**10.3** (The MyInteger class) Design a class named MyInteger. The class contains:

(Mylnteger 类）设计一个名为 Mylnteger 的类。这个类包括：

■ An int data field named value that stores the int value represented by this object.

一个名为 value 的 int 型数据域.存储这个对象表示的 int 值。

■ A constructor that creates a MyInteger object for the specified int value.

一个为指定的 int 值创建 Mylnteger 对象的构造方法。

■ A getter method that returns the int value.

一个返回 int 值的 getter 方法。

■ The methods isEven(), isOdd(), and isPrime()that return true if the value in this object is even, odd, or prime, respectively.

如果值分别为偶数、 奇数或素数，那么isEvenO、isOdd()和 isPrimeO 方法都会返回

true

■ The static methods isEven(int), isOdd(int), and isPrime(int)that return true if the specified value is even, odd, or prime, respectively.

如果指定值分别为偶数、 奇数或素数，那么相应的静态方法 isEven(int)、isOdd(int)和

isPrime(int)会返回 true

■ The static methods isEven(MyInteger), isOdd(MyInteger), and isPrime(MyInteger)that return true if the specified value is even, odd, or prime, respectively.

如果指定值分别为偶数、 奇数或素数，那么相应的静态方法 isEvenCMylnteger)、

isOdd(Mylnteger)和 isPrime(Mylnteger)会返回 true。

■ The methods equals(int)and equals(MyInteger)that return true if the value in this object is equal to the specified value.

如果该对象的值与指定的值相等，那么 equals(int)和 equals(Mylnteger)方法返回 true

■ A static method parseInt(char[])that converts an array of numeric characters to an int value.

静态方法 parselnt(charl：[])将数宇字符构成的数组转换为一个 int 值。

■ A static method parseInt(String)that converts a string into an int value.

静态方法 parselnt(String)将一个字符串转换为一个 int 值。

Draw the UML diagram for the class and then implement the class. Write a client program that tests all methods in the class.

画出该类的 UML 图并实现这个类。编写客户程序测试这个类中的所有方法

答：

Mylnteger类：Mylnteger.java

public class Mylnteger {

int value;

public Mylnteger(int value) {

this.value = value;

}

public int getValue() {

return value;

}

public boolean isEven() {

if (value % 2 == 0)

return true;

else

return false;

}

public boolean isOdd() {

if (value % 2 == 1)

return true;

else

return false;

}

public boolean isPrime() {

if (value <= 3) {

return value > 1;

}

for (int i = 2; i < value; i++) {

if (value % i == 0) {

return false;

}

}

return true;

}

public static boolean isEven(int i) {

if (i % 2 == 0)

return true;

else

return false;

}

public static boolean isOdd(int i) {

if (i % 2 == 1)

return true;

else

return false;

}

public static boolean isPrime(int ii) {

if (ii <= 3) {

return ii > 1;

}

for (int i = 2; i < ii; i++) {

if (ii % i == 0) {

return false;

}

}

return true;

}

public static boolean isEven(Mylnteger mi) {

int i = mi.value;

if (i % 2 == 0)

return true;

else

return false;

}

public static boolean isOdd(Mylnteger mi) {

int i = mi.value;

if (i % 2 == 1)

return true;

else

return false;

}

public static boolean isPrime(Mylnteger mi) {

int ii = mi.value;

if (ii <= 3) {

return ii > 1;

}

for (int i = 2; i < ii; i++) {

if (ii % i == 0) {

return false;

}

}

return true;

}

public boolean equals(Mylnteger mi) {

if (value != mi.value)

return false;

return true;

}

public boolean equals(int i) {

if (value != i)

return false;

return true;

}

public int parseInt(char[] arr) {

int i = 0;

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

if (arr[j] < 48 && arr[j] > 57)

return 0;

i = i \* 10 + arr[j] - 48;

}

return i;

}

public int parseInt(String str) {

int i = 0;

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

if (str.charAt(j) < 48 && str.charAt(j) > 57)

return 0;

i = i \* 10 + str.charAt(j) - 48;

}

return i;

}

}

Mylnteger测试类：MylntegerText.java

public class MylntegerText {

public static void main(String[] args) {

Mylnteger mi = new Mylnteger(123);

System.out.println("value:" + mi.getValue());

System.out.println("isEven():" + mi.isEven());

System.out.println("isOdd():" + mi.isOdd());

System.out.println("isPrime():" + mi.isPrime());

System.out.println("isEven(int: 4):" + Mylnteger.isEven(4));

System.out.println("isPrime(int: 7):" + Mylnteger.isPrime(7));

System.out.println("isOdd(int: 3):" + Mylnteger.isOdd(3));

Mylnteger mi2 = new Mylnteger(12345);

System.out.println("isEven(Mylnteger: 12345):" + Mylnteger.isEven(mi2));

System.out.println("isPrime(Mylnteger: 12345):" + Mylnteger.isPrime(mi2));

System.out.println("isOdd(Mylnteger: 12345):" + Mylnteger.isOdd(mi2));

System.out.println("equals(Mylnteger: 12345) :" + mi.equals(mi2));

System.out.println("equals(int: 123) :" + mi.equals(123));

char[] arr = {'7','3','5','6'};

System.out.println("parseInt(char[] arr) :"+mi.parseInt(arr));

System.out.println("parseInt(String arr) :"+mi.parseInt("7458"));

}

}

MyInteger

value: int

+ Mylnteger ()

+ getValue(): int

+ isEven(): boolean

+ isOdd (): boolean

+ isPrime (): boolean

+ isEven(i: int): boolean

+ isOdd (i: int): boolean

+ isPrime (ii: int): boolean

+ isEven(mi: Mylnteger): boolean

+ isOdd (mi: Mylnteger): boolean

+ isPrime (mi: Mylnteger): boolean

+ equals(mi: Mylnteger): Boolean

+ equals(i: int): boolean

+ parseInt(char[]: arr): int

+ parseInt(String: str): int

**10.4** (The MyPoint class) Design a class named MyPointto represent a point with x- and y-coordinates. The class contains:

(MyPoint 类）设计一个名为 MyPoint 的类，表示一个带x坐标和y坐标的点。该类包括：

■ The data fields x and y that represent the coordinates with getter methods.

两个带 get 方法的数据域 x 和 y 分别表示它们的坐标

■ A no-arg constructor that creates a point (0, 0).

一个创建点(0,0)的无参构造方法。

■ A constructor that constructs a point with specified coordinates.

一个创建特定坐标点的构造方法。

■ A method named distance that returns the distance from this point to a specified point of the MyPoint type.

一个名为 distance 的方法，返回从该点到 MyPoint 类型的指定点之间的距离。

■ A method named distance that returns the distance from this point to another point with specified x- and y-coordinates.

一个名为 distance 的方法，返回从该点到指定 x 和y坐标的指定点之间的距离。

Draw the UML diagram for the class and then implement the class. Write a test program that creates the two points (0, 0) and (10, 30.5) and displays the distance between them.

画出该类的 UML 图并实现这个类。编写一个测试程序，创建两个点（0,0)和（10,30.5),

并显示它们之间的距离。

MyPoint测试类：MyPoint.java

public class MyPoint {

double x;

double y;

public MyPoint() {

x = 0;

y = 0;

}

public MyPoint(double x, double y) {

this.x = x;

this.y = y;

}

public double getX() {

return x;

}

public double getY() {

return y;

}

public double distance(MyPoint mp) {

double result = Math.sqrt((mp.getX() - x) \* (mp.getX() - x) + (mp.getY() - y) \* (mp.getY() - y));

return result;

}

public double distance(double x, double y) {

double result = Math.sqrt((this.x - x) \* (this.x - x) + (this.y - y) \* (this.y - y));

return result;

}

}

MyPoint测试类：MyPointText.java

public class MyPointText {

public static void main(String[] args) {

MyPoint mp1 = new MyPoint(0,0);

MyPoint mp2 = new MyPoint(10,30.5);

System.out.printf("distance:%.2f",mp1.distance(mp2));

}

}

MyPoint

x: double

y: double

+ MyPoint ()

+ MyPoint (x: double,y: double)

+ getX(): double

+ getY(): double

+ distance(x: double,y: double): double

+ distance(mp: MyPoint): double