

ממשקי המחלקות שבספר הלימוד עצמים תחילה, בהוצאת מבט לחלונות**המחלקה Circle (עמ' 71)**

המחלקה Circle מגדירה 'מעגל' המאופיין על ידי רדיוס וצבע.

<code>public Circle(double radius, string color)</code>	פעולה בונה המקבלת רדיוס וצבע מעגל ומאתחלת עצם חדש שנוצר מטיפוס המחלקה Circle בהתאם לערכי הפרמטרים.
<code>public double GetRadius()</code>	הפעולה מחזירה את הרדיוס של המעגל
<code>public string GetColor()</code>	הפעולה מחזירה את הצבע של המעגל
<code>public void SetRadius(double radius)</code>	הפעולה מעדכנת את רדיוס המעגל
<code>public void SetColor(string color)</code>	הפעולה מעדכנת את צבע המעגל
<code>public void IncRadius()</code>	הפעולה מגדילה את רדיוס המעגל ב-1
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם (את ערכי תכונות המעגל)

המחלקה Book (עמ' 73)

המחלקה Book מגדירה 'ספר' המאופיין על ידי שם ספר, שם סופר, מספר עמודים ומחיר ספר.

<code>public Book(string name, string writer, int pages, double price)</code>	פעולה בונה המקבלת שם ספר, שם סופר, מספר עמודים ומחיר ספר ומאתחלת עצם חדש שנוצר מטיפוס המחלקה Book בהתאם לערכי הפרמטרים.
<code>public string GetName()</code>	הפעולה מחזירה את שם הספר
<code>public string GetWriter()</code>	הפעולה מחזירה את שם הסופר
<code>public int GetPages()</code>	הפעולה מחזירה את מספר העמודים בספר
<code>public double GetPrice()</code>	הפעולה מחזירה את מחיר הספר
<code>public void SetPrice(double price)</code>	הפעולה מעדכנת את מחיר הספר בהתאם לפרמטר.
<code>public void ChangePrice(double percent)</code>	הפעולה מקבלת ערך המציין אחוז העלאה (מספר חיובי) או הורדה (מספר שלילי) במחיר הספר ומשנה את מחיר הספר בהתאם
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את כל פרטי הספר

המחלקה Student (עמ' 74)

המחלקה Student מגדירה 'תלמיד' המאופיין על ידי שם ורשימת ציונים.

<code>public Student(string name)</code>	פעולה בונה המקבלת שם תלמיד ומאתחלת את העצם החדש שנוצר מטיפוס המחלקה Student.
<code>public string GetName()</code>	הפעולה מחזירה את שם התלמיד
<code>public void AddGrd(int grd)</code>	הפעולה מקבלת ציון ומוסיפה אותו לרשימת הציונים של התלמיד
<code>public int MaxGrd()</code>	הפעולה מחזירה את הציון הגבוה ביותר של התלמיד
<code>public int MinGrd()</code>	הפעולה מחזירה את הציון הנמוך ביותר של התלמיד
<code>public double Avg()</code>	הפעולה מחזירה את ממוצע הציונים של התלמיד
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את כל נתוני התלמיד

המחלקה Snack (עמ' 76)

המחלקה Snack מגדירה 'חטיף' המאופיין על ידי שם החטיף, שם החברה ומחיר החטיף.

<code>public Snack (string name, string comp, double price)</code>	הפעולה בונה המקבלת שם חטיף, שם חברה ומחיר. הפעולה מאתחלת עצם חדש מטיפוס Snack בהתאם לערכי הפרמטרים.
<code>public void SetPrice (double price)</code>	הפעולה מקבלת כפרמטר מחיר חדש ומעדכנת את מחיר החטיף.
<code>public double GetPrice ()</code>	הפעולה מחזירה את מחיר החטיף.
<code>public string Tostring()</code>	פעולה המחזירה מחרוזת המתארת את מצב העצם (ערכי התכונות של החטיף)

המחלקה A (עמ' 77)

המחלקה מגדירה את הטיפוס A המאופיין על ידי מספר שלם ותו.

<code>public A (int num, char tav)</code>	פעולה בונה המקבלת כפרמטרים מספר שלם ותו ומאתחלת את מצב העצם החדש בהתאם לערכים אלו.
<code>public void Add(int num2)</code>	הפעולה מקבלת כפרמטר מספר שלם ומוסיפה למספר שבעצם
<code>public void SetNum(int num)</code>	הפעולה משנה את ערך המספר שבעצם בהתאם לפרמטר.
<code>public void PrintTav()</code>	הפעולה מדפיסה את התו tav num פעמים
<code>public int GetNum()</code>	הפעולה מחזירה את המספר
<code>public char GetChar()</code>	הפעולה מחזירה את התו

המחלקה Coin (עמ' 78)

המחלקה *Coin* מגדירה 'מטבע' המאופיין על ידי ערך המטבע באגורות (מספר שלם) וצבע המטבע "*Silver*" או "*Gold*".

<code>public Coin(int value, string color)</code>	פעולה בונה המקבלת מספר שלם (ערך המטבע) ומחרוזת (צבע המטבע) ומאתחלת בהתאם את המצב של העצם החדש מטיפוס מטבע.
<code>public void UpdateColor()</code>	הפעולה משנה את צבע המטבע (מזהב לכסף או מכסף לזהב)
<code>public int GetValue()</code>	הפעולה מחזירה את ערך המטבע (מטיפוס <i>int</i>)
<code>public string GetColor()</code>	הפעולה מחזירה את צבע המטבע (מטיפוס <i>string</i>)
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם (את המטבע)

המחלקה Computer (עמ' 79)

המחלקה *Computer* מגדירה 'מחשב' המאופיין על ידי שם המעבד, מהירות, גודל הזיכרון והמחיר.

<code>public Computer(string name, double speed, int memory, int price)</code>	פעולה בונה המקבלת שם מעבד, מהירות, גודל זיכרון ומחיר ומאתחלת בהתאם את המצב של העצם החדש מטיפוס מחשב.
<code>public void UpdateMem(int memory)</code>	הפעולה מקבלת גודל של זיכרון נוסף ומגדילה את זיכרון המחשב בגודל זה.
<code>public int GetMem()</code>	הפעולה מחזירה את גודל הזכרון במחשב
<code>public void SetMem(int memory)</code>	הפעולה מקבלת גודל של זיכרון ומשנה את גודל זיכרון המחשב לגודל זה.
<code>public string ToString()</code>	פעולה המחזירה מחרוזת המתארת את כל נתוני המחשב

המחלקה Couple (עמ' 90)

המחלקה *Couple* מגדירה טיפוס 'זוג מספרים' המאופיין על ידי שני מספרים שלמים.

<code>public Couple(int num1, int num2)</code>	פעולה בונה המקבלת שני מספרים שלמים ומאתחלת בהתאם את מצב העצם החדש מטיפוס <i>Couple</i> .
<code>public int GetNum1()</code>	הפעולה מחזירה את ערך המספר הראשון בזוג.
<code>public int GetNum2()</code>	הפעולה מחזירה את ערך המספר השני בזוג.
<code>public void SetNum1(int num1)</code>	הפעולה מקבלת מספר שלם ומשנה את ערך המספר הראשון לערך זה.
<code>public void SetNum2(int num2)</code>	הפעולה מקבלת מספר שלם ומשנה את ערך המספר השני לערך זה.
<code>public string ToString()</code>	פעולה המחזירה מחרוזת המתארת את ערכי זוג המספרים במבנה: <code>num2=....</code> <code>num1=....</code>

המחלקה Car (עמ' 91)

המחלקה Car מגדירה את הטיפוס מכונית, המאופיינת על ידי שם החברה היצרנית, שם הדגם, מספר הרישוי, המחיר ומספר הבעלים הקודמים שהיו לרכב.

<code>public Car(string comp, string model, string id, double price, int owners)</code>	פעולה בונה המקבלת שם חברה, שם דגם, מספר רישוי, מחיר ומספר הבעלים הקודמים שהיו לרכב ומאתחלת בהתאם את מצב העצם החדש מטיפוס Car
<code>public double GetPrice()</code>	הפעולה מחזירה את מחיר המכונית.
<code>public int GetOwners()</code>	הפעולה מחזירה את מספר הבעלים הקודמים
<code>public void SetPrice(double price)</code>	הפעולה משנה את הכותרת של הקובייה הנוכחית לפי הערך המתקבל כפרמטר.
<code>public void sellCar()</code>	הפעולה מבצעת מכירה של המכונית (מגדילה את מספר הבעלים ב-1)
<code>public string ToString()</code>	פעולה המחזירה מחרוזת המתארת את נתוני המכונית

המחלקה Candidate (עמ' 98)

המחלקה Candidate מגדירה 'מועמד לעבודה' המאופיין על ידי שם המועמד, גיל, מספר שנות הוותק וקוד המשרה אליה הוא משויך (אם יתקבל).

<code>public Candidate(string name, double age, double exp)</code>	פעולה בונה המקבלת שם מועמד, גיל, ומספר שנות הוותק ומאתחלת בהתאם את מצב העצם החדש מטיפוס Candidate. (קוד המשרה אינו מאותחל – כלומר null)
<code>public string GetName()</code>	הפעולה מחזירה את שם המועמד.
<code>public double GetAge()</code>	הפעולה מחזירה את גיל המועמד
<code>public double GetExp()</code>	הפעולה מחזירה את מספר שנות הוותק של המועמד.
<code>public void SetCode(string code)</code>	הפעולה מקבלת כפרמטר קוד מקצוע (מטיפוס String) ומעדכנת את קוד המקצוע של המועמד בהתאם.
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את נתוני המועמד

המחלקה *Triangle* (עמ' 99)

המחלקה *Triangle* מגדירה 'משולש' המאופיין על ידי אורכי 3 צלעות וגודל שתיים מתוך הזוויות שלו.

<code>public Triangle(double side1, double side2, double side3, double angle1, double angle2)</code>	פעולה בונה המקבלת אורכים של 3 צלעות וגדלים של 2 זוויות משולש ומאתחלת בהתאם את גדלי הצלעות והזוויות במשולש.
<code>public double GetSide1()</code>	הפעולה מחזירה את אורך הצלע הראשונה.
<code>public double GetSide2()</code>	הפעולה מחזירה את אורך הצלע השנייה.
<code>public double GetSide3()</code>	הפעולה מחזירה את אורך הצלע השלישית.
<code>public double GetAngle1()</code>	הפעולה מחזירה את גודל הזווית הראשונה.
<code>public double GetAngle2()</code>	הפעולה מחזירה את גודל הזווית השנייה.
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת אורכי הצלעות וגדלי הזוויות של המשולש.

המחלקה BankAccount (עמ' 103)

המחלקה *BankAccount* מגדירה טיפוס נתונים 'חשבון בנק' המאופיין על ידי מספר חשבון, שם הבעלים והיתרה (סכום הכסף שיש בחשבון הבנק).

<code>public BankAccount(string name, string number, int money)</code>	פעולה בונה המקבלת מספר חשבון, שם הבעלים ויתרת הכסף בחשבון ומאתחלת בהתאם את מצב העצם החדש מטיפוס <i>BankAccount</i> . (יתרת הכסף יכולה להיות חיובית, 0, או שלילית)
<code>public int GetMoney()</code>	הפעולה מחזירה את סכום הכסף שבחשבון
<code>public void WithdrawalMoney(int moneyToWithdrawal)</code>	הפעולה מקבלת כפרמטר את סכום הכסף שיש למשוך מחשבון הבנק, ומעדכנת את היתרה בחשבון בהתאם.
<code>public void DepositMoney(int moneyToDeposit)</code>	הפעולה מקבלת כפרמטר סכום כסף לפקדה בחשבון הבנק, ומעדכנת את היתרה בחשבון בהתאם
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם (נתוני חשבון הבנק)

המחלקה Numbers (עמ' 105)

המחלקה *Numbers* מגדירה טיפוס נתונים 'מספרים' המאופיין על ידי שלושה מספרים שלמים.

<code>public Numbers(int n1, int n2, int n3)</code>	פעולה בונה המקבלת כפרמטרים שלושה מספרים שלמים ומאתחלת בהתאם את מצב העצם החדש מטיפוס <i>Numbers</i> .
<code>public int GetN1()</code>	הפעולה מחזירה את המספר הראשון
<code>public int GetN2()</code>	הפעולה מחזירה את המספר השני
<code>public int GetN3()</code>	הפעולה מחזירה את המספר השלישי
<code>public int Max3()</code>	הפעולה מחזירה את המספר הגדול מבין השלושה
<code>public int Min3()</code>	הפעולה מחזירה את המספר הקטן מבין השלושה
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם (שלושת המספרים)

המחלקה Time (עמ' 120)

המחלקה *Time* מגדירה את טיפוס הנתונים 'זמן' המאופיין על ידי זמן הנתון בשעות ודקות.

<code>public Time(int hour, int minute)</code>	פעולה בונה המקבלת כפרמטרים שני מספרים שלמים המציינים שעות ודקות ומאתחלת עצם מטיפוס <i>Time</i> .
<code>public int GetHour()</code>	הפעולה מחזירה את מספר השעות
<code>public int GetMinutes()</code>	הפעולה מחזירה את מספר הדקות
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם (הזמן בשעות ודקות)

המחלקה Jumper (עמ' 142)

המחלקה Jumper מגדירה טיפוס 'קופץ לגובה' המאופיין על ידי מספר תעודת הזהות שלו, השיא האישי שלו בס"מ, ותוצאות שלושת הקפיצות האחרונות שלו.

<code>public Jumper(string id)</code>	הפעולה בונה עצם שמספר תעודת הזהות שלו מתקבלת כפרמטר. ערכי כל הקפיצות והשיא מאותחלים ל-0.
<code>public double GetRecord()</code>	הפעולה מחזירה את תוצאת השיא של הקופץ
<code>public double GetJump1()</code>	הפעולה מחזירה את תוצאת הקפיצה האחרונה
<code>public double GetJump2()</code>	הפעולה מחזירה את תוצאת הקפיצה הלפני אחרונה
<code>public double GetJump3()</code>	הפעולה מחזירה את תוצאת הקפיצה השניים לפני אחרונה
<code>public void SetRecord(double record)</code>	הפעולה מעדכנת את תוצאת השיא
<code>public void SetJump1(double jump1)</code>	הפעולה מעדכנת את תוצאת הקפיצה האחרונה
<code>public void SetJump2(double jump2)</code>	הפעולה מעדכנת את תוצאת הקפיצה הלפני אחרונה
<code>public void SetJump3(double jump3)</code>	הפעולה מעדכנת את תוצאת הקפיצה השניים לפני אחרונה
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם

המחלקה CubeGame (עמ' 142)

המחלקה CubeGame מגדירה טיפוס 'משחק הטלת קוביות' בין זוג שחקנים, המאופיין על ידי שמות שני שחקנים והמספרים שעל גבי הקוביה של כל אחד מהם.

<code>public CubeGame(string player1, string player2)</code>	פעולה בונה המקבלת את שמות שני השחקנים ומאתחלת את מצב העצם החדש. לכל משתמש, הפעולה מאתחלת את המספר שעל גבי הקוביה שלו באופו אקראי (בערך שבין 1 ל-6)
<code>public int GetDice1()</code>	הפעולה מחזירה את המספר שעל גבי הקוביה של השחקן הראשון
<code>public int GetDice2()</code>	הפעולה מחזירה את המספר שעל גבי הקוביה של השחקן השני
<code>public string GetPlayer1()</code>	הפעולה מחזירה את שם השחקן הראשון
<code>public string GetPlayer2()</code>	הפעולה מחזירה את שם השחקן השני
<code>public void SetDice1(int dice1)</code>	הפעולה מעדכנת את המספר שעל גבי הקוביה הראשונה
<code>public void SetDice2(int dice2)</code>	הפעולה מעדכנת את המספר שעל גבי הקוביה השנייה
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם

המחלקה *BagrutSubject* (עמ' 150)

המחלקה *BagrutSubject* מגדירה 'מקצוע בגרות' המתאר נתוני בית ספר במקצוע בגרות מסוים, ומאופיין על ידי שם המקצוע, סמל שאלון הבחינה, ממוצע הבחינה, מספר הנבחנים ומספר הנכשלים.

<code>public BagrutSubject(string subject, int code, double testAvg, double schoolAvg, int students, int failed)</code>	פעולה בונה המקבלת את שם המקצוע, סמל שאלון הבחינה, ממוצע הבחינה, ממוצע ציוני ההגשה, מספר הנבחנים ומספר הנכשלים ומאתחלת את מצב העצם החדש.
<code>public int GetCode()</code>	הפעולה מחזירה את מספר השאלון
<code>public string GetSubject()</code>	הפעולה מחזירה את שם המקצוע
<code>public double GetTestAvg()</code>	הפעולה מחזירה את ממוצע הבחינה
<code>public double GetSchoolAvg()</code>	הפעולה מחזירה את ממוצע ציוני ההגשה
<code>public double GetStudents()</code>	הפעולה מחזירה את מספר הנבחנים
<code>public double GetFailed()</code>	הפעולה מחזירה את מספר הנכשלים
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם (נתוני הבגרות של בית ספר במקצוע מסוים)

המחלקה *Camp* (עמ' 159)

המחלקה *Camp* מגדירה 'קייטנת טבע' המאופיינת על ידי מספר הילדים המשתתפים בשתי קבוצות גיל: הראשונה – גילאי 8-11 והשנייה גילאי 12-15.

<code>public Camp(int num1, int num2)</code>	פעולה בונה המקבלת את מספר הילדים בשתי קבוצות גיל ומאתחלת את מצב העצם החדש.
<code>public int GetNum1()</code>	הפעולה מחזירה את מספר הילדים בקבוצת הגיל הראשונה
<code>public int GetNum2()</code>	הפעולה מקבלת את מספר הילדים בקבוצת הגיל השנייה
<code>public void Setnum1(int num)</code>	הפעולה מעדכנת את מספר הילדים בקבוצת הגיל הראשונה
<code>public void Setnum2(int num)</code>	הפעולה מעדכנת את מספר הילדים בקבוצת הגיל השנייה
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם (מספר הילדים בשתי קבוצות הגיל)

המחלקה FootballTeam (עמ' 150)

המחלקה *FootballTeam* מגדירה 'קבוצת כדורגל' המאופיינת על ידי שם הקבוצה, הליגה אליה משתייכת הקבוצה, מספר המשחקים בהם ניצחה בעונה הנוכחית, מספר המשחקים בהם הפסידה בעונה הנוכחית, ומספר המשחקים שסיימה בתיקו.

<code>public FootBallTeam(string name)</code>	פעולה בונה המקבלת את שם הקבוצה ומאתחלת את מצב העצם שנוצר כך שמספר הניצחונות, ההפסדים ותוצאות התיקו הוא 0, והליגה שבה משחקת הקבוצה היא ליגת העל.
<code>public string GetName()</code>	הפעולה מחזירה את שם קבוצת הכדורגל
<code>public string GetLeague()</code>	הפעולה מחזירה את הליגה שבה משחקת הקבוצה
<code>public int GetWins()</code>	הפעולה מחזירה את מספר הניצחונות של הקבוצה בעונה הנוכחית
<code>public int GetLosses()</code>	הפעולה מחזירה את מספר ההפסדים של הקבוצה בעונה הנוכחית
<code>public int GetDraws()</code>	הפעולה מחזירה את מספר תוצאות התיקו של הקבוצה בעונה הנוכחית
<code>public void IncWins()</code>	הפעולה מעלה ב-1 את מספר הניצחונות של הקבוצה
<code>public void IncLosses()</code>	הפעולה מעלה ב-1 את מספר ההפסדים של הקבוצה
<code>public void IncDraws()</code>	הפעולה מעלה ב-1 את מספר תוצאות התיקו של הקבוצה
<code>public void SetLeague(string league)</code>	הפעולה מעדכנת את הליגה שבה משחקת הקבוצה בהתאם לפרמטר
<code>public string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם (שם הקבוצה, הליגה, ומספר הניצחונות, ההפסדים והתיקו של קבוצת הכדורגל)

המחלקה Park (עמ' 173)

המחלקה *Park* מגדירה 'חניון' המורכב משתי קומות שבכל אחת מהן, 100 מקומות חניה. המחלקה מאופיינת על ידי מספר המקומות הפנויים בכל קומה.

<code>public Park()</code>	פעולה בונה שאינה מקבלת פרמטרים ומאתחלת את מצב העצם כך שמספר מקומות החניה הפנויים בכל קומה הוא בהתאם למספר מקומות החניה שבקומה.
<code>public int GetFloorA()</code>	מחזירה את מספר מקומות החניה הפנויים בקומה א
<code>public int GetFloorB()</code>	מחזירה את מספר מקומות החניה הפנויים בקומה ב
<code>public void InFloorA()</code>	מפחיתה ב-1 את מספר המקומות הפנויים בקומה א עקב כניסת רכב לקומה זו
<code>public void OutFloorA()</code>	מעלה ב-1 את מספר המקומות הפנויים בקומה א עקב כניסת רכב לקומה זו
<code>public void InFloorB()</code>	מפחיתה ב-1 את מספר המקומות הפנויים בקומה ב עקב כניסת רכב לקומה זו
<code>public void OutFloorB()</code>	מעלה ב-1 את מספר המקומות הפנויים בקומה ב עקב כניסת רכב לקומה זו
<code>public string ToString()</code>	מחזירה מחרוזת המתארת את מצב העצם (מספר מקומות החניה הפנויים בכל קומה)

המחלקה MultiCamp (עמ' 180)

המחלקה MultiCamp מייצגת סניף מסוים בקיטנה שבה יש מספר סניפים.

<code>public MultiCamp(string name)</code>	פעולה בונה המקבלת את שם הסניף ומחזירה עצם ובו מספר הנרשמים לכל קבוצת גיל מאותחל ל-0.
<code>Public public string GetName()</code>	הפעולה מחזירה את שם הסניף
<code>public int GetNum1()</code>	הפעולה מחזירה את מספר הילדים שנרשמו לקבוצת הגיל הראשונה (גילאי 8-11 כולל)
<code>public int GetNum2()</code>	הפעולה מחזירה את מספר הילדים שנרשמו לקבוצת הגיל השנייה (גילאי 12-15 כולל)
<code>public void Setnum1()</code>	מעלה באחד את מונה מספר הילדים בקבוצת הגיל הראשונה
<code>public void Setnum2()</code>	מעלה באחד את מונה מספר הילדים בקבוצת הגיל השנייה
<code>public override string ToString()</code>	מחזירה מחרוזת המתארת את מצב העצם (שם המחנה ומספר הילדים בכל אחת מקבוצות הגיל)

המחלקה TomAndJerry (עמ' 181)

המחלקה TomAndJerry מגדירה את המשחק טום וג'רי.

<code>public TomAndJerry()</code>	פעולה בונה שאינה מקבלת פרמטרים. הפעולה בונה עצם בו המיקום ההתחלתי של תום הוא 10, המיקום ההתחלתי של ג'רי הוא 40, ומספר המהלכים מאותחל ל-0.
<code>void SetTom(int num)</code>	מעדכן את המיקום של תום על הלוח
<code>void SetJerry(int num)</code>	מעדכן את המיקום של ג'רי על הלוח.
<code>public void SetNum()</code>	הפעולה מעדכנת את מספר המהלכים שהיו במשחק
<code>int GetTom()</code>	מחזיר את המיקום של תום
<code>int GetJerry()</code>	מחזיר את המיקום של ג'רי
<code>public int GetNum()</code>	הפעולה מחזירה את מספר המהלכים שהיו במשחק
<code>public override string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את מצב העצם (כלומר מצב השחקנים על הלוח)