



מבוא לתכנות מכוון עצמים - תח"ע

עצם עצם תרדוף



פוגה הכלבה ומיקי הכלב נפגשו כמדי יום ביומו בשעת בוקר מוקדמת בדשא שליך אשכול הפיס. כמובן שהם דנו על דברים שברומו של עולם, כגון מה כל אחד חלם בשנתו. למרבה ההפתעה הוברר להם ששניהם חלמו אותו חלום בדיוק, והוא "עצם עסיסית הצונחת מבעד העננים ונוחתת לידם". "העצם שנחתה לידי" אמרה פוגה "אורכה היה 20 ס"מ צבעה אפור ומשקלה כ 200 גרם אך כשבאתי לנגוס ממנה התעוררתי מהחלום". "העצם שלי", אמר מיקי, "אורכה כ 30 ס"מ, צבעה אפרפר ומשקלה כ 300 גרם אך כשבאתי לנגוס ממנה הגיע הכלב של השכנים וחטף לי אותה".
בווא ונבדוק את מושא חלומותיהם של הכלבים, העצם.

מהו עצם?

עצם (אובייקט) הוא המפתח להבנת הגישה התכנותית הקרויה **תמ"ע - תכנות מכוון עצמים** (Object Oriented Programming - OOP). אנו מוקפים בעצמים ממשיים בחיי היום-יום, כמו כלב, עט, מכונית, רדיו וכד'. עצמים ממשיים אלו חולקים שני מאפיינים: לכולם יש **תכונות** (attribute) המתארות את מצב העצם ולכולם יש **התנהגויות** (behavior). לדוגמא, לכלבים יש תכונות כמו שם, צבע, גזע, נובח, ויש להם גם התנהגות - נביחה, ריצה, כשכוש בזנב. גם למכונית יש תכונות כמו צבע, דגם, מספר רישוי ומהירות, ויש לה התנהגות כמו האצה, האטה, הדלק-אורות, קבע-מהירות (מהירות מבוקשת).

גם לעצם בתכנות מכוון עצמים, בדומה לעצם ממשי, יש תכונות והתנהגויות. תכונות העצם בתכנה נשמרות **במשתנה** (variable) אחד או יותר. משתנה הוא פריט המחזיק נתון או נתונים, בעלי שם מזהה. התנהגות העצם בתכנה מיושמת על ידי **פעולות** (methods).

ניתן לייצג עצמים ממשיים באמצעות עצמים של תכנה. נוכל לדאוג לייצג למשל את הכלב הממשי כעצם של תכנה בתכנית אנימציה, או לייצג את המכונית הממשית כעצם תוכנה בתכנית המדמה מרוץ מכוניות. ניתן גם להשתמש בעצם של תוכנה כדי לייצג עצמים מופשטים.

כל דבר שאובייקט התכנה יודע (תכונות העצם) ומסוגל לעשות (התנהגות העצם) בא לידי ביטוי על ידי המשתנים והפעולות השייכים לו. באובייקט תכנה המדמה את המכונית הממשית יכולו המשתנים ערכים שמציינים את מצבה הנוכחי של המכונית: הדגם הוא מאזדה לאנטיס, הצבע הוא לבן, המהירות היא 95 קמ"ש. ערכים אלו מייצגים את **מצב העצם** מפני שהם מכילים את התכונות של אובייקט מכונית מסויימת. ובמונחי תכנות מכוון עצמים, אובייקט שנקבע מצבו נקרא גם **מופע** (instance).

בנוסף למשתנים, יש למכונית המדומה בתכנה גם פעולות: לבלום, לשנות את המהירות ולהדליק אורות. פעולות אלו ידועות בשם **פעולות מופע** (instance methods) מפני שהן בודקות ומשנות את התכונות של מופע מסויים של המחלקה מכונית.

נשים לב שמשנתני העצם מכילים אך ורק תכונות הקשורות לעצם הספציפי. לעצם מסוג מכונית לא תהינה תכונות מסוג נובח? ואין לו שם, כשם שלעצם מסוג כלב לא תהיה תכונה מסוג מצב הבלמים או דגם.

כל העולם עצמים

אם נבחן פעולות יומיומיות, נראה שאנו מוקפים בעצמים. למשל, כשאנו יוצאים מהבית, אנו לוקחים עימנו טלפון-סלולארי, מפתחות וארנק.

מהו טלפון-סלולארי?

טלפון סלולארי הוא עצם שיש לו תכונות כמו מספר-מנוי, זכרונות וסוללה. על הטלפון ניתן לבצע פעולות כמו לשלף מספר מהזיכרון, לחייג, לסיים שיחה ולבדוק את מצב הסוללה.

מהו ארנק?

ארנק הוא עצם שיש לו תכונות כמו צבע, וסוג-חומר ובתוכו יש מטבעות ושטרות. הפעולות שניתן לבצע הן להוציא כסף מהארנק, להכניס כסף לתוכו ולבדוק מהו סכום-הכסף הכללי.

מהו מפתח?

גם מפתח הוא עצם שיש לו תכונות כמו סוג וצורה. אולי בהתחלה קשה לזהות פעולות מעבר לפתיחה וסגירת דלת, אך בהמשך נראה שגם למפתח יש פעולות.

גם בחיי היום-יום בבית הספר אנו מוקפים בעצמים, חלקם ממשיים כמו ספר, מחברת, ילקוט וקלמר, וחלקם מופשטים כמו ציונים ומקצועות לימוד.

מהו ספר?

לספר יש תכונות הכוללות שם, הוצאה ומספר עמודים. ניתן לעטוף את הספר, לפתוח בעמוד מסויים, לקרוא בו ולחשב כמה עמודים נשאר לנו לקרוא בו עד לסופו.

מהו ציון?

ציון הוא עצם מופשט המשוך למקצוע, ולו תכונות כמו סוג (מבחן, בוחן, עבודה), תאריך (מתי הושג הציון) וערך הציון. ניתן לברר מהו הציון (פעולת אחזור) ולתקן ציון (פעולת עדכון). מעצם היותו משויך למקצוע מסויים, הרי שהציון הינו תכונה של עצם אחר הקרוי מקצוע-לימוד.

אז מהו בעצם "עצם" ?

עצם הינו יחידה סגורה ועצמאית המוגדר על ידי **תכונות** המייצגות את מצבו ו**פעולות** המייצגות את הפעולות שניתן לבקש מהעצם לבצע, כלומר התנהגויות האובייקט.

בהקשר התכנותי מדובר על מאפיינים שנשמרים בתוך משתנים בזיכרון המחשב, ובפעולות שניתן לבצע על משתנים אלו.

עצם (אובייקט) הוא אוסף של **משתנים** ו**פעולות** השייכים לו.

תכונות העצם הם פרטי מידע המאפיינים את העצם. לא די להגיד שלטלפון יש מספר זכרונות וסוללה. כדי לתאר את הטלפון של חיים יש לציין שמדובר בטלפון שמספרו 9876543, מצב הסוללה שלו הוא "מלא" ויש לו 150 זכרונות שרק 30 מתוכם מלאים. לעיתים תכונה של עצם יכולה להיות בעצמה עצם, לדוגמה – כל איש קשר השמור בזיכרון הטלפון שלנו הינו עצם שתכונותיו הן פרטי איש הקשר.

מכיוון שעצם הוא יחידה סגורה, נזדקק לפעולות מיוחדות שיספרו לנו מהן תכונות העצם, כלומר – פעולת **אחזור** למידע השמור במאפייני העצם. למשל, דיווח על מצב הסוללה או שליפה (פעולת אחזור) של מספר טלפון מתוך הזיכרון. לעיתים נרצה לעדכן את מאפייני העצם, למשל עדכון מספר הטלפון של ברוריה ששמור בזיכרון המכשיר של חיים.

קיימת פעולה מיוחדת המאתחלת עצם, כלומר יוצרת עצם חדש בעל תכונות מסוימות שנקבעו מראש, למשל רכישת טלפון חדש שמספרו 7575757 וזיכרונותיו ריקים ומצב הסוללה שלו "ריק".

מהי מחלקה?

מחלקה (class) מתארת עצם. תבנית שלפיה ניתן ליצור עצמים מסוג מסוים או מטיפוס מסוים (טיפוס המחלקה). המחלקה אינה מייצגת עצם מסוים אלא הגדרה כללית של סוג של עצם. המחלקה תכיל את תכונות העצם ללא פירוט הערכים שלהם, ואת הפעולות המתארות את התנהגות העצם.

כאשר יוצרים עצם חדש מהטיפוס המוגדר במחלקה, יוצרים **מופע (instance)** של עצם. כלומר – הטלפון של חיים, הטלפון של ברוריה והטלפון של מנחם הם שלושה מופעים של המחלקה **טלפון-סלולארי**. לכל מופע יש אותם מאפיינים, ואותן פעולות, אך אחזור מספר הטלפון של חיים ייתן ערך שונה מאחזור מספר הטלפון של ברוריה.

מופע (instance) עצם שנקבע מצבו.

מהו כימוס?

המחלקה הינה "קופסא שחורה" המספקת למשתמשים בה התנהגויות אולם מסתירה מהם את המבנה הפנימי שלה. המשמעות של "קופסא שחורה" היא שאין שום צורך להכיר ולהבין את המבנה הפנימי של המחלקה על מנת להשתמש בה, כמו שהשתמש בה, כמו שהשתמש בטלפון הסלולארי אינו צריך להכיר את המבנה הפנימי שלו על מנת לבצע שיחה.

מאפייני העצם מהווים המבנה הפנימי של העצם. הפעולות של העצם מקיפות מאפיינים אלו ומסתירות אותם מפני עצמים אחרים בתכנית, כך שניתן להגיע אליהם רק באמצעות פעולות של העצם. אריזת מאפייני העצם תחת הגנת הפעולות נקראת **כימוס (encapsulation)**.

העטיפה של המשתנים בצורה זו מספקת שני יתרונות עיקריים למתכנתים:

- **מודולריות:** ניתן לבצע שינויים במבנה הפנימי של העצם מבלי שהדבר ישפיע על עצמים אחרים ותכניות המשתמשות בו. למשל, הטכנולוגיה של הטלפונים הסלולאריים הראשונים שונה מהטכנולוגיה של הטלפונים המתקדמים של היום אך הדבר אינו משפיע על האופן שבו אנו מבצעים שיחה.
- **הסתרת מידע:** לאובייקט יש **ממשק ציבורי (public interface)** שבאמצעותו אובייקטים אחרים יכולים לתקשר עימו. כדי לבצע שיחת טלפון כל שעלינו לעשות הוא ללחוץ על הכפתורים המתאימים, מבלי להתעסק בטכנולוגיה של המכשיר.

כיצד עצמים מתקשרים ביניהם?

ברוב המקרים נמצא שעצם יחיד הוא חסר תועלת, לכן עצם בדרך כלל מופיע כרכיב בתכנית או ביישום גדול יותר, שמכיל עצמים רבים. באמצעות תקשורת בין העצמים, מתכנתים יכולים להשיג התנהגויות מורכבת של העצמים. הטלפון אינו מסוגלים לפעול בעצמו. הטלפון הופך להיות שימושי רק כשעצם אחר מפעיל אותו – למשל, מתקבלת שיחה ממכשיר טלפון אחר, או כשאתה מתקשר לטלפון אחר באמצעות הטלפון הנוכחי.

אובייקטים של תוכנה מתקשרים זה עם זה באמצעות הפעולות שלהם. כאשר עצם A רוצה שעצם B יבצע אחת מהפעולות של B, הוא שולח אליו בקשה מתאימה. לפעמים העצם המופעל צריך יותר מידע כדי לדעת במדויק מה עליו לבצע. לדוגמא, כשאתה (עצם A) רוצה להתקשר למנוי טלפון אחר (שמספרו 3456789), עליך לקחת את מכשיר הטלפון שלך (עצם B) ולתת לו מספר טלפון זה. המידע הזה מועבר עם הבקשה כפרמטר (parameter) אחד או יותר.

שלושת הרכיבים שיוצרים בקשה הם :

- העצם שהבקשה מופנית אליו – מכשיר הטלפון שלי : myPhone
- שם הפעולה שיש להפעיל – חייג dialNumber
- הפרמטר שצריך להעביר לפעולה – 3456789

myPhone.dialNumber (3456789)

שלושת רכיבים אלו מספקים את המידע הדרוש לעצם המופעל, כדי לבצע את הפעולה המבוקשת.

מהי הורשה?

- מה המשותף למכונית, משאית, רכבת ומטוס? - כולם כלי תחבורה.
- מה המשותף לחולצה, מכנסיים, גרביים ושמלה? - כולם פרטי לבוש.
- מה המשותף לכלב, ציפור, דג, ונחש? - כולם בעלי חיים.
- מה המשותף לעיפרון, עט, טוש וגיר? - כולם כלי כתיבה.

על מנת להתגבר על עודף המידע המקיף אותנו, אנו נוטים לקבץ עצמים שונים בעלי מאפיינים משותפים ולתת להם סיווג משותף. כאשר דני מספר לחברו יגאל שהוא התחיל ללמוד נהיגה על מכונית, יתעניין יגאל האם המכונית אוטומטית או בעלת הילוכים, כי הוא יודע מהי מכונית ומה ניתן לעשות בה.

מערכות מכוונות עצמים מאפשרות להגדיר מחלקות במונחים של מחלקות אחרות. לדוגמא, המכונית, משאית, רכבת ומטוס הם סוגים שונים של כלי-תחבורה. במונחי תכנות מכוון עצמים, כל אמצעי התעבורה האלה הם **תת-מחלקות** (subclasses) או **מחלקות נגזרות** (derived classes) של המחלקה כלי-תחבורה. באופן דומה, המחלקה כלי-תחבורה היא **מחלקת-על** (superclass) או **מחלקת הבסיס** (base class) של מכונית, משאית, רכבת ומטוס.

כל תת-מחלקה **יורשת** (inherits) תכונות והתנהגויות ממחלקת העל. למכונית יש מאפיינים המבדילים אותה ממשאית, ומטוס מכיל תכונות שאין לרכבת, אולם לכל ארבעת האובייקטים הללו יש תכונות משותפות: לכולם יש תכונה המתארת מהירות, לכולם יש תכונה המתארת שנת ייצור, מספר רישוי, שם יצרן וכד'. לכל אותם העצמים יש גם התנהגות משותפת (עצירה, האצה, התנעה, כיבוי וכו'). כל ארבעת העצמים הם מקרים פרטיים של כלי תחבורה.

עם זאת, המחלקה אינה מוגבלת רק למצבים ולהתנהגויות שמספקת מחלקת-העל שלה. תת-מחלקה יכולה להוסיף מצבים והתנהגויות לאלו שירשה ממחלקת-העל. לדוגמא, לאופניים דו-מושביים יש שני מושבים ושני כידונים, ולכמה מהסוגים של אופני הרים יש מערכת נוספת של הילוכים נמוכים.

באמצעות הורשה נוכל להגדיר מחלקות המתבססות ומשכללות מחלקות קדומות יותר. ההורשה מספקת למחלקה יכולת לרשת תכונות והתנהגויות ממחלקה קודמת.

מהו פולימורפיזם?

אחד השימושים העיקריים במנגנון ההורשה ובמימוש הממשקים הוא **פולימורפיזם** (polymorphism) – **רב-צורתיות**. פולימורפיזם הוא היכולת לבצע פעולה מסויימת (הפעולה הפולימורפית) על עצמים (מופעים) של מחלקות שונות וכל עצם יתנהג בהתאם לפעולות שהוגדרו אצלו.

מחלקות יורשות מחלקה בסיסית משותפת כאשר יש להן מכנה משותף. המכנה המשותף יכול לבוא לידי ביטוי במאפיינים משותפים, כמו גם בהתנהגויות משותפות (פעולות). מכאן ניתן להסיק שלמרות שלמחלקות נגזרות יש את אותה ההתנהגות, כל אחת מהן יכולה להגדיר דרכים שונות על מנת לממש אותה.

פעולה פולימורפית (polymorphic operation) היא פעולה המוגדרת במספר מחלקות שונות. בכל מחלקה יהיה לה שם זהה, תהייה לה משמעות לוגית זהה, אך יהיה לה **מימוש** שונה (המתאים לאותה מחלקה). למשל, המחלקות עיגול, מלבן, ריבוע, מעוין, כולם מחלקות השייכות למחלקת-על צורות-הנדסיות. פעולת חישוב השטח מוגדרת לכל אחת מהמחלקות, והמשמעות הלוגית של הפעולה זהה בכל אחת מהמחלקות, אולם המימוש של כל אחת מהן היא שונה, חישוב שטח העיגול שונה מאופן חישוב שטח המלבן, הריבוע או המעוין.

סיכום

פרק זה מהווה מבוא לתכנות מכוון עצמים, וכולל מספר רב של מושגים חדשים אשר יובהרו בספר זה. חלק מהמושגים שייכים לתכנית הלימודים **יסודות מדעי המחשב** וחלק לתכנית הלימודים **עיצוב תכנה ותכנות מכוון עצמים**.

במסגרת ספר זה, נכיר את עקרונות השפה ואת השימוש בעצמים קיימים, ואף נלמד לבנות עצמים חדשים ולהשתמש בהם.

הוראת הירושה והפולימורפיזם מהווים פרקי המשך לתלמיד שיבחר להתמחות במקצוע מדעי המחשב ברמה של 4 ו-5 יחידות לימוד.

הדגש בתכנית הלימודים כפי שהוא בא לידי ביטוי בספר זה, הוא על פתרון בעיות באמצעות אלגוריתמים ומבני נתונים הקיימים בשפה, תוך שימוש בשפה מבוססת אובייקטים Object Base Programming.

תודות:

בהכנת פרק זה נעזרנו בספרים הבאים:

1. צוות מדעי המחשב, אוניברסיטת ירושלים, **תכנות מונחה עצמים בשפת ג'אווה**, תשס"ד. 2003.
2. ארז קלר, **תכנות מונחה עצמים ב-C#**.
3. **The Java Tutorial – סדנת לימוד**, הוצאת הוד-עמי.
4. **C# - סדנת לימוד**, הוצאת הוד-עמי.

תודה לעפרה ברנדס, ארז קלר וזוהר עמיהוד על רשותם להשתמש בחומרים אלו.

תרגילים

1. דנה החליטה לערוך מסיבה לכבוד פתיחת שנת הלימודים. כמו בכל מסיבה, יש להכין כיבוד ולכן יוצאת דנה לקניות. ברשימת הקניות שלה רשום שיש לקנות שתייה וחטיפים.

בחנות מוצאת דנה הרבה סוגים של בקבוקי שתייה, לכל בקבוק: שם המשקה, תכולת הבקבוק ומחירו. גם במדף החטיפים מוצאת דנה הרבה סוגים של חטיפים, לכל חטיף: שם החטיף, משקל האריזה ומחירה.

דנה לוקחת עגלת קניות ומתחילה למלא אותה במוצרים השונים.

 - א. להלן רשימת של מושגים הקשורים לקניותיה של דנה בחנות. עבור כל מושג ציין האם הוא עצם, או מופע של עצם?

א.	בקבוק-שתייה	עצם / מופע
ב.	שקית-במבה	עצם / מופע
ג.	עגלת-קניות	עצם / מופע
ד.	חטיף	עצם / מופע
ה.	בקבוק-קולה	עצם / מופע
 - ב. ברצוננו להגדיר מחלקה עבור כל אחד מהמושגים שקבעת שהם עצמים. עבור כל מחלקה, ציין תכונות אחדות ופעולות עיקריות המאפיינות אותה. עבור כל פעולה, ציינו האם היא מקבלת ערכים כפרמטר.
 - ג. תן דוגמא לשלושה מופעים שונים של כל אחת מהמחלקות שהגדרת.
 - ד. האם עגלת הקניות של דנה היא מופע או עצם? נמק.
 - ה. מהן תכונותיה של עגלת הקניות? במה היא נבדלת מחטיף או מבקבוק שתייה?
2. יואב מתחיל את לימודיו בבית הספר. נרגש כולו הוא אורז ילקוט לבית הספר. הילקוט כולל מחברת, קלמר, ובתוכו עיפרון מחק וצבעים. בכיס מיוחד בילקוט מונח הטלפון הסלולארי של יואב. יואב עדיין אינו יודע זאת, אך הילקוט שלו וכל תכולתו הינם עצמים. הגדר מחלקה עבור כל אחד מעצמים אלו. עבור כל מחלקה הגדר תכונות ופעולות מאפיינות.
3. במהלך 12 שנות לימודיו בבית הספר יצבור יואב ציונים במקצועות שונים. האם ציון הינו עצם? הגדר עבורו מחלקה.
4. ציון משוייך למקצוע לימוד. הגדר מחלקה מסוג מקצוע-לימוד, וקבע את מאפייניה ואת הפעולות שלה.

דף עבודה 1 לצייר עם צבים

1. פתח את סביבת העבודה וייבא לתוכה את הפרויקט Chap1. הפרויקט Chap1 נמצא בקישור הבא: <http://www.kadman.net/cs4site/Chap1Ex.rar>. בתוך הפרויקט נמצאות מספר מחלקות. פתח את המחלקה: DrawingWithTurtles והרץ.
2. התכנית שבמחלקה יוצרת אובייקט חדש מסוג צב (Turtle). שם האובייקט - t1. לזנב הצב מחובר עיפרון. כשהעיפרון "מונָד" והצב מטייל בחלון הציור, הוא גורר את העיפרון ומשאיר אחריו עקבות.
כדי לראות מה צייר הצב, מחק את שני הלוכסנים שלפני ההוראה: `t1.TailDown()`; (השורה איבדה את הצבע המסמן הערה שאינה לביצוע, וקיבלה צבע של הוראה לביצוע). הרץ שוב את התכנית.
שים ♥: לאחר כל שינוי בתכנית, יש לבצע שמירה לפני ההרצה.
3. הציור שצייר הצב חרג ממסגרת החלון.
מחק את סימני ההערה (שני הלוכסנים) שלפני ההוראה: `t1.MoveBackward(200)`; והרץ שוב. במידת הצורך, הגדל את חלון התצוגה (ע"י גרירת אחת הפינות) כך שיכיל את כל הציור.

להלן טבלה המפרטת חלק מהפעולות אותן ניתן לבצע באמצעות אובייקט מסוג צב:

שם הפעולה	תאור הפעולה
Turtle ()	פעולה בונה היוצרת עצם חדש מטיפוס צב. (לאחר שנוצר הצב, לא ניתן להפעיל את הפעולה באמצעותו).
MoveForward (x)	פעולה המקדמת את הצב x צעדים קדימה.
MoveBackward (x)	פעולה המקדמת את הצב x צעדים אחורה.
TurnRight (x)	פעולה הגורמת לצב לפנות ימינה x מעלות.
TurnLeft (x)	פעולה הגורמת לצב לפנות שמאלה x מעלות.
TailUp()	הרם זנב.
TailDown()	הורד זנב.

- הפעולות הן פעולות של אובייקט מסוג צב. כדי שהצב t1 יבצע הוראה, יש לרשום את שמו לפני ההוראה, ולקשר ביניהם באמצעות נקודה: `t1`; שם-הפעולה. שם-האובייקט.
- כל הוראה מסתיימת בסוגריים. לפעמים יש לרשום ערך בתוך הסוגריים, למשל - מספר המהווה את גודל הצעד שעל הצב להתקדם, או מספר המציין את זווית הסיבוב.
- כל הוראה מסתיימת בסימן ; (נקודה-פסיק).

אם היו לנו כמה צבים בתכנית, ההוראה: `t1.TurnLeft(25)`; תחול על הצב t1 בלבד ותגרום רק לו לפנות 25 מעלות שמאלה.

4. מה יש לשנות בהוראות התכנית כדי לשנות את גודל הצעד ו/או את זווית הפנייה ? עבור כל תת-סעיף שלהלן - בצע את השינוי, שמור והרץ לבדיקת התוצאות.
- שנה את גודל הצעד האופקי ל-125.
 - שנה את גודל הצעד האנכי ל-75.
 - שנה את זווית הסיבוב ימינה ל-45 מעלות.
 - שנה את זווית הסיבוב שמאלה ל-25 מעלות.
5. צור צב חדש בשם $t2$, וגרום לו לצייר ציור שיהיה תמונת ראי לציור שצייר הצב $t1$? למשל, עבור הציור המקורי של הצב $t1$ תתקבל התוצאה:

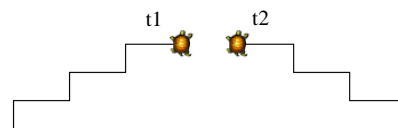


באמצעות ההוראות $TailUp()$ ו- $TailDown()$ והוראות סיבוב ותזוזה תוכל למקם את הצבים במקומות שונים בחלון הציור, מבלי להשאיר סימנים.

6. מקם את אחד הצבים במרחק 20 מהצב האחר כך שיתקבל הציור הבא:



7. מקם את הצבים כך שיציירו את הציור שלהם זה מול זה.



8. נסה לצייר את הציורים שציירת במשימות 5 - 7 באמצעות צב אחד בלבד.
9. צור צב חדש בשם $t3$, מקם אותו בחלק הימני תחתון של המסך וגרום לו לצייר **ריבוע** שאורך כל צלע שלו 80.
10. הזז את $t3$ 200 צעדים שמאלה (מהריבוע של סעיף 8), וגרום לו לצייר **מחומש משוכלל** (מצולע בעל 5 צלעות שוות).

הערה: ניתן להסתיר את הצב באמצעות ההוראה: $t3.SetVisible(false);$



עבודה נעימה !



דף עבודה 2 מים בדליים

פתח את סביבת העבודה וייבא לתוכה את הפרויקט Chap1.

הפרויקט Chap1 נמצא בקישור הבא: <http://www.kadman.net/cs4site/Chap1Ex.rar>.
בתוך הפרויקט תמצא מספר מחלקות. פתח את המחלקה: FillBuckets והרץ.

התכנית שבמחלקה יוצרת שלושה אובייקטים (עצמים) מסוג דלי (Bucket).

- שם האובייקט המהווה את הדלי הראשון - _____ והוא בקיבולת של _____ ליטרים.
 - שם האובייקט המהווה את הדלי השני - _____ והוא בקיבולת של _____ ליטרים.
 - שם האובייקט המהווה את הדלי השלישי - _____ והוא בקיבולת של _____ ליטרים.
- שים ♥: אין לבלבל בין שם האובייקט, והשם המופיע בחלון התצוגה.

להלן טבלה המפרטת חלק מהפעולות אותן ניתן לבצע באמצעות אובייקט מסוג דלי:

שם הפעולה	תאור הפעולה
Bucket (capacity, "name")	פעולה בונה היוצרת דלי ריק בעל קיבולת capacity (מספר שלם) ושם הרשום בין סימני הגרשיים. (לאחר שנוצר הדלי, לא ניתן להפעיל את הפעולה באמצעותו).
Empty ()	פעולה המרוקנת את הדלי הנוכחי.
Fill (x)	פעולה הממלאת את הדלי הנוכחי בכמות מים השווה ל-x ליטרים. אם כמות המים היא מעבר לקיבולת הדלי, הדלי יתמלא ויתרת המים תישפך החוצה. הנחה: x הוא מספר לא שלילי, לא בהכרח שלם.
GetCurrentAmount ()	פעולה המחזירה מספר השווה לכמות המים שבדלי הנוכחי.
PourInto (otherBucket)	פעולה המעבירה את כמות המים המרבית האפשרית מהדלי הנוכחי לדלי ששמו otherBucket.

- הפעולות הן פעולות של אובייקט מסוג דלי. כדי שהדלי b1 יבצע הוראה, יש לרשום את שמו לפני ההוראה, ולקשר ביניהם באמצעות נקודה: (); שם-הפעולה.שם-האובייקט.
- כל הוראה מסתיימת בסוגריים. לפעמים יש לרשום ערך בתוך הסוגריים (למשל - מספר המהווה את כמות המים שיש למלא, או שם של עצם אחר מסוג דלי אליו יש להעביר מים מהדלי הנוכחי) ערך זה קרוי פרמטר.
- כל הוראה מסתיימת בסימן ; (נקודה-פסיק).

ההוראה `b1.Fill (10);` תמלא את הדלי הנוכחי (דלי b1) ב-10 ליטרים מים (בתנאי שהקיבולת שלו לפחות 10 ליטרים).

וההוראה `b1.PourInto (b2);` תעביר מים מהדלי הנוכחי (b1 המפעיל את הפעולה) לדלי b2 (ששמו רשום בתוך הסוגריים).

1. התכנית ממלאה את הדלי b ב-16 ליטרים מים.
 מהי כמות המים המרבית שבה ניתן למלא את הדלי ?
 מה תהיה כמות המים בדלי אם נרשום הוראה למלא כמות מים הגדולה מקיבולת הדלי?
 מה מבצעת כל הוראה בקטע ההוראות הבא :
 b1.PourInto(b2);
 b2.PourInto(b3);
 b3.Empty();
 b1.PourInto(b3);
 מה מבצע קטע ההוראות כולו ?
3. בסוף התכנית יש 3 הוראות המתחילות במילים System.out.println
 מה מבצעות הוראות אלו, והיכן ?
4. שנה את ההוראות שבקטע התכנית המצוין בסעיף 2, כך שדלי b3 יתמלא רק בעזרת דלי b2.
 כמה פעמים היה צורך להעביר מים מדלי b2 לדלי b3 כדי למלא את המשימה?
 האם הפעולה גרמה לאבדן מים ? הסבר :
5. צור דלי חדש בשם b6 בעל קיבולת 6 ליטרים ושם b6.
 הוסף לתכנית סדרת הוראות שבסופה יהיו בדלי b6 בדיוק 5 ליטרים מים ?
6. לרשותך שני דליים. האחד בעל קיבולת 3 ליטרים והשני בעל קיבולת 5 ליטרים.
 רשום סדרת הוראות שתגרום לכך שבדלי בעל קיבולת 5 ליטרים יהיו בדיוק 4 ליטרים של מים.
7. בעזרת שני הדליים משאלה 6 תוכל למלא סידרה של 5 דליים בעלי קיבולת של 5 ליטרים בכמות של 1, 2, 3, 4 ו-5 ליטרים בהתאמה. רשום סדרת הוראות לביצוע המשימה. בסיום הצג את חמשת הדליים החדשים.



עבודה נעימה !

תרגילים 6 ו-7 מבוססים על דף עבודה שפותח ע"י דורית ליקרמן.