

1. הטיפוס T חייב לממש את `operator<` כדי שנוכל לעשות השוואה בין איברים על מנת למיין את `SortedList`. כמו כן נהיה חייבים לממש בנאי העתקה כדי שנוכל לייצר איברים חדשים שאנחנו מקבלים ב`insert`.

2. במידה ונרצה לממש איטרטור `non-const` שיחזיר `T&` כאשר מפעילים `*operator`. ייווצרו בעיות במימוש ובהתנהגות ה`SortedList`. אם נשנה איבר ב`SortedList` אז או שהרשימה הממוינת כבר לא תהיה ממוינת במקרה והאיבר לא אמור להיות במקום שבו הוא נמצא או שהיא תשנה את סדר האיברים בצורה מחוכמת ואז למעשה כאשר נרצה לקרוא מהאיבר במיקום הנתון הוא כבר יהיה איבר אחר.

3. דרך אחת היא לממש את פרדיקט כפונקציה שמקבלת `const T&` או `T` במקרה הפחות טוב ומחזירה `true` או `false` להאם האיבר צריך להיות כחלק מהרשימה הממוינת החדשה. `filter` יקבל מצביע לפונקציה כזאת וזאת דרך אחת. דרך שנייה תהיה לממש מחלקה שמממשת `operator()` שמתפקד כפרדיקט. וה`filter` יקבל רפרנס קבוע לאובייקט המחלקה. בגלל שאנחנו ממשים את `filter` שיקבל טיפוס שהוא מוגדר על ידי `template`, הקומפילר יקמפל את `filter` לקבל את שני הטיפוסים השונים כל עוד ניתן להפעיל על שניהם אופרטור סוגריים.