

## Anteprima di test

parte pratica 20 Gennaio 2015

Data: Mon Jan 26 12:55:31 2015 Punteggi massimi: 40

### 1. Java visitor pattern (5 Punti)

---

Una espressione (numerica) può essere o un numero (NumberLiteral) o la somma di due espressioni o il prodotto. Scrivi le classi per rappresentare le espressioni di questo tipo.

Si vuole realizzare un valutatore di espressioni (cioè data una espressione calcola il valore numerico). Realizzano in due modi:

1. con un metodo dell'espressione value che resituisce il valore
2. mediante il visitor design pattern

Scrivi un main in cui costruisci l'espressione (costruendo gli opportuni oggetti) "3+(4\*5)". Calcola il suo valore usando l'approccio 1 e 2.

Crea un progetto in eclipse con nome COGNOME\_NOME\_JAVA. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome COGNOME\_NOME\_JAVA.zip. Consegna il file ZIP qui:

(5 Punti)

### 2. Dangling pointer in C con free (4 Punti)

---

Fa un esempio in C in cui hai un dangling pointer dovuto all'uso sbagliato di free. Costruisci uno scenario in cui riesci ad usarlo ugualmente e poi ti accorgi di qualche errore.

Crea un progetto in eclipse con nome COGNOME\_NOME\_DP. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome COGNOME\_NOME\_PP.zip. Consegna il file ZIP qui:

(4 Punti)

### 3. Puntatore a puntatore in C - int (4 Punti)

---

Fa un esempio in C in cui usi il puntatore a puntatore per definire un funzione del1Int che serve per togliere il primo intero in un puntatore a buffer di interi (int\*). Nota che del1Int dovrà modificare il puntatore e non il buffer. Esempio se ho int a[] = {1,2,3}; int\* p = a; e faccio del1Int(p), p punterà a {2,3} ma a sarà invariato.

Crea un progetto in eclipse con nome COGNOME\_NOME\_PP. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome COGNOME\_NOME\_PP.zip. Consegna il file ZIP qui:

(4 Punti)

### 4. definizione di funzione in SCALA - contaPari Collezioni (6 Punti)

---

- Scrivi una funzione contaPari che data una lista di interi (List[Int]) restituisce il numero di interi pari in essa. Puoi fare diverse versioni (in ordine di difficoltà):
  - contandoli con un ciclo for semplice (for (n <- nums))
  - contandoli ma usando un foreach (nums.foreach(countPar)) e una funzione countPar
  - in modo ricorsivo (usa isEmpty head e tail)
  - usando folderLeft, del tipo (nums.foldLeft( ...)(...))
  - usando filter (nums.filter ..)
- Scrivi una higher order function contaP che generalizza il conteggio con un predicato generico P (funzione da int a boolean) che dato un intero x mi dice se x soddisfa P. contaP prede una lista di interi (List[Int]) e conta il numero di interi nella lista che soddisfano P. Usa il currying se riesci.
- Riscrivi (con nome contaPari2) la funzione contaPari usando contaP (sia con funzione anonima per sapere che un numero è pari sia con funzione esplicita).
- Scrivi un po' di chiamate definendo un Object e usando lo schema seguente:

```
1 object prova {
2
3   def ....
4
5   def main(args: Array[String]) {
6     println( ....)
7   }
8 }
```

**Crea un progetto in eclipse con nome COGNOME\_NOME\_SCALA. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome COGNOME\_NOME\_SCALA.zip. Consegna il file ZIP qui:**

**(6 Punti)**

## Introduction to Functional Programming with Scala

Angelo Gargantini

INFO 3A AA 2013/14  
credits: Pramode C.E  
<https://class.coursera.org/progfun-00>

December 9, 2013

### Workshop Plan

Here is what we will do:

- Learn a bit of functional programming in Scala
- Learn some important concepts like (NOT ALL): closures, higher order functions, purity, lazy vs strict evaluation, currying, tail calls/TCO, immutability, persistent data structures, type inference etc!

Workshop material (slide/code samples) sul sito.

### 5. Tipi opachi in C - Cane (4 Punti)

**Fa un esempio in C in cui usi un puntatore a tipo opaco per realizzare un'implementazione di un Cane. Aggiungi il campo nome e il metodo abbaia (che stampa <nome>:BAU). Crea un metodo creaabbaia (nel modulo principale quello col main) che crea una cane con nome "fido" e lo fa abbaire.**

**Nel un main chiama creaabbaia 2 volte. Assicurati di non avere memory leak.**

**Crea un progetto in eclipse con nome COGNOME\_NOME\_OP. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome COGNOME\_NOME\_OP.zip. Consegna il file ZIP qui:**

**(4 Punti)**

### 6. Ricorsione - C - contaPositivi (12 Punti)

**Scrivi una funzione contaPositivi in C che dato in ingresso un array di interi restituisce il numero di array positivi (>0) in esso.**

**Al solito, scrivi tre versioni: una non ricorsiva, una ricorsiva senza tail recursion e una ricorsiva con tail recursion.**

**Specifica esattamente i parametri che passi alla procedura, il tipo di passaggio utilizzato e il loro significato. Definisci funzioni ausiliarie di aiuto se necessario, per tenere la segnatura della funzione contaPositivi più semplice.**

**Scrivi anche un main di esempio in cui chiami la funzione con l'array {1,-1} invocala in modo assegni il risultato ad una variabile locale al main POS. Non usare alcuna variabile globale.**

**Disegna il record di attivazione per tutte e tre le versioni fino alla massima estensione del record di attivazione. Nel caso di tail recursion, spiega nel codice quali ottimizzazioni hai adottato o potresti adottare.**

Leggi le istruzioni qui:

## Istruzioni per l'esame di info 3 – funzioni ricorsive

### Uso di C

Crea un nuovo progetto C con eclipse. Metti TUTTO dentro il progetto in eclipse (anche il foglio elettronico).

### Record di attivazione

Se vuoi puoi anche disegnare il record di attivazione usando openoffice (oocalc)

Usa questo formato. Ogni record disegnalolo come blocco con bordi di celle così:

Etichetta	Nome variabile	valore	Indirizzo Puntato
foo(x,y) [1]	CL		
	x	6	
	y	7	
[6]	AL		[1]
blockinline	CL		
	...		

Nella prima colonna metti la descrizione del tipo di RA. Metti anche un numero o label se vuoi riferirti a quell'indirizzo

Qui metti descrizione della cella (ad esempio nome variabile, CL ...)

Metti il valore ospitato (se non è un indirizzo)

Metti il valore (o etichetta) se è un indirizzo

Aggiungi liberamente commenti a destra

Se devi fare vedere come evolve, copia, incolla a dex e prosegui in basso:



**Crea un progetto in eclipse con nome COGNOME\_NOME\_RICORSIONE. metti nel progetto il foglio openoffice. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome COGNOME\_NOME\_RICORSIONE.zip. Consegna il file ZIP qui:**

**(12 Punti)**

### 7. Java Lista - uso di generics -ListaComputer (5 Punti)

Scrivi una classe **Computer** (Comparable a seconda del ID intero, suo unico campo) e una classe **Notebook** che estende **Computer**. Usando i **generics** in Java, definisci una **ListaComputer** che rappresenta una lista di tipi generici T che siano **Computer** o sotto tipi di persone (ad esempio **Notebook**) usando ereditarietà della classe **ArrayList**.

Scrivi una classe **Algoritmi** che ha un metodo (generico) **statico getFirst** che prenderà una **ListaComputer** e restituisce il primo elemento della lista.

Scrivi un **main** in cui dichiari due **ListaComputer<T>** con T computer e T Notebook.

Chiama il metodo **getFirst** per entrambe le liste.

Crea un progetto in eclipse con nome **COGNOME\_NOME\_GEN**. Per la consegna produci un file zip usando la funzione **export archive file** di eclipse. Lo zip abbia nome **COGNOME\_NOME\_GEN.zip**. Consegna il file ZIP qui:

**(5 Punti)**