

《程式語言》

試題評析	<p>今年地方特考程式語言都以基本分為主，包括BNF、物件導向的多型與繼承及例外處理。另外雖有雲端運算概念的HDFS、MapReduce及No SQL，但都屬於MIS或資料庫中常見的概念，同學若有完整準備，應也不難拿分。</p> <p>值得一提的是，這次的物件導向題目整合繼承與多型概念，並要完全了解多型概念，才能順利改寫考題中的程式碼，因此難度應為整張考卷中最高者，此外，也出現了幾年不見的UML題目，要拿到完整分數不算容易。</p> <p>綜觀而論，今年題目普通考生應可以拿到60分以上，程度好的考生拿到90分也絕非難事。</p>
-------------	---

一、定義下述文法(grammar)：

expression \rightarrow expression + expression | expression - expression | type

type \rightarrow int | double

(一)根據上面之文法，請畫出int+int+int兩個不同的剖析樹(parsing tree)。(10分)

(二)根據上面之文法，請導出所有只包含2個type之expression的結果。該expression一個type為int，另一type為double。(10分)

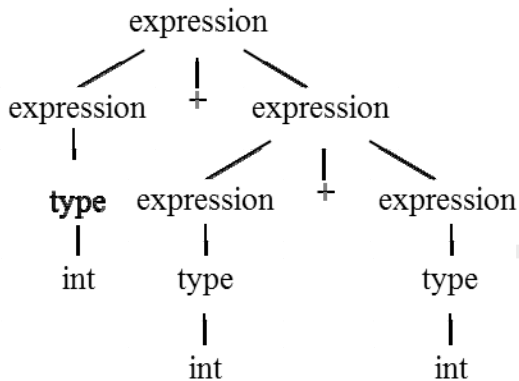
(三)請說明何謂模稜兩可的文法(ambiguous grammar)?(5分)

考點命中	<p>《高點程式語言講義》第六回，金乃傑編撰，頁6-36。</p> <p>《高點程式語言總複習講義》，金乃傑編撰，考點27，頁78-82。</p> <p>《高點103地特考猜程式語言》，金乃傑編撰，考點8，頁17-18。</p>
-------------	--

答：

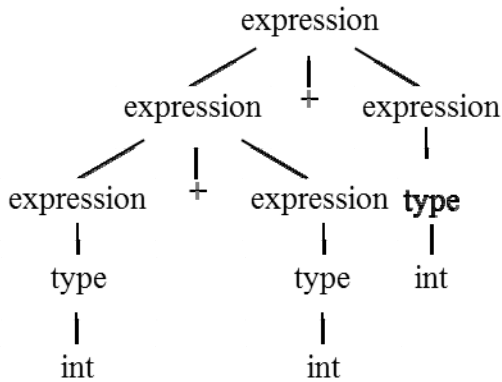
(一)兩棵不同的剖析樹如下：

剖析樹一：



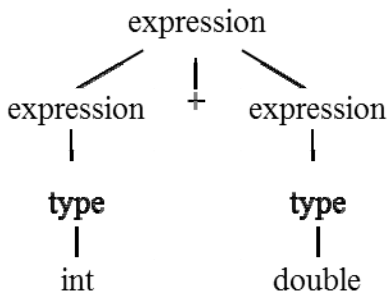
剖析樹二：

【版權所有，重製必究！】

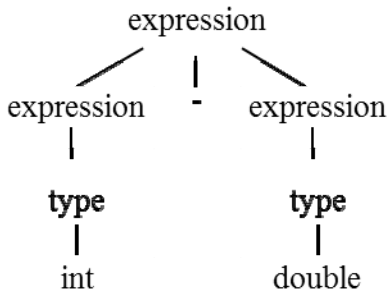


(二) 只包含2個type為int與double的剖析樹共2棵，如下：

第一棵：



第二棵：



(三) 模稜兩可的文法 (ambiguous grammar)：指一語句可畫出兩棵0以上剖析樹的BNF文法。如題目中文法若要表達int+int+int可以畫出兩棵剖析樹來表示 (第一子題)，則該文法便屬之。

二、網際網路程式與應用的發展，結合了許多技術與分析方法，請用數行文字，說明下列幾個網際網路應用技術的專有名詞：

- (一) 說明HDFS英文全名，其架構元素與基本運作特性。(8分)
- (二) 說明MapReduce的特性及Map和Reduce的運作方式。(8分)
- (三) 說明NO SQL的英文全名及其特性。(9分)

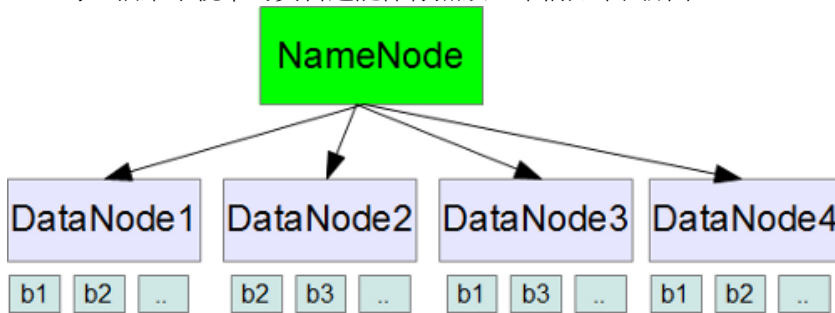
考點命中

《高點資訊管理與資通安全講義》第一回，金乃傑編撰，頁21。
 《高點資訊管理與資通安全講義》第一回，金乃傑編撰，頁28。

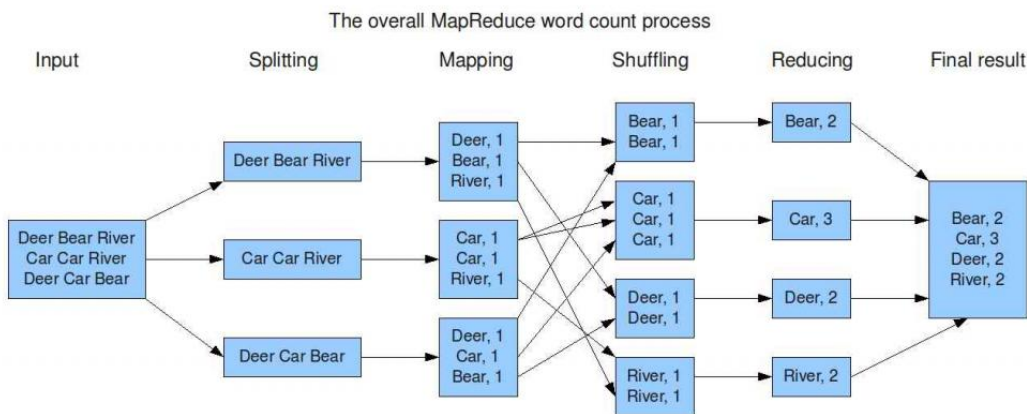
答：

(一) HDFS：全名為：Hadoop Distributed File System，是Hadoop的三大主要元件之一，提供Hadoop的檔案儲存

系統。其功能是将分散的儲存資源整合成一個具容錯能力、高效率且超大容量master/slave架構的儲存環境，由Name node及Data nodes組成，Name node儲存檔案屬性權限等資訊；資料檔會被切割成數個較小的區塊儲存在不同的Data node上，每一個區塊還會有數份副本存放在不同節點，這樣當其中一個節點損壞時，檔案系統中的資料還能保存無缺。架構如下圖所示：



(二) MapReduce：是Google提出的一個軟體架構，用於大規模數據集的平行運算。顧名思義包含Map（映射）和Reduce（化簡）兩個概念。Map函數把一組Key-Value映射成一組新的Key-Value，再由Reduce函數對重排後的Key-Value進行化簡。由於每個動作都可以分散在不同的電腦中完成，因此可以藉由平價的電腦整合產生強大的運算能力。其運作方式如下圖所示：



該程式使用MapReduce計算一input文件每個單字的出現次數。當系統將文件切為不通區段後，交由不同的電腦進行Mapping運算，計算每個區段中單字個別的出現次數；最後再由Reducing進行化簡，將相同單字的出現次數進行加總，快速計算出文件每個單字的總出現次數。

(三) NO SQL：英文全名是Not Only SQL，指的是不同於傳統的關聯式資料庫的資料庫管理系統。NoSQL的初衷是成為現代網路規模的資料庫，主要目的為建立非關聯式、分散式、開放原始碼和平行擴展的資料。NoSQL一詞最早出現在1998年，是Carlo Strozzi開發的一個輕量、開放原始碼和不提供SQL功能的關聯式資料庫。而NoSQL運動正式開始於2009年初，並迅速發展起來，如Google BigTable、Amazon S3的Dynamo資料庫，或是MongoDB。根據nosql-database.org統計，目前NoSQL資料庫大約有150種，並可以分成5種類型：簡單鍵值存儲（Key-Value）、列存儲（Column Family）、文件存儲（Document Stores）、圖形資料庫（Graph Database），其共同特色如下：

1. 模式自由（schema-free）：不需要像關聯資料庫在一開始時就定義schema，因此不會因為需要擴展schema時造成困擾。NO SQL在需要時新增schema即可，而且不需要每筆資料都使用相同的欄位來儲存。
2. 支援簡易複製（easy replication support）：很容易將資料複製到多個資料節點，不需要像關聯式資料庫進行複雜的同步設定。

3. 提供簡單API (simple API)：讓應用程式容易串接，如直接與json格式對應，方便網頁應用程式讀取資料。
4. 最終一致性 (eventually consistent)：使用BASE模型 (而不是ACID模型)，指不同資料節點的資料在同一時間可能存在不相同，但是最終同步完成後便會使資料相同。
5. 支援海量資料 (huge amount of data)：不使用關聯式儲存，降低隨機讀取記憶體造成的瓶頸，使存取數TB資料時亦有高的執行效能。

三、請以物件導向，多型 (polymorphism) 的技術，改寫下列計算薪水的程式，其中經理 (manager) 薪水是40000，工程師 (engineer) 薪水是35000。

```
int getPay(String employeeType){
    if(employeeType=="manager"){
        return 40000;
    }
    else if(employeeType=="engineer"){
        return 35000;
    }
}
```

(一)設計Employee, Manager, Engineer等三個類別及其屬性 (attribute) 與方法 (method)，方法包括建構子 (constructor)。其中Employee是抽象類別，有一個保護 (protected) 屬性薪水 (salary)；Manager和Engineer是Employee的子類別。(18分)

(二)畫出UML類別圖。(7分)

考點命中

《高點程式語言講義》第三回，金乃傑編撰，頁31-59。
 《高點程式語言總複習講義》，金乃傑編撰，考點18、19、21。
 《高點103地特考猜程式語言》，金乃傑編撰，考點5、考點6。

答：

(一)以Java撰寫如下：

【版權所有，重製必究！】

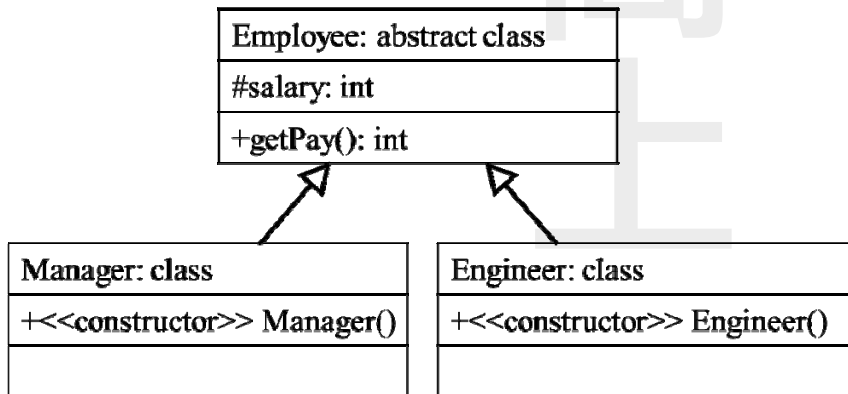
```

1 package PL104;
2
3 abstract class Employee{
4     protected int salary; //薪水屬性
5
6     public int getPay(){ //改寫為多型
7         if(this.getClass() == Manager.class) return 40000;
8         else if(this.getClass() == Engineer.class) return 35000;
9         return 0;
10    }
11 }
12
13 class Manager extends Employee{
14     Manager(){}; //建構子
15 }
16
17 class Engineer extends Employee{
18     Engineer(){}; //建構子
19 }
20
21 public class Exam_34460_3 {
22     public static void main(String[] args) {
23         Employee em[] = new Employee[2];
24         em[0] = new Manager(); //經理物件
25         em[1] = new Engineer(); //工程師物件
26
27         for(Employee ee: em){ //以迴圈檢查薪水
28             System.out.println(ee.getPay());
29         }
30     }
31 }

```

輸出：
40000
35000

(二) UML類別圖如下：



四、例外 (exception) 處理的應用：

(一) 試說明Java對於例外處理的抓取或宣告規則 (catch or declare rule)。(7分)

(二) 填寫下列空格，完成以下inputScore方法。該程式要求使用者輸入成績，若輸入為非數字，則拋出例外 (exception) 警示，迴圈繼續，直到使用者輸入合理的數字成績。若輸入的成績沒有介於0-100間，則把例外向上拋給呼叫的方法。(18分)

```

public static void inputScore () ____ (1) ____ {
    String str;
    boolean valid;

```

```

double score = 0;
Scanner sc = new Scanner(System.in);
do {
    valid = true;
    System.out.println("Input the score");
    str = sc.next();
    try {
        score = Double.parseDouble(str);
    } catch (____ (2) ____ ) {
        System.out.println(e);
        valid = false;
    }
} while (____ (3) ____);
if (score > 100 || score < 0) throw ____ (4) ____ ("分數未介於0-100間");
System.out.println(score);
}

```

考點命中

《高點程式語言講義》第三回，金乃傑編撰，頁63-68。
 《高點程式語言總複習講義》，金乃傑編撰，考點14，頁36-38。
 《高點103地特考猜程式語言》，金乃傑編撰，考點3，頁6。

答：

(一) 例外處理指的是當系統偵測到軟體或硬體引發無法執行正常指令的事件（例外）所進行的程式單元（exception handler），使程式回復到可以繼續執行的狀態稱之。在Java語言中，可以透過try...catch抓取例外。在try中撰寫可能會出現例外的程式碼，使用catch抓取例外，一個try...catch結構中可以使用多個catch來抓取不同情況的例外，當例外被catch抓取後，會執行catch區段中的敘述，執行完後會跳出整個try...catch結構。如下所示：

```

try{
    //可能產生例外的程式碼
}catch(抓取例外一){
    //處理例外一的程式碼
}catch(抓取例外二){
    //處理例外二的程式碼
}

```

值得一提的是，當例外物件被catch接到後，便不會再被後面的catch接住，所以當例外物件有繼承關係時，要把所有例外的父類別Exception放在最後一個catch，以免其他catch都無法發揮效用。此外，若拋出例外的程式碼外面沒有try...catch結構，例外可以往函數的呼叫者拋出，只要在呼叫函數的地方有try...catch結構，則例外一樣可以被catch接住。

(二)填寫如下：

1. throws Exception
2. Exception e
3. valid == false
4. new Exception

【版權所有，重製必究！】