בעדכון האחרון למערכת הבחירות הממוחשבת נוסיף לכל **סבב בחירות** בדיקת שגיאות ושימוש בחריגות, יחד עם עדכון הקוד לשימוש בתבניות ומבני הנתונים והאלגוריתמים הסטנדרטיים של ++C.

תבניות

בחלק זה נעדכן את הקוד כך שיעשה שימוש בתבניות.

עליכם לזהות אילו מחלקות מייצגות או מכילות מבני נתונים או אלגוריתמים כלליים, ולעדכן אותן לשימוש בתבניות. ניתן לייצר מחלקות תבנית חדשות ולהשתמש בהן. שימו לב: מחלקות אלו אינן צריכות להיות ספציפיות למערכת הבחירות ועיצוב המחלקות הכולל ייבדק. לדוגמא, בהינתן מערך דינמי יש ליצור מחלקת תבנית של מערך דינמי כללי, ולא ספציפי לסבב בחירות.

בחלק זה עליכם לממש לפחות אלגוריתם תבנית אחד. בנוסף, עליכם להשלים את DynamicArray מהרצאה 11 כך שתתמוך באיטרטורים ואלגוריתמים של STL באופן מלא ולהשתמש בה, או לממש מבנה נתונים אחר משלכם (עם שימוש בתבנית וכולל איטרטור והתאמה ל-STL).

כלומר, יש לממש: iterator, const_iterator, begin/end, cbegin/cend, rbegin/rend, insert, erase x2.

חריגות

בחלק זה נעדכן את המערכת לזיהוי וטיפול בשגיאות.

עליכם להמיר את כל בדיקות השגיאות לשימוש בחריגות, כך שכל שגיאה אפשרית תיבדק במקום הנכון ותיתפס (והטיפול בה יתבצע) במקום הראוי. בכל הקוד כולו אין לעשות שימוש בערכי החזר, אלא בחריגות בלבד. כמו-כן יש להוסיף בדיקות תקינות לכל הבנאים וזריקת חריגות בהתאם, היכן שיש צורך ומתאפשרת שגיאה בארגומנטים המתקבלים בבנאי.

יש לדאוג לשחרר זיכרון באופן נכון בכל מקרה של חריגה, ולבדוק ולטפל נכון גם בשגיאות של הקצאת זיכרון (וכמובן גם גישה לקבצים וכו'). כל חריגה צריכה להיתפס באיזשהו שלב – כך שלא תתאפשר חריגה שתקריס את התוכנית.

הוסיפו למערכת את הבדיקות הבאות:

- כל הקצאת זיכרון מצליחה.
- תעודת הזהות של אזרח מורכבת מ-9 ספרות.
- שנת הלידה של אזרח תקינה כך שגילו מעל 18 (בהתאם למועד של סבב הבחירות הנוכחי). יש לבצע השוואה לפי שנת הלידה לשנה של סבב הבחירות, אין צורך להתייחס ליום וחודש.
- בדיקות תקינות נתונות של לוגיקת המערכת (לדוגמא, כל אזרח מצביע פעם אחת בלבד).
- יש לוודא שקלט של מספרים לאורך התוכנית (לדוגמא, תאריך סבב הבחירות) הוא תקין. עבור התאריך ניתן לקבל כל שנה שהיא מספר חיובי, וניתן להניח שבחודש פברואר תמיד יש 28 יום.
- בכל בנאי (בו יש מגבלה על ערכי הארגומנטים) יש לוודא שניתן ערך תקין לכל ארגומנט.
- בחישוב תוצאות הבחירות נזרוק חריגה בכל מקרה בו לא ניתן לחשב את התוצאות.
- בדיקות נוספות ושימוש בחריגות בהתאם לעיצוב המחלקות שלכם וללוגיקה הנתונה.

STL

בחלק זה נעדכן את הקוד כך שיעשה שימוש במחרוזות ובמבני הנתונים והאלגוריתמים הסטנדרטיים.

יש להמיר בקוד כל שימוש במחרוזות (*char) ומבני נתונים (פרט לאחד שמימשתם בסעיף של תבניות) לטיפוסים המתאימים מתוך STL, היכן שאפשר. כמו-כן יש לעדכן כל אלגוריתם כללי (פרט לאחד שמימשתם בסעיף של תבניות) לאלגוריתם המתאים מתוך STL, היכן שאפשר.

שימו לב לכל קוד שהופך למיותר בעקבות שינויים אלו ועליכם להוריד.

הנחיות

עליכם לבצע תיקונים בעיצוב ומימוש הקוד כמו בהנחיות התרגיל הקודם, בהתאם למשוב של תרגיל 2. קריטריונים אלו יבדקו שוב בתרגיל 3.

שימו לב לעיצוב נכון של המחלקות. כרגיל, עליכם לעצב את המחלקות באופן מלא וכקופסה שחורה – לכל מחלקה תחום אחריות מוגדר משלה, ממשק פומבי ברור, ועיצוב המחלקות צריך לאפשר שימוש בהן באופן מלא גם במערכות אחרות, ללא תלות חזקה בתפריט או במערכת הספציפית (אלא רק לפי הגדרת הישות שניתנה).

שימו לב למועד ההגשה של התרגיל (אשר מכיל כבר דחיה של שבוע) ותחילת תקופת המבחנים. עליכם לנהל את הזמן נכון כדי להגיש את התרגיל לפני מועד ההגשה. לא יתקבלו דחיות להגשה!

עבור פתרון התרגיל ניתן להשתמש בכל הנושאים שנלמדו בכיתה. יש לבדוק שגיאות של הקצאת זיכרון ולהימנע משימוש בערכי החזר עבור שגיאות, כפי שפורט בסעיף של חריגות. עבור מימוש מבנה הנתונים יש להשתמש בחריגות ולבדוק שגיאות רק היכן שנדרש כפי שנלמד (לדוגמא, אופרטור [] לא בודק את תקינות האינדקס אך פעולת at כן בודקת).

מעבר לכך, הנחיות ההגשה וההנחיות הכלליות זהות לתרגילים הקודמים.

בהצלחה!