**מבוא לתכנות מונחה עצמים:**

[**http://blog.csit.org.il/UpLoad/FilesUpload/CS\_cp1.pdf**](http://blog.csit.org.il/UpLoad/FilesUpload/CS_cp1.pdf)

**אוספים(חזרה)**

**אוסף(collection) של אובייקטים בעלי אותו סוג מגדיר בעצמו מושג חדש(אובייקט).**

**דוגמאות: אוסף ילדים בגן, אוסף תלמידים בכיתה, אוסף תלמידים הלומדים מחשבים מאבן יהודה, אוסף שחקנים כדור סל בבית הספר, ....**

**אוספים מיועדים לשמירה בטוחה של איברים(עצמים/אובייקטים) בעלי אותו סוג/טיפוס וביצוע פעולות על האוסף או מספר אוספים.**

**לדוגמה: פעולות הבאות הן הכי נפוצות באוספים: הוספת עצם לאוסף, הוצאת עצם מאוסף, עדכון תוכן העצם באוסף, פעולות חיפוש עצמים באוסף, מיונים של אוספים, ...**

**כדי לאפשר עבודה עם האוספים בשפת תכנות C# אנו נשתמש במערך ככלי הפשוט ביותר הידוע לנו עד כה, המאפשר ליצור אוספים ולטפל בהם בכלות(יחסית).**

**מהם היתרונות של מערכים ומהם החסרונות ?**

**משימה:**

**יש ליצור אוסף שיכיל לכל היותר 100 מספרים שלמים ויאפשר לנו להוסיף מספרים שלמים לאוסף, להוציא אותם מן האוסף, להשוואת ביניהם באוסף, לחפש מספרים באוסף, למיין אוסף. וכן, לאחד אוספים, לחתוך האוספים ופעולות נוספות.**

**רגע חושבים!!!**

**מהן התכונות העצם "אוסף מספרים שלמים" IntColl ???**

**מהן הפעולות שמטפלות ביצירת הצם ותפקודו ??? ובין העצמים "אוספים" ???**

**שלב1. טבלת UML :**

|  |
| --- |
| **IntColl** |
| **אוסף מספרים שלמים: - int[] coll**  **מספר איברים באוסף - current** |
| **+ IntColl(int length)**  **+ bool IsFull()**  **+ bool IsEmpty()**  **+ bool Search(int x)**  **+ int CountXNumbers(int x)**  **+ bool Insert(int x)**  **+ int PlaceInCol(int x)**  **+ bool BinSearch(int x)**  **+ bool InsertSorted(int x)**  **+ bool Remove(int x)**  **+ void DeleteAllXNumbers(int x)**  **+ void DeleteAllDuplicatesForNumber(int x)**  **+ int GetMaxValue()**  **+ int GetMinValue()**  **+ override string ToString()**  **----------------------------- לפעולות בין האוספים --------------------------**  **+ IntColl Merge(IntColl other)**  **+ IntColl Unite(IntColl other)**  **+ IntColl Intersection(IntColl other)**  **+ IntColl Subtraction(IntColl other)**  **+ IntColl Equals(IntColl other)**  **+ bool IsSubIntColl(IntColl other)** |

**שלב2. יישום מחלקה IntColl:**

**public class IntColl**

**{**

**private int[] coll; // אוסף מספרים הממוין לפי סדר עולה**

**private int current; // מספר איברים באוסף**

**public IntColl(int length) // length - מספר איברים מקסימלי באוסף**

**{**

**this.coll = new int[length + 1];**

**this.current = 0;**

**}**

**public bool IsFull()**

**{**

**return this.current + 1 == this.coll.Length;**

**}**

**public bool IsEmpty()**

**{**

**return this.current == 0;**

**}**

**public bool Search(int x)**

**{**

**for (int i = 0; i < this.current; i++)**

**if (this.coll[i] == x)**

**return true;**

**return false;**

**}**

**public int CountXNumbers(int x)**

**{**

**int count = 0;**

**for (int i = 0; i < this.current; i++)**

**if (this.coll[i] == x)**

**count++;**

**return count;**

**}**

**public bool Insert(int x)**

**{**

**if (this.IsFull())**

**return false;**

**this.coll[this.current++] = x;**

**return true;**

**}**

**private int PlaceInCol(int x)**

**{**

**for (int i = 0; i < this.current; i++)**

**if (this.coll[i] == x)**

**return i;**

**return -1;**

**}**

**public bool Remove(int x)**

**{**

**int place = PlaceInCol(x);**

**if (place == -1)**

**return false;**

**for (int i = place; i < this.current; i++)**

**this.coll[i] = coll[i + 1];**

**this.current--;**

**return true;**

**}**

**public void DeleteAllXNumbers(int x)**

**{**

**while(Search(x))**

**Remove(x);**

**}**

**public void DeleteAllDuplicatesForNumber(int x)**

**{**

**DeleteAllXNumbers(x);**

**InsertSorted(x);**

**}**

**public int GetMaxValue()**

**{**

**int max = int.MinValue;**

**for (int i = 0; i < this.current; i++)**

**if (max < this.coll[i])**

**max = this.coll[i];**

**return max;**

**}**

**// הנחה: מספר איברים באוסף שונה מאפס**

**public int GetMinValue()**

**{**

**int min = int.MaxValue;**

**for (int i = 0; i < this.current; i++)**

**if (min > this.coll[i])**

**min = this.coll[i];**

**return min;**

**}**

**}**

**מסקנות:**

פתחנו בקלות מחלקה אשר מטפלת באוסף של מספרים שלמים.

באותה דרך אפשר גם לטפל באוסף של מספרים ממשיים, אוסף תווים, אוסף מחרוזות, אוסף של עצם כלשהו. כלומר לכל עצם ליצור מחלקה מתאימה.

מהו ההבדל בין כל המחלקות האלה?

ההבדל היחיד הוא טיפוס של האיברים באוסף. כלומר יש שינוי בהצהרת הטיפוס הנדרש. כל הפעולות הן אותן פעולות בלי שינוי ממש.

האם ניתן להתגבר על כפילויות של הקוד ? התשובה היא כן!

**משימה למחר: תבדקו את הפעולות האלה במחלקה Program.**