

## תרגיל בית 3- חלק יבש

תאריך: 24.7.2024

מספר קורס: 234124

שם: לינוי גבע ת.ז: 213518194

שם: עדי וולפמן ת.ז: 211402789

### 3.1 שאלות על SortedList

1. במבנה שבנינו הדרישות ההכרחיות שעל הטיפוס T לקיים הם:  
**בנאי העתקה** בהוספת אובייקט לרשימה עלינו ליצור עותק של האובייקט שאנו רוצות להוסיף על מנת ליצור מבנה עצמאי מבחינת זיכרון ומידע, ועל כן עליו להיות בעל בנאי העתקה שיאפשר זאת.

**אופרטור >** האובייקט שאנחנו מייצרים ממיין את הטיפוס T לפי האופרטור >, ולכן לכל אובייקט שנרצה ליצור לו רשימה האופרטור > חייב להיות ממומש.

**הורס-** אנחנו משתמשים באובייקט גנרי. בקוד שלנו אנו לא משתמשים בהקצאות דינאמיות של T, ועדיין על האובייקט להיות אחראי על הזיכרון שלו ולכן יש צורך בהורס בין אם דיפולטי ובין אם מוגדר.

\*במימוש הספציפי שלנו לא נדרש אופרטור השמה על T (גם לא default) כיוון שבאופרטור ההשמה של list אנו משתמשים ב-c'tore של T.

2. במידה והיינו מממשים במחלקה שלנו non-const iterator בלבד. אזי במקרה בו היינו עובדים עם SortedList<const T> הקוד לא היה עובר קומפילציה, שכן אנו מנסים להחזיר רפרנס רגיל עבור אובייקט שהוא const וזו המרה שאינה מתאפשרת.

3. בהינתן הפרמטרים הבאים:

```
int n; //the modulo parameter
SortedList<int> list; // the list with all the data
```

נשתמש ב-lambda בקריאה לפונקציית filter באמצעת הפקודה הבאה:

```
SortedList<int> newList = list.filter([](int i) { return (i % n == 0); });
```

כאמור הפקודה filter מסננת רשימה עבור פונקציה המחזירה ערך בוליאני עבור כל איבר שלה.

יצרנו כפרמטר לפונקציה, מתודה המקבלת int (טיפוס הרשימה) ובודקת את שארית החלוקה במספר n (מספר שיכל להתקבל ולהתעדכן בזמן ריצה).

הפונקציה filter תיצור רשימה חדשה בה ישארו רק המספרים שיתחלקו ב-n כנדרש.