

The **Java** Tutorial

סדנת לימוד

הוראות התקנה והפעלה של התקליטור
בעמוד 19 ובתקליטור

עורך ראשי: **יצחק עמיהוד**
עריכה מקצועית: **משה יצחקי, זהר עמיהוד**
תרגום: **איציק מנשרוף, יעל בצלאל**
עריכה לשונית ועיצוב: **עפרה שפי, שרה עמיהוד**
עיצוב עטיפה: **שרון רז**

שמות מסחריים

שמות המוצרים והשירותים המוזכרים בספר הינם שמות מסחריים רשומים של החברות שלהם. הוצאת הוד-עמי והוצאת Addison Wesley עשו כמיטב יכולתן 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 והוצאת Addison Wesley אינן אחראיות כלפי יחיד או ארגון עבור כל אובדן או נזק אשר ייגרם, אם ייגרם, מהמידע שבספר זה, או מהתקליטור המצורף.

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

☐ פקס: 09-9571582

☐ דואר אלקטרוני: info@hod-ami.co.il

☐ אתר באינטרנט: www.hod-ami.co.il

The **Java** Tutorial

סדנת לימוד

Mary Campione
Kathy Walrath
Alison Huml



The Java Tutorial, Third Edition

A Short Course on the Basics

By **Mary Campione, Kathy Walrath, Alison Huml**

Editor: **I. Amihud**

Copyright © 2000, All Rights Reserved

Published by arrangement with the original publisher, ADDISON WESLEY LONGMAN, a Pearson Education Company.

Translation copyright © 2001 by Hod-Ami Computer Book Publisher LTD.

(C)

כל הזכויות שמורות

הוצאת הוד-עמי

לספרי מחשבים בע"מ

ת.ד. 6108 הרצליה 46160

טלפון: 09-9564716 פקס: 09-9571582

info@hod-ami.co.il

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 2001

All Rights Reserved

HOD-AMI Ltd.

P.O.B. 6108, Herzliya

ISRAEL, 2001

מסת"ב 965-361-275-1 ISBN

תוכן עניינים מקוצר

19	התקליטור המצורף.....
21	הקדמה.....
25	פרק 1: הכרת טכנולוגיית Java
67	פרק 2: מושגים בתכנות מוכוון עצמים (OOP).....
91	פרק 3: יסודות השפה.....
145	פרק 4: יסודות האובייקט ואובייקטים של נתונים פשוטים.....
207	פרק 5: מחלקות והורשה.....
259	פרק 6: ממשקים וחבילות.....
275	פרק 7: טיפול בשגיאות בעזרת חריגים - Exceptions ..
	פרק 8: ביצוע שתי משימות ויותר בו-זמנית בעזרת מטלות (Threads).....
301	
343	פרק 9: קלט/פלט – קריאה וכתיבה של נתונים.....
379	פרק 10: רכיבים לבניית ממשק משתמש גרפי – AWT ..
461	פרק 11: פריסת רכיבים בתוך מכולה.....
483	פרק 12: ממשק משתמש גרפי – Swing
523	פרק 13: רישות בפלטפורמת Java

567	נספח א': בעיות שכיחות ופתרון
583	נספח ב': יישומנים לאינטרנט (Applets)
641	נספח ג': אוספים (Collections)
701	נספח ד': שיטות Thread שהוצאו משימוש
709	נספח ה': Reference
721	אינדקס

תוכן העניינים

19	התקליטור המצורף.....
19	Acrobat Reader - התקנה
20	התקנת ערכת הפיתוח של Java בסביבת חלונות
21	הקדמה.....
21	למי מיועד הספר?.....
22	כיצד לקרוא בספר.....
22	הצוות המקצועי.....
25	פרק 1: הכרת טכנולוגיית Java.....
26	אודות טכנולוגיית Java (Java Technology)
26	שפת התכנות Java
27	הפלטפורמה של Java (Java Platform)
29	מה יכולה טכנולוגיית Java לעשות?.....
31	במה יכולה טכנולוגיית Java לעזור לי?.....
32	צעדים ראשונים (Win32).....
32	מה נדרש לכתובת תוכנית.....
32	יצירת התוכנית הראשונה.....
37	יצירת היישומון (applet) הראשון.....
38	שגיאות נפוצות (Win32).....
40	צעדים ראשונים (UNIX/Linux).....
40	מה נדרש לכתובת תוכנית.....
40	יצירת התוכנית הראשונה.....
44	יצירת היישומון (applet) הראשון.....
46	השגיאות הנפוצות ביותר (UNIX/Linux).....
47	צעדים ראשונים (MacOS).....
47	מה נדרש לכתובת תוכנית.....
47	יצירת התוכנית הראשונה.....
52	יצירת היישומון הראשון.....
54	הסבר על שגיאה (MacOS).....
55	מבט מעמיק ב-HelloWorld.....
55	מהו יישום?.....
59	האנטומיה של יישומון.....
64	שאלות ותרגילים.....

66דוגמאות קוד

פרק 2: מושגים בתכנות מכוון עצמים (OOP) 67

68 מהו אובייקט?

70 מהי הודעה (message)?

71 מהי מחלקה (class)?

74 ההבדל בין אובייקטים למחלקות

74 מהי הורשה (inheritance)?

76 מהו ממשק (Interface)?

77 מהו פולימורפיזם (Polymorphism)?

81 כיצד לתרגם את המושגים לקוד?

81 קוד מקור ותווית יישומון של ClickMe

82 אובייקטים ביישומון ClickMe

82 מחלקות ביישומון ClickMe

84 הודעות ביישומון ClickMe

84 הורשה ביישומון ClickMe

85 ממשקים ביישומון ClickMe

86 תיעוד ממשק התכנות - API

87 סיכום

88 שאלות ותרגילים

89 דוגמאות קוד

פרק 3: יסודות השפה 91

92 מזהים - Identifiers

94 משתנים

96 סוגי נתונים

98 שמות משתנים

99 טווח הכרה (Scope)

100 אתחול משתנים

101 משתנים סופיים

101 משתנים - סיכום

102 משתנים - שאלות ותרגילים

103 אופרטורים

104 אופרטורים אריתמטיים

108 אופרטורי יחס ואופרטורי תנאי

112 אופרטורים לוגיים ואופרטורי סיביות

115 אופרטורי השמה

117 אופרטורים נוספים

117 אופרטורים - סיכום

119 אופרטורים - שאלות ותרגילים

121	ביטויים, משפטים ובלוקים
121	ביטויים
124	משפטים
124	בלוקים
125	ביטויים, משפטים ובלוקים - סיכום
125	ביטויים, משפטים ובלוקים - שאלות ותרגילים
126	משפטי בקרת זרימה
127	משפטי while ו-do-while
128	משפט for
129	משפטי if-else
132	משפט switch
134	משפטים לטיפול בחריגים
135	משפטי הסתעפות
140	משפטי בקרת זרימה - סיכום
142	זרימת בקרה - שאלות ותרגילים
144	דוגמאות קוד

פרק 4: יסודות האובייקט ואובייקטים של נתונים פשוטים 145

146	מחזור חיים של אובייקט
148	יצירת אובייקט
153	שימוש באובייקטים
156	ניקוי אובייקטים לא נחוצים
157	אובייקטים - סיכום
158	אובייקטים - שאלות ותרגילים
159	תווים (characters) ומחרוזות (strings)
159	תווים
161	מחרוזות וחוצצי מחרוזות (string buffers)
162	יצירת מחרוזות וחוצצי מחרוזות
164	קבלת האורך של מחרוזת או חוצץ מחרוזת
165	קבלת תווים על פי אינדקס ממחרוזת או מחוצץ מחרוזת
166	חיפוש תו או תת-מחרוזת בתוך מחרוזת
168	השוואת מחרוזות וחלקי מחרוזות
170	טיפול במחרוזות
171	שינוי חוצצי מחרוזת
172	מחרוזות והמהדר
173	תווים ומחרוזות - סיכום
176	מספרים ומחלקות עוטפות
177	המחלקות של Number
180	המרת מחרוזות למספרים
181	המרת מספרים למחרוזות

181	קביעת פורמט למספרים
182	קביעת פורמט לשערי מטבע
183	קביעת פורמט למספרים על ידי פורמטים מותאמים
185	אריתמטיקה בסיסית ויותר
190	מספרים - סיכום
191	מספרים - שאלות ותרגילים
192	מערכים
192	יצירת מערכים ושימוש בהם
196	מערכים של אובייקטים
199	מערך של מערכים - מטריצות
203	מערכים - שאלות ותרגילים
204	דוגמאות קוד

פרק 5: מחלקות והורשה 207

208	יצירת מחלקות
209	הכרזה על מחלקה
211	הכרזה על משתנה חבר
212	הגדרת שיטות
216	אספקת בנאים למחלקות
217	העברת מידע לשיטה או לבנאי
220	החזרת ערך משיטה
222	שימוש במילת המפתח this
223	בקרת גישה לחברי מחלקה
228	הבנת חברי מופע וחברי מחלקה
231	אתחול חברי מופע וחברי מחלקה
232	יצירת מחלקות - סיכום
233	יצירת מחלקות: שאלות ותרגילים
234	ניהול הורשה
234	דריסה והסתרה של שיטות
237	הסתרת משתנים חברים
237	שימוש ב-super
239	להיות צאצא של Object
243	כתיבת מחלקות, שיטות ומשתנים סופיים
245	כתיבת מחלקות ושיטות מופשטות
247	דוגמה מסכמת, והאופרטור instanceof
249	ניהול הורשה - סיכום
250	ניהול הורשה - שאלות ותרגילים
251	מחלקות מקוננות
253	מחלקות פנימיות
255	עובדות נוספות על מחלקות מקוננות

255.....	מחלקות מקוננות - סיכום
255.....	מחלקות מקוננות - שאלות ותרגילים
257.....	דוגמאות קוד

פרק 6: ממשקים וחבילות..... 259

260.....	יצירת ממשקים ושימוש בהם
261.....	הגדרת ממשק
263.....	מימוש של ממשק
264.....	שימוש בממשק כסוג נתונים
264.....	ממשקים אינם יכולים לגדול
265.....	ממשקים - סיכום
265.....	ממשקים - שאלות ותרגילים
266.....	יצירת חבילות ושימוש בהן
267.....	יצירת חבילה
268.....	שם חבילה
273.....	יצירת חבילות ושימוש בהן - סיכום
273.....	יצירת חבילות ושימוש בהן - שאלות ותרגילים
274.....	דוגמאות קוד

פרק 7: טיפול בשגיאות בעזרת חריגים - Exceptions .. 275

276.....	מהו חריג (exception)?
277.....	כלל Handle or Declare
278.....	לכידת החריגים והטיפול בהם
279.....	הבלוק try
280.....	הבלוק catch
281.....	הבלוק finally
282.....	להרכיב הכל ביחד
287.....	הכרזה על חריגים הנזרקים על ידי השיטה
287.....	כיצד לזרוק חריגים
288.....	המשפט throw
289.....	המחלקה Throwable ותת-המחלקות שלה
290.....	יצירת מחלקות חריגים משלך
292.....	חריגי זמן-ריצה - המחלוקת
292.....	יתרונות מנגנון החריגים
293.....	יתרון 1: הפרדת הקוד לטיפול בשגיאות מהקוד ה"רגיל"
295.....	יתרון 2: הולכת שגיאות במעלה מחסנית הקריאות
296.....	יתרון 3: יצירת קבוצות של שגיאות והפרדה בין סוגים שונים של שגיאות
297.....	חריגים - סיכום
298.....	שאלות ותרגילים
300.....	דוגמאות קוד

פרק 8: ביצוע שתי משימות ויותר בו-זמנית

301	בעזרת מטלות (Threads)
303	מהי מטלה (thread)?
305	המחלקות Timer ו-TimerTask
306	עצירת מטלות של מוני זמן
307	ביצוע משימה שוב ושוב
309	מידע נוסף אודות מוני זמן
309	יישום השיטה run
310	הרחבת המחלקה Thread ודריסת השיטה run
311	מימוש הממשק Runnable
313	מתי להשתמש בממשק Runnable
313	מחזור החיים של מטלה
314	יצירת מטלה
315	הפעלת מטלה
316	איך להפוך מטלה ל-Not Runnable
317	עצירת מטלה
318	השיטה IsAlive
318	מהי קדימות (priority) של מטלה?
319	מירוץ המטלות
320	מטלות אנוכיות (Selfish Threads)
321	מנות זמן (Time Slicing)
323	סיכום של קדימות מטלה
323	סינכרון מטלות
324	דוגמת יצרן/צרכן
326	נעילת אובייקט
328	השיטות wait ו-notifyAll
330	הרצת הדוגמה יצרן/צרכן
331	איך למנוע הרעבה (Starvation) ונעילה ללא מוצא (Deadlock) של מטלות
333	הקבצת מטלות
334	קבוצת מטלות
334	יצירת מטלה ושיוך מפורש לקבוצה
335	איתור הקבוצה של מטלה
335	השימוש במחלקה ThreadGroup
339	סיכום מטלות
340	תמיכת חבילות במטלות
340	תמיכת Java במטלות
340	תמיכה למטלות בזמן ריצה
341	שאלות ותרגילים
342	דוגמאות קוד

פרק 9: קלט/פלט – קריאה וכתיבה של נתונים 343

344	מבט כללי על תזרימי קלט/פלט
345	תזרימי תווים
346	תזרימי בתים
347	מחלקות-על לטיפול בקלט/פלט
348	השימוש בתזרימים
350	שימוש בתזרימי קובץ
352	השימוש בתזרימי צינור
354	כריכת תזרים
355	שרשור קבצים
357	עבודה עם תזרימי מסנן
364	רישום אובייקטים
365	איך לרשום אובייקטים
366	רישום אובייקטים עבור מחלקותיך
370	קבצים לגישה אקראית
371	שימוש בקבצים לגישה אקראית
372	כתיבת מסננים עבור קבצים לגישה אקראית
375	מחלקות וממשקים נוספים
376	לסיכום נושאי קריאה וכתיבה
376	שאלות ותרגילים
378	דוגמאות קוד

פרק 10: רכיבים לבניית ממשק משתמש גרפי – AWT .. 379

380	רכיבי AWT
382	כללים לשימוש ברכיבים
385	לחצנים
387	בדי ציור
389	תיבות סימון
391	תיבות רשימה נפתחות
393	תיבות דו-שיח
396	חלונות מסגרת
398	תוויות
399	תיבות רשימה
403	תפריטים
411	פנלים
412	פסי גלילה
414	חלונות נגללות
417	אזורי טקסט ושדות טקסט
419	כתיבת מטפלים באירועים (Event handlers)
422	כללים לכתיבת מאזינים לאירועים

425	כתיבת מאזין לפעולה
427	כתיבת מאזין לכוונון
429	כתיבת מאזין לרכיב
432	כתיבת מאזין למכולה
434	כתיבת מאזין למיקוד
437	כתיבת מאזין לפריט
439	כתיבת מאזין למקש
443	כתיבת מאזין לעכבר
447	כתיבת מאזין לתנועת עכבר
449	כתיבת מאזין לטקסט
452	כתיבת מאזין לחלון
456	יצירת מאזינים בעזרת מחלקות אנונימיות
458	ארכיטקטורת הרכיבים
459	הטיפול של עמיתים באירועים

פרק 11: פריסת רכיבים בתוך מכולה 461

461	מנהל פריסה (layout manager) ותפקידיו
462	שימוש במנהלי פריסה
464	כללים לשימוש במנהלי פריסה
466	הפעלת BorderLayout
467	הפעלת CardLayout
469	הפעלת FlowLayout
470	הפעלת GridLayout
471	הפעלת GridBagLayout
479	יצירת מנהל פריסה מותאם אישית
480	סידור רכיבים ללא מנהל פריסה (מיקומים מוחלטים)

פרק 12: ממשק משתמש גרפי – Swing 483

484	מבט כללי על Swing והמעבר מ-AWT
484	המעבר מ-AWT ל-Swing
485	הידור והרצת תוכניות Swing
485	תוכנית Swing הראשונה
488	דוגמה שנייה : SwingApplication
488	מראה ותחושה (Look and Feel)
489	הצבת לחצנים ותוויות
490	טיפול באירועים
492	הוספת גבולות סביב רכיבים
493	דוגמה שלישית : CelsiusConverter
494	הוספת HTML
495	הוספת סמל (Icon)

496 LunarPhases : דוגמה רביעית :
498 גבולות מורכבים
498 (Combo Boxes) תיבות משולבות
500 תמונות מרובות
500 VoteDialog : דוגמה חמישית :
501 (Radio Buttons) לחצני אפשרויות
502 (Dialogs) תיבות דו-שיח
507 ניהול פריסה
507 השימוש במנהלי פריסה
510 Swing ו- מטלות
510 כלל המטלה הבודדת
512 איך לבצע קוד במטלה משגרת אירועים
513 Swing ויזואלי של רכיבי
519 סיכום
520 שאלות ותרגילים
522 דוגמאות קוד

פרק 13: רישות ב- Java

524 (Networking) סקירה כללית על רישות
524 שירותי הרישות ב- Java
525 יסודות הרישות
530 URL עבודה עם כתובות
531 URL מהו ?
532 URL יצירת
535 URL פירוק כתובת
536 URL קריאה ישירות מ-
538 URL התחברות ל-
538 קריאה מאובייקט URLConnection וכתובה אליו
543 (Sockets) הכל אודות שקעים
544 מהו שקע ?
545 קריאה משקע וכתובה אליו
548 כתיבת צמד לקוח/שרת
556 (Datagrams) הכל אודות צורות נתונים
557 מהו צרור נתונים
557 כתיבת לקוח ושרת של צרור נתונים
558 QuoteServerThread המחלקה
560 QuoteClient המחלקה
562 הרצת השרת והלקוח
563 (Broadcasting) שידור ללקוחות מרובים

נספח א': בעיות שכיחות ופתרון 567

567	בעיות בתחילת הדרך
568	בעיות מהדר
571	שגיאות סמנטיות
571	בעיות מפרש
572	בעיות תכנות כלליות
573	בעיות עם יישומונים
575	בעיות בממשק משתמש
575	רכיבי Swing
578	פריסה (Layout)
579	טיפול באירועים
579	גרפיקה ב-Swing
581	הסבת תוכניות ל-Swing

נספח ב': יישומונים לאינטרנט (Applets) 583

584	מבט כללי על יישומונים
584	יישומון פשוט
585	מחזור הפעילות של יישומון
587	שיטות לציוני דרך
588	שיטות לציון ולטיפול באירועים
590	שיטות להוספת רכיבי UI
590	שיטות לשימוש ברכיבי UI
592	מה יכולים יישומונים לעשות ובאילו מגבלות
592	מבחן דרך ליישומון
593	סיכום
594	רכיבי AWT
595	סקירת רכיבי AWT העיקריים
598	ניצול רכיבי API של יישומון
599	איתור וטעינה של קבצי נתונים
600	הצגת הודעות מצב קצרות
600	הצגת מסמכים בדפדפן
602	שליחת הודעות ליישומונים אחרים
606	השמעת צלילים
608	הגדרת פרמטרים של יישומון והשימוש בהם
613	התגית <APPLET>
617	שיקולים מעשיים בכתובת יישומונים
617	מגבלות אבטחה
619	יצירת ממשק משתמש
621	הצגת מידע אבחון בתזרימים סטנדרטיים של פלט ושגיאות
622	קריאת תכונות מערכת

624	מטלות ביישומוני AWT
629	השלמת היישומון
629	לפני הפצת היישומון
630	היישומון המושלם בהחלט
632	יישומוני Swing
632	הרצת יישומוני Swing
633	כתיבת יישומוני Swing
636	הפעלת יישומון עם תוסף Java
637	המרת יישומוני AWT ליישומוני Swing
638	מידע נוסף
638	דוגמאות קוד

641(Collections) אוספים

642	מבוא
642	מהי מסגרת אוספים?
643	יתרונות העבודה באוספים
644	חסרונות מסגרת האוספים
644	ממשקים
646	ממשק Collection
649	ממשק Set
653	ממשק List
661	ממשק Map
670	סידור אובייקטים
677	הממשק SortedSet
680	ממשק SortedMap
682	מימושים
682	מימושים למטרות כלליות
685	מימושי עטיפה
688	מימושי נוחות
689	אלגוריתמים
690	מיון (Sorting)
692	ערבוב (Shuffling)
693	טיפול שגרת בנתונים
693	חיפוש
693	מציאת ערכים קיצוניים
694	מימושים מותאמים אישית
694	הסיבות לכתיבת מימוש משלך
695	איך לכתוב מימושים מותאמים אישית

697	יכולת פעולה הדדית (Interoperability)
697	תאימות
699	תכנון API

701 נספח ד': שיטות Thread שהוצאו משימוש

701	מדוע Thread.stop הוצאה משימוש?
702	האם ניתן לתפוס את ה ThreadDeath ולתקן את האובייקט הפגום?
704	מדוע Thread.resume ו- Thread.suspend?
704	במה עלי להשתמש במקום Thread.resume ו- Thread.suspend?
707	מה בדבר Thread.destroy?
708	מדוע Runtime.runFinalizersOnExit הוצאה משימוש?

709 נספח ה': Reference

709	מה הקשר בין Java לקפה?
710	מילות מפתח בשפת Java.
711	סדר העדיפויות של האופרטורים.
712	התגית <APPLET>
714	מוסכמות POSIX עבור ארגומנטים של שורת הפקודה.
715	סביבות פיתוח משולבות.
715	עזרה לעדכון הנתביב.
716	עדכון המשתנה PATH (Win32).
718	עדכון המשתנה PATH (Unix).

721 אינדקס

התקליטור המצורף

התקליטור המצורף לספר זה מכיל:

- ♦ את הספר באנגלית, בגרסת HTML ובגרסת PDF.
- ♦ כל התוכניות המוזכרות בספר (ראה הפניות בספר).
- ♦ דוגמאות הקוד מהספר (לא כולן). רובן בתיקה JavaTutorial. הקבצים בפרק 10-11 נמצאים בתיקה Chap10-11 והקבצים בפרק 13 נמצאים בתיקה Chap13.
- ♦ טבלאות "דוגמאות קוד" המפרטות את מראי המקום של התוכניות בתקליטור. לעיתים, במקום בו יש צורך בכך, יש הפניה לאתר באופן ישיר.
- ♦ JDK בגרסאות שונות, המותאמות לעבודה בסביבות Windows, UNIX או Mac.
- ♦ מחלקות Java חיוניות.
- ♦ תוכנת Adobe Acrobat Reader לקריאת קבצי PDF.

העתקת קבצים מהתקליטור

בעת העתקת קבצים מתקליטור לכוונן הדיסק הקשיח במחשב מקבלים הקבצים המועתקים את התכונה **קריאה בלבד** (Read Only). כדי שניתן יהיה לבצע בקבצים אלה שינויים ולשמור אותם לאחר מכן, יש להקפיד ולבטל את התכונה **קריאה בלבד**:

העתק את הקבצים הרצויים מהתקליטור לכוונן הדיסק הקשיח; סמן את הקובץ/ים; לחץ לחיצה ימנית על הקבצים המסומנים; מתפריט הקיצור בחר **מאפיינים**; בטל את הסימון ליד תיבת הסימון **לקריאה בלבד**; לחץ **אישור**.

Acrobat Reader - התקנה

כדי לקרוא ולהדפיס את קבצי PDF שבתקליטור המצורף יש להתקין תוכנה זו. התוכנה גם מאפשרת חיפוש בעברית ובאנגלית במסמך המוצג. בנוסף, בעזרת תוכנה זו תוכל לקרוא את המסמכים שההוצאה מפרסמת באתר האינטרנט. התוכנה פועלת בסביבת מערכות הפעלה **Windows 95 ומעלה**. להתקנת התוכנה פעל כך:

לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**; בתיבת הטקסט **פתח את** הקלד את הפקודה **X:\Adobe Acrobat\ARME4Heb.exe** (החלף את האות X באות המייצגת את כונן התקליטורים שלך D, E או כל אות אחרת) ולחץ על **אישור**; אשף ההתקנה מתקין את הרכיבים הנדרשים. עליך ללחוץ על **Next**, **Accept**, ו-**Next** פעם נוספת כדי לבצע את ההתקנה; בסיום ההתקנה עשויה להופיע על המסך תיבת דו-שיח **התנגשות בין גירסאות** ומייד אחר כך להיעלם. במקומה תופיע על המסך תיבת הודעה של תוכנית ההתקנה. לחץ על **אישור** ובתיבת הדו-שיח **התנגשות בין גירסאות** ששבה להופיע לחץ על **כן**, כדי לשמור את גרסת הקובץ שלך.

התקנת ערכת הפיתוח של Java בסביבת חלונות

אם ברשותך Windows 9x/ME/NT/2000/XP, פעל באופן הבא :

שים את התקליטור בכונן ; פתח את **סייר Windows** (Windows Explorer) ; עבור לכונן התקליטורים ; עבור לתיקיה **SDKs** שבתקליטור ; עבור לתיקיה **j2se13** ; עבור לתיקיה **win32** ; לחץ לחיצה כפולה על קובץ **j2sdk1_3_0-win.exe** . פעל בהתאם להנחיות שעל המסך . להלן סדר הלחצנים שיש לחוץ בהתקנה רגילה (מימין לשמאל) : Yes ,Next ,Next ,Next , המתן לסיום העתקת הקבצים ולסיום לחץ Finish .

עבור לשורש כונן C ; לחץ לחיצה ימנית בעכבר על קובץ **Autoexec.bat** ; מתפריט הקיצור בחר **ערוך** (Edit) . **פנקס הרשימות** (Notepad) נפתח ומציג את תוכן הקובץ ; במידת הצורך הקש **Ctrl+Left Shift** כדי שתוכן הקובץ יוצג צמוד לשמאל ; עבור לשורה האחרונה והוסף את השורה הבאה : `SET PATH=%PATH%;C:\JDK1.3\BIN` .

בתפריט **קובץ** (File) בחר **שמור** (Save) ; סגור את **פנקס הרשימות** (Notepad) ; בחר בתפריט **התחל** (Start) באפשרות **כיבוי** (Shut Down) ובתיבת הדו-שיח **סגירת Windows** בחר באפשרות **הפעלה מחדש** (Restart) .

כדי לבדוק אם ההתקנה הצליחה :

במערכת **Windows ME** : פתח את תפריט **התחל** (Start) , **תוכניות** (Programs) , **עזרים** (Accessories) ובחר באפשרות **שורת הפקודה של MS-DOS** .

במערכת **Windows 9x** : פתח את תפריט **התחל** (Start) , **תוכניות** (Programs) ובחר באפשרות **הפניה ל-MS-DOS** .

בחלון **MS-DOS Prompt** בעל הרקע השחור רשום **C:\Windows>** ולאחריו סמן מהבהב. הקלד **java** והקש על מקש Enter . אם המחשב מגיב בהודעה בסגנון :

```
C:\WINDOWS>java
Usage: JAVA.EXE [-options] class [args...]
           (to execute a class)


:where options include
help      print out this message-
version   print out the build version-
```

משמע שההתקנה הצליחה. אם מתקבלת ההודעה :

```
Bad command or file name
```

משמע שההתקנה לא הצליחה ועליך לחזור ולבצע החל מסעיף 9 (גמר התקנה). כדי לקבל פרטים נוספים על ההתקנה ניתן לפנות לאתר Sun באינטרנט :

<http://java.sun.com>

כדי לצאת מחלון MS-DOS Prompt יש לסגור את החלון בעזרת  או להקליד exit ולהקיש Enter .

הקדמה

ערכת הפיתוח (Java Development Kit) JDK יצאה לשוק בראשית 1995 ומאז פעל צוות הפיתוח של Sun Microsystems באופן עקבי לשפר את פלטפורמת Java. במקביל, הספר שלפניכם משקף התפתחות זו והינו תרגום של The Java Tutorial, מהדורה שלישית.

מהדורה זו מתעדת את ממשקי התכנות (APIs) של Java 2 Software Development Kit (או SDK), גירסה 1.3. מכיון שגרסאות תוכנה קודמות עדיין קיימות ויש לתחזק אותן, ספר זה מתאים גם עבור גרסאות 1.1 ו-1.2, כפי שתוכל לראות במהלך ההסברים והדוגמאות. במיוחד השארנו את הפרק הון ב-AWT, בנוסף להסבר המקיף על ממשק Swing שהינו חדש בפלטפורמת Java.

במהדורה זו הוספנו שאלות ותרגילים בפרקים המתאימים, כדי לסייע ללומד לחזור על החומר ולבחון את כישוריו לאחר הלימוד. כדי למנוע ממתחילים תלאות ותסכול בצעדיהם הראשונים, ניתן נספח המכיל רשימת שגיאות תכנות ותקלות נפוצות והצעת פתרונות להם. סיכומים שניתנו בחלקי הספר השונים מאפשרים ללומד לבחון את מה שנעשה ומה שצפוי בהמשך הלימוד.

ספר זה מבוסס על תוכנית הלימוד המקוון (online tutorial) עבור פלטפורמת Java, שניתן למצוא באתר Web של Sun Microsystems:

<http://java.sun.com/docs/books/tutorial/index.html>

(הערות הוצאה: שים לב שכתובות האינטרנט נכונות ליום הוצאת הספר. את כל התוכניות שבספר תוכל למצוא בתקליטור המצורף. ראה הפניות בספר).

כמו בגירסה המקוונת, גם ספר זה משקף את טכנולוגיית Java העדכנית ביותר. עם זאת, בשונה מהגירסה המקוונת, ספר זה מתמקד בכל ממשקי התכנות (APIs) אשר דרושים למתכנתים מתחילים ומתקדמים. לאחר שליטה בחומר שבספר זה, תוכל להמשיך וללמוד עוד על פלטפורמת Java מתוך המסמכים שבאתר.

כוונתנו היתה ליצור ספר שכיף לקרוא וללמוד בו, מדריך המותאם למטלות העבודה ולגישה של המתכנת בעולם מוכוון עצמים, ויש בו תרגילים רבים לתמיכה בתהליך הלמידה.

למי מיועד הספר?

הספר פונה אל מפתחים ומתכנתים, מתחילים ומתקדמים.

- ♦ **מתכנתים מתחילים** ימצאו תועלת רבה מקריאה מסודרת בספר מתחילתו ועד סופו. עליהם להקדיש תשומת לב להוראות התכנות הבנויות בשיטת צעד-אחר-צעד ולהריץ בעצמם את הדוגמאות שבמהלך הלימוד ולסיום – עליהם להשיב לשאלות ולהריץ את התרגילים המיועדים לחזרה על החומר שנלמד.
- ♦ **מפתחים ומתכנתים המנוסים בשפה פרוצדורלית**, כמו C או C++ למשל, יתחילו בוודאי בפרק העוסק בתפישת תכנות מוכוון העצמים (OOP) של Java.
- ♦ **מפתחים ומתכנתים מנוסים בתכנות מוכוון עצמים** (OOP) יעברו מייד לנושאים מתקדמים המעסיקים ומעניינים אותם, כמו יישומונים (applets), מחלקות (classes) מיוחדות, או ממשקי משתמש (כמו AWT ו-Swing).

ללא חשיבות לסיווג שתקבע לעצמך, תוכל למצוא נתיב לימוד ועניין שיש בו תכלית ותועלת עבורך ומתאים לצרכיך.

כיצד לקרוא בספר

הספר עוצב בדרך שתאפשר לקרוא לעבור בו מפרק לפרק או משלב לשלב באופן סדרתי. בכל מקרה שבו נושא מסוים נדון בפרק או בסעיף נוסף בספר, תמצא הפניה מתאימה. נצל זאת כדי להרחיב את ידיעותיך וללמוד בשיטה המעגלית.

דוגמאות הקוד הרבות שבספר (אך לא כולן) זמינות באופן מקוון באתר (נכון ליום הוצאת הספר), ולנוחותך הן נמצאות גם בתקליטור המצורף. בסוף הפרקים (אך לא בכלם) תמצא טבלת "דוגמאות קוד" המפרטת את מראי המקום של התוכניות בתקליטור. לעיתים יש הפניה לאתר באופן ישיר במקום שבו יש צורך בכך.

כפי שידוע לך, ארגונים נוטים לעדכן ולשנות את האתרים ולעיתים קשה לאתר את המידע, גם אם הוא עדיין קיים, אך "מוסתר" בדרך כלשהי. באתר תמצא חומר עדכני בנושאים הקשורים לפיתוח ב-Java. כדי לדעת מה יש ומה נוסף באתר, עיין בכתובת זו:

<http://java.sun.com/docs/books/tutorial/books/3e/index.html>

או בתקליטור

JavaTutorial/books/3e/index.html

הצוות המקצועי

מחברות הספר **Kathy Walrath , Mary Campione** ו- **Alison Huml** הן חברות בצוותי הפיתוח של Sun Microsystems וחברו ספרים וחומר הדרכה בתחום. במאמץ להוציא מתחת ידיהן מוצר איכותי, סייעו להן בקריאה ובבדיקת הטיוטה ובהרצת דוגמאות התכנות חברים וחברות רבים מצוותי הפיתוח של Sun. המחברות מודות לכל מי שעזר ותמך בהן בעבודתן.

היועץ המקצועי, **משה יצחקי**, הינו מרצה בכיר מוסמך Sun לקורסי Java (Sun Certified Java Programmer, Developer & Instructor) אשר מדריך במרכזי הדרכה מובילים בארץ. מחבר הספר "שפת C – נושאים מתקדמים ומולטימדיה" שיצא לאור על ידי הוד-עמי.

פרק 1

הכרת טכנולוגיית Java

בפרק זה נכיר את **טכנולוגיית Java** (Java™ technology), ולמה "טכנולוגיה"? – מפני ש-Java הינה יותר משפת תכנות, היא גם "סביבת עבודה", או "פלטפורמה". תחילה נסביר מהי פלטפורמת Java (Java platform) ומה השימושים שלה. בהמשך הפרק נלמד צעד-אחר-צעד כיצד להדר (compile) ולהריץ שתי תוכניות פשוטות בסביבות העבודה Win32, UNIX/Linux ו-MacOS¹. לאחר מכן נבחן את קטעי הקוד של התוכניות כדי להבין כיצד הן פועלות.

חברת Sun Microsystems מספקת **ערכת פיתוח תוכנה – SDK** (Software Development Kit) הכוללת אוסף כלים מינימלי. כלים אלה מאפשרים להדר ולהריץ תוכניות מתכנתים מתקדמים יעדיפו בוודאי להשתמש ב**ערכת פיתוח משולבת – IDE** (Integrated Development Environment)² המיועדת למקצוענים. לקבלת פירוט של ערכות הפיתוח מסוג IDE הזמינות, עיין בנספח ה'.

בפרק זה:

- ♦ אודות טכנולוגיית Java (Java Technology)
- ♦ במה יכולה טכנולוגיית Java לעזור לי?
- ♦ צעדים ראשוניים (Win32)
- ♦ צעדים ראשוניים (UNIX/Linux)
- ♦ צעדים ראשוניים (MacOS)
- ♦ מבט מעמיק ב-HelloWorld

¹ רשימה עדכנית של Java ports לסביבות עבודה נוספות תוכל למצוא בכתובת:

<http://java.sun.com/cgi-bin/java-ports.cgi>

² למעשה, Java 2 SDK, Standard Edition v.1.3 משולבת עם IDE בשם Forte™ for Java™ Community Edition. גירסה זו כלולה בתקליטור המצורף לספר זה.

אודות טכנולוגיית Java (Java Technology)

אנו שומעים רבות על הטכנולוגיה הנקראת **Java™**. נשאלת השאלה, מהי בדיוק טכנולוגיה זו? בסעיפים הבאים נסביר שטכנולוגיה זו היא למעשה שילוב של שפת תכנות ושל פלטפורמת הרצה גם יחד.

שפת התכנות Java

Java היא שפת תכנות עילית (high-level language) הניתנת לאפיון על ידי כל מילות המפתח הבאות¹:

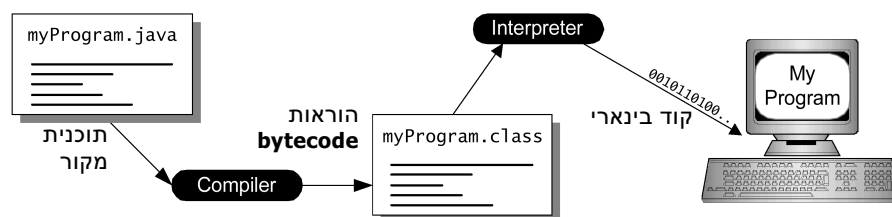
- ◆ שפה פשוטה (simple)
- ◆ שפה מוכוונת עצמים (object oriented)
- ◆ תומכת בתוכניות מבוזרות (distributed)
- ◆ מופעלת על ידי מפרש (interpreted)
- ◆ תוכניות אמינות ו"חסינות" (robust)
- ◆ שפה מאובטחת (secure)
- ◆ אינה תלויה בחומרה (architecture neutral)
- ◆ מאפשרת ניידות קוד (portable)
- ◆ בעלת ביצועים מהירים (high performance)
- ◆ מאפשרת ריבוי מטלות המתבצעות במקביל (multithreaded)
- ◆ תומכת בתוכניות דינמיות (dynamic)

ברוב שפות התכנות המקובלות, כדי לאפשר הרצת תוכנית במחשב עליך להדר (compile) את התוכנית לקובץ הרצה (executable file) הכתוב בשפת מכונה (machine language), שהינה רצפים של ספרות "0" ו-"1". שפת התכנות Java ייחודית בכך שתוכניות הכתובות בשפה זו אינן מהודרות לקובץ הרצה בשפת מכונה, כמקובל ב-Java, העברת תוכנית המקור משפת המקור של המתכנת לתוכנית הרצה נעשית בשני שלבים.

¹ כל אחת ממילות מפתח אלו מוסברת במפרט של ג'ימס גוסלינג והנרי מקגילטון, **סביבת השפה Java** (The Java Language Environment). תוכל למצוא מפרט זה באתר Java באינטרנט: <http://java.sun.com/docs/white/langenv/index.html>

באמצעות **מהדר Java** (Java compiler) ממירים את תוכנית המקור (קובץ עם סיומת .java). לשפת ביניים הקרויה **Java bytecodes**. בתום תהליך ההידור של התוכנית מקבלים קבצים בעלי סיומת .class, שבהם הוראות bytecode. על מנת להריץ את התוכנית צריך להפעיל **מפרש Java** (Java interpreter) שתפקידו לקרוא את קובץ bytecode, לנתח את הפקודות הרשומות בו, לפרש אותו ולהריץ את התוכנית במחשב.

תהליך ההידור משפת Java ל-bytecodes מתבצע פעם אחת בלבד, ואילו המפרש פועל בכל פעם שמריצים את התוכנית. בתרשים 1 מוצג התהליך:



תרשים 1: תוכניות הכתובות בשפת Java מהודרות תחילה ומפורשות בזמן ההרצה.

ניתן לראות את הוראות Java bytecodes כהוראות קוד מכונה עבור **המכונה הווירטואלית של Java** (Java Virtual Machine – Java VM). כל מפרש Java – אשר עשוי להיות כלי פיתוח של Java, או דפדפן אינטרנט שיכול להריץ יישומונים (applets) – מיישם את המכונה הווירטואלית.

הקורא בוודאי תוהה ושואל, מהי הסיבה שגרמה להחלטה שקוד מקור של Java לא יהודר ישירות לקובץ הרצה כמקובל, אלא יומר להוראות bytecode. ובכן, תהליך זה מקנה ל-Java יתרון רב. הוראות Java bytecodes מאפשרות להפוך את האימרה "כתוב פעם אחת, הרץ בכל מקום" לממשות. תוכל להדר תוכנית ל-Java bytecodes בכל סביבה בעלת מהדר Java, וקוד הביניים הזה, bytecode, יוכל לרוץ לאחר מכן בכל יישום של המכונה הווירטואלית. כלומר, כל זמן שבמחשב נמצאת המכונה הווירטואלית, אותה תוכנית שכתובה בשפת התכנות Java יכולה לרוץ על Windows NT, Solaris, או על iMac, כמוצג בתרשים 2.

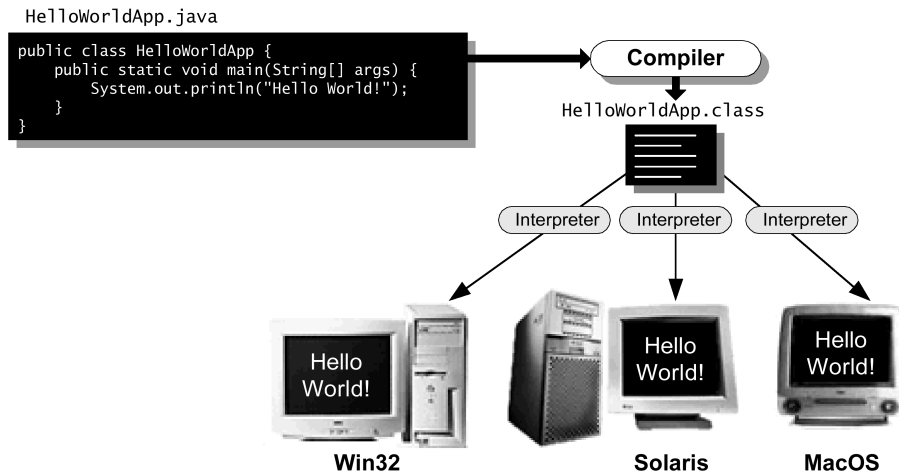
אפשר לומר שהמכונה הווירטואלית יודעת להפוך הוראות שאינן תלויות מערכת, להוראות שמערכת הפעלה הייחודית למחשב מסוים מבינה אותן ויכולה להפעילן.

הפלטפורמה של Java (Java Platform)

פלטפורמה היא סביבת התוכנה או החומרה שבה תוכנית רצה. לפניך כמה מהפלטפורמות המקובלות: Windows 2000, Linux, Solaris, ו-MacOS. את רוב סביבות העבודה ניתן לתאר כצירוף של מערכת הפעלה עם חומרה. הפלטפורמה של Java שונה מרוב סביבות העבודה האחרות, מכיון שהיא פלטפורמת תוכנה בלבד הפועלת על פלטפורמות חומרה שונות.

הפלטפורמה של java כוללת שני מרכיבים :

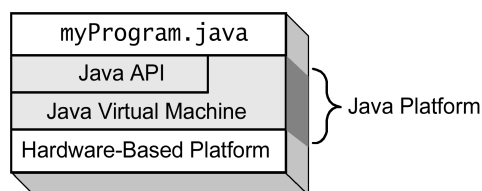
- ◆ המכונה הווירטואלית של Java – Java VM (Java Virtual Machine).
 - ◆ ממשק התכנות של Java – Java API (Java Application Programming interface).
- Java VM היא הבסיס לפלטפורמה של Java.



תרשים 2: תוכנית Java יכולה להיכתב פעם אחת, ולרוץ כמעט בכל מערכת הפעלה שפועל בה מפרש מתאים.

Java API הוא אוסף גדול של רכיבי תוכנה המספק יכולות שימושיות רבות, כגון פריטי **ממשק משתמש גרפי** – GUI (Graphic User Interface). Java API מקובץ בספריות רכיבים קשורים, המוכרות יותר בשם **חבילות** (packages). בסעיף הבא נפרט את הפונקציונליות שמספקות חלק מהחבילות הכלולות ב-Java API.

תרשים 3 מתאר תוכנית שרצה בפלטפורמת Java. כפי שהתרשים ממחיש, Java API והמכונה הווירטואלית מבודדים את התוכניות מתלות בחומרה.



תרשים 3: Java API והמכונה הווירטואלית מבודדים את התוכנית מתלות בחומרה.

קוד תלוי סביבה, או **קוד טבעי** (native code) הוא קוד, שלאחר הידורו, ניתן להריצו בסביבת חומרה מסוימת בלבד. כסביבה עצמאית, Java עשויה להיות קצת יותר איטית מקוד שתלוי בסביבה. עם זאת, מהדרים חכמים, מפרשים מתואמים היטב ומהדרי bytecodes בזמן אמת יכולים להתקרב לרמת ביצועי קוד שתלוי בסביבה, בלי לפגוע בניידות הקוד.

מה יכולה טכנולוגיית Java לעשות?

הסוגים השכיחים ביותר של תוכניות הנכתבות ב-Java הם **יישומונים** (Applets) ו**יישומים** (applications). אם גלשת באינטרנט, קרוב לודאי שאתה כבר מכיר יישומונים. יישומון הוא תוכנית שמצייתת למוסכמות מסוימות המאפשרות לה לרוץ בדפדפן תומך Java. כדי לראות יישומון בפעולה, ראה בתקליטור:

X:/JavaTutorial/getStarted/index.html

שם תוכל לראות את הקמע של Java, Duke, המנופף לך לשלום.



עם זאת, שפת התכנות Java אינה מיועדת לכתיבת יישומונים נחמדים ומשעשעים לאינטרנט בלבד. שפה זו היא שפת תכנות עילית לכל מטרה, היא גם פלטפורמה בעלת עוצמה רבה. השימוש בממשק תכנות היישומים העשיר מאפשר כתיבת סוגים רבים של תוכניות.

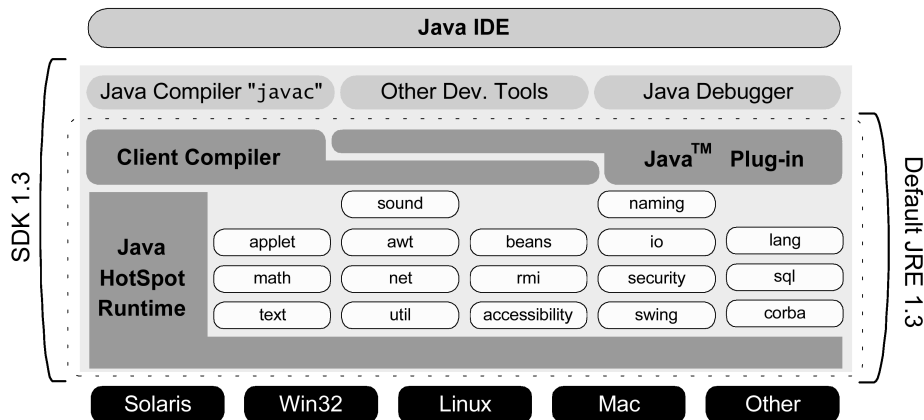
יישום (Application) הוא תוכנית עצמאית שרצה ישירות על פלטפורמת Java. **server** (תוכנית שרת) היא תוכנית ייחודית של שירות ותמיכה בלקוחות השרת. דוגמאות לתוכניות שרת הם שרתי web, שרתי דואר ושרתי הדפסה.

תוכנית ייחודית אחרת נקראת **servlet** (יישומון שרת). תוכנית servlet דומה ליישומון (applet), אלא שהיא פועלת בשרת. לעומת יישומונים שרצים בדפדפנים, תוכניות servlet רצות בתוך שרתי Java ומעצבים או מתאימים את השרת לדרישות המשתמש. מטלות אופייניות לשרת, המבוצעות באמצעות תסריטי CGI, ניתנות לביצוע גם באמצעות שפת התכנות Java אם בשרת פועלת המכונה הווירטואלית Java VM. היתרון בתוכניות servlet הוא ביכולתם של יישומוני שרת אלה לטפל בעומס תעבורה רב בשרת, ולהצליח בכך יותר מאשר יכולים תסריטי CGI. לפיכך, תוכניות servlet הן הבחירה המקובלת לבניית יישומי אינטרנט אינטרקטיביים, והם תופסים את מקום תסריטי CGI.

כיצד תומך ממשק התכנות Java API בכל סוגי התוכניות האלו? – הוא עושה זאת בעזרת חבילות רכיבי תוכנה שמספקות תחום פונקציונליות רחב. מימוש שלם של פלטפורמת Java מעניק את התכונות הבאות:

- ◆ **מרכיבים יסודיים** (the essentials): אובייקטים, מחרוזות, מטלות (threads), מספרים, קלט ופלט, מבני נתונים, מאפייני מערכת, תאריך וזמן, ועוד.
- ◆ **יישומונים** (applets): סט המוסכמות המשמשות ביישומוני Java.
- ◆ **רישיות** (networking): כתובות URL, TCP (Transmission Control Protocol), שקעי UDP (User Datagram Protocol) וכתובות IP (Internet Protocol).

- ◆ **התאמה אזורית** (Internationalization): תמיכה בכתיבת תוכניות מקומיות לאזורים שונים בעולם. תוכניות יכולות להתאים את עצמן אוטומטית למקומות מסוימים ולהיות מוצגות בשפה מתאימה.
 - ◆ **אבטחה** (Security): יכולת לספק אמצעי אבטחה ברמה הנמוכה והגבוהה, הכוללת חתימות אלקטרוניות, ניהול מפתח ציבורי ופרטי, בקרת גישה ואישורים (certificates).
 - ◆ **רכיבי תוכנה** (Software components): הידועים בשם JavaBeans™, יכולים להתאים לארכיטקטורת רכיבים קיימת, כגון ארכיטקטורות OLE/COM/Active-X של Microsoft, OpenDoc ו-Netscape של Live Connect.
 - ◆ **רישום אובייקטים** (Object serialization): מאפשר העברה של אובייקטים ברשת והתקשרות דרך RMI (Remote Method Invocation). כך הלקוח יכול להפעיל פונקציות השייכות לאובייקטים של תוכנית השרת.
 - ◆ **קישור למסדי נתונים של Java** – JDBC™ (Java Database Connectivity): מספק גישה אחידה לטווח מסדי נתונים יחסיים רחב.
- פלטפורמת Java כוללת ממשקי תכנות יישומים (APIs) לגרפיקה דו- ותלת-ממדית, נגישות, שרתים, שיתוף פעולה (collaboration), טלפוניה, דיבור, אנימציה ועוד. תרשים 4 מתאר מה כלול ב-Java 2 SDK.



תרשים 4: תיאור הערכה Java 2 SDK, Standard Edition v. 1.3. סביבת הריצה של Java 2 (JRE) מורכבת מהמכונה הווירטואלית (Java VM), ספריות הבסיס (core classes) של פלטפורמת Java וקבצי תמיכה. ערכת Java 2 SDK כוללת את JRE וכלי פיתוח, כמו מהדרים ותוכנות לניפוי שגיאות.

בספר זה נעסוק בשפה Java ובחלקים מתוך API הבסיסי שבהם משתמשים מתכנתים מתחילים, או מתקדמים, בתדירות הגבוהה ביותר. אם תצטרך מידע נוסף שאינו נמצא בספר זה, תוכל לבדוק בתקליטור:

X:/JavaTutorial/index.html

במה יכולה טכנולוגיית Java לעזור לי?

איננו יכולים להבטיח לך תהילה, עושר, או אפילו משרה אם תלמד Java. אבל סביר להניח ש-Java תהפוך את התוכניות שלך לטובות יותר ובפחות מאמץ בכתיבה מאשר בשפות אחרות. אנחנו מאמינים ש-Java תעזור לך לעשות את הדברים הבאים:

- ◆ **לימוד מהיר:** למרות ש-Java היא שפה מוכוונת עצמים בעלת עוצמה, היא קלה ללימוד, במיוחד למתכנתים שכבר מכירים את השפות C או ++C.
 - ◆ **כתיבת פחות קוד:** השוואת מדידות של מאפייני תוכנה (ספירת מספר המחלקות, השיטות או הפונקציות, וכו') מובילה למסקנה שתוכנית הכתובה ב-Java עשויה להיות קטנה פי ארבע מאותה תוכנית הכתובה ב- ++C.
 - ◆ **כתיבת קוד טוב יותר:** Java מעודדת הרגלי כתיבת קוד טובים. יכולתה לאסוף אשפה (garbage collection) עוזרת למנוע איבוד זיכרון. מכיון ש-Java היא שפה מוכוונת עצמים, ארכיטקטורת רכיבי JavaBeans וממשק תכנות היישומים הניתן בקלות להרחבה, מאפשרים שימוש חוזר בקוד שנכתב על ידי אחרים וכך למנוע באגים.
 - ◆ **פיתוח תוכניות מהיר יותר:** זמן הפיתוח ב-Java עשוי להיות קטן עד פי 2 מזמן הפיתוח של אותה התוכנית ב- ++C. מדוע? אתה כותב פחות שורות קוד עם Java ומכיון שזו שפת תכנות פשוטה יותר מ- ++C.
 - ◆ **הימנעות מתלות בפלטפורמה עם 100% Java טהורה:** תוכל לקיים ניידות של תוכנית על ידי הימנעות משימוש בספריות שנכתבו בשפות אחרות. לקבלת מידע נוסף, עיין במאגר המידע המקוון באתר: <http://java.sun.com/100percent/>
 - ◆ **כתוב פעם אחת, הרץ בכל מקום:** תוכניות המבוססות-Java בלבד מהודרות להוראות בקוד ביניים, bytecode, שאינן תלוי במכונה ולכן – יכולות לרוץ בכל פלטפורמה של Java.
 - ◆ **הפצת קלה יותר של תוכנה:** תוכל לשדרג את היישומים בקלות מתוך שרת מרכזי. יישומים מנצלים את תכונת Java, שמאפשרת למחלקות חדשות להיטען באופן דינמי ("on the fly"), ללא צורך בהידור כל התוכנית מחדש.
- הבה נתחיל את לימוד שפת Java בעזרת תוכנית פשוטה, המקובלת גם בלימוד של שפות אחרות: "Hello World". בהתאם למערכת ההפעלה שבה אתה משתמש, הינך מוזמן לקרוא את אחד מהסעיפים בפרק זה, הקרויים "צעדים ראשונים", שבהם תקבל הוראות מפורטות להידור התוכנית הראשונה בפלטפורמות Windows, בפלטפורמות UNIX ו-Linux, ובפלטפורמות MacOS. לאחר מכן, אל תחמיץ את הסעיף "מבט מעמיק ב-HelloWorld" שבו יש הסבר על מספר תכונות בסיסיות בשפת Java.

צעדים ראשונים (Win32)

בעזרת ההוראות המפורטות בהמשך תוכל לכתוב את התוכנית הראשונה שלך. הוראות אלו מתייחסות למשתמשים בפלטפורמות של Win32, הכוללות את מערכות Windows, החל בגירסה 95. נפתח ברשימת האמצעים הנחוצים לכתוב התוכנית הראשונה. לאחר מכן, נסקור את השלבים ביצירת יישום (application), השלבים ביצירת יישומון (Applet), וניתן הסברים על הודעות שגיאה שאתה עשוי להיתקל בהן.

מה נדרש לכתוב תוכנית

כדי לכתוב את התוכנית הראשונה אתה צריך:

1. **The Java 2 SDK, Standard Edition**: ערכה זו כלולה בתקליטור המצורף לספר זה. ניתן גם להוריד אותה מהאתר, או לבדוק אם זמינה גירסה עדכנית יותר: <http://java.sun.com/products/>
2. **עורך טקסט**: בדוגמה זו נשתמש בפנקס הרשימות (NotePad). כדי למצוא את פנקס הרשימות לחץ על לחצן **התחל**, בחר **תוכניות** < **עזרים** < **פנקס רשימות**. אפשר להשתמש גם בכל עורך טקסט פשוט אחר.

הוראות התקנה של Java 2 SDK נמצאות בעמודים 19-20.

הערה: אולי תשקול להיעזר בסביבת פיתוח משולבת (IDE) כדי לכתוב את התוכנית. ערכת Java 2 SDK, Standard Edition v. 1.3 נכללת עם IDE ונקראת Forte™ for Java™, Community Edition. גירסה זו נמצאת בתקליטור המצורף לספר זה.

יצירת התוכנית הראשונה

התוכנית הראשונה שאתה עומד לכתוב נקראת HelloWorldApp. זו תוכנית פשוטה המציגה את משפט הברכה "Hello World!". כדי ליצור תוכנית זו יהיה עליך לבצע כל אחת מהפעולות שלהלן:

- ♦ **יצירת קובץ מקור.** קובץ מקור מכיל טקסט הכתוב בשפת התכנות Java, כך שאתה ומתכנתים אחרים באותה השפה תוכלו לקרוא ולהבין אותו. תוכל להשתמש בכל עורך טקסט פשוט כדי ליצור ולערוך קבצי מקור.
- ♦ **הידור קובץ המקור לקובץ bytecodes.** המהדר javac מתרגם את הטקסט הכתוב בקובץ המקור להוראות שהמכונה הווירטואלית מבינה.

¹ לפני גירסה 1.2, נקראה ערכת הפיתוח שסופקה על ידי חברת Sun Microsystems בשם JDK (Java Developing Kit).

- ◆ **הרצת התוכנית הנמצאת בקובץ bytecode.** בשלב זה המפרש המותקן במחשב מיישם את המכונה הווירטואלית. בעזרת המכונה הווירטואלית המפרש לוקח את קובץ bytecode שיצר ומבצע את ההוראות (או הפקודות) אחת לאחת על ידי תרגומן להוראות בינאריות, בשפת מכונה, שהמחשב יכול להבין.

יצירת קובץ מקור

עומדות בפניך שתי אפשרויות ליצירת קובץ מקור. תוכל לשמור את הקובץ HelloWorldApp.java בדיסק הקשיח¹ ולחסוך הקלדה מרובה. לאחר מכן תוכל לגשת ישירות להידור קובץ המקור. לחילופין, תוכל לעקוב אחר ההוראות שלהלן.

תחילה הפעל את **פנקס הרשימות**. במסמך חדש הקלד את שורות הקוד האלו:

```
/**
 * The HelloWorldApp class implements an application that
 * displays Hello World! to the standard output.
 */
public class HelloWorldApp {
    public static void main(String[] args) {
        // Display "Hello World!"
        System.out.println("Hello World!");
    }
}
```

שים לב בעת ההקלדה: הקלד את כל הקוד, פקודות ושמות קבצים בדיוק כפי שהם מופיעים כאן. המהדר והמפרש **מבדילים** בין אותיות רגילות לרשיות (case sensitive) ולכן עליך להיות עקבי בשימוש בסוג האותיות בכל התוכנית. לדוגמה, HelloWorldApp אינו שקול ל-helloworldapp. אי שמירה על כלל זה גוררת שגיאות הידור.

לאחר מכן, שמור את הקוד בקובץ. משורת התפריטים בחר **קובץ < שמירה בשם**. בתיבת הדו-שיח **שמירה בשם** פעל כך:

- ◆ בתיבת הרשימה **שמור ב:** (Save As:) ציין את התיקיה שבה אתה רוצה לשמור את הקובץ. תוכל לשמור את הקובץ בכל תיקיה שתצפה. בדוגמה זו, הקובץ נשמר בתיקיה הנקראת java והנמצאת בכונן C.

- ◆ בתיבת הטקסט **שם הקובץ:** (File name:) הקלד "HelloWorldApp.java" ואל תשכח להקליד גם את הגרשיים משני צידי הטקסט. הגרשיים מאלצים את שמירת הקובץ כקובץ מסוג java ולא כקובץ מסוג ".txt".

¹ במהלך הספר אנו מציינים כאשר קוד המוזכר בטקסט **נמצא** בתקליטור. כמעט **בסוף כל פרק** תמצא טבלה ובה פירוט מיקום הקבצים בתקליטור המצורף.