

Indice generale

Introduzione	xi
Che cosa avete bisogno di sapere	xi
Come leggere questo libro	xi
Dove trovare il codice sorgente	xii
Di che cosa non parla questo libro	xii
Nota sulla traduzione	xii
Capitolo 1 Per iniziare.....	1
Un piccolo programma	1
Ciao mondo!	2
Stampa di combinazioni di stringhe letterali	4
Stampa di più righe di testo	4
Stampa di numeri interi	5
Commenti	6
Capitolo 2 Calcoli aritmetici	9
Somma di numeri interi.....	9
Altre operazioni tra numeri interi	10
Aritmetica in virgola mobile	11
Sequenze di istruzioni	13
Spezzare le stringhe letterali	13
Capitolo 3 Un nome per ogni cosa	17
Associare un nome ai valori	17
Variabili mutabili.....	19
Variabili mutabili non mutate	20
Variabili non inizializzate	21
Il segno di sottolineatura iniziale	22
Valori booleani	23
Espressioni booleane	25

Coerenza dei tipi negli assegnamenti	26
Inferenza di tipo	27
Cambiamento di tipo e di mutabilità	28
Operatori di assegnamento composti	28
Uso delle funzioni della libreria standard	29

Capitolo 4 Controllo del flusso di esecuzione.....31

Istruzioni condizionali (if)	31
Espressioni condizionali	33
Cicli condizionali (while)	35
Cicli infiniti (loop)	37
Cicli a conteggio (for)	37
Ambiti delle variabili	40

Capitolo 5 Uso di sequenze di dati43

Gli array	43
Gli attributi di Rust	45
Andare in panico	46
Utilizzo di sequenze di dati	47
Array mutabili	48
Array di dimensioni specificate esplicitamente	49
Array multidimensionali	50
Vettori	51
Altre operazioni sui vettori	53
Array e vettori vuoti	55
Stampe di debug	56
Copia di array e vettori	56

Capitolo 6 I tipi primitivi.....59

Basi numeriche non decimali	59
Il segno di sottolineatura nei letterali numerici	60
La notazione esponenziale	61
I vari tipi di numeri interi con segno	62
Tipi per numeri interi non negativi	63
Tipi di numeri interi dipendenti dal target	64
Inferenza del tipo	66
L'algoritmo di inferenza del tipo	67
Tipi numerici in virgola mobile	68
Conversioni esplicite	69
Suffissi di tipo per i valori letterali numerici	70
Tutti i tipi numerici	70
Booleani e caratteri	71
L'ennupla vuota	73
Tipi di array e vettori	74

	Le costanti	75
	Scoprire il tipo di un'espressione	75
Capitolo 7	I costrutti enum e match	77
	Le enumerazioni	77
	Il costrutto match	78
	Operatori relazionali ed enumerazioni.....	80
	Gestione di tutti i casi	81
	Utilizzo di match con i numeri	82
	Enumerazioni con dati	83
	Istruzioni match con variabili nei pattern	86
	Espressioni match.....	87
	Uso delle guardie nei costrutti match	88
	I costrutti if-let e while-let.....	89
Capitolo 8	Strutture di dati eterogenee	91
	Le ennuple.....	91
	Le strutture	92
	Le ennuple-strutture	95
	Convenzioni lessicali.....	96
Capitolo 9	Definizione di funzioni	99
	Definizione e invocazione di una funzione	99
	Funzioni definite dopo il loro utilizzo	100
	Funzioni che oscurano altre funzioni.....	101
	Passaggio di argomenti a una funzione	102
	Passaggio di argomenti per valore.....	103
	Restituzione di un valore da una funzione	104
	Uscita anticipata.....	105
	Restituzione di diversi valori.....	107
	Come modificare una variabile del chiamante	108
	Passaggio di argomenti per riferimento	109
	Utilizzo dei riferimenti	112
	Mutabilità dei riferimenti.....	113
Capitolo 10	Definizione di funzioni e di tipi generici.....	115
	Qual è la necessità di avere funzioni generiche?.....	115
	Definizione e utilizzo di funzioni generiche	116
	Inferenza del tipo dei parametri.....	117
	Definizione e utilizzo di strutture generiche.....	119
	Il meccanismo della genericità.....	120
	Array e vettori generici.....	122
	Enumerazioni generiche	122
	L'enumerazione standard Option<Tipo>	122

L'enumerazione standard Result<Tipo, Errore>	124
Funzioni di utilità standard per gli enum	125

Capitolo 11 Allocazione della memoria127

Le varie tipologie di allocazione	127
Indirizzamento lineare	127
Allocazione statica.....	129
Allocazione nello stack.....	131
Limiti dell'allocazione nello stack.....	135
Allocazione nello heap.....	137
Gestione dello heap	138
Il comportamento del tipo Box.....	139
Somiglianza con C e C++	140
Mettere in Box e togliere da Box	140
Allocazione nei registri del processore	142

Capitolo 12 Il formato dei dati in memoria143

Scoprire le dimensioni degli oggetti	143
La direttiva use.....	144
Le dimensioni dei tipi primitivi.....	145
La rappresentazione dei tipi primitivi	146
Posizione dei byte in memoria	147
Dimensioni dei tipi di dati composti	148
Allocazione dei vettori.....	149

Capitolo 13 Le closure.....153

La necessità di funzioni usa-e-getta	153
Catturare l'ambiente	154
Le closure	155
La sintassi di invocazione delle closure	157

Capitolo 14 Stringhe modificabili.....159

Stringhe statiche	159
Stringhe dinamiche.....	161
Implementazione del tipo String.....	162
Creazione di stringhe dinamiche	165
Conversione di stringhe statiche in stringhe dinamiche	165
Concatenazione di stringhe.....	166

Capitolo 15 I range e le slice169

I range	169
Motivazione per le slice	171
Le slice	172
Creare slice parziali ("slicing").....	175

Slicing fuori range	176
Slicing mutabile	177
Range aperti e slicing	178
Range inclusivi	179
Capitolo 16 Gli iteratori	181
I caratteri delle stringhe	181
Scansione di una stringa	182
Utilizzo degli iteratori per stringhe nei cicli for	183
Le edizioni di Rust	185
Iteratori su vettori, array e slice	185
Iteratori che generano riferimenti	187
Iterazioni senza modifiche	188
Iterazioni con modifica	188
Abbreviazioni per l'uso degli iteratori nei cicli	189
Generatori di iteratori	190
L'adattatore di iteratori filter()	191
L'adattatore map()	191
L'adattatore enumerate()	192
Il consumatore di iteratori any()	193
Il consumatore all()	195
Il consumatore count()	196
Il consumatore sum()	196
I consumatori min() e max()	197
Il consumatore collect()	198
Catene di iteratori	200
Gli iteratori sono pigri	201
Capitolo 17 Input, output e gestione degli errori	203
Gli argomenti della riga di comando	203
Il codice restituito dai processi	204
Le variabili d'ambiente	204
Lettura dalla console	205
Corretta gestione degli errori in fase di esecuzione	207
Scrittura sulla console	212
Conversione di un valore in una stringa	213
Conversione di una stringa in un valore	213
Input/output su file	214
Elaborazione di file di testo	216
Capitolo 18 Incapsulamento dei dati	219
La necessità dei metodi	219
La dichiarazione dei metodi	221
Le parole chiave self e Self	224

Le parole chiave mod e pub	226
La parola chiave type.....	229
Capitolo 19 Uso dei trait	231
La necessità dei trait	231
I trait vengono in soccorso.....	233
Funzioni generiche senza vincoli di parametri.....	236
Ambito dei trait.....	237
Trait con metodi multipli.....	238
Parametri generici con vincoli multipli	239
Ereditarietà dei trait	241
Aggiunta di metodi a tipi esterni.....	242
I trait standard Display e Debug	244
Trait generici	245
Utilizzo di tipi associati	249
Il trait standard Iterator	255
Capitolo 20 Programmazione orientata agli oggetti	259
Peculiarità dell'orientamento agli oggetti di Rust	259
Distruttori	259
Overloading degli operatori aritmetici	263
Overloading degli operatori aritmetici	268
Convertitori di tipo infallibili.....	270
Convertitori di tipo fallibili.....	272
Composizione invece di ereditarietà.....	274
Dispatch statico.....	276
Dispatch dinamico	278
Implementazione dei riferimenti a oggetti-trait	279
Dispatch statico o dinamico?.....	281
Capitolo 21 Le collezioni della libreria standard	283
Le collezioni	283
Misurare il tempo di esecuzione.....	284
Eeguire inserimenti e rimozioni arbitrarie.....	285
Le code	286
Le liste linkate.....	289
Gli heap binari.....	290
Gli insiemi ordinati e gli insiemi non ordinati.....	292
I dizionari ordinati e i dizionari non ordinati.....	294
Le collezioni in C++ e in Rust.....	296
Capitolo 22 I concetti di ownership, copia e spostamento	297
Distruzione deterministica	297
Ownership	299

Semantica degli assegnamenti	299
Prestazioni delle copie e degli spostamenti	302
Spostare e distruggere oggetti	303
Necessità della semantica di copia	305
Clonazione di oggetti	306
Rendere i tipi clonabili o copiabili	308
Capitolo 23 I concetti di prestito e di durata di vita	311
Errori di programmazione quando si utilizzano i riferimenti	311
Prendere in prestito	314
Prestiti mutabili e prestiti immutabili	314
Durata di vita degli oggetti	315
Come prevenire gli errori di “uso dopo la distruzione”	316
Come impedire gli errori di “uso dopo la modifica tramite alias”	317
La necessità di specificatori della durata di vita nei riferimenti restituiti	319
Utilizzo e significato degli specificatori di durata di vita	321
Verifica della validità degli specificatori di durata di vita	324
Utilizzo degli specificatori di durata di vita delle funzioni richiamate	327
Vantaggio dell’utilizzo di specificatori di durata di vita distinti per una funzione	328
Capitolo 24 Altre informazioni sulla durata di vita	331
Elisione della durata di vita	331
Elisione della durata di vita con la programmazione orientata agli oggetti	332
La necessità di specificatori di durata di vita nelle dichiarazioni di strutture	333
Possibili specificatori di durata di vita per le strutture	336
Specificatori di durata di vita per ennuple-strutture e per enumerazioni	338
Specificatori di durata di vita per riferimenti mutabili	339
Conclusioni	339
Indice analitico	341