

Ex3_Dry:

שני אמסלם ת"ז 322791724

היבא מלאק ת"ז 212744015

א. אם נגדיר את פעולות ה- Iterator הרגיל כ- `const`, כשנפעיל את ה- `Iterator` על תור רגיל, הדבר יגביל אותנו בפעולה על תור רגיל, כי לא נוכל לשנות את ערכי התור, לא נוכל להכניס ולהוציא איברים כי גם `*operator` יוגדר כ- `const`.

לדוגמה, באופרטור `*operator`, אנו רוצים עבור תור רגיל שיחזיר `reference` שנוכל לשנות את ערכו, בעוד עבור תור `const` עליו להחזיר `const` על מנת לשמור על ערכי התור ולכן לא ניתן להסתפק בשימוש ב- `Iterator` הרגיל כ- `const`.

ב. 1. בבנאי של `Queue` ובבנאי ההעתקה אנו מניחות שקיים בנאי חסר

ארגומנטים עבור הטיפוס הטמפלייטי `T`.

2. באופרטור השמה של `Queue`, אנו מניחות שקיים בנאי חסר

ארגומנטים ואופרטור השמה עבור הטיפוס הטמפלייטי `T`.

3. בפונקציה `popFront` מניחות שקיימים הורס עבור הטיפוס

הטמפלייטי `T`.

4. בהורס של `Queue` אנו מניחות שקיים הורס דיפולטיבי עבור הטיפוס

הטמפלייטי `T`, מכיוון שאנו קוראות ל- `popFront`.

5. בפונקציה `pushback` אנו מניחות שקיים בנאי דיפולטיבי (חסר

ארגומנטים) ואופרטור השמה עבור הטיפוס הטמפלייטי `T`.

6. בפונקציות `filter`, `transform` אנו מניחות כי לטיפוסים הטמפלייטים

שהם פונקטורים (`Function objects`) שמתקבלים כארגומנטים

לפונקציות יש אופרטור סוגריים שמגדיר את פעולתן.

ג. בזמן הקומפילציה, הקומפיילר צריך לקבל גם את המימוש של כל הפונקציות

שמשמשות בטמפלייט כדי לבצע את ההשמה של הטיפוס שבו מתבצע

השימוש. השגיאה תתקבל **בשלב הקישור (Linking)** כי הקומפיילר לא יכיר את הטיפוס `T` בקובץ ה-`cpp`. בעצם כשהקומפיילר מנסה ליצור את הפונקציות הממומשות עם `template`, כך שיהיו עבור טיפוס ספציפי במקום `T`, הוא לא יצליח ליצור את הקוד של ה `class` של התור כי המימוש לא נמצא בקובץ ה `h`. זאת כי אם המימוש עם `template` היה בקובץ `cpp`, בעצם בקובץ `h` יהיה רק את ההכרזות של הפונקציות בקלאס ולא מימושן, ואז הקומפיילר לא יוכל ליצור את המימוש של הפונקציות ונקבל שגיאה.

ד. הסטודנטית צריכה להגדיר פונקטור מתאים שמקבל `int`, להעמיס עליו אופרטור סוגריים כך שבזמן הריצה יהיה ניתן להגדיר את שדה החלוקה שלו לערך שיהיה ידוע בזמן ריצת התוכנית ויצטרך להחזיר ערך `bool`. בפועל האופרטור `()` בודק חלוקה של מספרים שברצונה למיין בערך הפנימי הזה שנקבל ואז כשהערך ידוע, ייווצר מופע חדש של ה `functor` עם הערך, ואז הפונקציה תשלח ל `filter`. בצורה זו תוכל לסנן תור של מספרים שלמים, כך שיישארו בתור רק מספרים המתחלקים במספר כלשהו שאינו ידוע בזמן קומפילציה.