**שפות תכנות מונחי עצמים**

**תרגיל תכנות מספר 3**

1. עליך לממש מחלקה ג'נרית בשם SortedArr המתחזק **מערך** ממוין של איברים שסוגם **פתוח** **אך הומוגני**. שדות המידע של המחלקה חייבות להיות private.

קוד לקוח המשתמש במחלקה חייב לספק שמות של 2 רוטינות המשוות בין 2 אברים מהסוג שבו מדןבר ומחזיר true או false אם (א) האיבר הראשון גדול מהשני, (ב) האיבאים שווים.

גודל המערך הוא בדיוק בגודל הנחוץ, כל פעם מקצים מערך חדש כל אימת שמוספים או גורעים איבר. הפעולות הפומביות (public) הנתמכות הן:

1. בנאי המקבל את שמות 2 הרוטינות שמצוינות לעיל.
2. מתודה

public void add(Type newMember)

המוסיפה איבר למערך

1. מתודה

bool delete(Type delMember)

המוחקת איבר מהמערך ומחזירה true או false אם הוא היה שם.

1. מאפיין N המאפשר קריאה בלבד של מספר האברים במערך.
2. מתודה

Type Geti(int i)

המחזירה את האיבר ה-i-יי במערך.

1. עליך לממש מחלקה ג'נרית בשם Priority הנגזרת מ-SortedArr והממש תור עדיפויות המחלצת בכל פעם את המקסימום. המימוש חייב להיות מינימלי ולהסתמך ככל האפשר על הקוד של המחלקה SortedArr.

המתודות הפומביות חייבות להיות:

1. בנאי עם פרמטרים כמו זה של SortedArr.
2. מתודה

Type removeMax()

הגורע את האיבר המרבי בתור ומחזיר את ערכו.

לדוגמא, הפלט של התוכנית הבאה:

class Class1

{

public static bool doubleIsGreater(object x, object y)

{

double u, v;

u = (double)x;

v = (double)y;

return (u > v);

}// doubleIsGreater

public static bool doubleIsEqual(object x, object y)

{

double u, v;

u = (double)x;

v = (double)y;

return (u == v);

}// doubleIsEqual

public static bool intIsGreater(object x, object y)

{

int u, v;

u = (int)x;

v = (int)y;

return (u > v);

}// doubleIsGreater

public static bool intIsEqual(object x, object y)

{

int u, v;

u = (int)x;

v = (int)y;

return (u == v);

}// intIsEqual

static void Main(string [] args)

{

int i;

SortedArr<double> srt1 = new

SortedArr<double>(doubleIsGreater, doubleIsEqual);

srt1.add(22.2);

srt1.add(66.6);

srt1.add(22.2);

srt1.add(33.3);

srt1.add(55.5);

srt1.add(77.7);

srt1.add(11.1);

srt1.add(44.4);

**srt1.delete(55.5);**

Console.WriteLine("\nsrt1:");

for(i=0; i < srt1.N; i++)

Console.WriteLine(srt1.Geti(i));

Priority<int> pri1 = new Priority<int>(intIsGreater,intIsEqual);

pri1.add(11);

pri1.add(33);

pri1.add(77);

pri1.add(44);

pri1.add(55);

pri1.add(66);

pri1.add(22);

Console.WriteLine("\npri1:");

while(pri1.N > 0)

Console.WriteLine(**pri1.removeMax()**);

} // Main

} // Class1

תהיה:

srt1:

11.1

22.2

33.3

44.4

66.6

77.7

pri1:

77

66

55

44

33

22

11