

מת"מ – תרגיל בית 2 – חלק יבש

שאלה 1:

1. אופרטור < חייב להיות מוגדר על איברי הטיפוס. אנחנו משתמשים באופרטור כדי למיין את הרשימה. אם אופרטור זה לא היה מוגדר אז אין משמעות לזה שהרשימה היא sorted.
2. ה-copy constructor של הטיפוס T חייב להיות מוגדר. אנחנו משתמשים בו כדי להעתיק את האיברים שהמשתמש נותן כקלט לפונקציה insert. אנחנו חייבים להעתיק את האיברים כדי שהמבנה שלנו לא יהיה תלוי במשתמש מבחינת זיכרון.

שאלה 2:

אם האיטרטור היה מחזיר T& אז המשתמש יכול לשנות את ערכי איברי הרשימה אחרי שהם כבר הושמו בה. זה יכול לגרום למצב שהרשימה אינה ממוינת. הדבר הזה בעייתי כיוון שבמימוש שלנו, אנחנו דואגים לזה שהאיברים יהיו מסודרים בסדר הנכון תוך כדי הכנסתם לרשימה. אם למשתמש יש גישה לשינוי איברי הרשימה אז במימוש הקיים לא תתקיים הדרישה שאיברי הרשימה הם ממוינים בכל רגע נתון.

שאלה 3:

שתי דרכים שונות לשינוי והעברת הפרדיקט:

1. מימוש: פונקציה שמחזירה bool ומקבלת פרמטר אחד מטיפוס T.
העברה: מעבירים ל-filter מצביע לפונקציה.
2. מימוש: עצם ממחלקה X כך של-X מוגדר operator(). חתימת העמסת האופרטור היא מהצורה:
`Bool operator()(T item);`
העברה: מעבירים ל-filter עצם מהמחלקה X.

ההבדל:

באפשרות 1, המשתמש כותב פונקציה רגילה ומעביר את הכתובת שלה למתודה filter. במימוש שלנו אנחנו פשוט מפעילים את הפונקציה על איברי ה-sorted_list.

באפשרות 2, המשתמש מעביר ממחלקה כמו שתואר קודם, ואנחנו במימוש שלנו של filter מפעילים את operator() על העצם עם פרמטר מטיפוס T שהוא איבר ב-sorted_list.

לא צריך לספק שני מימושים שונים כי:

C++ רואה את הפרמטר שמועבר ל-filter כ-object function. כל עצם שמוגדר עבורו אופרטור () הוא function object.

באפשרות השנייה – ברור למה אנחנו מעבירים עצם שמוגדר עבורו האופרטור.

באפשרות הראשונה – פונקציה רגילה היא בעצם function object שהוא built in ב-C++.