

Paradigmi di Programmazione – CORSO B

C.d.L. in Informatica – A.A. 2022-2023

Traccia degli argomenti per l'orale

Prof. Paolo Milazzo

Questo documento contiene un elenco dettagliato degli argomenti che potranno essere oggetto di domande durante la prova orale dell'esame del corso.

Questo elenco è **valido solo per il corso B (Prof. Milazzo)**

Attenzione: questo non è un elenco di domande. Le domande potranno entrare più nel dettaglio dei singoli argomenti, potranno richiedere di fare collegamenti tra i vari argomenti, potranno richiedere allo studente di formulare esempi dei vari concetti, oppure potranno richiedere allo studente di spiegare un concetto sulla base di un esempio dato dal docente.

Lambda-calcolo

1. Origini del λ -calcolo, rapporto con Macchine di Turing e nascita dei paradigmi imperativo e funzionale
2. Sintassi delle λ -espressioni, variabili libere e legate, α -equivalenza
3. Valutazione di λ -espressioni, capture avoiding substitution, β -riduzione e β -equivalenza
4. Confluenza e non terminazione, Teorema di Church-Rosser e la λ -espressione Ω
5. λ -calcolo e programmazione funzionale, funzioni Curryed e combinatore Y, riduzione *call-by-name* e *call-by-value*

Type Systems

1. Motivazioni per il controllo dei tipi e definizioni di *sistema di tipi*, *type safety*, *controllo dei tipi statico* e *controllo dei tipi dinamico*
2. Sistema di tipi per le espressioni: funzionamento generale, precisione e composizionalità, proprietà di progresso e conservazione
3. Lambda-calcolo tipato semplice: funzionamento generale, ambiente dei tipi e suo ruolo nei giudizi di tipo, proprietà di progresso e conservazione

Sviluppo di interpreti (e compilatori)

1. Compilazione, interpretazione e approccio misto
2. Il concetto di macchina astratta, le sue componenti e tipiche implementazioni
3. il Run-Time Support
4. Struttura di un compilatore e le varie fasi della compilazione
5. Introduzione allo sviluppo di interpreti (componenti principali e semantica SOS come base dell'implementazione)

Paradigma Object-Oriented

1. Nozioni preliminari: record e polimorfismo di sottotipo (subsumption di record e funzioni)
2. il concetto di Abstract Data Type e l'esempio dei moduli in OCaml
3. Definizione di oggetto e sue caratteristiche
4. Concetti di OOP: incapsulamento, astrazione, interfaccia, ereditarietà, sostituzione, polimorfismo
5. Object-based vs class-based, structural-(sub)typing vs nominal-(sub)typing
6. OOP in OCaml (a grandi linee)
7. Basi del linguaggio Java (a grandi linee), incapsulamento, interfacce e tipo apparente vs effettivo
8. Modello della memoria della JVM, suo funzionamento e collocazione dei membri statici e d'istanza
9. Il concetto di aliasing in Java
10. Ereditarietà in Java, overloading, overriding e dynamic dispatch con le tabelle dei metodi e lo sharing strutturale
11. Ereditarietà multipla: i problemi e le soluzioni di C++, Java e Python
12. Il concetto di mixin (a grandi linee)
13. Il principio di sostituzione di Liskov: concetto generale, relazione con le specifiche della classe, problemi di indecidibilità e regole indotte (della segnatura, dei metodi e delle proprietà)
14. Generici in Java: funzionamento generale (a grandi linee) e motivazioni per la non covarianza, type erasure
15. Java Collection Framework: funzionamento generale (a grandi linee), concetto di iteratore e suo funzionamento e utilizzo

Garbage collection

1. motivazioni e problematiche, reference counting, mark-and-sweep, copy collection e approcci generazionali

Programmazione concorrente

1. Concetto di esecuzione non sequenziale, concorrenza e parallelismo
2. Interleaving e motivazioni per la sua analisi a livello del linguaggio macchina
3. Shared-memory vs message-passing, thread e processi in un tipico scenario di concorrenza
4. Un linguaggio concorrente minimale e problematiche di concorrenza in caso di assenza di meccanismi di sincronizzazione
5. Estensione del linguaggio con primitive di mutua esclusione: scopo, funzionamento, coarse-grained vs fine-grained locking, deadlock e modi per risolverlo (es. 2PL)
6. Thread Java: funzionamento generale (a grandi linee) e primitive di sincronizzazione (metodi e blocchi sincronizzati, collezioni sincronizzate)