

חלק יבש - תרגיל 3

מגישים: מתן לוין (207010679) איתמר פרינטי (207387226)

תאריך הגשה: - 01/06/2022

שאלה א

עבור השאלה: למה לא ניתן להשתמש רק ב- ConstIterator ללא Iterator רגיל:

לא ניתן להשתמש רק ב-ConstIterator עם פעולות const משום שפעולות אלו יחזירו לנו איברים קבועים, וכך לא יהיה ניתן להשתמש באיטרטור עבור תור שאינו קבוע כי לא יהיה ניתן לשנות את ערכי האיברים בו כשנרוץ עליהם.

עבור השאלה: מדוע לא ניתן להגדיר את כל הפעולות של ה-Const Iterator כ-Const:

כאשר עובדים על תור קבוע (const Queue), הגדרת פעולות האיטרטור כ-const יבטיחו שלא יהיה שינוי לערך האיטרטור. באיטרטור אנחנו שומרים מצביע לתור, כלומר, פעולה שמוגדרת כ-const עדיין יכולה לשנות את ערכי התור כל עוד איננה משנה את המצביע לתור. לכן לא ניתן להסתפר בהגדרת הפעולות כ-const.

שאלה ב

- isEmpty: אנחנו משווים את ערכי הפוינטר (rear, front) ל-nullptr. תחת ההנחה שקיים אופרטור השוואה (==) בטיפוס הטמפלייטי.
- PushBack: אנחנו מניחים שקיים אופרטור השמה (=) בטיפוס הטמפלייטי.

פונקציות נוספות המשתמשות בפונקציית pushback ומסתמכות בעקיפין על אותה הנחה:

- copy constructor
- אופרטור השמה (=)
- filter
- PopFront: אנחנו מניחים שקיים אופרטור השוואה (==) ושקיים אופרטור השמה (=) בטיפוס הטמפלייטי.

פונקציות נוספות המשתמשות בפונקציית pushback ומסתמכות בעקיפין על אותה הנחה:

- emptyQueue

תחת מחלקת האיטרטור:

- אופרטור !=: אנחנו מניחים שקיים אופרטור שונה (!=) גם בטיפוס הטמפלייטי.

שאלה ג

השגיאה שמתקבלת: Error: undefined reference.

במילים פשוטות, לא ניתן לגשת למתודה של Queue מהטיפוס הרצוי.

זאת, משום שבעת שימוש בטמפלייט, הקומפיילר יוצר את המתודות של המחלקה עבור כל אחד מהטיפוסים הנדרשים.

העתקים אלו נוצרים בשלב הקומפילציה, אך משום שהמימוש נמצא בקבצי הקק, הקומפיילר לא ידע כיצד לממש אותם לפי הטמפלייט הרצוי ולא ייצור את ההעתקים הנדרשים. שכן הקשר והחיבור לקבצי הקק קורה מאוחר יותר בשלב linker - שלב מאוחר משלב הקומפילציה.

השגיאה תקפוץ בשלב Linking.

שאלה ד

על מנת לאפשר את מימוש פונקציית ה-filter עבור ערך n המוזן בזמן הריצה על ידי המשתמש וממומש בפונקציית filter אנחנו מציעים לסטודנטית להגדיר מחלקה נוספת DividesBy, כפי שראינו בהרצאה. בתוכה מוגדר אופרטור () ומוגדר שדה המחזיק את ערך המחלק הרצוי. כך תוכל להזין את ערך החלוקה בזמן הריצה על ידי הקונסטרקטור של המחלקה לפי הערך הרצוי ולהשתמש באופרטור () כדי לבדוק את החלוקה.

```
class DividesBy {  
    int n;  
public:  
    DividesBy(int n) : n(n) {}  
    bool operator()(int number) const {  
        return number % n == 0;  
    }  
};
```