this (): 呼叫此類別人既緣权)的 construction super()=分析文類的(無學報)自Foorsivictor 2. Fine 1) SE & F Th 22015 DAVE SINDING private = & A B dass + J to IR. T. + exa Class = 一般的難別。可象体化。不可有·克 6. d. Mit Bush Till Food a sa t. honson abstract class: 木丁被軍大大は、引汗するより取分が会 9. X.F town ) - testart () my Thread Time O dientes Three outs 10 X 6 hrani 2 4 16 8 16 1 x 9 1. Overriding: 繼承父類別的 method 之後,以相同名字、 相同缘软(type. 较量),相容的回伝值,只改 写 body 内容。(存取权限只能相同或效大) ex: protected test() {/-} > public test() {//不同內容} Overloading:在同一類列中,相同的method名,且多较一 定要不同(type、较量) ex: public test (int x) { ... } so him or long public test (double y) { -- } public test (int a, int b) { ... }

- 2. this():呼叫此類別(無緣較)的 constructor. super():呼叫父類別(無緣較)的 constructor.
- 3. public:任何 class 都可存取, protected:同 packege,或不同 packege 的子類別可存取。 default:同 packege 的 class 皆可存取。 private:只有同 class 才可存取.
- 4. class:一般的類別、可案体化、不可有abstract method. abstract class:不可被案体化、可同時有一般知抽象方法 interface: 其中的variable一定当public static final、 method 不能有body.由之後 implements此 interface 的 class 实体。

IV.

1. t. run() -> t. start() myThread implements Thread {
extends

2 B. java =

public class B extends A, Object &

public B (String name) {

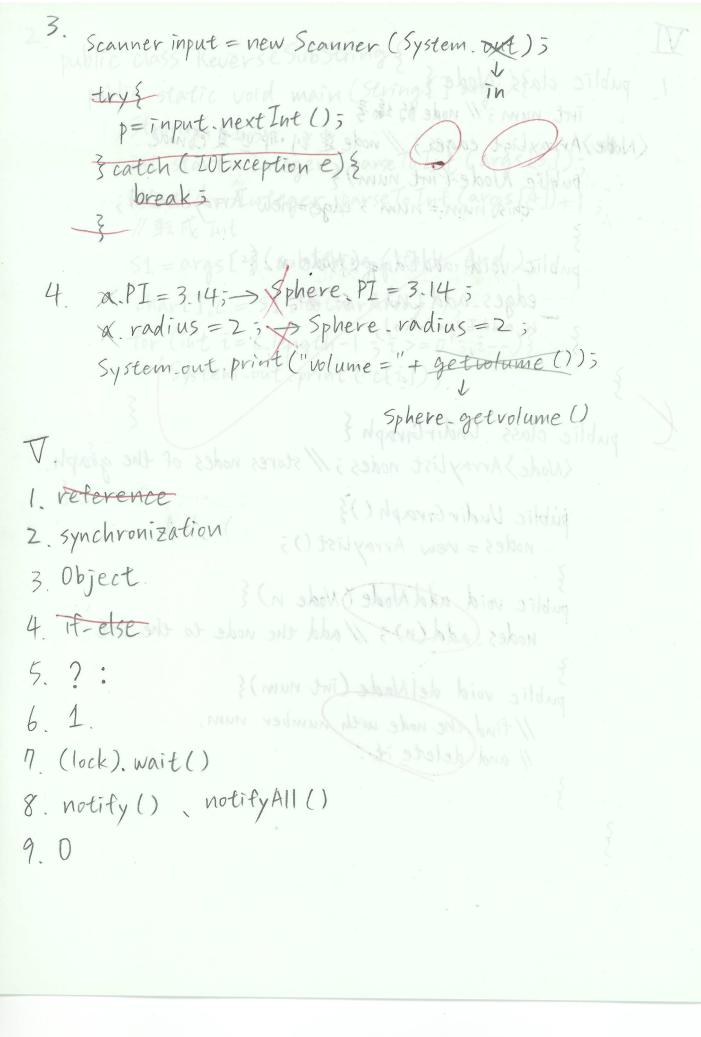
Super (name);

D-java: implements

public class D extends C {

public - private int get { region

public void set (int p) { }



```
public class Node {
      int num; // node 的编号
 (Node) Array list edges 5 // node 連到哪些其它 node.
 public Node (int num) {
           this num = num ; edges = new Array List ();
       public void add Edge (Node n) {
           edges.add (n); < 26
           h. add Edge (+his);
   public class Undir Graph }
       (Node) Array list nodes; // stores nodes of the graph.
    public Undir Graph () {
           nodes = new Arraylist ();
       public void add Node ( Node n) {
           nodes (add (n) 5 // add the node to the list.
        public void del Node (int num) {
           If find the node with number num,
           11 and delete it
```

```
public class Reverse Sub String {

public static void main (String[] avgs) {

String $1;

int start = Integer. parse To Int (args[3]);

int end = Integer. parse To Int (args[4])+1;

// $\frac{1}{2} \text{ Int}

$1 = args[2]. sub String (start, end);

char[] c = $1. to CharArray();

for (int i = c.length-1; i >= 0; i--) {

$ $ystem.out.print(c(i));

}
```