

מושגי יסוד

❖ מסמך זה הינו סיכום של החומר ולא מהווה את כולו. אם משהו לא ברור, אני מציע לעבור שוב על ההרצאות או לשאול במייל (bagrut.school@gmail.com).

פתיחת נושא

בפרק זה נלמד מושגים בסיסיים בכתיבת קוד ובכתיבת אלגוריתם.

מהו בכלל אלגוריתם?

אלגוריתם הינו שרשרת של צעדים לפתרון בעיה.

כלומר, כאשר אנו נתקלים בבעיה, האלגוריתם לפתרונה הוא הפעולות שנבצע כדי לפתור אותה.

לדוגמה, כאשר יש לנו בעיה קשה ואנחנו צמאים.

מהו האלגוריתם לפתרון הבעיה:

צעד 1 – ללכת למכולת

צעד 2 – לקנות בקבוק מים

צעד 3 – לשתות מהבקבוק

דוגמה מהעולם שלנו – חישוב של $3 * (4 + 5)$.

האלגוריתם לפתרון הבעיה:

צעד 1 – נחשב את $5 * 4 = 20$

צעד 2 – נחשב את $8 + 20 = 28$

צעד 3 – נחשב את $28 * 3 = 84$

איך אלגוריתם קשור לכתיבת קוד?

כתיבת קוד הוא בעצם מימוש של האלגוריתם.

כלומר, אנו כותבים קוד בכדי לפתור בעיה. כמו שלמדנו, נמצא אלגוריתם לפתרון הבעיה ונכתוב קוד שעושה מה שהאלגוריתם צריך לעשות.

לדוגמה, ניקח את החישוב שביצענו קודם $3 * (4 + 5)$ ונכתוב קוד שמממש את האלגוריתם שכתבנו:

- שורה 1 – מבצעת את צעד 1
- שורה 2 – מבצעת את צעד 2
- שורה 3 – מבצעת את צעד 3

עד עכשיו האלגוריתם היה בראש שלנו או על דף. עכשיו הוא כתוב בשפת מחשב.

בעיה ← פתרון ← אלגוריתם ← קוד

אז מה נלמד בפרק זה?

- פתיחת נושא
- ערכים וטיפוסי נתונים
- משתנים
- טבלאות מעקב
- מחרוזות על קצה המזלג
- קלט ופלט
- ביטויים חשבוניים
- פעולות מתמטיות
- פרק תרגול
- פעולות
- סיכום נושא

ערכים וטיפוס נתונים

במדעי המחשב, לכל ביטוי יש ערך. מהו הערך של $5 + 8 * 3$?

לתת הביטוי $8 * 3$ יש ערך של 24 ול- $5 + 24$ יש ערך של 29.

כך גם במדעי המחשב נרצה לתת ערך לכל ביטוי.

מהו גודל הערכים שניתן לייצג במתמטיקה? אינסופי. אפשר להרכיב מספרים עם מאות ספרות.

מהו גודל הערכים שניתן לייצג במחשב? לא אינסופי. או במילים אחרות – סופי. שכן למחשב יש זיכרון בגודל סופי היכול להכיל כמות מוגבלת של ערכים.

את הערכים נחלק לקטגוריות שונות הנקראות **טיפוס נתונים**.

בשיעור זה נכיר שניים מהם:

- **טיפוס הנתונים שלם** - כל הביטויים שלהם יש ערך של מספר שלם. כמו $5 + 12$. בשפת מדעי המחשב טיפוס נתונים זה נקרא **int**.
- **טיפוס הנתונים עשרוני** - כל הביטויים שלהם יש ערך של מספר עשרוני. כמו $5.2 + 12$. בשפת מדעי המחשב טיפוס נתונים זה נקרא **double**.
 - $15 + 6$? int כי 21 זה שלם.
 - $15 + 6.5$? double כי 21.5 זה עשרוני.
 - $5.5 + 4.5$? int כי 10 זה שלם.

למה צריך טיפוס נתונים int וגם טיפוס נתונים double?

בשביל לחסוך בזיכרון. מספר עשרוני במדעי המחשב דורש לרוב יותר זיכרון ממספר שלם ולכן אם אנו משתמשים בערך של חמש אז עדיף להשתמש ב-5 השלם מאשר ב-5.0 העשרוני.

זוכרים שדיברנו על גודל הערכים שניתן לייצג במחשב?

אז ב-int המספר הכי גדול הוא 2147483647. מספר גדול, אבל כולנו יודעים שיש מספרים גדולים ממנו.

וב-double המספר הכי גדול הוא בערך 18 ולאחריו מעל 300 אפסים!

משתנים

זוכרים ממתמטיקה את הנעלמים x, y ו- z ?

גם במדעי המחשב יש נעלמים הנקראים משתנים. לכל משתנה ניתן ערך.

הגיע הזמן שנלמד **פקודות חדשות!**

- `int x` – פקודה המאתחלת משתנה בשם x שהוא מטיפוס שלם.
- `int x = 5` – פקודה המאתחלת משתנה בשם x שהוא מטיפוס שלם ובעל ערך 5. כלומר $x=5$.
- `double y` – פקודה המאתחלת משתנה בשם y שהוא מטיפוס עשרוני.
- `double y = 1.2` – פקודה המאתחלת משתנה בשם y שהוא מטיפוס עשרוני ובעל ערך 1.2. כלומר $y=1.2$.
- `x = 10` – פקודה המשנה את הערך של המשתנה x ל-10. אפשר לבצע את הפקודה רק לאחר אתחול של x .

אם הגדרנו את x להיות `int` – אז הוא נשאר `int` לתמיד. כלומר אי אפשר לשים לו ערך של `double`.

אם הגדרנו את y להיות `double` – אז הוא נשאר `double` לתמיד. אפשר לשים לו ערך של `int`, אך הטיפוס שלו נשאר `double`. למשל `y = 5` `double` יכיל את $y=5.0$.

מודל האחסון של משתנה בזיכרון המחשב

כאשר אנו מאתחלים את משתנה x , אז מתפנה בזיכרון המחשב תא במיוחד עבורו.

הערך של x יכול להשתנות וכך גם התא בזיכרון המיועד ל- x משתנה. אך תמיד התא הזה שייך ל- x .

השמה

נרצה לשים ערך של משתנה אחד כערך של משתנה אחר.

פקודה חדשה:

- `y = x` – פקודה השמה ב- y את הערך של x .

דוגמה לשימוש:

```
int x = 10;
int y = 5;
y = x;
```

בשורה הראשונה אתחלנו את x ל-10 ובשורה השנייה את y ל-5. בשורה השלישית ביצענו השמה ל- y של הערך של x . כלומר מעכשיו הערך של y הינו 10.

אחרי שורה 2 – $y=5$

אחרי שורה 3 – $y=10$

שימו לב ש- x נשאר 10. זה לא שהעברנו את הערך 10 מ- x ל- y . העתקנו אותו.

תרגיל 1

עקבו אחר הקוד הבא ורשמו מה הערך של x, y ו- z לאחר כל שורה:

```
int x = 3;
int y = 5;
int z;
z = x;
z = 3;
y = z;
```

טבלאות מעקב

בדומה לתרגיל 1, נרצה לעקוב אחר קטע הקוד שלנו בין שורה לשורה.
לצורך זה נשתמש ב**טבלת מעקב** – כלי עזר המאפשר לעקוב אחר קטע הקוד ולבדוק את נכונותו.
בטבלה, ניצור עמודה עבור כל משתנה אחריו מעוניינים לעקוב. בכל שורה נכתוב את הערך העדכני של כל משתנה בצורה הבאה:

שם משתנה	X	Y	Z
ערך לאחר שורה 1			
ערך לאחר שורה 2			
ערך לאחר שורה 3			

תרגיל 2

כתבו טבלת מעקב עבור הקוד הבא:

```
int ben = 30;  
int yuval = 15;  
int michal = 10;  
michal = 4;  
yuval = michal;  
ben = yuval;
```

מחרוזות על קצה המזלג

בבגרות הזאת אנו מתמקדים בטיפוסי הנתונים int ו-double, אך ישנם טיפוסים רבים נוספים.

אחד שכדאי להכיר הוא טיפוס הנתונים String שכבר נתקלנו בו.

למשל, בפקודת הפלט:

```
System.out.println("shalom");
```

"shalom" הינה מחרוזת – String.

ניתן לשמור אותה במשתנה כמו שעשינו עם מספרים שלמים ועשרוניים, אך אנו נשתמש ב-String רק עבור פעולת הפלט והדפסה למשתמש.

קלט ופלט

למדנו בפרק הקודם על פעולת הפלט:

```
System.out.println("shalom");
```

ניתן להדפיס למסך גם משתנה:

```
int num = 5;  
System.out.println(num);
```

כאן, יודפס למסך 5. שימו לב שהפעם לא הוספנו מרכאות. אם היינו מוסיפים מרכאות:

```
System.out.println("num");
```

אז היה מודפס למסך "num" ולא הערך של המשתנה.

בדומה לפעולת הפלט, ישנה גם פעולת קלט. כאשר נרצה לקבל תוך כדי הרצת הקוד, נתונים מהמשתמש:

1. נוסיף את ה-import הבא לראש הקוד:

```
import java.util.Scanner;
```

2. בתוך ה-main נכתוב את שורת הקוד הבאה:

```
Scanner myObj = new Scanner(System.in);
```

אם נרצה לקבל מספר שלם: myObj.nextInt();

אם נרצה לקבל מספר עשרוני: myObj.nextDouble();

כמו תמיד, נשמור את התוצאה במשתנה, בכדי להשתמש בו לאחר מכן:

```
int age = myObj.nextInt();
```

הפקודה שומרת את הקלט מהמשתמש במשתנה age.

לא נתעמק כרגע בפעולות המורכבות של הקלט והפלט.

בינתיים, מתי שנצטרך לקבל קלט או להוציא פלט – נשתמש בפקודות האלה.

שרשור מחרוזות

בפקודת הפלט, ניתן להדפיס גם מחרוזת וגם משתנה בו זמנית.

לצורך זה נשתמש בסימן ה-+.

לדוגמה:

```
int age = 30;  
System.out.println("my age is "+age);
```

ידפיס למסך "my age is 30".

שימו לב ש-"my age is" זאת מחרוזת ולכן מוקפת בסוגריים, בעוד ש-age הינו משתנה ולכן ללא סוגריים.

ניתן להוסיף גם מספר +ים:

```
int age = 30;  
System.out.println("my age is "+age+" and I'm very old.");
```

ביטויים חשבוניים

ביטויים חשבוניים

ביטוי חשבוני הוא ביטוי המורכב ממשתנים וקבועים עליהם נפעיל פעולות חשבוניות כגון חיבור, חיסור, כפל, חילוק ועוד. כמו תרגיל בחשבון, גם לביטוי החשבוני יהיה ערך חשבוני כלשהו.

פעולות חדשות:

- $x + y$ – פעולת חיבור
- $x - y$ – פעולת חיסור
- $x * y$ – פעולת כפל
- x / y – פעולת חילוק

שימו לב ש-x ו-y יכולים להיות ערך (כמו 5, 10, 0.5) וכן יכולים להיות משתנה (כמו int או double).

ניתן לבצע מספר פעולות ביחד כמו שאנו מכירים מחשבון (סוגריים, חוקי חשבון...)

בנוסף, יש לשמור את התוצאה במשתנה בכדי לא לאבד אותה.

למשל הפקודה $x + y$ לא עושה דבר. אך $z = x + y$ שומרת את תוצאת הביטוי ב-z.

המרה

אם נחבר מספר int עם מספר double, מה תהיה התוצאה? int או double?

$10 + 5.5 = 15.5$, ומכאן שנקבל מספר double.

לכן, אם נחבר מספר עשרוני עם מספר שלם, המספר השלם "יומר" למספר עשרוני. לדוגמה, אצלנו 10 יהפוך ל-10.0 ובכך נוכל לחבר שני מספרים מאותו טיפוס נתונים.

תרגיל 3

כתבו קוד המקבל כקלט 3 משתנים: x , y ו- z , ופותר את המשוואה $15x + yz$. לבסוף מדפיס למסך את התוצאה של המשוואה.

תרגיל 4

הסתכלו על הקוד הבא:

```
int age = 30;
int year = 2020;
int birth = year - age;
System.out.println("I was born in:" + birth);
```

מה יודפס למסך?

מה הקוד הזה עושה?

שנו את המשתנה age לגיל שלכם. מה יודפס הפעם?

פעולות מתמטיות

בפרק זה נלמד על פעולות מתמטיות קצת יותר מורכבות.

פקודות חדשות:

- `Math.abs(num)` – פעולה המחזירה את הערך המוחלט של `num`. לדוגמה, עבור `num=-5` נקבל 5.
- `Math.pow(num1, num2)` – פעולה המחזירה את הערך של `num1` בחזקת `num2`. לדוגמה, `num1=3`, `num2=2` יהיה 3 בחזקת 2 שזה 9. שימו לב שהערך המוחזר הוא מטיפוס `double` אפילו אם מתקבל מספר שלם.
- `Math.sqrt(num)` – פעולה המחזירה את השורש של `num`. לדוגמה, עבור `num=9` הפעולה תחזיר 3.
- `Math.round(num)` – פעולה המחזירה את `num` מעוגל למספר שלם. לדוגמה, 3.15 יעוגל ל-3, בעוד 3.75 יעוגל ל-4 (קרוב יותר ל-4 מל-3).
- `Math.max(num1, num2)` – פעולה המחזירה את המספר הגדול מבין `num1` ו-`num2`. לדוגמה, עבור המספרים 4 ו-7, הפעולה תחזיר 7.
- `Math.min(num1, num2)` – פעולה המחזירה את המספר הקטן מבין `num1` ו-`num2`. לדוגמה, עבור המספרים 4 ו-7, הפעולה תחזיר 4.
- פעולת מספר רנדומלי דורשת 2 שורות:

```
Random random = new Random();
int age = random.nextInt(12);
```

- השורה הראשונה נדרשת בשביל להשתמש בפעולות של `Random` (נדבר על זה בהמשך).
- בשורה השנייה, הפעולה `random.nextInt(12)` מחזירה מספר רנדומלי בין 0 לבין 12 פחות 1, כלומר 11.
- אם היינו כותבים `3 + random.nextInt(12)` היינו מקבלים מספר רנדומלי בין 3 ל-14.
- מספר רנדומלי הכוונה לכך שייטכן שבהרצה אחת נקבל 3 ובהרצה אחרת נקבל 5.
- `num1 % num2` – שארית חלוקה של `num1` ב-`num2`. לדוגמה: `15%7=1`. שכן 15 מתחלק ב-7 עם שארית 1 (`7*2+1`).

ביטוי חשבוני עכשיו יכול להכיל: משתנים, קבועים, פעולות חשבוניות ופעולות מתמטיות.

למשל, `Math.max(4,5)+5*7` זה ביטוי חשבוני.

פעולות ספריות – שימו לב שהשתמשנו בספרייה שנקראת Math בספריה שנקראת Random. ספריה, היא אוסף של פעולות אשר אפשר להשתמש בהן.

לדוגמה, Math מכילה פעולות מתמטיות. עבור קריאה לפעולה בה – נוסיף נקודה ואחריה את שם הפעולה. לדוגמה, עבור הפעולה pow – Math.pow.

תרגיל 5

חשבו את ערכי הביטויים החשבוניים הבאים:

- Math.sqrt(2*8) + Math.min(5,6)
- Math.round(4.8) * 2
- Math.pow(Math.abs(-5),2)
- Math.max(1,17%6)
- Math.max(5,random.nextInt(12) + 1)

תרגיל 6

כתבו קוד המקבל כקלט שני משתנים מסוג int x ו-y ומדפיס כפלט את תוצאות הביטויים הבאים:

- סכום המשתנים
- מכפלת המשתנים
- המספר הגדול מבין הביטויים
- שארית החלוקה של x ב-y
- שארית החלוקה של y ב-x

תרגיל 7

ערכו 2 טבלאות מעקב על קטע הקוד הבא:

פעם ראשונה – כאשר x=5,y=3

פעם שנייה – כאשר x=5,y=6

```
Scanner myObj = new Scanner(System.in);
int x = myObj.nextInt();
int y = myObj.nextInt();
int sum = x + y;
System.out.println(Math.max(x + 4, y));
x = Math.min(x, y);
sum = x + y;
System.out.println(Math.max(x + 4, y));
System.out.println(Math.max(x, 5) * 8 + 4);
```

תרגיל 8

כתבו קוד אשר מקבל כקלט 3 משתנים ופולט את המספר הגדול ביותר מבינם.

תרגיל 9

כתבו קוד אשר מקבל כקלט 3 משתנים ופולט את המספר הקטן ביותר מבינם.

פרק תרגול

הגיע הזמן לתרגל קצת את כל מה שלמדנו על ידי דוגמאות לבעיות שיכולות להיווצר בקוד.

שגיאה לוגית – שגיאה באלגוריתם אשר גורם לתוצאות שגויות.

לדוגמה:

נניח שרצינו לחשב את סכום המשתנים x ו- y . ולהכפיל את התוצאה ב-8.

הפתרון לבעיה בכתב הינו: $(x+y)*8$.

מצד שני כתבנו בקוד: `int answer = x + y * 8;`

שמביא תוצאה שונה.

השגיאה הלוגית היא ששכחנו את הסוגריים וזה שינה את הערך של כל הביטוי.

תרגיל 10

זהו את השגיאה הלוגית בתרגיל הבא:

- בכיתה ביקשו לקבל את המספר הגדול מבין שני קלטים x ו- y .
- יוסי כתב את קטע הקוד הבא:

```
Scanner myObj = new Scanner(System.in);  
int x = myObj.nextInt();  
int y = myObj.nextInt();  
Console.WriteLine(Math.min(x, y));
```

מה הטעות בקטע הקוד? זהו ותקנו את השורה הרלוונטית.

שגיאה תחבירית – שימוש לא נכון לשפת התכנות, שהמחשב לא מצליח לעבד.

לדוגמה:

יוסי הבעייתי רצה לקבל קלט מהמשתמש וכתב את השורה הבאה:

```
int x = myObj.nextInt();
```

לאחר ההרצה, יוסי קיבל הודעה של שגיאה. למה?

כי הוא כתב `nextIn` במקום `nextInt`. טעות קטנה, אבל זה מספיק בשביל שהקוד לא ירוץ.

תרגיל 11

זהו את השגיאה התחבירית בתרגיל הבא:

- דני, חבר של יוסי, רצה להדפיס למסך את המשפט "I know how to code".
- הקוד שלו מכיל שורה אחת בלבד:

```
System.out.println(I know how to code);
```

למה דני קיבל שגיאה? זהו ותקנו את השורה הרלוונטית.

שגיאת זמן ריצה – שגיאה שנתפסת בזמן ריצת התוכנית.

כלומר, הקוד נכון לוגית ותחבירית, אך ישנה בעיה אשר גורמת לתוכנה להפסיק/לא לפעול.

תרגיל 12

הריצו את הקוד הבא:

```
int a = 20;
int b = 10;
int c = 10;
int x = a / (b - c);
int y = a / (b + c);
```

מה אתם מקבלים? מה הבעיה בקוד?

תרגילים נוספים לתרגול:

תרגיל 13

בכיתה ביקשו לכתוב קוד המקבל שני מספרים מטיפוס שלם (int), מחליף ביניהם את הערכים, ומדפיס למסך את ערך כל אחד מהם.

לדוגמה:

עבור $x=3$, $y=5$. נחליף את הערך של x להיות הערך של y , כלומר $x=5$. וכן y להיות הערך של x , $y=3$.

למסך יודפס:

x value is 5

y value is 3

מיכל פתרה את התרגיל באופן הבא:

```
Scanner myObj = new Scanner(System.in);
int x = myObj.nextInt();
int y = myObj.nextInt();
x = y;
y = x;
System.out.println("x value is " + x);
System.out.println("y value is " + y);
```

מהי השגיאה הלוגית בקוד של מיכל?

כתבו קוד משלכם הפותר את הבעיה.

פעולות

פעולה היא פיסת קוד המבצעת משימה מסוימת.

למשל, במשימת הצב turtle.left, turtle הינה **פעולה** המבצעת את המשימה של פנייה שמאלה.

כמו כן, Math.abs הינה פעולה המבצעת את המשימה של ערך מוחלט.

כרגע, נשים את הפעולות בתוך המחלקה שלנו, אך לפני ה-main:

```
public class Main
{
    // נכתוב את הפעולות
    public static void main(String[] args) {

    }
}
```

איך כותבים פעולה?

```
public static int getNumber(int num)
{
    return num;
}
```

בוא נבין מילה במילה מה כתוב כאן:

- **public** – זה אומר שהפעולה היא ציבורית – כולם יכולים להשתמש בה. למשל, private הינו סוג פרטי של פעולה. כרגע, מבחינתנו פעולה פרטית וציבורית זה בערך אותו הדבר ונרחיב על זה בהמשך.

- **Static** – נרשום את זה תמיד כמילה שנייה. בהמשך נלמד למה.

- **Int** – זהו הטיפוס המוחזר של הפעולה.
לדוגמה, במקרה של פעולת sqrt() הטיפוס המוחזר הוא מסוג מספר עשרוני ולכן שם כתוב double.

- איך מחזירים ערך? בעזרת return num. פקודת return מחזירה את הערך של המשתנה num. לכן צריך להיות מסוג int.
- איך שומרים את הערך? כמו שאנו כבר מכירים – בתוך משתנה:
- פעולת void – זאת פעולה אשר לא מחזירה ערך. כלומר, לא צריך return:

```
public static void printHello()
{
    System.out.println("Hello");
}
```

קוראים לה בלי לשמור במשתנה:

```
printHello();
```

- **שם הפעולה:** לאחר טיפוס הפעולה – רושמים את שם הפעולה: בפעולה הראשונה getNumber ובפעולה השנייה printHello. נבחר שם נוח שאנחנו רוצים.
למה לבחור שם טוב? כי כך יהיה לנו קל להשתמש בפעולות בעתיד. לדוגמה, פעולה הנקראת calculateSum זה שם טוב עבור פעולה המחשבת סכום.

- **שמות הפרמטרים:** לאחר שם הפעולה, נכתוב בסוגריים את הפרמטרים שנרצה להשתמש בהם: בפעולה הראשונה שלנו רצינו פרמטר אחד מסוג שלם, ולכן כתבנו: `int num`. בפעולה השנייה כתבנו סוגריים ריקים, כי לא היינו צריכים פרמטרים. אם נרצה יותר מפרמטר אחד – נפריד את הפרמטרים על ידי פסיק: `(parameter1,parameter2,parameter3...)` גם כאן, נשתמש בפרמטרים בעלי שמות נוחים. לדוגמה, בפעולה הראשונה שלנו השתמשנו בפרמטר בשם `num` כיוון שרצינו מספר (קיצור של `number`).
- **סוגריים מסולסלים וגוף הפעולה:** לאחר שורת הפתיחה, נכתוב סוגריים מסולסלים. ביניהם יש את גוף הפעולה - כלומר, תוכן הפעולה נכתב פה. השורה האחרונה תהיה פעולת `return`, במקרה והפעולה מחזירה טיפוס נתונים.

למה צריך פעולות בכלל? בשביל לשמור על סדר ונראות הקוד וכן לחלק למשימות.

למה כדאי לחלק משימה גדולה למשימות קטנות? בכדי שיהיה יותר קל לתקן, לשנות ולהוסיף.

העברה על פי ערך: נשים לב שכאשר אנו קוראים לפעולה, הערכים של הפרמטרים נקבעים על פי הערכים של המשתנים שלנו, אך לא משנים אותם.

לדוגמה, פעולה `getNum(int num)` הנקראת עם `num=5`. אפילו אם נשנה את `num` בתוך הפעולה – המשתנה מחוץ לפעולה לא משתנה.

(להסבר נוסף חזרו על ההרצאה)

תרגיל 14

כתבו פעולה המקבלת כקלט משתנה מסוג שלם המציין גיל, ומחזירה int המציין את השנה בה האדם נולד. לדוגמה: עבור קלט 24, היא מחזירה 1996 (כאשר השנה העכשווית היא 2020).
ב-main, קראו לפעולה 3 פעמים עם קלטים שונים והדפיסו את התוצאות למסך.

תרגיל 15

דני כתב את הקטע קוד המסורבל הבא:

```
public static void Main(String[] args)
{
    Scanner myObj = new Scanner(System.in);
    double num1 = myObj.nextInt();
    double num2 = myObj.nextInt();
    double num3 = myObj.nextInt();
    double max_num = Math.max(num1, num2);
    max_num = Math.max(max_num, num3);
    System.out.println("The biggest number is: " + max_num);
    double min_num = Math.min(num1, num2);
    min_num = Math.min(min_num, num3);
    System.out.println("The smallest number is: " + min_num);
}
```

כתבו מחדש את הקוד של דני כך שיכיל 2 פעולות קטנות המסדרות את הקוד בצורה נוחה יותר.

תרגיל 16

כתבו את התוכנית הבאה:

- התוכנית מקבלת כקלט 2 משתנים מסוג double.
- התוכנית כוללת 4 פעולות:
 - o פעולת חיבור:

```
double add(double x, double y)
```

המחזירה את הסכום של הערכים.

- o פעולת חיסור:

```
double sub(double x, double y)
```

המחזירה את החיסור של הערכים.

- o פעולת כפל:

```
double mul(double x, double y)
```

המחזירה את המכפלה של הערכים.

- o פעולת חילוק:

```
double div(double x, double y)
```

המחזירה את תוצאת החילוק של הערכים.

- התוכנית מדפיסה למסך את תוצאת הפעלת כל אחת מהפעולות על 2 המשתנים.

תרגיל 17

ערכו טבלת מעקב אחר קטע הקוד הבא:

- פעם אחת עבור $x=30, y=2, z=8$
- פעם שנייה עבור $x=12, y=5, z=100$

```
class Program
{
    private static double secret(int num1, int num2, int num3)
    {
        int what = Math.max(num1 * num2, num2 * num3);
        what = Math.max(what, num1 * num3);
        return what;
    }
    static void Main(String[] args)
    {
        Scanner myObj = new Scanner(System.in);
        int x = myObj.nextInt();
        int y = myObj.nextInt();
        int z = myObj.nextInt();
        System.out.println(secret(x, y, z));
    }
}
```

בטבלה, כתבו עמודה לכל המשתנים: $x, y, z, num1, num2, num3, what$ וכן עבור הפלט.

תרגיל 18

כתבו את התוכנית הבאה:

- התוכנית מקבלת כקלט מספר שלם תלת ספרתי.
- התוכנית כוללת פעולה אחת:
 - פעולת סכום ספרות:

`int sumDigits(int num)`

המחזירה את הסכום של הערכים.

רמז: השתמשו בחילוק (/) ובשארית (%).

- התוכנית מדפיסה למסך את תוצאת ההפעלה של הפעולה על גבי הקלט. הריצו את התוכנית על הקלטים הבאים: 123, 456, 999.

סיכום נושא

דיברנו בנושא זה על המון דברים חדשים: משתנים, ביטויים, פעולות מספריות ופעולות משלנו.

הרבה מאוד חומר שקצת קשה לזכור.

מציע לעבור שוב על החומר ולבדוק שאתם מבינים את הכול.

אם יש עוד שאלה, אני פה לשאלות:

bagrut.school@gmail.com

פתרונות לתרגילים

פתרון תרגיל 1

לאחר שורה 1 – $x=3$

לאחר שורה 2 – $x=3, y=5$

לאחר שורה 3 – $x=3, y=5$

לאחר שורה 4 – $x=3, y=5, z=3$

לאחר שורה 5 – $x=3, y=3, z=3$

פתרון תרגיל 2

שם משתנה	ben	yuval	michal
ערך לאחר שורה 1	30	-	-
ערך לאחר שורה 2	30	15	-
ערך לאחר שורה 3	30	15	10
ערך לאחר שורה 4	30	15	4
ערך לאחר שורה 5	30	4	4
ערך לאחר שורה 6	4	4	4

פתרון תרגיל 3

כתבו קוד המקבל כקלט 3 משתנים: x, y, z , ופותר את המשוואה $15 \cdot x + y \cdot z$. לבסוף מדפיס למסך את התוצאה של המשוואה.

```
Scanner myObj = new Scanner(System.in);
int x = myObj.nextInt();
int y = myObj.nextInt();
int z = myObj.nextInt();
int ans = 15 * x + y * z;
System.out.println("The answer is: " + ans);
```

פתרון תרגיל 4

מה יודפס למסך? 1990.

מה הקוד הזה עושה? מחשב את שנת הלידה.

שנו את המשתנה age לגיל שלכם. מה יודפס הפעם? שנת הלידה שלכם.

פתרון תרגיל 5

- 9
- 10
- 25
- 5
- אי אפשר לדעת. תלוי במספר הרנדומלי שיוגרל

פתרון תרגיל 6

```
Scanner myObj = new Scanner(System.in);
int x = myObj.nextInt();
int y = myObj.nextInt();
int sum = x + y;
int mul = x * y;
int max = Math.max(x, y);
int remainder_xy = x % y;
int remainder_yx = y % x;
Console.WriteLine("x + y =" + sum);
Console.WriteLine("x * y =" + mul);
Console.WriteLine("The maximum is " + max);
Console.WriteLine("x % y=" + remainder_xy);
Console.WriteLine("y % x=" + remainder_yx);
```

פתרון תרגיל 7

- כאשר x=5,y=3

שורה 1	x	y	Sum	פלט
שורה 2	5	-	-	-
שורה 3	5	3	-	-
שורה 4	5	3	8	-
שורה 5	5	3	8	9
שורה 6	3	3	8	-
שורה 7	3	3	6	-
שורה 8	3	3	6	7
שורה 9	3	3	6	44

- כאשר x=5,y=6

שורה 1	x	y	Sum	פלט
שורה 2	5	-	-	-
שורה 3	5	6	-	-
שורה 4	5	6	11	-
שורה 5	5	6	11	9
שורה 6	5	6	11	-
שורה 7	5	6	11	-
שורה 8	5	6	11	9
שורה 9	5	6	11	44

פתרון תרגיל 8

```
Scanner myObj = new Scanner(System.in);
int x = myObj.nextInt();
int y = myObj.nextInt();
int z = myObj.nextInt();
int max = Math.max(x, Math.max(y, z));
Console.WriteLine("The maximum number is: " + max);
```

פתרון תרגיל 9

```
Scanner myObj = new Scanner(System.in);
int x = myObj.nextInt();
int y = myObj.nextInt();
int z = myObj.nextInt();
int min = Math.min(x, Math.min(y, z));
Console.WriteLine("The minimum number is: " + min);
```

פתרון תרגיל 10

השגיאה הלוגית היא שיוסי השתמש בטעות בפעולת Math.min אשר תחזיר את המספר הקטן מבין השניים ולא הגדול מביניהם. בכדי להדפיס את המספר הגדול, נשתמש בשורה 3 בפעולה: Math.max.

פתרון תרגיל 11

זוהי שגיאה תחבירית ונקבל שגיאה מהתוכנה. צריך לתקן את שורת הקוד ולהוסיף סוגריים בכדי להפוך את המשפט למחרוזת:

```
System.out.println("I know how to code");
```

פתרון תרגיל 12

הרצת הקוד נופלת באמצע. הבעיה בקוד היא שבשורה 4 אנו מחלקים ב-0 וזהו דבר שלא אפשרי.

פתרון תרגיל 13

השגיאה הלוגית היא שלאחר שורה 3, הערך של x הוא כמו הערך של y, אך איבדנו את הערך המקורי של x ולכן y בשורה 4 יקבל את הערך שהוא נתן ל-x. כלומר, עבור x=3, y=5 נקבל x=5, y=5.

קוד מתוקן ישתמש במשתנה ביניים tmp השומר את ערכו המקורי של x:

```
Scanner myObj = new Scanner(System.in);
int x = myObj.nextInt();
int y = myObj.nextInt();
int tmp = x;
x = y;
y = tmp;
System.out.println("x value is " + x);
System.out.println("y value is " + y);
```

פתרון תרגיל 14

```
public static int yearOfBirth(int age)
{
    return 2020 - age;
}
```

נקרא לפעולה כמה פעמים:

```
System.out.println(yearOfBirth(14));
```

```
System.out.println(yearOfBirth(24));
System.out.println(yearOfBirth(30));
```

פתרון תרגיל 15

```
class Main
{
    public static double maxNumber(double x, double y, double z)
    {
        double max_num = Math.max(x, y);
        max_num = Math.max(max_num, z);
        return max_num;
    }
    public static double minNumber(double x, double y, double z)
    {
        double min_num = Math.min(x, y);
        min_num = Math.min(min_num, z);
        return min_num;
    }
    static void Main(String[] args)
    {
        Scanner myObj = new Scanner(System.in);
        double x = myObj.nextDouble();
        double y = myObj.nextDouble();
        double z = myObj.nextDouble();
        System.out.println("The biggest number is: " + maxNumber(num1,num2,num3));
        System.out.println("The smallest number is: " +minNumber(num1,num2,num3));
    }
}
```

פתרון תרגיל 16

```
class Main
{
    public static double add(double num1, double num2)
    {
        return num1 + num2;
    }
    public static double sub(double num1, double num2)
    {
        return num1 - num2;
    }
    public static double mul(double num1, double num2)
    {
        return num1 * num2;
    }
    public static double div(double num1, double num2)
    {
        return num1 / num2;
    }
    static void Main(String[] args)
    {
        Scanner myObj = new Scanner(System.in);
        double x = myObj.nextDouble();
        double y = myObj.nextDouble();
        Console.WriteLine("x + y = " + Add(x, y));
        Console.WriteLine("x - y = " + Sub(x, y));
        Console.WriteLine("x * y = " + Mul(x, y));
        Console.WriteLine("x / y = " + Div(x, y));
    }
}
```

פתרון תרגיל 17

- פעם אחת עבור $x=30, y=2, z=8$

שורה	x	y	z	Num1	Num2	Num3	What	פלט
Main1	30							
Main2		2						
Main3			8					
Secret1				30	2	8	60	
Secret2							240	
Main4								240

- פעם שנייה עבור $x=12, y=5, z=100$

שורה	x	y	z	Num1	Num2	Num3	What	פלט
Main1	12							
Main2		5						
Main3			100					
Secret1				12	5	100	500	
Secret2							1200	
Main4								1200

פתרון תרגיל 18

```
public static double sumDigits(int num)
{
    int sum = 0;
    sum += num % 10;
    num = num / 10;
    sum += num % 10;
    num = num / 10;
    sum += num;
    return sum;
}
static void Main(String[] args)
{
    Scanner myObj = new Scanner(System.in);
    int x = myObj.nextInt();
    Console.WriteLine("The sum of the digits is "+sumDigits(x));
}
```

עבור 123 – נקבל 6, עבור 456 – נקבל 15, ועבור 999 – נקבל 27.