

סעיף א'

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

איטרטור קבוע (Constlterator) מונע מאיתנו את האופציה לשנות את הערך שהאיטרטור מצביע עליו כרגע (רפרנס קבוע), כלומר, לא ניתן לשנות את התור בעודנו רצים עליו. לעומת זאת, איטרטור רגיל מאפשר לנו לשנות את הערך, בכך שהערך שהוא מצביע עליו הוא רפרנס לא קבוע. ברגע שאנחנו מגבילים את עצמנו לסוג אחד של איטרטור (לדוגמה רק const), לא תהיה לנו את האפשרות לשנות את ערכי התור, או להפך, באם נגדיר את כל הפונקציות של האיטרטור הרגיל בתור רגילות, אנחנו מקנים למשתמש את האופציה לערוך את ערכי התור באמצעות האיטרטור.

סעיף ב'

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

שם הפונקציה	הנחות
Queue<T> &operator=(const Queue<T> &toCopy)	existence of a constructor with no argument- T() existence of an = operator in T
Queue()	existence of a constructor with no argument- T() existence of an = operator(used in inner function in Queue::Node)
Queue(const Queue<T> &toCopy)	existence of a constructor with no argument- T() existence of an = operator(used in inner function in Queue::Node and in pushBack)
void pushBack(const T &nodeToAdd)	existence of an = operator (used in inner function in Queue::Node)
Queue<T> filter(const Queue<T> &queue, FileterFlag filterFlag)	calls Queue() so assumes all of Queue() assumptions hold: existence of a constructor with no argument- T() existence of an = operator
void transform(Queue<T> &queueToTransform, TransformOperation &transformOperation)	calls pushback so all assumptions hold: existence of an = operator

סעיף 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?

במקרה כזה תתרחש שגיאת קישור (לא יודע איך אומרים בעברית- Linker Error)

הנ"ל מתרחש בשלב Linking כאשר המחשב מנסה למצוא מימוש לפונקציות בקובץ ה-h המתאים ולא מוצא.

סעיף ד'

סטודנטית בקורס מבוא לתכנות מערכות סיימה לפתור את תרגיל בית 3, והחליטה להשתמש במימוש התור מהתרגיל לפרוייקט צד שהיא מפתחת בשעות הפנאי. במימוש פרוייקט הצד הסטודנטית נדרשה לסנן תור של מספרים שלמים, כך שישארו בתור רק מספרים המתחלקים במספר כלשהו שאינו ידוע בזמן קומפילציה אלא רק בזמן ריצה. הסבירו כיצד ניתן לממש את הפונקציונליות הדרושה בעזרת הפונקציה filter.

היינו ממליצים לסטודנטית לחזור על החומר של הרצאה 6, ובפרט על החלק בו מלמדים על שימוש ב-Function Objects.

ניתן ליצור Function Object אשר יועבר בתור FilterFlag לפונקציה filter שהגדרנו. הנ"ל מתאים למקרה המדובר היות וניתן לעשות overloading ל() כך שיתקבל פרמטר ובאופן כזה, להזין את המספר בו רוצים שאיברי התור יתחלקו, האובייקט ייוצר עם התנאי המתאים, ובאמצעות הפונקציה filter האיברים הרלוונטיים יסוננו.