

מבנה חוקי של מחלקה בשפת ג'אווה:

```

כותרת המחלקה -- השורה הפותחת את המחלקה
public class שם המחלקה

סארייט לפתיחת המחלקה
{

חתימה fe הפעולה הראשית לביצוע
    public static void main(String[] args)

סארייט לפתיחת הפעולה הראשית לביצוע
    {

הוראות לביצוע

סארייט לסגירת הפעולה הראשית לביצוע
    }

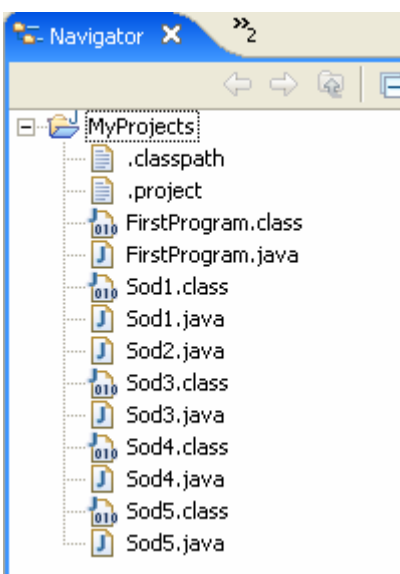
סארייט לסגירת המחלקה
}

```

- ❖ הפעולה הראשית main מופעלת אוטומטית עם הרצת המחלקה.
- ❖ לאחר כל הוראה לביצוע, מופיע התו ;
- ❖ יש הבחנה בין אותיות גדולות לבין אותיות קטנות.
- ❖ שם של מחלקה חייב להתחיל באות גדולה, ויכולים להכלל בו אותיות גדולות, אותיות קטנות, ספרות ומקף תחתון _.
- ❖ הוספת הערות לתכנית:

אפשרות א: הערה תיכתב בין סימני /* למשל: /*.....*/
אפשרות ב: הערה תיכתב החל משני סימני // ועד לסוף השורה //.....

פרויקט ומחלקה



- בחלון השמאלי של תכנת ג'אווה (חלון הקבצים) מופיעה רשימת הקבצים עימם אנו עובדים.
- הקבצים מאוגדים בפרויקטים. כל פרויקט מכיל מספר מחלקות.
- MyProjects הוא פרויקט שמכיל 6 מחלקות: FirstProgram , Sod1 , Sod2 , Sod3 , Sod4 , Sod5

הערות "טכניות"

- לאחר שמשנים תוכן של מחלקה, יש לבצע שמירה שלה. אם מריצים לפני ששומרים, מופיעה שאלה האם לשמור את המחלקה, ולאחר מכן התכנית מבצעת את הנדרש, אבל הפלט נעלם לפני שמספיקים לראות אותו!
- לאחר שמריצים מחלקה פעם אחת, אפשר להריץ אותה גם באמצעות לחיצה על כפתור (כפתור חמישי משמאל בשורת הכפתורים בחלק העליון של המסך).

הוראות הדפסה

S גדולה!

System.out.println (ערך להדפסה);

אפשרות ראשונה:

הערך להדפסה יכול להיות מחרוזת ויכול להיות מטיפוס מוכר בשפה: מספר שלם, מספר ממשי, תו, לוגי.

דוגמאות:

ההוראה להדפסה	הערך שיודפס
System.out.println (" we did it! ");	we did it!
System.out.println (88);	88
System.out.println ('t');	t
System.out.println (false)	false

- ❖ אפשר להשתמש ב- **print** במקום ב- println ואז ההדפסה הבאה תהיה באותה שורה.
- ❖ אם המספר להדפסה הוא מספר שאינו שלם, המספר מודפס בדיוק כפי שהוקלד.
- ❖ אם המספר שיש להדפיס הוא כתוצאה מחישוב של מספרים ממשיים, אז מודפסות 14 ספרות אחרי הנקודה.

שילוב של ערכים מטיפוסים שונים בהדפסת מחרוזת

- ❖ המחרוזת להדפסה נשאר בין סימני " ", מוסיפים את הערכים הרצויים באמצעות הסימן +.
- ❖ הסימן + לא יודפס. תפקידו להפוך טיפוסים שונים (מספר, תו, ערך בוליאני) למחרוזת, וליצור מחרוזת אחת מכל הערכים להדפסה.

דוגמאות:

ההוראה להדפסה	הערך שיודפס
System.out.println (" there are " + 10 + " boys");	there are 10 boys
System.out.println (" the average is " + 875/10);	the average is 87
System.out.println (" there are " + 10 + " students. the average is " + 875/10);	there are 10 students. the average is 87

- ❖ משמעות ההוראה System.out.println (9+9); היא: הדפסת הערך המספרי המוחזר מהביטוי 9+9 ולכן יודפס המספר 18.
- ❖ כאשר יש בהוראת ההדפסה מחרוזת (וגם מחרוזת ריקה "" היא מחרוזת!), שאר הערכים המשורשרים הופכים למחרוזות. לכן אם נרשום: System.out.println (""+9+9); יודפס 99. ואם נרשום: System.out.println (""+(9+9)); יודפס 18 כי קודם חושב ערך הביטוי שבסוגריים (9+9) ורק לאחר מכן הוא הפך למחרוזת.

תווי בקרה שולטים על עיצוב הפלט.
מבנה תו בקרה: הסימן \ ולאחריו אות.
תווי בקרה שימושיים:

תו הבקרה	תפקיד
\n	העברת הראש המדפיס לתחילת שורת ההדפסה הבאה.
\t	העברת הראש המדפיס לתחילת ה tab הבא.

ניתן לשלב תווי בקרה בכל מקום בתוך המחרוזת להדפסה. דוגמאות:

הערך שיודפס	ההוראה להדפסה
we learn	System.out.println (" \n we \t learn \n ");
how to print	System.out.println (" \t how to \nprint ");
7=3	System.out.println (" \n 7=3");
there are 10 students. the average is 87	System.out.println (" there are " + count + " students.\nthe average is " + sum/count);

❖ תווי בקרה אפשר לשלב רק בתוך הדפסה של מחרוזת. למשל: System.out.println ("\n"+9)

לכן, ההוראה System.out.println (\n+9+9) איננה תקינה.

בהוראה System.out.println ("\n"+9+9) מופיעה מחרוזת, לכן שאר הערכים המשורשרים הופכים למחרוזות, ולכן מודפסת המחרוזת 99.

גם בהוראה System.out.println ("\n"+(9+9)) מופיעה מחרוזת, וגם בה שאר הערכים המשורשרים הופכים למחרוזות, אבל הפעם הודפס 18 כי קודם חושב ערך הביטוי שבסוגריים (9+9) ורק לאחר מכן הוא הפך למחרוזת (כמו במקרה בו במקום תו הבקרה היתה מחרוזת ריקה).

אפשרות שניה: הדפסת באמצעות תווי המרה: System.out.format (להדפסה);

❖ בתוך המחרוזת להדפסה, במקום בו רוצים שיופיע התו או המספר, רושמים תו המרה.

❖ לאחר המרכאות " רושמים פסיק, ולאחריו את התו או המספר (או הביטוי החשבוני) הרצוי.

❖ אפשר לשלב בהוראת הדפסה מספר תווים ומספרים. במקרה כזה, הערך הראשון יכנס במקום תו ההמרה הראשון, הערך השני יכנס במקום תו ההמרה השני וכן הלאה.

תווי המרה שימושיים:

תו ההמרה	הטיפוס המודפס
%d	מספר שלם
%f	מספר ממשי
%c	תו

דוגמאות:

ההוראה	הערך שיודפס
System.out.format ("\\nthe temperature is %d degree" , 37);	the temperature is 37 degree
System.out.format (" \\n %d + %d = %d ", 2 , 3 , 2+3);	2 + 3 = 5
System.out.format (" \\n the average is \\t %f ", 89.93);	the average is 89.93
System.out.format (" \\n %c comes before %c", 'a' , 'b');	a comes before b

❖ אם משתמשים בתו ההמרה %c בהדפסת מספר שלם, יודפס התו שזהו הקוד האסקי שלו.

הדפסה מעוצבת

❖ הוספת מספר בתוך תו ההמרה קובעת את גודל שדה ההדפסה.

❖ במספרים ממשיים:

- מספר התווים כולל: נקודה עשרונית וסימן מינוס.
- ברירת המחדל: 6 ספרות אחרי הנקודה.
- אפשר להוסיף שני מספרים. המספר הראשון קובע את גודל שדה ההדפסה והמספר השני קובע את מספר הספרות אחרי הנקודה העשרונית (במידת הצורך, מתבצע עיגול של השבר).
- אם מספר הספרות המבוקש אחרי הנקודה, גדול ממספר הספרות אחרי הנקודה, יודפסו אפסים להשלמת מספר הספרות המבוקש.

❖ אם מספר התווים (לאחר העיגול) גדול מגודל שדה ההדפסה המבוקש, לא תהייה התחשבות בגודל שדה ההדפסה המבוקש.

❖ אם מספר התווים קטן מגודל שדה ההדפסה המבוקש, יודפסו רווחים לפני הערך המודפס כדי להגיע לגודל שדה ההדפסה המבוקש.

דוגמאות:

ההוראה	הערך שיודפס
System.out.format ("\\n %5d is integer", 37);	37 is integer
System.out.format ("\\n %10d is integer", 37);	37 is integer
System.out.format ("\\n %4f is real number", 3.65);	3.650000 is real number
System.out.format ("\\n %4.2f is real number", 3.65);	3.65 is real number
System.out.format ("\\n %9.3f is real number", 3.65);	3.650 is real number
System.out.format ("\\n %9c is char", 'x');	x is char

אותיות גדולות וקטנות

1. **מילים שמורות** של השפה נכתבות באותיות קטנות, למשל `int`, `double`.
2. מוסכמה 1: **שמות של משתנים** נכתבים באותיות קטנות. למעט תחילתה של כל מילה פנימית חדשה שתכתב באותיות גדולות. למשל, `numStudent`.
3. מוסכמה 2: **שמות של מחלקות** (`class`) נפתח באותיות גדולות (כמו `Sod1`). בהמשך, יופיעו רק ספרות ואותיות קטנות למעט תחילתה של כל מילה פנימית חדשה שתכתב באותיות גדולות (כמו `FirstProgram`).

קוד אסקי

(ASCII = American Standard code for Information Interchange)

המחשב "יודע" לקרוא רק ספרות בינאריות (0 ו-1). אולם שפת תכנות יודעת לטפל בתווים שונים (כולל ספרות שונות) – איך הדבר מתבצע?

כל תו "מיוצג" במחשב באמצעות רצף של ספרות בינאריות. למשל התו 'A' מיוצג באמצעות הרצף 01000001 ואילו התו 'B' מיוצג באמצעות הרצף 01000010. רצף הספרות שמייצג כל תו נקרא: קוד אסקי (ASCII) של התו המדובר.

לנוחיותנו, מתורגמים המספרים הבינאריים למספרים עשרוניים. כך למשל, הקוד 01000001 (המייצג את התו A) מתורגם למספר 65. לכן, נאמר כי הקוד האסקי של התו A הוא המספר 65. ובאותו אופן הקוד האסקי של התו B הוא המספר 66.