

### מת"מ חלק יבש – תרגיל בית 3

**שאלה א:** על מנת לאפשר איטרציה על תור קבוע (const) הממשק מאפשר לנו שימוש ב- ConstIterator בנוסף לאיטרטור הרגיל של התור. מדוע לא ניתן להסתפק בלהגדיר את פעולות האיטרטור הרגיל כ-const?

**תשובה:**

שני סוגי האיטרטורים נועדו ע"מ לספק פונקציונליות שונה:

- הקונסט איטרטור נועד ע"מ לעבור על מיכלים שהם קונסטים או באופן דומה להחמיר על מיכל שהוא לא קונסט ולא לתת למשתמש לשנות אותו.
- האיטרטור הרגיל נועד לעבור על אובייקטי המיכל עם האופציה לשנות את תוכנם.

מכיוון שאנו יצרנו תור גנרי, עלינו לממש את כלל הפונקציונליות של התור. כשהמשתמש של המחלקה Queue ירצה לעשות dereference ל-iterator מסוים על מנת לקבל את המידע השמור בתוכו הוא יקבל בחזרה T&.

גם איטרטורים שיעברו על תורים שהם קונסט וגם רגילים ע"מ לשנות את תוכן איברי התור לבקשת המשתמש.

כדוגמה מהתרגיל:

הפונקציה filter לא משנה את איברי התור ולכן נרצה לעבור על האיברים של התור עם איטרטור const.

לעומת זאת בפונקציה transform אנו רצים בלולאה על כל איברי התור בעזרת איטרטור ומשנים את איברי התור אחד אחד. אילו היינו מממשים אך ורק איטרטור const היינו מאבדים מהפונקציונליות של המיכל שלנו ולא היינו יכולים לממש בפשטות את הפונקציה אלא היינו צריכים לשכפל קוד. אחרת, אנחנו מכניסים לtransform ארגומנט מהצורה const T& ומקבלים שגיאה כי זה אומר ש-transform מקבלת const T& ואמורה לשנות אותו, כלומר השמה מ-const ל-non-const, שזו שגיאה.

**שאלה ב:** באילו מהפונקציות בממשק התור קיימות הנחות על הטיפוס הטמפלייטי? עבור כל אחת מהפונקציות הללו פרטו את הנחות.

**פתרון:**

- א. בבנאי של Node שנמצא בתוך התור יש הנחה כי קיים בנאי העתקה לT שמקבל רק ארגומנט אחד שהוא T&.
- ב. בפונקציה getReferenceToltem אנחנו מניחים שיש אופרטור השמה כי אנחנו מחזירים את מה שהפוינטר שבתוך האיטרטור מצביע עליו (כלומר item) והופכים אותו לT&, כלומר עושים השמה לT& חדש ומחזירים אותו.
- ג. בDestructor של Node יש הנחה כי קיים Destructor לitem ושלא עשו לו delete=.

חוץ מזה אין לנו עוד הנחות על הטיפוס הטמפלייטי מכיוון שעבדנו עם מצביעים ורפרנסים אליו ולא עם הטיפוס עצמו.

**שאלה ג:** סטודנט בקורס מבוא לתכנות מערכות שכח מהאזהרות שקיבל בתרגול ומימש את המחלקה Queue בקובץ cpp במקום בקובץ h. מהי השגיאה שיקבל כאשר ינסה לקמפל את התרגיל ובאיזה משלבי הקומפילציה היא מתרחשת?

**תשובה:**

מבוא:

מכיוון שהמחלקה Queue הינה מחלקה גנרית, ע"מ לממש אותה אנו נדרשים ב-templates. כפי שלמדנו 'templates צריכות להיות חשופות לקומפיליר כאשר אנו מנסים לבצע אינסטנטציה בכל זמן נתון (למעט שיטות עוקפות שמגבילות את היעילות הגנרית).

אילו היינו ממשים את Queue בקובץ cpp נפרד היינו מסתירים מהקומפיליר את ההגדרות של ה-template שצריכות לעבור אינסטנטציה והיינו מקבלים הודעת שגיאה בשלב ה-linking בסגנון של: 'to –FUNC reference undefined'.

פירוט: זאת, מכיוון שטמפלייט הוא כמו "מתכון" של הקומפיליר שמסביר לו איך לכתוב את המימוש של מחלקה מסוג מסוים.

מבחינת הקומפיליר, המחלקה Queue והפונקציות שלה לא קיימות בפני עצמן. הוא כותב את המימוש שלה לפי דרישת המשתמש, כלומר כשהמשתמש רוצה ליצור Queue<int> אז הקומפיליר מעתיק את המחלקה שכתבנו לו עם הטמפלייט ומחליף כל מופע של T ב-int.

ואז, לאחר שהקומפיליר עשה "העתק הדבק" והחליף את כל ה-T ב-int (לדוגמה, או בסוג האיבר שהמשתמש של המחלקה/פונקציה הגדיר) יש לו קטע קוד חדש עם ההכרזה והמימוש של המחלקה Queue<int>.

לכן מחלקה של Queue<int> ומחלקה של Queue<double> הינן שתי מחלקות שונות לגמרי כי בסופו של דבר לאחר החלפת ה-T יהיה מדובר בקטעי קוד נפרדים, כלומר ממש מחלקות אחרות.

כשהסטודנט מנסה לקמפל, הקומפיליר מקמפל כל קובץ cpp באופן עצמאי, אחד אחד. בבואו לקמפל את קובץ Queue.cpp, הוא יקבל מחלקה טמפלייטית (שכאמור היא מתכון לאיך לכתוב מימוש של מחלקה ולא מחלקה בפני עצמה) והוא לא יוכל לעשות העתק הדבק לתוך התבנית ולהחליף את ה-T כי הוא מקמפל רק את קובץ Queue.cpp, שבה אין שום הגדרה על מה צריך להדביק במקום T.

(וגם אם הקומפיליר היה מנסה לקמפל את קובץ הקסס בכל זאת ולהחליף את ה-T בכל סוג איבר שהוא מכיר, אז דבר ראשון הוא היה צורך המון זיכרון ודבר שני הוא לא מכיר את כל סוגי האיברים T שיכולים להיות כי המשתמש יכול לייצר מחלקה חדשה שהוא לא מכיר).

לסיכום, הלינקר יחזיר שגיאה על הפונקציות שמומשו בcpp כי T לא מוגדר עדיין.

המשתמש של המחלקה Queue הגדיר את T בmain.cpp (לדוגמה) וכאמור אנו מקמפלים כל קובץ בנפרד.

**שאלה ד:** סטודנטית בקורס מבוא לתכנות מערכות סיימה לפתור את תרגיל בית 3, והחליטה להשתמש במימוש התור מהתרגיל לפרויקט צד שהיא מפתחת בשעות הפנאי. במימוש פרויקט הצד הסטודנטית

נדרשה לסנן תור של מספרים שלמים, כך שיישארו בתור רק מספרים המתחלקים במספר כלשהו שאינו ידוע בזמן קומפילציה אלא רק בזמן ריצה. הסבירו כיצד ניתן לממש את הפונקציונליות הדרושה בעזרת הפונקציה filter.

### **פתרון:**

הפונקציה filter מקבלת תור ומצביע לפונקציה (כלומר תנאי, נקרא לו פרדיקט) ופולטת תור חדש המכיל איברים אשר עבורם התנאי (הפרדיקט) קיבל ערך true בלבד.

ע"מ לממש את הפונקציונליות שהסטודנטית רוצה, עליה לנהל פרדיקט שבזמן הריצה משתנה לכך שיוציא ערך true אך ורק לחלוקת המספר הרצוי.

עתה על הסטודנטית להשתמש בפרדיקט על הפונקציה filter, וזאת תפלוט את התור הרצוי.

מגשים:

איתי ברקוביץ 316088970

איל שטיין 208622142