Programming the .Net Framework using C#

# Sorted Pair – Generic Programming

Object Oriented programming in C#

## הוראות

1. כתבו מחלקה גנרית בשם SortedPair המשמשת ליצירת זוגות ממוינים של איברים (שני איברי הזוג הם מאותו הטיפוס). החלק הגנרי מגדיר את טיפוס האיברים כטיפוס המממש את הממשק IComparable<T>. כלומר כל שידוע על האיברים הוא שהם ניתנים להשוואה באמצעות המתודה CompareTo.
   1. המחלקה תכלול בנאי (constructor) המאפשר ליצור אובייקטים המייצגים זוג איברים מהטיפוס. הבנאי מקבל את האיברים בסדר כלשהו ויוצר מהם זוג ממוין. במקרה שהאיברים שווים יש לעורר מצב חריג (Exception) בשם IllegalPair (עליכם להגדיר מחלקה זו כתת-מחלקה של Exception).
   2. הוסיפו Properties First ו-Second להחזרת האיבר הקטן והגדול של הזוג
   3. ממשו (override object’s ToString method) את המתודה ToString להחזרת מחרוזת המייצגת את הזוג הממוין (קודם הקטן ולאחריו הגדול) .
   4. ממשו (override object’s Equals method)את המתודה Equals עבור מחלקה זו. שני זוגות הם שווים אם ורק אם כל אחד מהאיברים שלהם שווים.
2. בדקו את המחלקה הגנרית SortedPair עבור זוגות של מחרוזות (strings) . כתבו תכנית ראשית הקולטת זוגות של מחרוזת, יוצרת זוגות ממוינים ושומרת אותם ב-List. לבסוף, יש לבצע איטרציה על ה- List ולהציג את הזוגות הממוינים.
3. בדקו את המחלקה הגנרית SortedPair עבור זוגות של נקודות במישור (כל נקודה מורכבת מ-x ו-y). הוראות לבניית הSTRUCT POINT:

בהינתן שתי נקודות: נקודה תחשב כקטנה מנקודה אחרת אם רכיב ה-x שלה קטן מרכיב ה-x של הנקודה האחרת, ורכיב ה-y שלה קטן מרכיב ה-y של הנקודה האחרת. הנקודה תחשב כגדולה מהנקודה האחרת אם רכיב ה-x שלה גדול מרכיב ה-x של הנקודה האחרת, ורכיב ה-y שלה גדול מרכיב ה-y של הנקודה האחרת. בכל מקרה אחר, הנקודות תחשבנה שוות. לדוגמה, הנקודה (5,7) גדולה מהנקודה (3,2) ואילו הנקודות (2,4) ו- (1,5) ייחשבו כשוות.

כתבו תכנית ראשית המייצר 10 זוגות של נקודות עם ערכי x ו-y רנדומלים בין 0..100. עבור כל זוג, התכנית יוצרת זוג ממוין ומציגה אותו בפלט הסטנדרטי. במקרה של ניסיון ליצור זוג לא חוקי יש לתפוס את המצב החריג ולהציג הודעה מתאימה.

## Detailed Instructions

## Features in C# that may be used in this project:

* **All** C# features from our course, including Generics, Generic collections, Interfaces, Exceptions, structs classes etc.

## Important Notes

* Build your classes based on the principles we learned in class.
* Define private and public as needed.
* As always, make sure your code is readable, using meaningful names, indentation and documentation.
* Avoid code duplication
* Do not use Console methods in the SortedPair class – make sure your class can be added in the future to an application with Graphical User Interface without changes. ( you may use Console WriteLine etc in the Main function in Program)