Programmazione: un po' di terminologia

Corso di Programmazione 1A, 2021-22

Felice Cardone

L'attività di programmazione

Programmatore

Utente

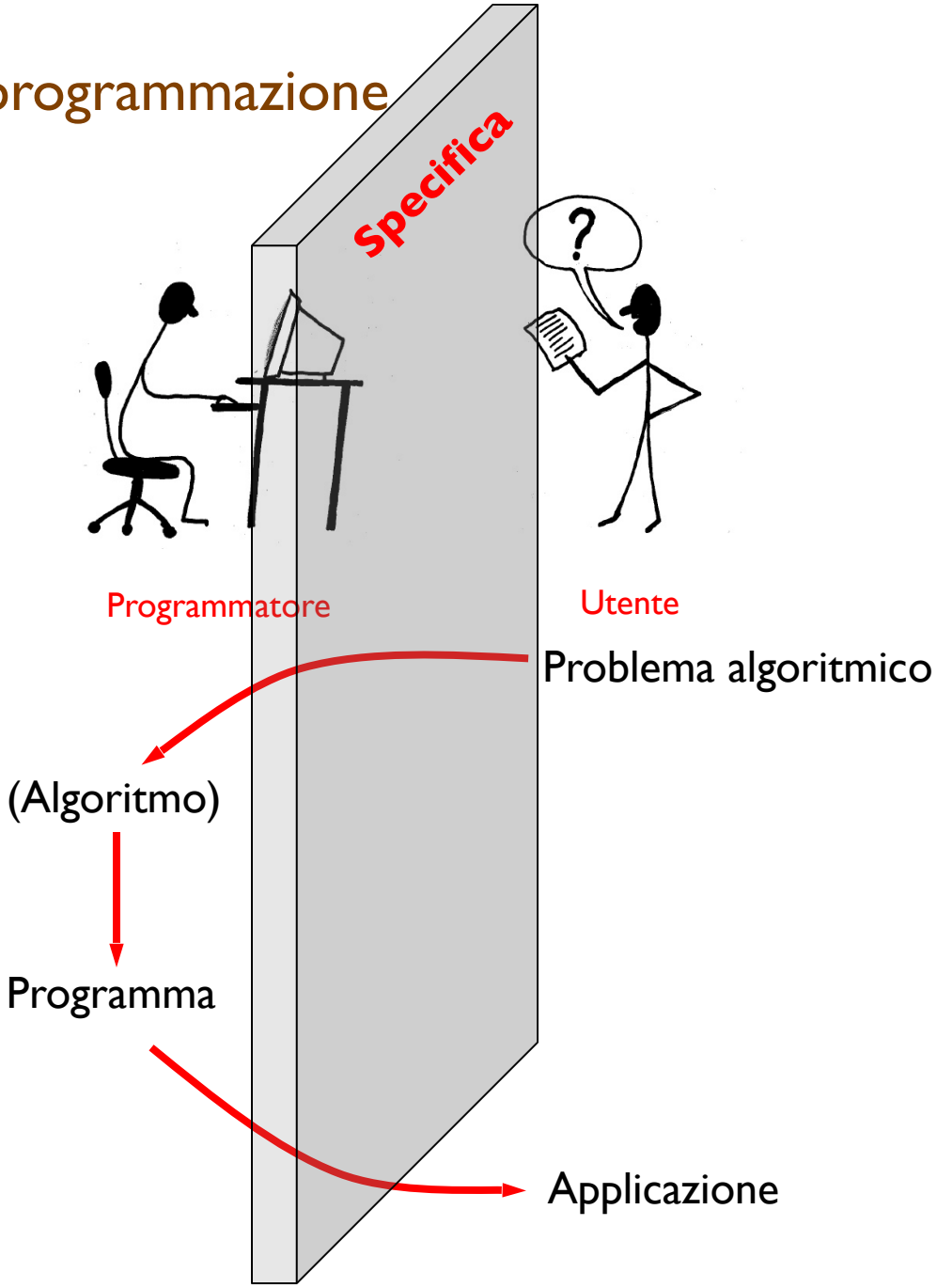
Problema algoritmico

(Algoritmo)

Programma

Applicazione

L'attività di programmazione



Piccolo lessico introduttivo

Specifica: Contratto tra Utente e Programmatore:

> **se** i dati in ingresso soddisfano la condizione di ingresso, **allora** i dati in uscita soddisfano la condizione di uscita

Quindi una specifica definisce: dati in ingresso; dati in uscita; condizione di ingresso (requisito sui dati in ingresso); condizione di uscita (relazione che lega dati in ingresso e dati in uscita)

Esempio (divisione intera)

dati in ingresso: interi ≥ 0 come dividendo N; interi > 0 come divisore D;

dati in uscita: interi $q \geq 0$ come quoziente q; interi $r \geq 0$ come resto;

condizione di ingresso: nessuna;

condizione di uscita: $N = qD + r, r < D$

Piccolo lessico introduttivo

Algoritmo: Piano di azione che prescrive le azioni da compiere per risolvere un problema. Le azioni devono poter essere eseguite in modo meccanico.

Esempio (divisione intera di N per D)

inizialmente q = 0;
inizialmente r = N;
fino a quando r ≥ D
    sottrai D a r;
    aumenta q di 1

Correttezza (parziale) di un algoritmo rispetto ad una specifica: Per ogni dato di ingresso che soddisfa la condizione di ingresso, **se** l'esecuzione termina **allora** i dati in uscita soddisfano la condizione di uscita.

Terminazione di un algoritmo: Per ogni dato di ingresso che soddisfa la condizione di ingresso, l'algoritmo restituisce il risultato dopo un numero finito di passi.

Correttezza (totale) di un algoritmo rispetto ad una specifica: Correttezza parziale + terminazione.

Piccolo lessico introduttivo

Linguaggio di programmazione: Collezione di costrutti componibili mediante regole di sintassi per formare, in particolare:

espressioni, che possono essere valutate (da un calcolatore) per ottenere il loro valore;

istruzioni che possono essere eseguite (da un calcolatore) per modificare lo stato della sua memoria.

Compilatore: Programma che traduce costrutti di un linguaggio di programmazione in costrutti di un altro linguaggio di programmazione di più basso livello.

Interprete: Programma che esegue le istruzioni e valuta le espressioni di un programma.

Piccolo lessico introduttivo

Paradigmi di programmazione:

Programmazione imperativa: un programma consiste di istruzioni che realizzano trasformazioni dello stato della memoria di un calcolatore, dove uno stato è identificato dai valori assunti ad un certo istante da un insieme di variabili;

Esempio: tutto il corso di Programmazione I

Programmazione funzionale: Un programma è un'espressione che viene valutata per ottenere il suo risultato (il valore);

Esempio: `f(5) where f(x) = x * 2` ha valore `10`

Altri esempi: il corso di Linguaggi e Paradigmi di Programmazione (3° anno)

Programmazione orientata agli oggetti: Un programma descrive la dinamica di una collezione di oggetti – esemplari di classi – che comunicano scambiandosi messaggi (= invocando metodi).

Esempio: tutto il corso di Programmazione 2

Memoria e istruzione di assegnamento