

Binding times

(Momenti di legame e legami di un L.P.)

Corso di Compilatori

Prof. Costagliola

Tratto da questo [riferimento](#) estratto dal testo:

[Programming Languages: Design and Implementation \(4th Edition\)](#)

Binding times ed attori

- Durante i vari «momenti» (binding time) **del ciclo di vita** di un linguaggio di programmazione (a partire dalla definizione del linguaggio fino all'esecuzione dei suoi programmi) ciascun elemento del linguaggio è «legato» (tramite binding) a delle «caratteristiche» da parte di un certo «attore».
- Momenti del ciclo di vita di un linguaggio di programmazione **compilato**:
 1. Definizione del linguaggio
 2. Implementazione del linguaggio (ovvero, scrittura del compilatore)
 3. Uso del linguaggio
 - a. Scrittura di un programma
 - b. Compilazione/Traduzione di un programma (compile time)**
 - c. Linking/caricamento di un programma oggetto
 - d. Esecuzione del programma eseguibile (run time)**
- Attori
 - Creatore del linguaggio (1.)
 - Sviluppatore del compilatore (2.)
 - Programmatore (3.a)
 - Compilatore/Traduttore (3.b)
 - S.O. (3.c)
 - Processore (3.d)

NOTA: *«legare un elemento ad una caratteristica»* si traduce in *«effettuare una scelta per quell'elemento»*

NOTA: nel libro anche la *scrittura del programma* ed il *linking/loader* vengono considerati come *«compile time»*

Bindings

- Ad ogni **momento**,
 - **elementi** di un linguaggio di programmazione o di un suo programma sono **legati** a delle **caratteristiche** e
 - delle **scelte** vengono **definite**.
- Esempi:
 1. la parola chiave *if* è legata (bound) al concetto di *condizione* a tempo di definizione del linguaggio.
 2. La scelta di dove memorizzare i parametri di una funzione è fatta a tempo di implementazione di un linguaggio
 3. Il legame fra una variabile ed il suo tipo è **definito** a tempo di scrittura del programma (in linguaggi tipati senza inferenze di tipo) e **realizzato** (bound) a tempo di traduzione (analisi semantica)
 4. Il legame fra una variabile ed il suo indirizzo relativo è **realizzato** (bound) a tempo di traduzione
 5. Il legame fra una variabile ed il suo indirizzo assoluto è **realizzato** (bound) a tempo di caricamento del programma in memoria
 6. Il legame fra una variabile ed il suo valore è **realizzato** (bound) a tempo di esecuzione

Esercizio

- **Momenti di un linguaggio di programmazione compilato:**
 1. Definizione del linguaggio
 2. Implementazione del linguaggio
 3. Uso del linguaggio
 - a. Scrittura di un programma
 - b. **Compilazione/Traduzione di un programma (compile time)**
 - c. Linking/caricamento di un programma oggetto
 - d. Esecuzione del programma eseguibile (run time)
- **Attori**
 1. Creatore del linguaggio
 2. Sviluppatore del compilatore
 3. Programmatore (3.a)
 4. Compilatore/Traduttore (3.b)
 5. S.O. (3.c)
 6. Processore (3.d)

- Data l'istruzione $X = X + 10$ individuare quando queste scelte e legami vengono definiti e da chi.
 1. Scelta dei possibili tipi primitivi di X: - a tempo di ... - attore
 2. Scelta del particolare tipo di X:
 3. Realizzazione* di legame (binding) fra X ed un particolare tipo (in linguaggi compilati):
 4. Realizzazione di legame (binding) fra X ed un particolare tipo (in linguaggi interpretati):... ..
 5. Scelta di quali valori possa assumere X se intero:
 6. Realizzazione di legame (binding) fra X ed il suo valore:
 7. Realizzazione di legame (binding) fra 10 e la sua rappresentazione in memoria (decimale, ottale, etc.):
 8. Scelta di quanti bits utilizzare per rappresentare 10:
 9. Realizzazione di legame (binding) fra l'operatore + ed i suoi possibili significati:
 10. Realizzazione di legame (binding) fra l'operatore + ed il codice che lo implementa:

Per le risposte studiare il riferimento indicato nella prima slide

*Per «realizzazione» si intende «messa in pratica», «effettiva implementazione»

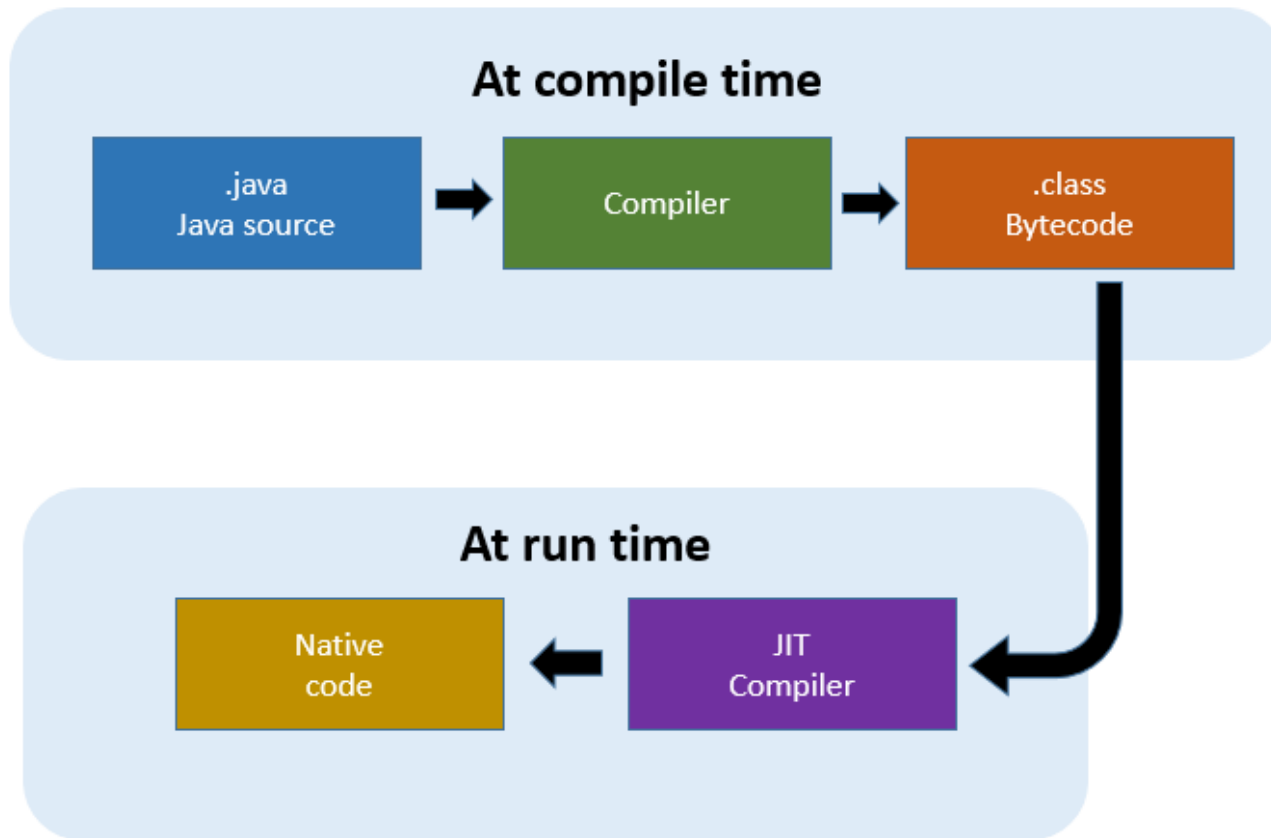
Comprehensible?

In summary, for a language like Pascal the symbol $+$ is bound to a set of *addition operations* at language definition time, each addition operation in the set is defined at language implementation time, each particular use of the symbol $+$ in a program is bound to a particular addition operation at translation time, and the particular value of each particular addition operation for its operands is determined only at execution time. This set of bindings represents one choice of possible bindings and binding times typical of a variety of programming languages. Note, however, that many other bindings and binding times are also possible.

Perchè parliamo di bindings?

- Per l'analisi ed il confronto dei linguaggi di programmazione
- Efficienza -> early binding (a tempo di compilazione)
 - Linguaggi compilati
 - La maggior parte dei bindings avviene a tempo di compilazione e ne restano pochi a tempo di esecuzione portando a maggiore efficienza
- Flessibilità -> late binding (a tempo di esecuzione)
 - Linguaggi interpretati
 - La maggior parte dei bindings avviene a tempo di esecuzione portando a maggiore inefficienza ma maggiore flessibilità (ad es. utilizzo di una stessa variabile con più tipi)
- I nuovi linguaggi e loro implementazioni stanno eliminando il gap (si veda ad esempio i just-in-time compilers (JIT))

Just in time compiler (JIT) – Java



The Just-In-Time (JIT) compiler is a component of the Java Runtime Environment that improves the performance of Java applications at run time.