C++ ו- OOP

למתכנת המקצועי

הוראות התקנה והפעלה של הדיסקט תמצא בנספח.
עיין בהן בתשומת לב לפני ההתקנה.
במהלך הספר תמצא התייחסות לקבצים.

עורך ראשי: **יצחק עמיהוד**
עריכה מקצועית: **צור ריכטר-לוין**
עריכה לשונית ועיצוב: **שרה עמיהוד**
עיצוב עטיפה: **סטודיו מצגר**

שמות מסחריים

שמות המוצרים והשירותים המוזכרים בספר הינם שמות מסחריים רשומים של החברות שלהם. הוצאת הוד-עמי עשתה כמיטב יכולתה למסור מידע אודות השמות המסחריים המוזכרים בספר זה ולציין את שמות החברות, המוצרים והשירותים. שמות מסחריים רשומים (registered trademarks) המוזכרים בספר צוינו בהתאמה.

הודעה

ספר זה מיועד לתת מידע אודות מוצרים שונים. נעשו מאמצים רבים לגרום לכך שהספר יהיה שלם ואמין ככל שניתן, אך אין משתמעת מכך כל אחריות שהיא.
המידע ניתן "כמות שהוא" ("as is"). הוצאת הוד-עמי אינה אחראית כלפי יחיד או ארגון עבור כל אובדן או נזק אשר ייגרם, אם ייגרם, מהמידע שבספר זה, או מהדיסקט שמצורף לו.

לשם שטף הקריאה כתוב ספר זה בלשון זכר בלבד. ספר זה מיועד לגברים ונשים כאחד ואין בכוונתנו להפלות או לפגוע בציבור המשתמשים/ות.

- **טלפון: 09-9564716 24 שעות**
- **פקס: 09-9571582 24 שעות**
- **דואר אלקטרוני: info@hod-ami.co.il**
- **אתר באינטרנט: www.hod-ami.co.il**

C++ ו- OOP

למתכנת המקצועי

שמעון כהן

הוצאת הוד-עמי
לספרי מחשבים

C++ and OOP for the Advanced Programmer

By **Shimon Cohen**

Editor: **I. Amihud**

(C)
כל הזכויות שמורות

הוצאת הוד-עמי
לספרי מחשבים בע"מ
ת.ד. 6108 הרצליה 46160
טלפון: 09-9564716 פקס: 09-9571582

אין להעתיק או לשדר בכל אמצעי שהוא ספר זה או קטעים ממנו בשום צורה ובשום אמצעי אלקטרוני או מכני, לרבות צילום והקלטה, אמצעי אחסון והפצת מידע, ללא אישור בכתב מאת ההוצאה, אלא לשם ציטוט קטעים קצרים בציון שם המקור.

הודפס בישראל
תשנ"ז, 1997

All Rights Reserved
HOD-AMI Ltd.
P.O.B. 6108, Herzliya
ISRAEL, 1997

ISBN 965-361-142-9 מסת"ב

תוכן עניינים מקוצר

תוכן העניינים 6

מבוא 17

פרק 1 סקירה של C 23

פרק 2 תכנות מוכוון אובייקטים 55

פרק 3 המחלקה 69

פרק 4 קלט ופלט 111

פרק 5 ירושה ככלי לשימוש חוזר בקוד 127

פרק 6 פולימורפיזם 163

פרק 7 תבניות ב-C++ 179

פרק 8 טיפול בשגיאות ועוד 221

פרק 9 חידושים ב-C++ 251

פרק 10 ספריית התבניות הסטנדרטיות - STL 273

פרק 11 מבני נתונים סדרתיים ב-STL 287

פרק 12 מבנים שאינם סדרתיים ב-STL 307

פרק 13 שילוב סוגי תכנות 335

נספח - דיסקט התוכניות המצורף 355

אינדקס עברי 359

אינדקס לועזי 373

תוכן העניינים

מבוא

על ++C 17
על הספר ומה שיש בו 19
הדיסקט המצורף 22

פרק 1 סקירה של שפת C

1.1 הקדמה והיסטוריה 23
1.2 תוכנית ראשונה 24
1.2.1 העברת ארגומנטים לתוכנית הראשית 26
1.2.2 הערות 27
1.3 משתנים ופעולות אריתמטיות 27
1.3.1 מערכים 29
1.3.1.1 מערכים כפולים 30
1.3.2 דוגמה 30
1.3.3 מצביעים 32
1.3.3.1 מצביעים ומערכים 33
1.3.3.2 שינוי הארגומנטים של פונקציה 34
1.3.3.3 מצביעים לפונקציות 35
1.3.4 אופרטורים להוספה והפחתה 36
1.3.5 ביטויים לוגיים 37
1.3.6 אופרטורים הפועלים על סיביות 38
1.3.6.1 פעולות לוגיות על סיביות 38
1.3.6.2 פעולות הזזה על סיביות 39
1.3.7 מחרוזות 39

1.4 משפטי בקרה.....40
1.4.1 משפט הלולאה while.....40
1.4.2 משפט התניה if.....42
1.4.3 משפט for.....43
1.4.4 משפט do-while.....44
1.4.5 משפט switch.....45
1.5 רשומות.....46
1.5.1 שימוש ב-typedef.....47
1.6 פונקציות רקורסיביות.....49
1.7 דוגמאות לקלט/פלט בסיסי.....49
1.7.1 ספירת תווים בקובץ נתון.....50
1.7.2 העתקת קבצים.....50
1.7.3 ספירת בתים ושורות.....51
1.8 קדם-מהדר.....52
1.8.1 הכללת קבצים.....52
1.8.2 הגדרות של קבועים.....52
1.8.3 הגדרת תבנית פונקציה.....53
1.9 סיכום.....53
1.10 שאלות.....54
1.11 מקור.....54

פרק 2 תכנות מוכוון אובייקטים

2.1 משבר התוכנה.....55
2.2 מה הם מודולים.....56
2.2.1 דוגמה: ספר טלפונים - ניסיון ראשון.....56
2.2.2 בניית ספר טלפונים בעזרת מודולים.....57
2.2.3 טיפול במספר מודולים.....58
2.2.4 החסרונות של המודולים.....61
2.3 המחלקה.....62
2.4 סיכום.....66
2.5 תרגילים.....67

פרק 3 המחלקה

3.1 מבנה המחלקה - אזורי גישה.....69
3.2 הבנאי - פונקציית האתחול של המחלקה.....70
3.2.1 בנאי ברירת המחדל.....70
3.2.2 בנאי עם פרמטרים.....71

3.2.2.1 בנאי עם קבועים 72
3.2.3 בנאי העתקה ומושג הייחוס 73
3.3 המפרק, פונקציית הניקוי של אובייקטים..... 74
3.4 מושג הייחוס 76
3.4.1 שימושי הייחוס 77
3.5 פונקציות מחלקה..... 79
3.5.1 פונקציות קבועות..... 79
3.5.2 מנגנון inline 80
3.5.2.1 פקודת מאקרו..... 82
3.6 אופרטורים 83
3.6.1 אופרטורים אונרים..... 84
3.6.2 אופרטורים בינארים 84
3.6.3 סיכום האופרטורים לפי עדיפות 86
3.7 שדות סטטיים המשותפים לאובייקטים 89
3.7.1 פונקציות סטטיות 90
3.7.2 דוגמה לשימוש בשדות ופונקציות סטטיות 91
3.8 מנגנון החבר (friend)..... 92
3.8.1 מנגנון החבר בשימוש 92
3.8.1.1 בנאים ל-String..... 93
3.8.1.2 אופרטורים 94
3.9 אובייקטים כשדות של אובייקטים אחרים 97
3.9.1 הגדרת תת-אובייקטים 97
3.9.2 העתקת אובייקטים מורכבים 98
3.9.3 דוגמה לשימוש במצביעים לתת-אובייקטים..... 99
3.10 הגדרה מקוננת של מחלקות..... 100
3.11 רשומות כמחלקות..... 101
3.11.1 אזורי גישה..... 102
3.11.2 פונקציות של רשומות 102
3.12 מחלקות חסכוניות..... 103
3.12.1 איחוד (union) 103
3.12.2 שדות סיביות (bits) 105
3.13 מצביעים לשדות ופונקציות במחלקה 106
3.13.1 מצביעים לשדות..... 107
3.13.2 מצביעים לפונקציות של אובייקטים..... 108
3.14 סיכום 109
3.15 שאלות 109

פרק 4 קלט ופלט

4.1 חסרונות הספרייה הסטנדרטית של C 111
4.2 קלט/פלט פשוט ב- C++ 113
4.2.1 העמסת אופרטורים 113
4.2.2 אובייקטים בסיסיים 114
4.3 קריאה מקבצים 115
4.3.1 קריאה עם עריכה 115
4.3.2 קריאה ללא עריכה 116
4.4 כתיבה לקבצים 117
4.4.1 כתיבה עם עריכה 118
4.4.2 כתיבה בינארית לקבצים 118
4.5 שילוב קלט ופלט 119
4.5.1 שילוב בין אובייקטי קלט לפלט 119
4.5.2 שימוש באובייקטי קריאה וכתיבה 120
4.5.3 קביעת מיקום מצביע הקריאה או הכתיבה 120
4.6 עריכת נתונים בזיכרון 122
4.7 שילוב פונקציות פלט/קלט 123
4.8 מניפולטורים 124
4.8.1 תמיכה במניפולטורים במחלקה נתונה 124
4.8.2 מניפולטורים במערכת קלט/פלט 125
4.9 סיכום 126
4.10 שאלות 126

פרק 5 ירושה ככלי לשימוש חוזר בקוד

5.1 מהי ירושה 127
5.2 ירושה יחידה 128
5.2.1 סוג הירושה 131
5.2.1.1 ירושה פרטית 132
5.2.1.2 ירושה מוגנת 133
5.2.1.3 הורשה ציבורית 134
5.2.2 דוגמה - רשימה מעגלית כפולה 135
5.2.2.1 צומת ברשימה 135
5.2.2.2 האיטרטור 136
5.2.2.3 הרשימה 138
5.2.2.4 שימוש ברשימה בעזרת ירושה 141
5.2.3 ירושה מהרשימה 144
5.2.3.1 מחסנית 144
5.2.3.2 תור 144

5.2.4 ניהול זיכרון והגדרת אופרטורים 145
5.2.4.1 הגדרת אופרטור new 146
5.2.4.2 הגדרת האופרטור delete 146
5.2.4.3 שימושים באופרטורי ניהול הזיכרון..... 146
5.3 ירושה מרובה..... 148
5.3.1 הגדרת ירושה מרובה..... 148
5.3.2 אתחול של מחלקות בסיסיות..... 150
5.3.3 ירושה וירטואלית 151
5.3.3.1 אתחול במצב של ירושה וירטואלית..... 152
5.4 סדר בנייה ופירוק של אובייקטים 154
5.4.1 סדר בניית אובייקטים..... 154
5.4.2 סדר פירוק אובייקטים 155
5.4.3 דוגמה לסדר בנייה ופירוק של אובייקטים 155
5.5 המרות..... 157
5.5.1 המרות של סוגים בסיסיים 157
5.5.2 המרות של מחלקות 158
5.5.3 חיתוך אובייקטים 158
5.5.4 המרות של מצביעים או ייחוס..... 159
5.5.5 המרות המוגדרות על ידי המשתמש 159
5.5.6 סדר ההמרות..... 160
5.6 סיכום..... 161
5.7 שאלות..... 162

פרק 6 פולימורפיזם

6.1 דוגמה - הדרך הישנה 163
6.1.1 הגדרת המסך..... 164
6.1.2 הגדרת הצורות..... 165
6.1.3 הגדרת הפונקציות והפעולות 167
6.1.4 ניתוח חסרונות..... 168
6.2 מהו פולימורפיזם..... 168
6.2.1 הפעלה וירטואלית..... 169
6.3 כיצד מיישמים פונקציות וירטואליות?..... 170
6.4 אילו פונקציות יכולות להיות וירטואליות?..... 172
6.5 מחלקות מופשטות טהורות..... 173
6.5.1 נקודת התורפה של מערכת פולימורפית 174
6.5.1.1 ייצור אובייקטים מופשטים (Abstract Factory)..... 175
6.6 סיכום..... 177
6.7 שאלות..... 177

פרק 7 תבניות ב-++C

7.1 לשם מה תבניות 179
7.2 תבניות ב-++C..... 180
7.2.1 פונקציות תבנית 180
7.2.1.1 שימוש במספר פרמטרים בתבנית..... 181
7.2.1.2 תכנות גנרי 184
7.2.2 מחלקות תבנית 190
7.2.2.1 אובייקטי פונקציות 192
7.2.2.2 שימוש בפרמטר קבוע למחלקת תבנית 193
7.2.2.3 דוגמה - רשימה מקושרת..... 194
7.2.3 אפליקטורים..... 205
7.2.3.1 הדפסה בעזרת האפליקטור 206
7.2.3.2 פירוק בעזרת האפליקטור..... 207
7.2.4 דוגמה - מערך דינמי 208
7.2.4.1 מבנה המערך 208
7.3 ירושה ותבניות..... 213
7.3.1 ירושה מהמחלקה Array 214
7.3.2 שימוש באובייקט מסוג מערך בטוח..... 215
7.3.2.1 פונקציות וירטואליות..... 215
7.3.3 מתאמים 216
7.3.3.1 תור..... 216
7.3.3.2 מחסנית..... 217
7.3.3.3 מתאמים גנרים..... 218
7.4 סיכום..... 220
7.5 שאלות..... 220

פרק 8 טיפול בשגיאות ועוד

8.1 מדוע צריך לטפל בשגיאות..... 221
8.2 פונקציות טיפול בשגיאות..... 223
8.3 טיפול בסיסי בעזרת מבני ++C..... 224
8.3.1 לעורר שגיאה (Throwing an Error) 224
8.3.2 אופן הטיפול בשגיאה 225
8.3.3 דוגמה - מערך בטוח..... 226
8.3.4 חריגים וירושה..... 228
8.3.4.1 ירושה ללא פונקציות וירטואליות 228
8.3.4.2 שימוש בפונקציות וירטואליות..... 230
8.3.4.3 לכידת כל השגיאות..... 231
8.3.4.4 לעורר מחדש את השגיאה..... 232

8.3.4.5 סימון מצבים חריגים 232
8.3.4.6 טיפול בחריגים שאינם צפויים 234
8.4 שימוש וטכניקות טיפול במצבים חריגים 235
8.4.1 טיפול בזיכרון 236
8.4.1.1 מצביעים אוטומטיים לאובייקטים 237
8.4.1.2 מצביע אוטומטי למערך של אובייקטים 239
8.4.2 שימוש במשאבים 241
8.5 שילוב בין תבניות למצבים חריגים 243
8.5.1 מחלקת התבנית - מחסנית 244
8.5.2 השימוש במחלקה Stack 245
8.5.3 טיפול במצב1ים חריגים 246
8.5.3.1 מצבים חריגים מקומיים 247
8.5.3.2 מצבים חריגים חיצוניים 247
8.6 סיכום 249
8.7 שאלות 250

פרק 9 חידושים ב-++C

9.1 מידע בזמן ריצה 251
9.1.1 האופרטור typeid 252
9.1.2 המחלקה type_info 253
9.1.3 האופרטור dynamic_cast 257
9.2 המרות נוספות של מצביעים 259
9.2.1 המרות סטטיות static_cast 259
9.2.2 המרות המבטלות קביעות const_cast 260
9.2.3 המרה חוזרת reinterpret_cast 261
9.3 מרחבי שמות namespace 263
9.3.1 שימוש בהגדרות מקוננות 263
9.3.1.1 המונח namespace 264
9.3.1.2 שימוש במילת המפתח using 265
9.3.2 כינון של מרחבי שמות 266
9.3.3 הרחבת מרחב שמות 267
9.4 מילת המפתח mutable 268
9.5 מילת מפתח explicit (מפורש) 269
9.6 סיכום 270
9.7 שאלות 271

פרק 10 ספריית תבניות סטנדרטיות STL

10.1 עקרונות הספרייה STL 273
10.2 מכולות ב-STL 275
10.3 איטרטורים 276
10.3.1 שימוש באלגוריתמים ואיטרטורים 277
10.3.2 תווית של איטרטור 278
10.4 אלגוריתמים 281
10.5 מתאמים (Adaptors) 283
10.6 אובייקטי פונקציות (Function Objects) 284
10.7 סיכום 285
10.8 מקורות 285

פרק 11 מבני נתונים סדרתיים ב-STL

11.1 וקטור 287
11.1.1 מבנה הווקטור 287
11.1.2 האיטרטור 289
11.1.3 פונקציות חשובות 289
11.2 רשימה כפולה - list 291
11.2.1 מבנה הרשימה 291
11.2.2 האיטרטור 291
11.2.3 פונקציות שימושיות 292
11.3 תור כפול 293
11.3.1 מבנה התור הכפול 293
11.3.2 האיטרטור 294
11.4 מבנה נתונים - דוגמה 295
11.4.1 מבנה היומן 295
11.4.2 פונקציות עזר 296
11.4.3 התוכנית הראשית 299
11.4.4 תכנות גנרי 304
11.5 סיכום 306
11.6 שאלות 306

פרק 12 מבנים שאינם סדרתיים ב-STL

12.1 הגדרת קבוצות 307
12.1.1 פעולות על קבוצות 308
12.1.2 מבני נתונים אפשריים לקבוצה 308
12.1.2.1 טבלת ערבול 308
12.1.2.2 עצים בינאריים 310
12.1.2.3 עץ מאוזן - Red Black Tree 311
12.2 הקבוצה set 312
12.2.1 מבנה הקבוצה 312
12.2.2 פונקציות חשובות 313
12.2.3 האיטרטור 314
12.3 קבוצה עם כפילויות 315
12.3.1 פונקציות חשובות 315
12.4 דוגמה - שימוש בקבוצה 316
12.4.1 כניסה ביומן 316
12.4.2 פונקציות עזר 317
12.5 מפות (map, multimap) 323
12.5.1 מפה (map) 323
12.5.1.1 פונקציות חשובות 323
12.5.1.2 האיטרטור 326
12.5.2 מפה כפולה (multimap) 326
12.5.2.1 פונקציות חשובות 326
12.5.2.2 האיטרטור 326
12.5.3 דוגמה - שימוש במפה 327
12.5.3.1 פונקציות עזר 327
12.5.3.2 פונקציות קלט 328
12.5.3.3 פונקציות עדכון למילון 330
12.5.3.4 קלט ופלט מקבצים 331
12.5.3.5 פונקציות הניהול 332
12.6 סיכום 334
12.7 שאלות 334

פרק 13 שילוב סוגי תכנות

13.1 מתאמים (Adaptors) 335
13.1.1 תיאום בין ספריות שונות 335
13.1.2 המרת ממשק אובייקט לקבלת פונקציונליות אחרת 337
13.1.2.1 שימוש ברשימה 338
13.1.2.2 גמישות במימוש המחסנית 338

13.1.3 המחלקה stack ב-STL 341
13.2 דוגמה - ניהול משימות (tasks) 342
13.2.1 היררכיית הפעילויות 342
13.2.1.1 המחלקה Task 342
13.2.1.2 המחלקה CopyFileTask 343
13.2.1.3 המחלקה WCTask 344
13.2.2 מנגנון יצירת האובייקטים 345
13.2.2.1 המחלקה MetaClass 345
13.2.2.2 המחלקה class_vector 346
13.2.3 לב המערכת - אוסף המשימות 347
13.2.4 פונקציות הניהול 349
13.3 סיכום 352
13.4 שאלות 352

נספח

1 דיסקט התוכניות המצורף 355
2 התקנה 355
3 הרצה - כללי 355
4 הידור והרצה תחת חלונות 356
4.1 הגדרת פרויקט 356
4.2 הידור התוכנית 357
4.3 הרצת התוכנית 357
5 הידור והרצה בסביבת DOS 357
5.1 חוקים בסיסים של make 357
5.2 הידור והרצה 358

אינדקס

עברי 359
לועזי 373

מבוא

על C++

כשכותבים ספר, נותנים את הדעת למי הוא מיועד, כי קהל היעד קובע את הרמה והסגנון של הספר. **הספר שלפניכם מיועד למתכנתים בשפת C++ שרוצים להגיע לרמת מיומנות גבוהה ולהתקדם להיות מתכנתים מקצועיים באמת. הוא גם מיועד למתכנתי C ושפת פסקל, שאינם מכירים את שפת התכנות C++**. לטובת מתכנתי C ניתן פרק הגירה המסכם את אפשרויות שפת C, עם הצבעה על המקבילות להן בשפת C++. במהלך הלימוד יש הסברים והתייחסויות הקשורים לשפת C, מכיון ששפת C++ התפתחה כהמשך לשפת זו. מתכנתים ב**שפת פסקל** יוכלו גם כן להשתמש בספר, אך ייתכן שקצב הלימוד שלהם יהיה איטי יותר בתחילה. לאלה האחרונים, ההמלצה היא לקרוא בעיון את הפרק העוסק בשפת C.

שפת התכנות C++ והמושג OOP - תכנות מוכוון אובייקטים - חד הם. אפשר לומר ששפת C++ מבטאת את המימוש של עקרונות OOP, ולכן לא ניתן להפריד בין השניים. **התכונות והכלים של C++ נגזרים מעקרונות OOD (עיצוב מוכוון אובייקטים) ו-OOP** ולכן, אי אפשר ללמוד C++ ללא OOP, ואי אפשר ללמוד OOP ללא C++. במהלך הלימוד והרחבת הידע נעסוק בשני תחומים: **תחום התכנות בשפת C++ ותחום תפישת התכנות המתקדמת - OOP**, אשר מהווה כאמור, את התשתית לשפת התכנות.

קיימים ספרים לא מעטים על C++, רובם מתארים את השפה ברמה בסיסית ואחרים מתארים אותה ברמה מתקדמת יותר. ספר זה מתחיל מרמה בסיסית של C++, אך מניח שלקורא יש רקע בסיסי כלשהו בתכנות. על כן, הוא ממשיך מיד לנושאים מתקדמים, למושגים חדשים ולחידושים בשפה שטרם מצאו ביטויים ברבים מהספרים המתקדמים. מטרת הספר היא להפוך **מתכנתי C++ "רגילים" למתכנתים מקצועיים** בשפה זו, השולטים בטכניקות תכנון ותכנות חדשות ומתקדמות.

C++ היא שפה מתפתחת. התקן שלה אינו גמור עדיין (לפחות בעת כתיבת שורות אלו), אך קרוב לסיומו. שיטות וטכניקות חדשות עדיין מתוספות ולכן, ספרים חדשים המלמדים טכניקות ומונחים חדשים הם כורח המציאות.

שפות התכנות C ופסקל הן שפות פשוטות ובעלות יכולות מוגבלות. מתכנת ממוצע יכול להשתלט ולהתמחות בשפות אלו בזמן קצר יחסית, במספר חודשים מועט. עם C++, לעומת זאת, המצב שונה. גם מתכנתים המשתמשים בשפה זו מספר שנים, אינם מכירים את כל אפשרויותיה. הזמן המשוער למתכנת ממוצע להשתלט על יכולות השפה מוערך בשנים. מתכנת שמכיר את C יכול לכתוב תוכניות ב-C++ כמעט באופן מיידי, אך כדי לכתוב תוכניות יעילות ובאיכות גבוהה, דרוש זמן לימוד רב. **ספר זה מלמד את השיטות והטכניקות הדרושות למתכנת כדי שיוכל לנצל את השפה ביעילות.** לימוד עקבי יכול לקצר את זמן הלימוד באופן משמעותי.

C++ היא שפה הייברידית, כלומר, שפה שממזגת מספר גישות. היא יורשת מ-C את כל המבנים והתכונות שלה, אך בנוסף לאלה היא מציגה יכולות חדשות, כגון **תכנות מוכוון אובייקטים** (Object Oriented Programming) ו**תכנות גנרי** (Generic Programming). למרות ששני המונחים האחרונים נשמעים זהים, הם שונים. הסבר על כך תמצא בפרקים המתאימים.

שפת C++ היא שפה מוכלאת (הייבריד) ולכן היא מאפשרת לכתוב תוכניות דמויות C בשפה C++. רבים חושבים שאפשרות זו היא חיסרון. קיימות שפות מוכוונות אובייקטים אחרות, שאינן מאפשרות כתיבה אחרת, פרט לכתיבה בשיטה זו. חסידי שפות אלו טוענים ששפת C++ מאפשרת חופשיות רבה מידי למתכנת, ואף מאפשרת להשתמש במבנים שאינם טובים מבחינת הנדסת תוכנה. מבנים אלה הן **פונקציות** שאינן קשורות לאובייקטים.

בשפת אובייקטים טהורה קיימים רק אובייקטים, כל הפונקציות נקראות **שיטות (methods), ואינן** עומדות בפני עצמן, אלא **שייכות לאובייקט כלשהו**. אובייקט מייצג מבנה נתונים כלשהו ופעולות המבוצעות עליו, ואלה קשורים בקשר בלתי ניתן להפרדה. תכונה זו היא יתרון במקרים רבים, אבל ייתכנו גם מקרים בהם נרצה לכתוב פונקציות **שאינן** קשורות לאובייקט. הדבר יעיל במיוחד כשרוצים לכתוב פונקציות גנריות (כלליות), הפועלות על מגוון רחב של אובייקטים, ללא תלות בסוג. היכולות לבצע דברים כאלה אינה נתונה בשפות אחרות. יכולת זו הולידה את המונח **תכנות גנרי**, שנתמך ב-C++, אבל לא בשפות אובייקטים אחרות.

תכנות גנרי (Generic Programming) מאפשר אם כן לכתוב פונקציות, שהפרמטרים שלהן וסוג האובייקטים שהן מקבלות משתנה ומותאם לפי סוג השימוש שנעשה בהם. תכונה זו מאפשרת לכתוב פונקציות גנריות הפועלות על סוגים רבים של אובייקטים שאינם קשורים זה לזה, כמו למשל, פונקציה שמעתיקה תחום אחד למשנהו. ה"תחום" יכול להיות מערך, רשימה, עץ חיפוש וכדומה. פעולת ההעתקה אינה תלויה בסוגים אלה, כל עוד הם מספקים ממשק קבוע של פעולות. הדבר חוסך מהמתכנת שכפול של פונקציות מספר רב של פעמים, אחת לכל סוג של מבנה נתונים, או לכל סוג אובייקט.

בשפות אובייקטים טהורות אחרות כמו **SmallTalk**, כל סוגי הנתונים הם מחלקות היורשות מאובייקט גנרי כלשהו. לא כך הדבר ב-C++, שבה יש את כל סוגי הנתונים שקיימים ב-C, וגם סוגי נתונים חדשים, או הרחבות של השפה. סוגי נתונים אלה הן **מחלקות** (Classes) אשר נקראות גם: **סוגי נתונים המוגדרים על ידי המשתמש** (User Defined Types).

מתכנני C++ הקדישו מחשבה רבה ליעילות השפה. C++ אינה נופלת משפת התכנות C, שנחשבת לשפה היעילה ביותר בין שפות התכנות המקובלות, ויש מקרים שאפשר להגיע בה לרמת יעילות גבוהה יותר. עובדה זו נראית מפתיעה, ומרבית הקוראים יתקשו לקבל עובדה זו. **ספר זה מדגים את המקרים ואת השיטות שמאפשרים להגיע ליעילות כזו**, המושגת הודות ליכולת לכתוב פונקציות גנריות.

כדי להגיע ליעילות המתקרבת לשפת התכנות C חייבים להכיר את C++ היטב ולשלוט ברזי השפה (רבים לא מנצלים אותה בצורה יעילה). בין שפות האובייקטים, C++ היא היעילה ביותר. היא מאפשרת גישה לסיביות בודדות במילה, לפעול על מצביעים, וגם מאפשרת גישה לאוגרים של המחשב.

העיקרון העומד מאחורי C++ הוא שאי-שימוש ביכולות מסוימות של השפה לא יפגע במתכנת, או בתוכנית שהוא מפתח. מתכנתים יכולים להשתמש רק בחלקי השפה שהם מכירים, ללא כל צורך להכיר את כולה על בוריה.

על הספר ומה שיש בו

הנושאים הנלמדים בספר זה רבים, ומכסים את כל היבטי השפה. הפרקים מחולקים ועולים ברמת הקושי, ככל שמתקדמים בקריאת הספר. ההנחה הבסיסית היא שקורא שמתקדם מפרק לפרק רוכש יותר ידע, ומאפשר לעלות ברמת הנושאים שהוא לומד.

הפרק הראשון מלמד על **המבנים ומשפטי הבקרה** ש- C++ יורשת מ-C. פרק זה מביא את הקורא לרמה הידע הדרושה בשפת C כדי שיוכל ללמוד את הנושאים האחרים שבספר.

הפרק השני מציג את המושג **תכנות מוכוון אובייקטים - OOP**. הפרק מתחיל בהצגת המונח **מודול**, שקדם לתכנות מוכוון אובייקטים. הפרק מציג את הסיבות והמוטיבציה שהולידו את הטכניקות של תכנות מוכוון אובייקטים.

הפרק השלישי מציג את המונח **מחלקה**. מונח זה התפתח מהמונח **מודול**. מחלקה ב-C++ היא סוג נוסף שמגדיר המתכנת ובכך מרחיב את השפה. כלומר, C++ מאפשרת להגדיר אופרטורים למחלקות, או סוגים חדשים המוגדרים על ידי המתכנת, כך שסוגים אלה ייראו כמו סוגים בסיסיים של השפה. זהו מונח מרכזי בשפה זו ומהחשובים בשפות אובייקטים.

הפרק הרביעי מציג את **ספריית הקלט/פלט** של C++. שפת התכנות עצמה אינה כוללת מבנים המאפשרים טיפול בקבצים. במקום זאת, מסופקת ספרייה תקנית

המאפשרת את הטיפול בקלט/פלט. שימוש בספרייה זו מאפשר לכתוב תוכניות ניידות שיכולות לרוץ על סוגים שונים של מחשבים ומהדרים, ללא צורך בשינוי.

הפרק החמישי מלמד את המושג **ירושה** (Inheritance). הפרק מציג את המונחים **ירושה, ירושה מרובה וירושה וירטואלית**. הפרק מציג נושאים רבים הקשורים לירושה, כגון סדר בנייה של אובייקטים, סדר פירוק של אובייקטים, מתי להשתמש בירושה וכיצד להשתמש בירושה בצורה טובה.

הפרק השישי מלמד את הנושא **פונקציות ווירטואליות**. פונקציות וירטואליות קשורות למונח פולימורפיזם, כלומר ריבוי-צורתיות. מונח זה, יחד עם מחלקות וירושה, מהווה בעיני רבים את הבסיס למונח תכנות מוכוון אובייקטים. פרק זה מציג גם את מודל האובייקטים וכיצד מושגת קשירה מאוחרת של פונקציות לאובייקטים.

הפרק השביעי מתאר את המונח **תבניות** (Templates) ב-C++. מונח זה הינו מהחשובים בשפה, אשר מאפשר **תכנות גנרי**, אחת מטכניקות התכנות החדשות שהופיעה לאחרונה ב-C++. פרק זה מסביר את עקרונות התכנות הגנרי.

הפרק השמיני מדגים את המונח **מצבים חריגים** (Exception Modes). מצבים חריגים הם תוספת חדשה יחסית ל- C++. מתכנתים מעטים יודעים לטפל במצבים חריגים, בדרך נאותה. נושא זה אינו מוסבר בדרך כלל בספרי לימוד אחרים של השפה. בדרך כלל, מוסברות רק המכניקה ומילות המפתח של השפה. בספר זה, לעומת זאת, בנוסף למכניקה ומילות המפתח, מוסברות טכניקות תכנות וטיפול במצבים חריגים. גם מוצגות מספר מחלקות העוזרות בטיפול במצבים חריגים.

הפרק התשיעי מציג **נושאים חדשים** שהתוספו לשפה ושאינם נתמכים על ידי רוב המהדרים. בין הנושאים כלולים **מרחבי שמות**, **המרות של מצביעים**, **זיהוי סוגי אובייקטים בזמן ריצה** ועוד.

הפרק העשירי מהווה מבוא לספריית התבניות הסטנדרטית של C++, **STL**. ספרייה זו הביאה לעולם את המונח **תכנות גנרי** ויצרה מהפכה באופן החשיבה והשימוש ב-C++. STL היא ספרייה יעילה במיוחד, יחסית לספריות האחרות הקיימות ב-C++ וכדי להשתמש בה ביעילות דרוש ידע נרחב בתבניות. הספרייה יעילה בפעולותיה פי שתיים או שלוש מספריות אחרות. הספרייה תומכת ב**מכולות**, שהם אובייקטים המכילים אובייקטים רבים אחרים (כגון, רשימות). הספרייה תומכת באיטרטורים שמאפשרים לפעול על מכולות בצורות שונות, ומספקת מגוון רחב של אלגוריתמים גנרים. אלגוריתמים כמו sort, שממיין תחום נתונים, או אלגוריתם כגון unique, שמבטל מופעים כפולים של אובייקטים בתחום נתון. אלגוריתמים אלה פועלים על כל סוג נתונים כל עוד סוג הנתונים מקיים ממשק מסוים. כלומר, האלגוריתמים המסופקים בספרייה יכולים לתמוך במכולות של ספריות אחרות!

הפרק האחד עשר מציג את הנושא **מבנים סדרתיים** שמספקת STL. מבנים אלה כוללים **וקטור** (מערך חד-ממדי), **רשימה** ו**תור כפול**. כמו כן, מציג הפרק את האיטרטורים של מחלקות אלו. הפרק נותן הסבר על המבנה הפנימי של המחלקות וכללים מתי ואיך להשתמש בהן ביעילות.

הפרק השניים עשר דן ב**מבנים שאינם סידרתיים** ושמספקים חיפוש יעיל ומהיר של אובייקטים בתוכם. מבנים אלה מבוססים על עצי חיפוש בינארים מאוזנים. המחלקות שמתוארות בפרק זה הן: **set**, **multiset**, **map** ו-**multimap**. כמו כן, מתוארים האיטרטורים שלהם ומוצגות דוגמאות לשימוש במחלקות אלו.

בשני הפרקים האחרונים מודגמת, ומוצגת, שיטת התכנות הגנרית שהביאה STL לעולם. השיטה מוצגת לעומק ועקרונותיה מוסברים בפירוט.

בפרק השלושה עשר מתוארים מתאמים (adaptors) ושילוב בין תכנות גנרי לבין תכנות מוכוון אובייקטים. פרק זה מציג דוגמה המשלבת בין עקרונות התכנות מוכוון האובייקטים והתכנות הגנרי, ניהול מטלות ועוד.

רוב המתכנתים הלומדים את C++, הופכים לחסידים נלהבים שלה. צריך להזכיר לקורא, **שהשפה היא כלי להשגת המטרה ולא המטרה עצמה**. המטרה היא לכתוב תוכניות שמקיימות דרישות מסוימות בצורה יעילה ובאיכות גבוהה, תוכניות עמידות בשינויים, שקל לשנותן בעת הצורך. C++ אינה המטרה, אלא **הכלי** להשגתה. כך למשל, אין להשתמש בפונקציות וירטואליות, אלא אם הן תומכות בפתרון בעיית התכנות.

התחלתי לעבוד בשפת C++ בינואר 1989 בסידני שבאוסטרליה, ומאז ועד כתיבת שורות אלו אני משתמש בה. למרות מספר השנים הרב בהם אני עוסק בתכנות מוכוון אובייקטים אני עדיין ממשיך ללמוד דברים חדשים לגבי C++. באוסטרליה עבדנו על פרויקט שנחשב, בשעתו, מהפכני בסוגו. כאשר הציגו לנו את השפה חשבתי שזו טעות להשתמש בשפה חדשה בפרויקט חדש שבו יש טכנולוגיות חדשות ורמת הסיכון בו כה גבוהה. מאוחר יותר התברר ש- C++ עזרה לנו לפתח תוכנה באיכות גבוהה יותר ובזמן קצר יותר.

כיום אני עובד בחברת אורבוטק, ובה אנו מנצלים את השפה, ואת הטכניקות החדשות האלו, לפיתוח של פרויקטים מתקדמים. אני יכול להבטיח לקורא ש- C++ מספקת כלים יעילים מאוד למי **שיודע להשתמש** בהם! למי שאינו יודע להשתמש בכלי השפה, אייעץ שלא להשתמש בהם. כדאי לו, למשתמש, לנצל את אפשרויות השפה שהוא מכיר ובהן הוא שולט היטב.

קוראים המוצאים צורך בייעוץ, או מעוניינים לשאול שאלות, מוזמנים לפנות אלי. ניתן לעשות זאת בשתי דרכים: דרך **הוצאת הוד-עמי לספרי מחשבים** המוציאה לאור של ספר זה, או להשתמש בכתובת הדואר האלקטרוני שלי: