java源文件的文件名应与文件中的类名一致

C． Java字节码文件的扩展名应为．java

D． 一个Java源文件中只能包含一个Java类

76.下列哪项不属于Java语言的基本数据类型？

A．int

B．String

C．double

D．boolean

77.下列哪项不是int类型的字面量？

A．\u03A6

B．077

C．OxABBC

D．20

78.下列哪项不是有效的标识符？ D

A．userName

B．2test

C．$change

D．\_password

79.下列哪项是Java语言中所规定的注释样式？（选三项） A C D

A．／／单行注释

B．--单行注释

C． ／\*

\*单行或多行注释

\*/

D． /\*\*

\*文档注释

\*/

80.下列哪项不是Java语言的关键字？

A．goto

B．sizeof

C．instanceof

D．volatile

81.现有如下五个声明：

Linel: int a\_really\_really\_really\_long\_variable\_name=5 ;

Line2: int \_hi=6；

Line3: int big=Integer. getlnteger("7”);

Line4：int $dollars=8;

line5: int %opercent=9;

哪行无法通过编译？ D

A．Line1

B．Line3

C. Line4

D. Line5

82.现有：

1. class Top {

2． static int x=l；

3. public Top (int y) {

x\*=3;

}

4. }

5. class Middle extends Top {

6. public Middle() {

x+=1;

}

7. public static void main (String [] args) {

8. Middle m = new Middle();

9. System. out .println (x);

IO. }

II. }

结果为：D

A.1

B. 2

C．3

D．编译失败

83.现有代码片段：

String s="123"；

String sl=S+456;

请问sl的结果是哪项？ A

A. 123456

B. 579

C．编译错误

D．运行时抛出异常

84.基本数据类型float的包裹类是哪项？ C

A． Integer

B． Double

C． Float

D． Character

85.现有：

1． class Test4 {

2. public static void main (String [] args) {

3． boolean X = true;

4． boolean y = false;

5． short Z = 42;

7. if((z++ == 42) ＆＆ (y = true)){

z++;

}

8. if((x = false) || (++z == 45)){

z ++;

}

10. System. out.println(¨z=”+z);

II. }

12. }

结果为： D

A. Z=42

B. z=44

C． Z= 45

D． z= 46

86.现有：

class Test2 f

public static void main (String [] args) {

short a，b,c；

a=l;

b=2;

c=a+b；

a+=2：

}

}

以上代码中，哪一句是错误的？ B

A．a=1：

B．C=a+b；

C． a+=2;

D. short a，b,C；

87.现有：

5． class Wrench2 {

6． int size;

7. public static void main(String [] args) {

8. Wrench2 w=new Wrench2();

9． w.size=11;

IO. Wrench2 w2=go(w, w.size);

II. System. out .print (w2.size);

12． }

13. public static Wrench2 go(Wrench2 wr. int s) {

14. s=12;

15。 return wr;

16. }

17. }

结果为： 11

A. 11

B. 12

c．编译失败。

D．运行时异常被抛出

88.现有：

1． class Passer f

2． static final int x=55；

3. public static void main (String [] args) {

4． new Passer()．go (x);

5. System. out .print (x);

6， }

7． void go (int x) {

8． System. out .print(x++);

9． }

10. }

结果是什么？ A

A．55

B．56

C．65

D．66

89.现有：

1． class Wrench {

2. public static void main(String [] args) {

3. Wrench w=new Wrench();

Wrench w2=new Wrench();

4． w2=go (w, w2)；

5. System.out.print (w2==w);

6． }

7. static Wrench go (Wrench wrl, Wrench wr2) {

8. Wrench wr3=wrl;

wrl=wr2;

wr2=wr3;

9． return wr3;

10． }

11． }

结果是什么？ B

A. false

B． true

C.编译失败

D．运行的时候有异常抛出

90. class TestApp{

public static void main (String[] args){

for (int i=0; i<lO;i++){

if(i==3)

break;

System. out .print (i);

}

}

}

程序运行后的输出是哪项？D

A．0123

B．012456789

C． 0123456789

D． 012

91. class TestApp{

public static void main (String[] args){

int x=6；

if (x>l)

System. out. println("x>l");

else if (x>5)

System. out .println("x>5");

else if (x<10)

System. out. println("x<lO");

else if (x<29)

System. out .println( "x<29");

else

System. out．println（“以上都不是”）;

}

}

上述程序运行后的结果是哪项？ B

A．x>5

B．x>l

C． x<10

D．x<29

92.现有：

class TestApp{

public static void main (String[] args){

int [5] myarray={ 10, 11, 12, 13, 14);

int sum=0;

for (int x ： myarray)

sum+=x;

System. out. println( "sum= "+sum);

}

}

上述程序运行后的结果是哪项？C

A． sum=10

B． sum=70

C． sum=60

D． 运行时抛出异常

93.下列有关数组的声明中，正确的是哪项？（选两项） C D

A． int s[10];

B． int[10] s；

C． int[5] s={l,2,3,4,5};

D． int s[];

94.已知数组array，其最后一个元素的下标是？ A

A． array. size

B． array. length-l

C． array. size-l

D． array．length

95.程序：

class TestApp{

public static void main (String[] args){

int X=5：

switch (x){

case l：

case 2：

case 3：

System. out．println（“一季度”）;

break;

case 4：

case 5：

case 6：

System. out．println（“二季度”）；

break;

default：

System. out．println（“三季度以上”）;

break;

)

}

)

上述程序运行后的结果是哪项？B

A. 一季度

B．二季度

c．三季度以上

D．无输出

96.为将数组myArray的长度由3改为6，现采取以下编码： D

int[]myArray=new int [3];

myArray=new int[6]；

代码执行后，以下叙述哪项是正确的？

A．数组myArray的长度已由3改为6，其中前3个元素的值不变，后3个元素

的值为空。

B．数组myArray的长度已由3改为6，其中前3个元素的值不变，后3个元素

需再经过初始化后才能使用。

C．数组myArray的长度没有变化。

D．数组myArray的长度已由3改为6，原来3个元素的值全部丢失。

97.现有：

1． class Ifs {

2． public static void main (String [] args) {

3． boolean state=false;

4． int i=2；

5． if( (++i>2) ＆＆ (state=true))

6． i++;

7． if( (++i>4) l l (state=false))

8． i++;

9． System.out .println (i);

10． }

11． }

结果为： A

A． 6

B． 5

C． 4

D．编译失败

98.现有：

3． public class Tester {

4. public static void main (String[] args) {

5． int x = 5;

6. Integer xl = x; Integer x2 = x;

7. int x3=new Integer(5);

8. system.out.print(x1.equals(x));

9. system.out.print(xl==x);

lu. system.out.print(x2.equals(xl));

11. system.out.print(x2==xl);

12- system.out.print(x2==x3);

13- system.out.print(x2.equals(x3));

}

15．}

结果为： C

A．编译失败

B. falsefalsetruetruetruetrue

C． truetruetruetruetruetrue

D. falsefalsetruetruetruefalse

E. truefalsetruefalsefalsetrue

F.运行时异常被抛出

99.现有：

1． class Rectangle {

2' public static V.id main(string [] args) {

3． int[]x=(1，2，3);

4． x[1]= (x[1]>1) ?x[2]: O;

5． System.out．println(x[1]);

6． }

7． }

结果为：A

A．3

B．2

C．1

D．0

100.现有：

1．class Output (

2． public static void main (String[] args) {

3． int i=5：

4． System.out.print( "4"+i+"");

5． System.out.print (i+5+"7");

6． System.out.println (i+"8");

7． }

8． }

结果为：C

A． 9 9722

B． 9 55758

C． 45 10758

D． 45 9722

101.以下哪种初始化数组的方式是错误的？

A． String[]names={"zhang", "wang", "li");

B． String names[] =new String [3];

names [O] ="zhang"; names [1] ="wang"; names [2] ="li";

C． String[3] names={"zhang", "wang", "li"};

D． 以上皆正确

102.现有：

1． class WhileTests {

2． public static void main (String [] args) {

3． int X=5；

4． while (++x<4) {

5． --x;

6． }

7． System.out.println( "x="+x);

8． }

9． }

结果是什么？ A

A.X=6

B. X=5

C.X=2

D．编译失败

103.现有：

1． class Test2 {

2． public static void main (String [] args) {

3． boolean X= true;

4． boolean y= false;

5． short Z=20;

7． if((x==true) &＆ (y=true)) z++;

8． if((y==true) || (++z==22)) z++;

10． System. out .println( "z="+z)；

11． }

12． }

结果是什么？C

A.Z=21

B. z=22

C. z=23

D． Z= 24

104.现有：

1． class Foo {

2． public static void main (String [] args) {

3． int x = O；

4． int y = 4；

5． for (int z = 0; z < 3； Z++, X++) {

6． if( x > 1 & ++y < 10)

7． y++;

8． }

9． System. out .println (y);

10． }

11． }

结果是什么？ A

A．7

B．8

C．10

D．12

105.配置PATH、CLASSPATH、JAVA\_HOME的作用？(10分)

保证javac命令可以在任意目录下运行。

保证class文件可以在任意目录下运行

方便修改内容是JDK的安装路径

106.java的基本数据类型分为四型八种，分别指那些？我们选择数据类型的时候按照那种原则选用？(10分)

数据类型：byte、short、Int、long

浮点类型：float、double

布尔类型：boolean

字符类型：char

107.在JAVA中，下列哪些说法是正确的（ ） A C

A．java源文件的扩展名为.java

B．写好的java源程序可以直接运行

C．编写的源程序必须先编译后才能运行

D．程序员可以读懂扩展名为.class的文件

108.下列标识符哪个是合法的（ ） B

A．class

B．$abc

C．1234

D．\_name

109.下面哪些赋值语句是正确的?( ) D

A. long test = 012;

B. float f = -412;

C. int other = (int) true;

D. double d = 0x12345678;

E. byte b = 128;

110.在Java中，下列语句不能通过编译的有( ) B D

A. String s= “join”+ 3;

B. int a= “join”+3;

C. int a= ‘a’+5;

D. float f=5+5.5;

111.设 a = 3，则表达式 (--a )<<a的结果是（ ） B

A. 16

B. 8

C. 24

D. 12

112.下列代码运行的结果是( ) C

public static void main(String[] args) {

int x = 4;

int y = 5;

if(x++ > 4 & ++y > 5) {

x++;

}

System.out.println("x=" + x + ",y=" + y);

}

A. x=4,y=5

B. x=4,y=6

C. x=5,y=6

D. x=6,y=6

113.尝试运行下面的代码会发生什么？( )

public class Test {

public static void main(String[] args) {

boolean m = true;

if (m = false) {

System.out.println("true");

} else {

System.out.println("false");

}

}

}

A. 输出字符串”true”

B. 输出字符串”false”

C. 编译时错误

D. 运行时错误

114.编译并运行以下代码将发生什么？ （ ） E

public static void main(String args[]){

char digit = 'a';

for (int i = 0; i < 10; i++){

switch (digit)

{

case 'x' :

{

int j = 0;

System.out.println(j);

}

default :

{

int j = 100;

System.out.println(j);

}

}

}

int i = j;

System.out.println(i);

}

A. 输出11次100

B. 输出10次100，然后抛出运行期异常

C. 编译错误，因为变量i不能在main() 方法中被声明2次

D. 编译错误，因为变量j不能在main() 方法中被声明2次

E. 以上都不对

115.class Demo B

{

public static int fun(int c)

{

return c+=2;

}

public static void main(String[] args)

{

int temp = fun(2);

System.out.println(temp);

}

}

A. 2

B. 4

C. 6

D. 8

116.下面哪些函数是public void aMethod(){...}的重载函数?( ) B D

A. void aMethod( ){...}

B. public int aMethod(int a, float b){...}

C. public void aMethod (){...}

D. public float aMethod (int m){…}

117.在java中，关于数组描述正确的是（ ） B D

A．数组中的索引下标从1开始

B．存储在数组当中的数据都属于同一数据类型

C．通过数组名.length()能获得数组的长度

D．数组的最大索引下标是数组的长度减1

118.下面程序的输出结果是什么( ) D

int[] arr = new int[10];

System.out.println(arr[0]);

A．编译不通过

B．运行时出错

C．输出null

D．输出0

119.下面哪个语句正确地声明一个整型的二维数组( )

A. int a[][] = new int[][];

B. int b[10][10] = new int[][];

C. int c[][] = new int[10][10];

D. int []d[] = new int[10][10];

120.以下代码输出是（ ） D

class Demo {

public static void main(String[] args) {

int i = 0;

int sum = 0;

while (i <= 10) {

i++;

if (i % 2 != 0)

continue;

sum += i;

}

System.out.println(sum);

}

}

A. 55

B. 45

C. 35

D. 30

121.和下面代码能完成相同的选项是（） C

class Demo {

public static void main(String[] args) {

int i = 1;

int sum = 0;

while (i <= 100) {

if (i % 2 == 0) {

sum = sum + i;

}

i++;

}

System.out.println(sum);

}

}

A. for (int x =1; x<=100;x++){ sum=sum+x;}

B. for (int x =0; x<=100;x+=2){ sum=sum+x;}

C. for (int x =1; x<=100;x+=2){ sum=sum+x;}

D.上述全对