חלק יבש תרגיל בית 2

1. במבנה הגנרי SortedList, הדרישות ההכרחיות שעל הטיפוס T לקיים הן:
   1. Constructor
   2. Destructor
   3. Copy constructor
   4. Operator=
   5. Operator<
   6. Operator<<

נסביר למה כל דרישה הכרחית לפעולת התוכנה:

* 1. הבנאי הכרחי ליצירת איבר גנרי חדש, בלעדיו אי אפשר ליצור רשימה גנרית ממוינת.
  2. ההורס הכרחי להריסת האיבר הגנרי, ובלעדיו ההורס של הרשימה המקושרת הממוינת לא תוכל לשחרר את הזיכרון כראוי.
  3. הבנאי המעתיק הכרחי להעתקת איבר גנרי, בלעדיו הבנאי המעתיק של הרשימה המקושרת הממוינת לא תוכל להעתיק איבר גנרי מהרשימה הממוינת.
  4. אופרטור ההשמה הכרחי להשמת איבר גנרי, ובלעדיו אופרטור ההשמה של הרשימה המקושרת הממוינת לא יוכל להשים את האיברים ברשימה המקושרת הממוינת.
  5. אופרטור > הכרחי להשוואת שני איברים גנריים, פעולה הכרחית לשמירת הרשימה המקושרת ממוינת כפי שנדרש בתרגיל.
  6. אופרטור >> הכרחי להדפסת האיבר הגנרי. באמצעותו בלבד תוכל הרשימה המקושרת הממוינת להדפיס את איבריה.

1. אם היינו ממשים איטרטור לא קבוע (non-const) עבור SortedList, היינו מקבלים שעבור הפעולה \* , ערך ההחזרה יהיה &T במקום const T&. מימוש זה בעייתי כיוון שמשתמש יכול בטעות לשנות את ערך אחד הפרמטרים של האיטרטור כך שערכיהם יהיה לא חוקי. לדוגמה, משתמש יכול לשנות את הindex של האיטרטור שמצביע על רשימה בעלת 5 איברים להיות 10, ואז האיטרטור מצביע על איבר לא חוקי. כאשר המשתמש ישתמש שנית בפעולת \* הוא יקבל שגיאה, או אם ייקרא ההורס של האיטרטור אז התוכנה תנסה למחוק זכרון שלא הקצתה ובכך תגרום לשגיאה.
2. שתי הדרכים השונות למימוש והעברת הפרדיקט הן באמצעות טמפלייטים כפי שהתמשנו בתרגיל או להשתמש בפוינטר לפונקציה, כפי שהשתמשנו בתרגיל בית הקודם. ההבדל בין שתי השיטות: בטמפלייטים מעבירים אובייקט מסוג פונקציה (FuncObject) ובעצם מעבירים משתנה מסוג הפונקציה. בשימוש באמצעות מצביע לפונקציה הוא באמצעות פוינטר מסוג void\*, ומשם מגיעה הגנריות של הפונקציה. נשים לב שבשני המקרים אנו מעבירים את משתנה הפונקציה באותה צורה: עבור הפונקציה func מסוג FuncType אנו נעביר ע"י filter(Functype func). לכן לא צריך לממש באמצעות שתי השיטות.