

מטלה - עץ בינארי

מטרת המטלה היא לאמן אתכם במחלקות, בניה, פירוק, ניהול זיכרון, מבני-נתונים ובדיקות-יחידה.

המשימה: כיתבו מחלקה בשם `Tree`, המייצגת עץ-חיפוש בינארי לשמירת מספרים בסדר עולה, כפי שלמדתם בקורס מבני נתונים (לא חייב להיות עץ מאוזן).

פעולות

המחלקה צריכה לאפשר את הפעולות הבאות:

- `insert` - מקבלת כקלט מספר `i` ומכניסה את `i` לעץ במקום המתאים.
- `remove` - מקבלת כקלט מספר `i` ומסירה את `i` מהעץ.
- `size` - מחזירה את מספר המספרים בעץ.
- `contains` - מקבלת כקלט מספר `i` ומחזירה "אמת" אם `i` נמצא בעץ.
- `root` - מחזירה את המספר הנמצא בשורש העץ.
- `parent` - מקבלת כקלט מספר `i` ומחזירה את המספר שנמצא מעל `i` בעץ.
- `left` - מקבלת כקלט מספר `i` ומחזירה את המספר שהוא הבן השמאלי של `i` בעץ.
- `right` - מקבלת כקלט מספר `i` ומחזירה את המספר שהוא הבן הימני של `i` בעץ.
- `print` - הדפסת מבנה העץ, לצורך הצגה וניפוי שגיאות. פורמט ההדפסה - לשיקולכם.

יש לזרוק חריגות במקרים הבאים:

- `insert` - כשמנסים להכניס מספר שכבר נמצא בעץ.
- `remove` - כשמנסים למחוק מספר שלא נמצא בעץ.
- בשאר הפקודות - לזרוק חריגה בכל מצב שנראה לכם חריג - תפעילו שיקול דעת.

קבצים

בנוסף לקבצים של הפתרון, יש להכין קובץ בשם **`TreeTest.cpp`** הכולל בדיקות-יחידה מפורטות.

מצורפים לתרגיל זה הקבצים:

- `TreeDemo.cpp` - תוכנית ראשית לדוגמה.
- `TreeTest.cpp` - תוכנית ראשית הכוללת בדיקות-יחידה לדוגמה; המשימה שלכם היא להרחיב את הקובץ הזה כך שיכלול בדיקות-יחידה מפורטות למצבים שונים.

שתי התוכניות הראשיות צריכות להתקמפל עם הקבצים שתכתבו.

הגשה

צרו מאגר (repository) חדש בגיטהאב והעלו לשם את הקבצים בתיקה הראשית.
הגישו בטופס-ההגשה קישור-שיבוט למאגר - הקישור שרואים כשלוחצים על הכפתור clone בגיטהאב.

בדיקה אוטומטית

אנחנו נבצע את הפקודות הבאות ממחשב עם לינוקס:

1. `git clone <הקישור שלכם>`

2. נעתיק לתוך התיקה שלכם תוכנית `TreeTest.cpp` משלנו, הכוללת בדיקות אוטומטיות נוספות וחישוב ציון.

3. `clang++-5.0 -std=c++17 TreeTest.cpp Tree.cpp`

4. `./a.out`

אתם יכולים לפתור את התרגיל בכל סביבת-פיתוח שאתם רוצים, אבל לפני ההגשה, וודאו שהפקודות האלו רצות בלי שגיאות על מחשב לינוקס אחר כלשהו. אם אין לכם `clang` תשתמשו ב-`g++` במקום.

שלבי ההגשה

יש להגיש את הפתרון בשני שלבים.

שלב א - שבוע לאחר פירסום המטלה, יש להגיש פתרון המקבל ציון של 50 לפחות בבדיקות האוטומטיות שלנו. זה אומר שאתם צריכים לממש נכון חלק מהפונקציות (לשיקולכם), ובשאר הפונקציות לשים `stub` - קוד מינימלי שיאפשר לקוד שלכם להתקמפל.

במקביל, יש להגיש קובץ `TreeTest.cpp` הכולל בדיקות-יחידה מקיפות ומפורטות משלכם. בשלב זה כמובן חלק גדול מהבדיקות שלכם לא יעברו.

שלב ב - שבועיים לאחר פירסום המטלה, יש להגיש פתרון מלא, המקבל 100 בבדיקות האוטומטיות שלנו.

בדיקה בתירגולים

בתירגולים תתבקשו להציג תוכנית-דוגמה היוצרת עץ בינארי ומדפיסה אותו (בעזרת `print` שכתבתם). בנוסף תתבקשו להסביר על הקוד שלכם ולהראות שהוא מקיים כללים בסיסיים של הנדסת תוכנה: חלוקה לקבצים, תיעוד, שמות משמעותיים למשתנים, בדיקות תקינות ומניעת דליפת-זיכרון.

דגשים

- יש לחזור על החומר של ההרצאות לפני שמתחילים לכתוב, ולהשתמש בו לפי הצורך.
- מותר להשתמש בתכונות מתקדמות של שפת C++ גם אם עדיין לא נלמדו בהרצאות.
- אין להעתיק תרגילים שלמים מסטודנטים אחרים. מותר להיעזר בקטעי קוד מהאינטרנט, אולם **יש לציין בבירור את המקור**, לוודא שהקוד עובד, ולוודא שאתם מבינים למה הוא עובד.