1. 语法错误：创建代码时发生了错误，导致编译时出现的错误。

运行错误：当运行应用程序时，环境检测到一个不可能执行的操作时，出现的错误。

逻辑错误：程序未按预期方式执行的错误。(P17~19)

1. 语法：拼错关键字、忽略标点符号、左花括号对应无右花括号等。

运行：输入错误、0作为除数等。

逻辑：假设将摄氏35度转化为华氏度，写成（9/5）\*35+32只能得到67，因为java中整数相除返回除法的整数部分，所以9/5只能得到1.要得到正确结果需要使用9.0/5，才能得到1.8

1. 变量和方法命名采用小写字母，如果一个名字包含多个单词，就将它们连在一起，第一个单词字母小写，后面的每个单词首字母大写。（redius，area，方法print）

类名中每个单词首字母大写（System,Computer）.

常量里所有的字母都要大写，两个单词之间用下划线连接（MAX\_VALUE）.

MAX\_VALUE可作为常量，Test作为类，read作为变量，readDouble作为方法

1. package palindromenumber;

import java.util.Scanner;

public class PalindromeNumber {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.print("Enter the three\_digit interger:");

int number=input.nextInt();

if(number/100==number%10){

System.out.println(number + " is a palindrome");

}

else{

System.out.println(number + " is not a palindrome");

}

}

}

5.

package randomnumber;

public class RandomNumber {

public static void main(String[] args) {

System.out.println((char)(65+(Math.random()\*26)));

}

}

6.

package dividedby5or6;

public class DividedBy5or6 {

public static void main(String[] args) {

final int NUMBER\_OF\_NUMBERS\_PER\_LINE=10;

int count=0;

int number=0;

for(number=100;number<=200;number++){

boolean IsNumber=false;

if((number%5==0)&&(number%6!=0)){

IsNumber=true;

}

else if((number%6==0)&&(number%5!=0)){

IsNumber=true;

}

if(IsNumber){

count++;

if(count%NUMBER\_OF\_NUMBERS\_PER\_LINE==0){

System.out.println(number);

}

else{

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

}

}

}

}

}

7.方法重载是指可以在一个类中定义多个同名的方法，但它们具有不同的参数类型和参数列表。 所以可以。 被重载的方法必须有不同的参数列表，而且不能基于不同的修饰符和返回值。 所以不可以。

8．

package sumdigits;

import java.util.Scanner;

public class SumDigits {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.print("Enter a number:");

int number=input.nextInt();

int sum=sumDigits(number);

System.out.println("The sumdigits of number " + number + " is " + sum);

}

public static int sumDigits(long n){

int sum=0;

while(n>0){

sum+=n%10;

n/=10;

}

return sum;

}

}

9.

package methodtotestprime;

public class MethodToTestPrime {

public static void main(String[] args) {

int count=0;

for(int number=2;number<10000;number++){

if(isPrime(number)){

count++;

}

}

System.out.println("The number of prime is " + count);

}

public static boolean isPrime(int number){

for(int divisor=2;divisor<=number/2;divisor++){

if(number%divisor==0){

return false;

}

}

return true;

}

}

10.

package assigngrades;

import java.util.Scanner;

public class AssignGrades {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.print("Enter the number of students:");

int number=input.nextInt();

double scores[]=new double[number];

System.out.print("Enter " + number + " sores :");

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

scores[i]=input.nextDouble();

}

double max=scores[0];

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

if(scores[j]>max)

max=scores[j];

}

for(int k=0;k<scores.length;k++){

char level;

if(scores[k]>=max-10){

level='A';

}

else if (scores[k]>=max-20){

level='B';

}

else if (scores[k]>=max-30){

level='C';

}

else if(scores[k]>=max-40){

level='D';

}

else{

level='F';

}

System.out.println("student " + k + " score is " + scores[k] + " and grade is " + level);

}

}

}

11.

package arrayaverage;

import java.util.Scanner;

public class ArrayAverage {

public static int average(int[] array){

int sum1=0;

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

sum1+=array[i];

}

return (int)(sum1/array.length);

}

public static double average(double[] array){

double sum2=0;

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

sum2+=array[j];

}

return sum2/array.length;

}

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.println("Enter 10 int values:");

int array1[]=new int[10];

for(int i=0;i<array1.length;i++){

array1[i]=input.nextInt();

}

System.out.println("Enter 10 double values:");

double array2[]=new double[10];

for(int j=0;j<array2.length;j++){

array2[j]=input.nextDouble();

}

System.out.println("The average of the 10 int values is: " + average(array1));

System.out.println("The average of the 10 double values is: " + average(array2));

}

}

12.

package findsmallestelement;

import java.util.Scanner;

public class FindSmallestElement {

public static double min(double[] array){

double min=array[0];

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

if(array[i]<min)

min=array[i];

}

return min;

}

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.println("Enter ten numbers:");

double[] array=new double[10];

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

array[i]=input.nextDouble();

}

double min=min(array);

System.out.println("The mininum number is:" + min);

}

}

13.

package sumofmajordiagonal;

import java.util.Scanner;

public class SumOfMajorDiagonal {

public static double sumMajorDiagonal (double[][] m){

double sum=0;

for(int row=0;row<m.length;row++){

for(int col=0;col<m[row].length;col++){

if(row==col){

sum+=m[row][col];

}

}

}

return sum;

}

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.println("Enter a 4-by-4 matrix row by row:");

double [][] array = new double[4][4];

for(int row=0;row<array.length;row++){

for(int col=0;col<array[row].length;col++){

array[row][col]=input.nextDouble();

}

}

double sum=sumMajorDiagonal(array);

System.out.println("Sum of the elements in the major diagonal is " + sum);

}

}

14.

package sortstudentsongrades;

import java.util.Arrays;

public class SortStudentsOnGrades {

public static void main(String[] args) {

char[][] answers={

{'A','B','A','C','C','D','E','E','A','D'},

{'D','B','A','B','C','A','E','E','A','D'},

{'E','D','D','A','C','B','E','E','A','D'},

{'C','B','A','E','D','C','E','E','A','D'},

{'A','B','D','C','C','D','E','E','A','D'},

{'B','B','E','C','C','D','E','E','A','D'},

{'B','B','A','C','C','D','E','E','A','D'},

{'E','B','E','C','C','D','E','E','A','D'}};

char [] keys={'D','B','D','C','C','D','A','E','A','D'};

int []count=new int[8];

for(int row=0;row<answers.length;row++){

int correctcount=0;

for(int col=0;col<answers[row].length;col++){

if(answers[row][col]==keys[col]){

correctcount++;

}

}

count[row]=correctcount;

}

Arrays.sort(count);

for(int k=0;k<count.length;k++){

System.out.println("Student " + k + " 's correct count is " + count[k]);

}

}

}

15.

(1)构造方法没有返回类型，普通方法有返回类型。

(2)构造方法名必须具备和所在类相同的名字，普通方法的方法名不能和类名相同。

（3）构造方法是在创建一个对象使用new操作符时调用的，它的作用是初始化对象。如果没有给一个类定义构造方法，方法在加载这个类的时候，会自动添加一个无参构造方法。所以，常见的 Student s=new Student()；里的s这个实例，是通过构造方法初始化的；而普通方法不行。

（4）普通方法仅仅是类成员，构造方法也是类成员，但是，构造方法不仅可以创建一个对象，还可以初始化对象的值。

16. 访问器方法：是访问类中私有数据域的方法，返回数据域的值。

修改器方法：是修改类中私有数据域值的方法，为数据域设置新值。

命名习惯：访问器get+数据域名，修改器set+数据域名

17.传递基本类型参数时，传递的为实参的值。传递引用类型参数时，传递的是对象的引用。结果为：count=101，times=0；

18.(1)引用隐藏的数据域即成员变量，通过this可以引用所调用的实例方法的对象。

Public Class Student {

String name; *//定义一个成员变量name*

private void SetName(String name) { *//定义一个参数(局部变量)name*

this.name=name; *//将局部变量的值传递给成员变量*

}

}

(2)调用构造方法，当一个类存在多个构造方法时，需要用this（参数列表）来表示到底调用的是哪一个构造方法，如果this关键字后面加上了一个参数，那么就表示其引用的是带参数的构造方法。this调用本类中的其他构造方法，调用时要放在构造方法的首行。

19.