

חומרים שהוכנו על-ידי מורי הניסוי תש"ע להוראת "יסודות מדעי המחשב"

ניתן להשתמש בחומרים לצורך הוראה בלבד.
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי
ללא קבלת אישור מראש מצוות הפיתוח

עצמים ומחלקות – הגדרות וסיכום

כתיבה ועריכה:
דורית ליקרמן

עצמים ומחלקות הגדרות וסיכום

• טיפוס:

סוג של ערך - באמצעותו ניתן לייצג נתונים שונים בסביבה ממוחשבת.
קיימים מספר טיפוסים המוגדרים בשפה:
int – טיפוס פשוט המייצג ערך מספרי שלם.
double – טיפוס פשוט המייצג ערך מספרי ממשי.
char – טיפוס פשוט המייצג ערך מסוג תו בודד.
boolean – טיפוס פשוט המייצג את הערכים "אמת" או "שקר"
String – טיפוס המייצג באוסף של תווים המאוגדים יחד.

הגדרת הטיפוס קובע את הערכים ואת הפעולות שניתן לבצע על ערכים אלו.

• משתנה:

משתנה הוא תא זיכרון אשר במהלך ביצוע הוראות ניתן לשמור בו ערך מטיפוס פשוט, ולקרוא את הערך השמור בו. למידע השמור בתוך המשתנה קוראים ערך המשתנה. פנייה למשתנה נעשית באמצעות שמו, שהוא שם המשתנה.

ב-Java מגדירים משתנה באופן הבא:

```
שם_משתנה טיפוס;
```

הפקודה:

```
double height;
```

מגדירה משתנה ששמו height מטיפוס מספר ממשי.

• תכונה:

הגדרה מילונית - מה שיש לאדם או לדבר מה מבחינת צורתו או כישרונו או אופיו וכד', כגון גובה, יעילות, אכזריות, טיב, איכות.

התכונה היא ייחודית ולכן מוגדרת באופן פרטי.

ב-Java מגדירים תכונה באופן הבא:

```
שם_תכונה טיפוס private;
```

הפקודה:

```
private int num;
```

מגדירה תכונה פרטית (המוכרת רק בתוך המחלקה) מטיפוס שלם ששמה num.

- **מחלקה (class):**

כדי לפתח תוכנות מורכבות, נרצה להגדיר בעצמנו טיפוסים חדשים נוספים. הטיפוסים החדשים יאופיינו על ידי שם הטיפוס החדש, התכונות שלו והפעולות שניתן לבצע על עצמים מטיפוס זה. נהוג לקרוא לטיפוסים החדשים שאנו יוצרים בשם: **טיפוס מורכב**.

מחלקה היא תבנית המגדירה טיפוס נתונים.

- **עצם (object):**

הגדרה מילונית- דבר מוחשי, גוף, כל דבר הקיים בטבע. במדעי המחשב **עצם** הוא מופע מוחשי של **אוסף תכונות** שמאפיינות אותו.

לכל העצמים מטיפוס מחלקה מסוימת יש אותם האיפיונים. הם נבדלים זה מזה במצבם, כלומר בערכי התכונות שלהם

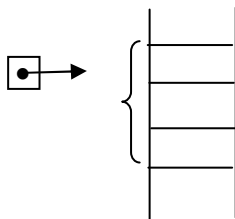
- **הפנייה:**

משתנה המכיל הפנייה לקבוצת תאים בזיכרון המחשב, בהם מאוחסנים אוסף התכונות של עצם טיפוס מחלקה.

הפקודה:

```
MyType t;
```

מגדירה משתנה t מטיפוס MyType המכיל הפנייה לעצם מטיפוס זה. וכך זה נראה:



• פעולה בונה (constructor):

פעולת יצירה הגורמת להיווצרות עצם במצב תחילי רצוי.

הפקודה:

```
t = new MyType ();
```

- מייצרת הפנייה t לעצם מטיפוס MyType במצב תחילי רצוי.
- המצב התחילי קובע ערכי ברירת מחדל לתכונות העצם בתוך המחלקה.
 - מהפקודה שבדוגמא זו לא ניתן ללמוד דבר על תכונות העצם.

באופן דומה הפקודה:

```
t = new MyType ("blue", 21, 'C', true);
```

- מייצרת הפנייה t לעצם מטיפוס MyType במצב תחילי שנקבע באופן חיצוני.
- מהפקודה שבדוגמא זו ניתן ללמוד כי לעצם קיימות התכונות הבאות:
מחרוזת שערכה לכחול, מספר שלם שערכו 21, תו שערכו C ותכונה בוליאנית שערכה "אמת".
 - העצם יכול להכיל תכונות נוספות שערכיהן קבועים בתוך המחלקה.

ב-Java מגדירים חדש ע"י יצירת מחלקה באופן הבא:

```
public class MyType → הצהרה על מחלקה מטיפוס חדש
{
    private String type;
    private String color;
    private int num;
    private char tav;
    private boolean flag;
}

public MyType ()
{
    this.type="TV";
    this.color=" ";
    this.num=0;
    this.tav=' ';
    this.flag=false;
}
```

הצהרה על תכונות הטיפוס

הצהרה על בנאי שמייצר עצמים מהטיפוס החדש וקובע ערכי ברירת מחדל לתכונותיו

```
public MyType(String col,int x,char ot,boolean fg)
{
    this.type="TV";
    this.color=col;
    this.num=x;
    this.tav=ot;
    this.flag=fg;
}
} → סוגר את המחלקה
```

הצהרה על בנאי שמייצר עצמים מהטיפוס החדש. הבנאי קובע את התכונה type להכיל ערך ברירת מחדל TV, שאר התכונות נקבעות באופן חיצוני

הפקודה:

```
MyType t = new MyType();
```

הפקודה:

```
MyType t = new MyType("blue",21,'C',true);
```

מגדירות, מייצרות ומאתחלות עצם מטיפוס MyType בעזרת ההפנייה t.

זימון הפעולה נעשה בתוך הפעולה הראשית.

• פעולות נוספות:

על מנת שניתן יהיה להשתמש בעצמים החדשים, להתייחס לתכונותיהם הפרטיות וליישם באמצעותם פרויקטים שונים, עלינו להוסיף לתכונות ולבנאים אוסף של פעולות שניתן לבצע על עצמים אלו.

קיימים שני סוגים של פעולות:

- פעולות מאחזרות ערכים:

פעולות שמחזירות מידע על העצם או על אחת מתכונותיו.

```
public int getNum()
{
    return this.num;
}
```

בכותרת הפעולה עלינו להצהיר על הערך שהפעולה מחזירה. בגוף הפעולה יש להחזיר את המידע הרצוי בעזרת הפעולה return.

- פעולות מעדכנות ערכים

פעולות שתפקידן לשנות את מצבו של העצם ולשנות תכונה אחת או יותר פעולות אלו לא מחזירות ערך, אך עליהן לקבל כפרמטר ערך חדש (או מספר ערכים חדשים) לעדכון התכונה.

```
public void setNum(int x)
{
    this.num=x;
}
```

פעולה נוספת שמיחסים לה חשיבות נפרדת היא הפעולה `toString`.
פעולה זו מחזירה מחרוזת המתארת את טיפוס הנתונים החדש כפי שהמתכנת ניסח.

```
public String toString()  
{  
    return ".....";  
}
```

• מוסכמות הכתיבה:

- ב Java, כמו בכל שפה, ישנן מוסכמות לגבי סגנון הכתיבה בשפה. מוסכמות אלה הן כללים שהמתכנתים בשפה קיבלו על עצמם ומומלץ לציית להן.
 - שם מחלקה מתחיל באות גדולה.
 - שימוש בשמות בעלי משמעות. כדי לאפשר קריאה נוחה של קוד (אפילו אם הוא מכיל שמות מורכבים וארוכים), ייכתב שם המחלקה, כך שכל מילה חדשה תתחיל באות גדולה לדוגמא:

```
public class PinkWormWithBlueDots
```

- שמות התכונות ושמות הפעולות ייכתבו גם הם בהתאם לאותו כלל אלא שהמילה הראשונה תתחיל באות קטנה. לדוגמא:

```
int numOfLegs;           תכונה  
  
public int getNumOfLegs()  כותרת פעולה
```

כל הכתוב לקוח מהספרים של תוכנית הלימודים:
יסודות מדעי המחשב 1 ו-2 / מכון ויצמן
ציצוב תוכנה / אוניברסיטת ירושלים