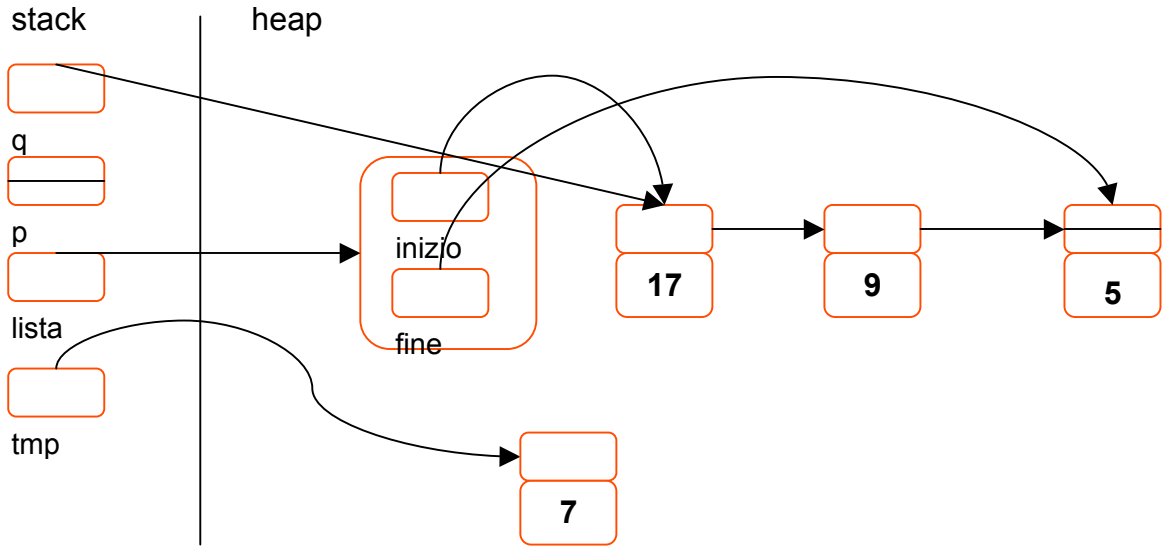
Corso di laurea in Informatica AA 2006/07 Programmazione, M.A. Alberti

Pre-appello 2006/07 / VI Appello 2005/06 - 29 gennaio 2007

Cognome **Nome**

Matricola

1



Osservate la situazione dello stack delle chiamate dei metodi e dello heap schematizzato dalla figura, in cui si fa riferimento alla classe `ListaOrdinata` con la classe interna `NodoLista`:

```
public class ListaOrdinata {
    private NodoLista inizio, fine;

    private static class NodoLista {
       Comparable dato;
       NodoLista pros;
    }    ……
}
```

La variabile `lista` sullo stack e' il riferimento a una lista dopo un certo numero di inserimenti. Definite e inizializzate le due variabili `p` e `q` usate come indice di scorrimento della lista:

e le due istruzioni per far avanzare gli indici sulla lista:

Disegnate inoltre la struttura dopo aver inserito il nodo con il dato `7` e lo stato delle variabili al termine. Si noti che la lista e' ordinata in senso decrescente. Le variabili `p` e `q` servono a individuare la posizione esatta in cui inserire il nuovo nodo `tmp` e non perdere le informazioni sui legami.

2

Dato il metodo `int f(int, int, int)` seguente, tracciare lo stato dello stack delle chiamate dei metodi durante l'esecuzione di `f(6, 0, 1)` (si ricodi che `rit` indica l'indirizzo e `val` il valore di rientro)

```
int f(int n, int f0, int f1) {
  if (n==0)
    return f0;
  else if (n==1)
    return f1;
  else
    return (f(n-1, f1, f0+f1));
}
```

n									
f0									
f1									
val									
rit									

Record 1 Record 2

Quale valore riporta la chiamata di `f(6, 0, 1)`?
Quante chiamate ricorsive vengono effettuate?

3

Scrivere un metodo per computare **iterativamente** la funzione di Fibonacci `f(n)`, dove `n` e' un numero intero non negativo, definita come segue:

```
       | 0 per n = 0
f(n)=  | 1 per n = 1
       | f(n-1) + f(n-2)) per n > 1
```

Calcolare `f(6)`:

4

funzione ricorsiva numeri Fibonacci

```
int f(int n) {
  if (n==0 || n==1)
    return n;
  else
    return (f(n-1)+f(n-2));
}
```

1. Numero delle chiamate ricorsive per calcolare f(6) ?
2. Disegnare l'albero delle chiamate

5
Esiste una relazione tra la funzione ricorsiva dell'esercizio **4** e quella dell'esercizio **2**?

Cosa si può concludere?

6
Data l'espressione aritmetica (2 + 6) / (2 * 2) + 5 * 3 trascriverla in forma postfissa:

Quindi valutare l'espressione facendo uso di una struttura a stack.
Disegnare l'andamento dello stack durante la valutazione dell'espressione nella griglia proposta (ogni colonna rappresenta una fotografia dello stack ad un passo della computazione, il fondo dello stack sia in basso)

7
Dato il seguente spezzone di codice:

```
public class Giochi {
   private int pedine;
   public int muovi() {
      return pedine--;
   }
}
```

```
public class Scacchi extends Giochi {
   int scacchiera;

}
```

Dire se le seguenti istruzioni sono lecite o no:

a. `Giochi g = new Scacchi();` SI NO

b. `Giochi f = new Giochi(); int p = f.muovi();` SI NO

c. `Scacchi s; int p = s.muovi();` SI NO

d. `g.scacchiera = 8;` SI NO

In caso di risposta NO, correggete, se possibile, le istruzioni per renderle lecite:

a.

b.

c.

d.

8
In una implementazione dinamica della struttura dati stack, definita come segue

```
public class Stack  {
   private NodoStack cima;
   private class NodoStack  {
      Object dato;
      NodoStack pros;
   }

   public Stack()  {
      cima = null;
   }
```

```
public void push(Object o)  {
      NodoStack t = new NodoStack();
      t.dato = o;
      t.pros = cima;
      cima = t;
}
```

Implementare la funzione `top()` che accede al valore del nodo in cima allo stack e lo riporta all'ambiente senza modificare la struttura e un'eccezione non controllata `EmptyStackException()`.

Si scriva una istruzione `try-catch` in cui si cattura l'eventuale eccezione lanciata dal metodo `top()` nel contesto di un programma in cui sia stata introdotta la variabile `pila`, onde poter recuperare e inserire almeno un nodo. Gli oggetti da inserire nella pila siano di classe `Integer`.

```
Stack pila = new Stack();
```

9

Avendo definito `class Settimana implements java.util.Iterator { };` dire se sono lecite le seguenti istruzioni:

a. `Settimana s = new Settimana();`

   `while(s.hasNext()) <computa s.next()>` SI NO

b. `Iterator i = new Settimana();` SI NO

c. Qual'è il concetto della programmazione a oggetti per cui diversi oggetti che hanno una interfaccia comune rispondono in modo diverso quando un metodo di quell'interfaccia è invocato:

   ……..

d. Casting è appropriato quando si riceve un riferimento a una classe antenato e si sappia che l'oggetto è una particolare sotto-classe e si voglia usare l'oggetto nella sua funzionalità completa. SI NO