

### לפותרים בשפת Java

בגן החיות "חיים בשלום" פותחה מערכת ממוחשבת ובה שומרים נתונים על בעלי החיים שבגן. המערכת כוללת את המחלקות האלה: בעל חיים – **Animal**, ציפור – **Bird**, תוכי – **Parrot**, דג – **Fish**, נחש – **Snake**. נתון חלק ממבנה המחלקות במערכת הממוחשבת:

```
public class Animal {
    private String color;
    private int weight;
    public Animal (String color, int weight)
    {
        this.color = color;
        this.weight = weight;
    }
    public String toString() {
        return "My color is " + this.color + "! And I weigh " + weight + " kilos!";
    }
}

public class Bird {
    private String color;
    private int weight;
    private boolean isFlying;
    public void zoo () {
        System.out.println ("Hello");
    }
}

public class Parrot {
    private String color;
    private int weight;
    private boolean isFlying;
    private boolean isTalking;
    public void zoo () {
        System.out.println ("Hello");
    }
}

public class Fish {
    private String color;
    private int weight;
    private String waterType;    // דג ים/דג מים מתוקים
}

public class Snake {
    private String color;
    private int weight;
    private int length;
    private boolean isVenomous;  // האם הנחש ארסי?
}
```

הניחו שיש פעולות get ו-set לתכונות המחלקות.

א. כתבו מחדש את הכותרות, התכונות והפעולות של המחלקות: **Fish**, **Parrot**, **Bird** ו- **Snake**, לפי עקרונות תכנות מונחה עצמים (אין לשנות את המחלקה Animal). בסעיף זה אין צורך לכתוב פעולות בונות. הניחו שהפעולות get ו- set קיימות.

נוסף על כך, סרטטו תרשים הייררכייה בין כל המחלקות. יש לסמן ירושה באמצעות החץ .

ב. נתונה המחלקה Tester:

```
public class Tester {  
    public static void main (String[] args) {  
        Animal[] animals = new Animal [5];  
        animals [0] = new Bird ("white", 4, false);  
        animals [1] = new Fish ("blue", 3, "sweet water");  
        animals [2] = new Parrot ("brown", 12, true, true);  
        animals [3] = new Snake ("gray", 2, 6, true);  
        animals [4] = new Snake ("black", 3, 4, false);  
        /**/  
    }  
}
```

הוסיפו פעולות בונות למחלקות **Bird**, **Parrot**, **Fish** ו- **Snake**, המאתחלות את תכונות המחלקות (כך שהפעולה main תרוץ ללא שגיאות).

הערה: אין לשנות את הפעולה הבונה הנתונה של **Animal**.

ג. (1) בפעולה main נוסף קטע הקוד שלפניכם (במקום המסומן ב- \*\*):

```
for (int i = 0; i < animals.length; i++) {  
    System.out.println (animals[i]); }
```

קטע הקוד תקין ומדפיס את הפלט שלפניכם:

```
I'm a bird! My color is white! And I weigh 4 kilos!  
My color is blue! And I weigh 3 kilos!  
I'm a parrot! My color is brown!  
I'm a snake! I'm venomous, be careful! My color is gray! And I weigh 2 kilos!  
I'm a snake! I'm not venomous! My color is black! And I weigh 3 kilos!
```

הוסיפו את הפעולה toString במחלקות (רק היכן שיש צורך) לפי עקרונות תכנות מונחה עצמים, כך שהפלט יהיה מתאים.

הערה: אין לשנות את הפעולה toString במחלקה **Animal**. תשובה שבה יהיה שינוי לא תזוכה בנקודות.

(2) במחלקה Tester נוספה הפעולה שלפניכם:

```
public static void hello (Animal [] arr) {  
    for (int i = 0; i < arr.length; i++) {  
        arr[i].zoo(); }  
}
```

אם הפעולה תקינה, כתבו את הפלט שלה עבור המערך animals (הנתון לעיל), ואם הפעולה לא תקינה – תקנו אותה (כך שהקריאה ל- zoo בתוך הלולאה תהיה תקינה) וכתבו את הפלט עבור המערך animals. הערה: – אין לשנות את המחלקות עצמן. תשובה שבה יהיה שינוי במחלקות לא תזוכה בנקודות.

בגן החיות "חיים בשלום" פותחה מערכת ממוחשבת ובה שומרים נתונים על בעלי החיים שבגן. המערכת כוללת את המחלקות האלה: בעל חיים – **Animal**, ציפור – **Bird**, תוכי – **Parrot**, דג – **Fish**, נחש – **Snake**.  
נתון חלק ממבנה המחלקות במערכת הממוחשבת:

```
public class Animal {
    private string color;
    private int weight;
    public Animal (string color, int weight)
    {
        this.color = color;
        this.weight = weight;
    }
    public override string ToString() {
        return "My color is " + this.color + "! And I weigh " + weight + " kilos!";
    }
}

public class Bird {
    private string color;
    private int weight;
    private bool isFlying;
    public void Zoo () {
        Console.WriteLine ("Hello");
    }
}

public class Parrot {
    private string color;
    private int weight;
    private bool isFlying;
    private bool isTalking;
    public void Zoo () {
        Console.WriteLine ("Hello");
    }
}

public class Fish {
    private string color;
    private int weight;
    private string waterType;    // דג ים/דג מים מתוקים
}

public class Snake {
    private string color;
    private int weight;
    private int length;
    private bool isVenomous;    // האם הנחש ארסי?
}
```

הניחו שיש פעולות Get ו- Set לתכונות המחלקות.

א. כתבו מחדש את הכותרות, התכונות והפעולות של המחלקות: **Fish**, **Parrot**, **Bird** ו- **Snake**, לפי עקרונות תכנות מונחה עצמים (אין לשנות את המחלקה Animal). בסעיף זה אין צורך לכתוב פעולות בונות. הניחו שהפעולות Get ו- Set קיימות.

נוסף על כך, סרטטו תרשים הייררכייה בין כל המחלקות. יש לסמן ירושה באמצעות החץ .

ב. נתונה המחלקה Tester:

```
public class Tester {  
    public static void Main (string[] args) {  
        Animal[] animals = new Animal [5];  
        animals [0] = new Bird ("white", 4, false);  
        animals [1] = new Fish ("blue", 3, "sweet water");  
        animals [2] = new Parrot ("brown", 12, true, true);  
        animals [3] = new Snake ("gray", 2, 6, true);  
        animals [4] = new Snake ("black", 3, 4, false);  
        /**/  
    }  
}
```

הוסיפו פעולות בונות למחלקות **Bird**, **Parrot**, **Fish** ו- **Snake**, המאתחלות את תכונות המחלקות (כך שהפעולה Main תרוץ ללא שגיאות).

הערה: אין לשנות את הפעולה הבונה הנתונה של **Animal**.

ג. (1) בפעולה Main נוסף קטע הקוד שלפניכם (במקום המסומן ב- \*\*\*):

```
for (int i = 0; i < animals.Length; i++) {  
    Console.WriteLine (animals[i]); }
```

קטע הקוד תקין ומדפיס את הפלט שלפניכם:

```
I'm a bird! My color is white! And I weigh 4 kilos!  
My color is blue! And I weigh 3 kilos!  
I'm a parrot! My color is brown!  
I'm a snake! I'm venomous, be careful! My color is gray! And I weigh 2 kilos!  
I'm a snake! I'm not venomous! My color is black! And I weigh 3 kilos!
```

הוסיפו את הפעולה ToString במחלקות (רק היכן שיש צורך) לפי עקרונות תכנות מונחה עצמים, כך שהפלט יהיה מתאים.

הערה: אין לשנות את הפעולה ToString במחלקה **Animal**. תשובה שבה יהיה שינוי לא תזוכה בנקודות.

(2) במחלקה Tester נוספה הפעולה שלפניכם:

```
public static void Hello (Animal [] arr) {  
    for (int i = 0; i < arr.Length; i++) {  
        arr[i].Zoo(); }  
}
```

אם הפעולה תקינה, כתבו את הפלט שלה עבור המערך animals (הנתון לעיל), ואם הפעולה לא תקינה – תקנו אותה (כך שהקריאה ל- Zoo בתוך הלולאה תהיה תקינה) וכתבו את הפלט עבור המערך animals. הערה: – אין לשנות את המחלקות עצמן. תשובה שבה יהיה שינוי במחלקות לא תזוכה בנקודות.