

《程式設計》

一、請使用Java、C#、C++或Python等物件導向程式語言，建立矩陣相關運算的系統，系統一共有三個類別，分別為抽象類別Matrix，實作類別MatrixMultiply及MatrixAdd，其中：（30分）

*抽象類別Matrix中所有的變數跟方法皆為protected，主要包含了：

- matrix1, matrix2, resultM三個實數值二維 matrices資料
- 一個抽象的checkDimension()方法，用來檢查matrix1跟matrix2是否適合做所指定的運算
- 一個抽象的matrixOperation()方法，用來對matrix1跟matrix2進行運算
- 一個setMatrix()方法，可以將所傳入的兩個matrices複製到matrix1, matrix2
- 改寫物件的toString()方法，可以顯示matrix1, matrix2及resultM的內容

*類別MatrixMultiply，繼承類別Matrix：

- 實作Matrix中的兩個抽象方法，分別可以實現父類別內 matrix1 及 matrix2 進行乘法時的維度檢查及乘法運算，進行運算前會主動呼叫 checkDimension()，確定是否可以進行相關運算，如無法運算，resultM會被設定為null並回傳，否則進行運算，結果存於父類別的resultM並回傳
- 一個空建構子及一個可以接受兩個二維matrices的建構子，建構子中呼叫父類別的setMatrix()方法，設定matrix1及matrix2（空建構子會將matrix1及matrix2設為null）

*類別MatrixAdd，同MatrixMultiply，僅實作抽象方法時，是以matrices加法為對象，進行維度的檢查及運算

試題評析

本試題是物件導向的問題，以抽象類別為父類別，另外產生兩個子類別繼承父類別後改寫函數與實作抽象函數。其中，要注意的是矩陣乘法，平常有練習的考生會快速完成矩陣乘法函數。本題只要按照題目指示的方式撰寫程式就可以獲取高分。

答：

Matrix.java

```
public abstract class Matrix {
    protected double [][]matrix1;
    protected double [][]matrix2;
    protected double [][]resultM;

    protected abstract boolean checkDimension();
    protected abstract void matrixOperation();

    protected void setMatrix(double [][]matrix1, double [][]matrix2) {
        this.matrix1=matrix1;
        this.matrix2=matrix2;
    };
    @Override
    public String toString() {
        System.out.println("matrix1:");
```

```

for(int i=0; i<matrix1.length; i++) {
    for(int j=0; j<matrix1[i].length; j++) {
        System.out.print(matrix1[i][j] + " ");
    }
    System.out.println("");
}
System.out.println("matrix2:");
for(int i=0; i<matrix2.length; i++) {
    for(int j=0; j<matrix2[i].length; j++) {
        System.out.print(matrix2[i][j] + " ");
    }
    System.out.println("");
}
System.out.println("resultM:");
for(int i=0; i<resultM.length; i++) {
    for(int j=0; j<resultM[i].length; j++) {
        System.out.print(resultM[i][j] + " ");
    }
    System.out.println("");
}
return "";
}
}

```

MatrixAdd.java

```

public class MatrixAdd extends Matrix {
    public MatrixAdd() {
        this.matrix1=null;
        this.matrix2=null;
    }
    public MatrixAdd(double [][]matrix1, double [][]matrix2) {
        super.setMatrix(matrix1, matrix2);
    }
}

```

@Override

```

protected boolean checkDimension() {
    if(matrix1.length==matrix2.length)
        return true;
    else
        return false;
}

```

@Override

```

protected void matrixOperation() {
    resultM = new double[matrix1.length][matrix1[0].length];
    if(checkDimension()) {
        for(int i=0; i<matrix1.length; i++)
            for(int j=0; j<matrix1[i].length; j++)
                resultM[i][j]=matrix1[i][j]+matrix2[i][j];
    }
}
}

```

MatrixMultiply.java

```

public class MatrixMultiply extends Matrix {
    public MatrixMultiply() {
        this.matrix1=null;
        this.matrix2=null;
    }
    public MatrixMultiply(double [][]matrix1, double [][]matrix2) {
        super.setMatrix(matrix1, matrix2);
    }
    @Override
    protected boolean checkDimension() {
        if(matrix1[0].length==matrix2.length)
            return true;
        else
            return false;
    }
    @Override
    protected void matrixOperation() {
        resultM = new double[matrix1.length][matrix2[0].length];
        if(checkDimension()) {
            for(int i=0; i<matrix1.length; i++)
                for(int j=0; j<matrix2[0].length; j++)
                    for(int k=0; k<matrix1[0].length; k++)
                        resultM[i][j]+=matrix1[i][k]*matrix2[k][j];
        }
    }
}

```

【版權所有，重製必究！】

二、請使用Java、C、C++、C#或Python，分別使用iterative跟recursive方法，撰寫二元搜尋法，搜尋已排序的整數值數列。（20分）

註：假設數列資料是以具有array性質的list來存放

*模組程式應能接受欲搜尋的資料及已排序數列的相關資料

* 模組程式應回傳所欲搜尋的資料是否在數列資料中

試題評析

二元搜尋法(Binary Search)是資料結構內的基本考題，對於本試題，大部分的考生應該都很熟悉演算法。二元搜尋法的資料數列需要先排序，每次從數列的中間數值進行比較，比較的結果有相同或比較小或比較大，分別採取不同的後續處理機制，只要用心撰寫程式就可得到高分。

答：

```
int binarySearch_loop(int data[], int left, int right, int key){
    int mid;

    while(left<=right){
        mid=(left+right)/2;
        if(key<data[mid]) right=mid-1;
        else if(key>data[mid]) left=mid+1;
        else return(mid); /* found */
    }
    return(-1); /* not found */
}

int binarySearch_rec(int data[], int left, int right, int key){
    int mid;
    if(left<=right){
        mid=(left+right)/2;

        if(data[mid]==key)
            return(mid);
        else if(data[mid]>key)
            return(binarySearch_rec(data, left, mid-1, key));
        else
            return(binarySearch_rec(data, mid+1, right, key));
    }
    else return(-1); /* not found */
}
```

三、請使用Java、C、C++、C#或Python撰寫相關程式模組，使用stack（先進後出的線性資料結構）來完成preorder的深度優先（Depth First Search）樹狀圖追蹤（traversal）。（25分）

註：假設樹狀圖的節點資料可以為任意型別

註：假設樹狀圖的節點結構內含三個全域變數：資料、父節點、所有子節點串接的linked list

【版權所有，重製必究！】

* 模組程式應能接受樹狀圖的樹根

* 模組程式應以字串數列方式，回傳樹狀圖追蹤的結果（以空白、逗號或換行符號區隔資料字串）

* 樹狀圖中的節點需另以獨立的class定義節點資料、相關的建構子（Constructor）與存取子（Accessor/Matador）

試題評析

本試題以物件導向的程式撰寫二元樹的前序追蹤法(Preorder Traversal)。首先先定義節點(Node)類別，包含資料成員，左參考(Reference)與右參考(Reference)。接下來定義二元樹與前序追蹤法(Preorder)函數，利用主程式呼叫前序追蹤法函數就可以完成程式。本試題也是資料結構的問題之一，演算法並不難，有程度的考生可輕鬆獲取高分。

答：

Node.java

```
import java.util.ArrayList;
```

```
public class Node {
    int item;
    Node parent;
    ArrayList<Node> children;
    public Node(int key) {
        item = key;
        parent = null;
        children = new ArrayList<Node>();
    }
}
```

Tree.java

```
import java.util.Iterator;
```

```
public class Tree {
    Node root;
    Tree() {
        root = null;
    }
    void preorder(Node node) {
        if (node == null)
            return;
        System.out.print(node.item + ",");
        Iterator<Node> it = node.children.iterator();
        while(it.hasNext()) {
            preorder(it.next());
        }
    }
}
```

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

```
    Tree tree = new Tree();
    Node n1 = new Node(1);
    Node n2 = new Node(2);
    Node n3 = new Node(3);
    Node n4 = new Node(4);
```

```
Node n5 = new Node(5);
Node n6 = new Node(6);
Node n7 = new Node(7);
Node n8 = new Node(8);
```

```
tree.root = n1;
n1.children.add(n2);
n1.children.add(n3);
n2.children.add(n4);
n2.children.add(n5);
n3.children.add(n6);
n3.children.add(n7);
n3.children.add(n8);
System.out.println("\nPreorder traversal ");
tree.preorder(tree.root);
```

```
}
```

```
}
```

結果: Preorder traversal

1,2,4,5,3,6,7,8,

四、請使用 Java、C、C++、C# 或 Python 撰寫相關模組程式，可以找出某一整數的所有 generator(s)。(25分)

digit-sum：正整數N的digit-sum定義為「該正整數本身跟構成該整數個別數字的總和」，如245的digit-sum為256，因為 $245+2+4+5$ 等於256

generator：正整數M是正整數N的digit-sum時，則N為M的generator

如245是256的generator

如1979是2005的generator，因為 $1979+1+9+7+9$ 等於2005

如198及207皆是216的generator，而198是216最小的generator

* 模組程式應能接受使用者由標準輸入裝置所輸入的資料

* 模組程式中應控制使用者輸入資料的正確性與合法性，如輸入資料非正整數，程式應顯示輸入資料不正確的訊息，然後結束程式

* 輸入資料無誤，執行程式後，應顯示該輸入資料的所有generator(s)

試題評析

本試題是一個普通的邏輯問題，試題內定義generator的意義，也就是數值本身加上數值內所有的數字總和。因此，只要搭配迴圈就可以完成本程式，考生只要小心作答就可以獲取高分。

答：

```
void generator(){
    int num, digits=0, i, n, digit_sum=0;
    printf("enter a positive integer.");
    scanf("%d", &num);
    if(num<=0){
        printf("the number is not positive.\n");
        return;
```

```
}  
digits=0; n=num;  
while(n){  
    n/=10;  
    digits++;  
}  
  
for(i=num-1; i>=num-10*digits; i--){  
    digit_sum=i, n=i;  
    while(n){  
        digit_sum += n%10;  
        n/=10;  
    }  
    if(num==digit_sum)  
        printf("generator: %d\n", i);  
}  
}
```

高
點
·
高
上

【版權所有，重製必究！】