Class 声明

Public都可以访问/Protected子类可访问/Private只有类自己能访问

Static静态方法和静态属性由类来调用，只在静态方法区加载一次

//成员属性和成员方法都是静态的，但属于对象、由对象来调用

实例化对象：

类名称+对象名称 = new 类名称（）；

Byte a,byte b;

Byte c=a+b则会报错 因为a+b的默认的类型为int

Byte等数据类型会被包装成类，其本身不是类

调用Byte包装类的类成员

import java.util.Scanner;

public class Hello

{

public static void main(String[] arg)

{

System.out.println(Byte.MIN\_VALUE);

System.out.println(Short.MIN\_VALUE);

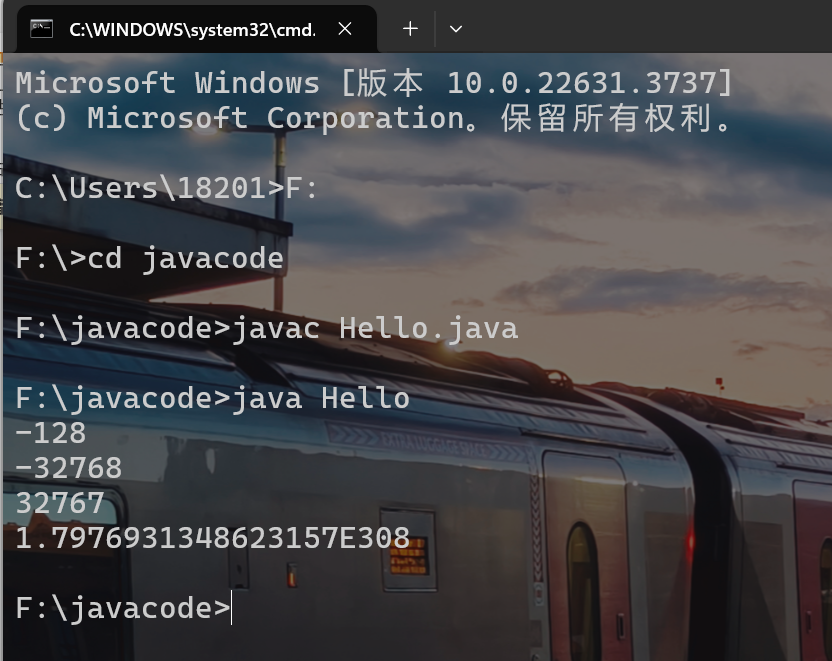
System.out.println(Short.MAX\_VALUE);

System.out.println(Double.MAX\_VALUE);

}

}

结果如下



十进制转二进制

import java.util.Scanner;

public class Hello

{

public static void main(String[] arg)

{

Scanner s=new Scanner(System.in);

System.out.println("输入十进制数：");

int a=s.nextInt();

while(a>0)

{

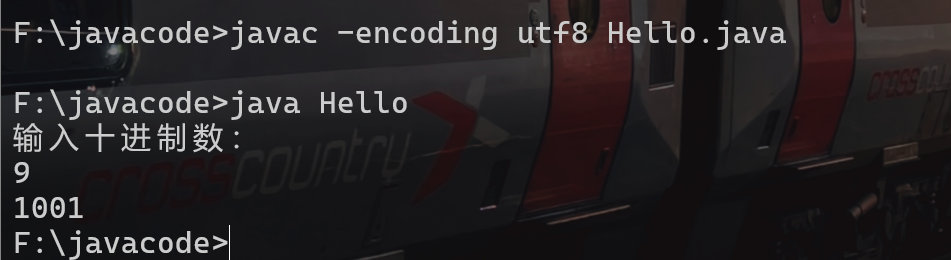
System.out.print(a%2);

a/=2;

}

}

}



分支判断三角形

import java.util.Scanner;

public class Hello

{

public static void main(String[] arg)

{

Scanner s=new Scanner(System.in);

System.out.println("分别输入三条边的值：");

int a=s.nextInt();

int b=s.nextInt();

int c=s.nextInt();

if((a+b)>c&&(a+c)>b&&(b+c)>a)

{

System.out.println("可构成三角形");

if(a==b&&b==c)

{

System.out.println("且是等边三角形");

}

else if(a==b||b==c||a==c)

{

System.out.println("且是等腰三角形");

}

if(a\*a+b\*b==c\*c||a\*a+c\*c==b\*b||b\*b+c\*c==a\*a)

{

System.out.println("且是直角三角形");

}

}

else

{

System.out.println("不是三角形");

}

}

}





吃馒头

import java.util.Scanner;

public class Hello

{

public static void main(String[] arg)

{

int a=4;

int x=9;

while(x>0)

{

System.out.println("第"+x+"天吃了"+(a+1)\*2+"个馒头");

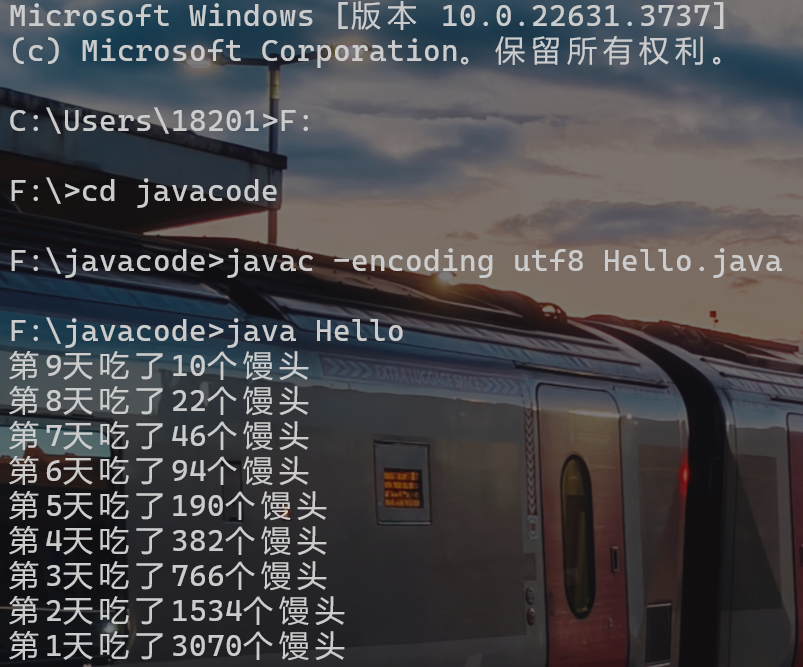
a=(a+1)\*2;

x--;

}

}

}



集成开发环境：IDEA

Project包含module，module包含包（package），包包含类

快捷方式：psvm sout

new Scanner(System.in).var+enter->Scanner xx=new Scanner(System.in)

java数组

类名称[] 数组名=new int[];

new int[5].var🡪int [ ]xx=new int[5];

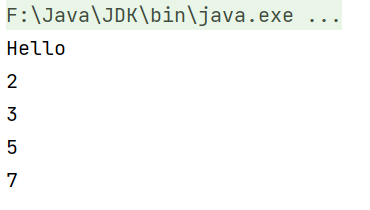
int[] i = new int[]{0,1};

简写：int[]j={0,2,3,5};

冒泡排序

public class Student {  
 public static void main(String[] args) {  
 System.*out*.println("Hello");  
 Scanner s = new Scanner(System.*in*);  
 int[]num={3,5,7,2};  
 boolean flag=true;  
 while(flag)  
 {  
 flag=false;  
 for(int i=0;i<3;i++)  
 {  
 if(num[i]>num[i+1])  
 {  
 int t=num[i];  
 num[i]=num[i+1];  
 num[i+1]=t;  
 flag=true;  
 }  
 }  
 }  
 for(int i=0;i<4;i++)  
 {  
 System.*out*.println(num[i]);  
 }  
 }  
}

结果



输出100以内质数及斐波那契数列：

package Users;  
  
import java.util.ArrayList;  
  
import java.util.Scanner;  
  
public class Student {  
 public static void main(String[] args) {  
 ArrayList<Integer> list = new ArrayList<>();  
 list.add(1);  
 list.add(2);  
 list.add(3);  
 Student s = new Student();  
 s.pr();  
 s.ph();  
 }  
 public void pr()  
 {  
 for(int i=0;i<101;i++)  
 {  
 if(isp(i))  
 System.*out*.println(i);  
 }  
 }  
 public boolean isp(int i)  
 {  
 if(i==0||i==1)  
 return true;  
 for(int j=2;j<i;j++)  
 {  
 if(i%j==0)  
 return false;  
 }  
 return true;  
 }  
  
 public void ph()  
 {  
 int a=1;  
 int b=1;  
 int sum=2;  
 System.*out*.println(1);  
 System.*out*.println(1);  
 while(sum<100)  
 {  
 int t=sum;  
 System.*out*.println(sum);  
 a=b;  
 b=t;  
 sum=a+b;  
 }  
 }  
}

判断字符串

public int finds(String m,String s)  
{  
 int ml=m.length();  
 int sl=s.length();  
 int cnt=0;  
 for(int i=0;i<=ml-sl;i++)  
 {  
 if(m.substring(i,i+sl).equals(s))  
 cnt++;  
 }  
 return cnt;  
}