

שאלה 1 (35 נק')

בקובץ Tree.h נתונה לכם המחלקה Tree. לכל אובייקט Tree יש id וילדים כ `vector<Tree*>`. בתוך המחלקה Tree ישנה המחלקה iterator – זהו האיטרטור הייעודי של העץ.

עליכם לממש בקובץ Tree.cpp את המתודות הבאות.

עבור Tree:

- האופרטור `+=` . בהינתן id האופרטור יוסיף תת-עץ חדש עם ה id הזה.

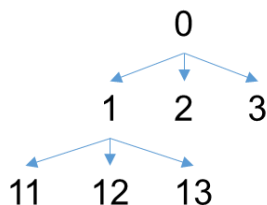
עבור iterator:

- בנאי שמקבל `Tree*`
- אופרטור `*` שמחזיר את העץ עליו מצביע האיטרטור
- המתודה `up` שמעלה את האיטרטור במעלה העץ steps פעמים
 - אם מס' הצעדים עובר את שורש העץ יש להישאר בשורש
- המתודה `down` שמורידה את האיטרטור לבן הראשון אם הוא קיים,
 - אחרת, האיטרטור יישאר במקום
- האופרטור `++` (שמאלי. למשל `++it`) עובר לבן הבא של האבא אם הוא קיים (ראו דוגמה בהמשך) אחרת, האיטרטור יישאר במקום
- האופרטור `--` (שמאלי. למשל `--it`) עובר לבן הקודם של האבא אם הוא קיים (ראו דוגמה בהמשך) אחרת, האיטרטור יישאר במקום

בקובץ Tree.h מותר לכם לערוך אך ורק את ה data members של המחלקה iterator (האזור שבתוך הכוכביות). אין לשנות דבר בקוד מסביב ובפרט אין לשנות חתימות של מחלקות ומתודות.

טיפ: ככל הנראה תצטרכו לעשות שימוש במחסנית.

מוד אימון:



מייצר את העץ הבא, מטייל על קודקודי העץ באמצעות האיטרטור ובודק שהוא אכן מצביע על האיבר הנכון בעץ לאחר הפעולות השונות.

לדוגמה, כאשר האיטרטור מצביע על "1" ומפעילים `++` הוא יצביע על "2" פעולה נוספת של `++` תגרום לו להצביע על "3" ואילו פעולה נוספת של `++` תשאיר את האיטרטור באותו המקום. הפעלה של אופרטור `--` תחזיר את האיטרטור ל "2". הפעלה של `up(1)` תחזיר אותו ל "0" ואילו 2 הפעלות של `down` יגרמו לאיטרטור להצביע על "11".

מוד הגשה:

אופי הבדיקה במוד הגשה דומה, אך הבדיקה תכלול עץ שנראה אחרת וסדר שונה של פעולות. יש לכתוב קוד שעונה להגדרות לעיל ולא רק לדוגמה הפרטית של מוד האימון.

שאלה זו תיבדק באופן אוטומטי בלבד. לכן, חובה שהקוד שלכם יתקמפל ללא אזהרות או שגיאות ולא יכלול שגיאות ריצה כלשהן. אחרת השאלה לא תוכל להיבדק אוטומטית וכל הנק' ירדו.

שאלה 2 (30 נק')

בקובץ covid19.h נתונות המחלקות Employee ו Manager (שאר הקוד שייך לשאלה 3).

אין לערוך קובץ זה ובפרט הוא אינו להגשה.

ה Manager הוא סוג של Employee בעצמו אך מחזיק מערך דינאמי של עובדים מסוג Employee**. כך שכל תא Employee*[i] יכול להיות בתורו Employee או Manager (שגם לו יש עובדים וכו').

לכל עובד ניתן לבדוק ע"י המתודה test האם הוא לצערנו חיובי לקורונה.

עליכם לממש בקובץ Q2.h את הפונקציה deepPositiveCount אשר בהינתן Employee* היא תחפש באופן עמוק ותחזיר כ int את מספר העובדים החיוביים לקורונה.

לדוגמה אם הזנו עובד שהוא מסוג manager ולו 5 עובדים ש 2 מהם מסוג Manager ולכל אחד מאלה 2 עובדים אז בסך הכל יש 10 עובדים שצריך לבדוק.

במוד האימון נבדקת היררכיית עובדים קבועה ופשוטה, אולם במוד ההגשה נבדקת היררכית עובדים בגודל אקראי והדבקה אקראית בקורונה.

ניקוד: ככל שהתשובה רחוקה יותר מהתשובה הנכונה כך ירדו יותר נק' (ומקסימום 30 נקודות).

שאלה זו גם תיבדק באופן ידני, אולם הקנס על שגיאת קומפילציה או ריצה הוא 10 נק'.

שאלה 3 (35 נק')

בסוף הקובץ covid19.h נתונה המחלקה Covid19. עליכם לממש את המחלקות הבאות בקובץ Q3.h:

- BrCovid20 סוג של Covid19 שמוסיף מידע של double ph.
 - בבנאי עליו לקבל int id ו double ph לאתחול.
- SaCovid20 סוג של Covid19 שמוסיף מידע של char m.
 - בבנאי עליו לקבל int id ו char m לאתחול.
- IlCovid21 סוג של BrCovid20 ו SaCovid20 שמוסיף את המידע int x. תשימו לב שרכיב ה Covid19 באובייקט ה IlCovid21 צריך להופיע רק פעם אחת
 - בבנאי עליו לקבל רק int x לאתחול
 - ה id צריך להיות 972
 - ה ph צריך להיות 2.0
 - ה m צריך להיות 's'

הבדיקה במוד האימון בודקת את ההגדרות לעיל, מוד ההגשה זהה למוד האימון.

שאלה זו גם תיבדק באופן ידני, אולם הקנס על שגיאת קומפילציה או ריצה הוא 10 נק'.

הגשה

הגשה: יש להגיש את Q2.h, Q3.h, Tree.cpp, Tree.h בלבד ל <https://cktest.cs.colman.ac.il/> קורס OOP תיבה מועד א. עד לתום הבחינה עליכם להגיש את הקוד למוד הגשה ואז למוד הגשה סופית ולקבל מס' אסמכתא בן 4 ספרות.

בהצלחה!