

## תרגיל מספר 3

תרגיל זה עוסק בנושאי הרצאות 3 ו-4. מומלץ לעבור תחילה על הרצאות אלה. תרגיל זה מהווה **תרגיל המשך לתרגיל 2** ובו נמשיך לעסוק בטיפול בנתוני בעלי החיים בגן החיות "קוקו". לצורך כתיבת התרגיל ניתן להשתמש בקוד של תרגיל מספר 2 אולם יש **לערוך** בו שינויים וכן **להוסיף** קוד התואם לתרגיל זה.

נעבור לטיפול בשני סוגי בעלי חיים:

- בעלי חיים בריאים הזקוקים לטיפול שגרתי (אוכל, מים וכו') יטופלו באמצעות מבנה נתונים **תור**.
- תור – מבנה נתונים התומך בעיקרון FIFO-First In First Out הראשון שנכנס הוא הראשון שיצא.
- בעלי חיים חולים הזקוקים לטיפול תרופתי יטופלו באמצעות מבנה נתונים **מחסנית**.
- מחסנית – מבנה נתונים התומך בעיקרון LIFO-Last In First Out האחרון שנכנס הוא הראשון שיצא.
- יש לפתח תוכנה לפיה נטפל בבעלי החיים בגן כך שניתן יהיה לבצע את הפעולות הבאות:
  - הוספה של בעל חיים חדש – לתור או למחסנית תלוי בדחיפות
  - טיפול בבעל חיים
  - הדפסת נתוני בעלי החיים

לבעל חיים המאפיינים הבאים :

- קוד
- שם
- זן (Male/Female) – יש להגדיר **enum** המכיל קבועים אלה
- משקל בק"ג
- האם בעל החיים חי במים? – שדה בוליאני
- מיקום בתור – position
- דחיפות – שדה בוליאני, urgent
- מונה בעלי חיים בריאים – שדה **סטטי**

- מונה בעלי חיים חולים – שדה **סטטי**

1. הוסף למחלקה Animal מתרגיל מספר 2 את השדות הנוספים.
2. כתוב לכל שדה **properties** בהתאם לשדה (ישנם שדות שדורשים גם get וגם set ואחרים דורשים רק אחד מהם. החלט בעצמך לפי צרכי התרגיל).
3. כתוב בנאי המקבל את כל הנתונים (למעט השדות הסטטיים).
4. אל תשכח לעדכן בבנאי את השדה הסטטי הרלבנטי.
4. כתוב מתודה printAnimalInfo אשר מדפיסה את פרטי בעלי החיים.
5. כתוב את המחלקה Queue המטפלת בבעלי חיים בריאים

המחלקה תכיל את השדות הבאים:

- Animals, מערך בעלי החיים
  - emptyCell, מיקום התא הפנוי הבא
- המחלקה תכיל את המתודות הבאות:
- בנאי המקבל את גודל המערך
  - בנאי ריק אשר מפעיל את הבנאי מהסעיף הקודם (שולח גודל = 1)
  - Enqueue – מתודה המוסיפה בעל חיים לסוף התור, במידה והתור אינו מלא
  - Dequeue – מתודה המוציאה בעל חיים מתחילת התור, במידה והתור אינו ריק
  - IsEmpty – מתודה בוליאנית הבודקת האם התור ריק
  - IsFull – מתודה בוליאנית הבודקת האם התור מלא
  - PrintQueue – מדפיסה את בעלי החיים בתור
  - getLastPos – מחזירה את מיקום התא הפנוי הבא בתור
  - setDownAllPos – מקטינה את המיקום של כל בעלי החיים

6. כתוב את המחלקה Stack המטפלת בבעלי החיות החולים

המחלקה תכיל את השדות הבאים:

- Animals, מערך בעלי החיים
  - Pointer, מיקום התא הפנוי הבא במחסנית
- המחלקה תכיל את המתודות הבאות:
- בנאי המקבל את גודל המערך
  - בנאי ריק אשר מפעיל את הבנאי מהסעיף הקודם (שולח גודל = 1)

- Push – מתודה המוסיפה בעל חיים לראש המחסנית (הכנסה למיקום ה-Pointer)
- Pop – מתודה המוציאה את בעל החיים שנמצא בראש המחסנית (הוצאה ממיקום ה-Pointer)
- isEmpty – מתודה בוליאנית הבודקת האם המחסנית ריקה
- isFull – מתודה בוליאנית הבודקת האם המחסנית מלאה
- printStack – מדפיסה את בעלי החיים במחסנית
- getLastPos – מחזירה את מיקום התא הפנוי הבא במחסנית
- setUpAllPos – מגדילה את המיקום של כל בעלי החיים
- setDownAllPos – מקטינה את המיקום של כל בעלי החיים

7. כתוב את המחלקה Service אשר מכילה שני שדות: רשימת בעלי החיים הבריאם (תור) ורשימת הלקוחות החולים (מחסנית) המחלקה תכיל את המתודות הבאות:

- בנאי ריק היוצר את הרשימות – לצורך התרגיל ניתן לייצר תור ומחסנית בגודל לפי בחירתך.
- addNewAnimals – מבקשת מהמשתמש את נתוני בעל החיים ומכניסה אותו לרשימה הרלבנטית לפי רמת הדחיפות
- printUrgentAnimals – מדפיסה את בעלי החיים החולים שזקוקים לטיפול תרופתי
- printRegularAnimals – מדפיסה את בעלי החיים הבריאם שזקוקים לטיפול שגרתי
- TakeCareNextAnimal – מקבלת כפרמטר האם דחוף ומטפלת בבעל החיים הבא

8. מצורף main המכיל תפריט עם האופציות הבאות:

1. הוספת בעל חיים חדש
2. טיפול בבעל החיים הבא – יש לבקש מהמשתמש רמת דחיפות ולטפל בבעל החיים הבא
3. הדפסת בעלי החיים בהתאם לרמת הדחיפות
4. יציאה

עליכם להשתמש ב-main זה ולא לערוך בו שינויים.

```

namespace exe3
{
    enum MenuAnimal { ADD_ANIMAL, TAKE_CARE, PRINT, EXIT };
    class Program
    {
        static void Main(string[] args)
        {
            MenuAnimal menuChoice;
            Service service = new Service();
            do
            {
                Console.Clear();
                string Menu = "\nAnimal Manager \n\n\t1 - Add New Animal \n\t2
- Take care of next animal\n\t3 - Print animals List \n\t4 -
Exit\n\t\t\t\t\t*\n\t\t\t\t\t*\n\t\t\t\t\t*";
                Console.WriteLine(Menu);
                menuChoice = (MenuAnimal)(int.Parse(Console.ReadLine()));
                if (menuChoice == MenuAnimal.EXIT)
                {
                    Console.WriteLine("Have a good day!");
                    break;
                }
                switch (menuChoice)
                {
                    case MenuAnimal.ADD_ANIMAL: // Add New animal
                    {
                        Console.WriteLine("\n\t Add New Animal ");
                        service.addNewAnimal();
                        break;
                    }
                    case MenuAnimal.TAKE_CARE: // take care next animal
                    {
                        Animal ani;
                        Console.Write("\n\tpress R for regulal animals or U
for urgent animals: ");
                        char urgent = char.Parse(Console.ReadLine());
                        if (urgent == 'R')
                            ani = service.TakeCareNextAnimal(false);
                        else
                            ani = service.TakeCareNextAnimal(true);
                        if (ani != null)
                        {
                            Console.WriteLine("Take care of animal:");
                            ani.printAnimalInfo();
                        }
                        else
                            Console.WriteLine("\n\t - There are no waiting
animals in list");
                        break;
                    }
                    case MenuAnimal.PRINT: // Print animals
                    {
                        Console.WriteLine("Which animal list do you want to
display?");
                        Console.WriteLine("Regular animal list - press
R\nUrgent animal list - press U");
                        char type = char.Parse(Console.ReadLine());
                        if (type == 'R')
                            service.printRegularAnimals();
                        else
                            service.printUrgentAnimals();
                    }
                }
            }
        }
    }
}

```

```
        break;
    case MenuAnimal.EXIT: // Exit
        Console.WriteLine("Have a nice day!");
        break;
    default:
        Console.WriteLine("\n\t{{ Wrong Selection }}");
        break;
    }
    Console.WriteLine("\n\n\tPress Anykey For The Menu");
    Console.ReadLine();
} while (menuChoice != MenuAnimal.EXIT);
}
}
```

לפניכם דוגמת הרצה אחת

עליכם לכתוב קוד כך שיתקבל **בדיוק** הפלט כפי שמוצג בדוגמא זו.

```
C:\Users\זינו\source\repos\ConsoleApp1\ConsoleApp1\bin\Debug\ConsoleApp1.exe

Animal Manager

    1 - Add New Animal
    2 - Take care of next animal
    3 - Print animals List
    4 - Exit

                                     *
                                     *
                                     *

1

    Add New Animal
insert code number: 1
insert name: lion
insert weigth number: 150
  Female/Male animal?
Male
The animal has to live in the water?-press Y
other - press N
N
Is it an urgent issue? press Y/N
Y

    Press Anykey For The Menu
```

```
Animal Manager

1 - Add New Animal
2 - Take care of next animal
3 - Print animals List
4 - Exit

*
*
*

1

    Add New Animal
insert code number: 2
insert name: dolphin
insert weight number: 120
    Female/Male animal?
Female
The animal has to live in the water?-press Y
other - press N
Y
Is it an urgent issue? press Y/N
Y

    Press Anykey For The Menu
```

```
Animal Manager

1 - Add New Animal
2 - Take care of next animal
3 - Print animals List
4 - Exit

*
*
*

2

    press R for regular animals or U for urgent animals: U
Take care of animal:
Position in list:1, name: dolphin, code: 2, type animal: Female, is sea?: True, weight: 120

    Press Anykey For The Menu
```

```
Animal Manager

1 - Add New Animal
2 - Take care of next animal
3 - Print animals List
4 - Exit

*
*
*

2

press R for regulal animals or U for urgent animals: R

- There are no waiting animals in list

Press Anykey For The Menu
```

```
Animal Manager

1 - Add New Animal
2 - Take care of next animal
3 - Print animals List
4 - Exit

*
*
*

1

Add New Animal
insert code number: 3
insert name: monkey
insert weigth number: 50
Female/Male animal?
Male
The animal has to live in the water?-press Y
other - press N
N
Is it an urgent issue? press Y/N
N

Press Anykey For The Menu
```



```

Animal Manager

    1 - Add New Animal
    2 - Take care of next animal
    3 - Print animals List
    4 - Exit

                                     *
                                     *
                                     *
3
Which animal list do you want to display?
Regular animal list - press R
Urgent animal list - press U
U
There are 2 animals in urgent list.
Animals in Stack:
Position in list:1, name: dolphin, code: 2, type animal: Female, is sea?: True, weighth: 120
Position in list:1, name: lion, code: 1, type animal: Male, is sea?: False, weighth: 150

Press Anykey For The Menu

```

```

Animal Manager

    1 - Add New Animal
    2 - Take care of next animal
    3 - Print animals List
    4 - Exit

                                     *
                                     *
                                     *
3
Which animal list do you want to display?
Regular animal list - press R
Urgent animal list - press U
U
There are 1 animals in urgent list.
Animals in Stack:
Position in list:1, name: lion, code: 1, type animal: Male, is sea?: False, weighth: 150

Press Anykey For The Menu

```

## הנחיות הגשה

### 1. יש להגיש במודל 2 קבצים:

- **תיקייה בסיומת zip** כאשר שם התיקייה יהיה "שם מלא סטודנט 1\_שם מלא סטודנט 2". התיקייה תכיל את הקבצים Animal.cs, Service.cs, Queue.cs, Stack.cs

- **קובץ pdf** - הקובץ צריך להיות מסודר, בתחילת הקובץ יש לרשום את שמות המגישים ולהכיל כותרת מסודרת עבור כל מחלקה, העתקה של הקוד (לא צילום)

- מסך) ובסוף צילום מסך של מסך הפלט (צילום איכותי שניתן לראות- צילומים לא ברורים יגוררו הורדת נקודות).
2. אופן ההגשה המלא מפורט בדף נוהל הגשת מטלות אשר נמצא באתר הקורס.
3. יש להגיש את התרגיל עד התאריך 3.5.2020

**עבודה נעימה!!!**