

מכללת הדסה, החוג למדעי המחשב
תכנות מונחה עצמים ופיתוח משחקים
סמסטר ב', תשפ"ג

תרגיל 3

תאריך אחרון להגשה:

הנביאים - יום א', 07/05/23, ט"ז אייר ה'תשפ"ג, בשעה 23:59
שטראוס-גברים - יום א', 07/05/23, ט"ז אייר ה'תשפ"ג, בשעה 23:59
שטראוס-נשים - יום א', 07/05/23, ט"ז אייר ה'תשפ"ג, בשעה 23:59

מטרת התרגיל:

תרגול חריגות (exceptions).

תיאור כללי:

תרגיל זה יתבסס על הפתרון של תרגיל 1 (שתקבלו מאיתנו), כאשר אתם נדרשים לשנות ו"לשדרג" את הפתרון הזה באמצעות טיפול בחריגות, ולהפוך אותו לעמיד בפני טעויות של משתמשים לא זהירים. השינוי יכלול שיפור של מימוש הפקודות הקיימות כדי לזרוק שגיאות במקרה של קלט שגוי. כמו כן נוסיף אפשרות לקריאת פקודות מקובץ נתון וביצוע שלהן. נשתמש בתרגיל זה רק בחריגות של השפה (std::exception) והמחלקות היורשות ממנה (או, במקרה הצורך, ניצור מחלקות שיורשות מהן. גם בתרגיל זה נשתמש במסוף (terminal) בלבד.

פירוט הדרישות:

- כל השגיאות שנזכיר כאן ונטפל בהן, **חייבות** להשתמש בחריגות (exceptions) כדי להודיע על הבעיה. בדרך כלל נעדיף שהחריגה תיזרק מתוך המתודה או המחלקה המממשת את הפונקציה ושהיא תיתפס ותטופל באחת הרמות מעליה, ברמה הכי גבוהה האפשרית.
- כשהמשתמש מכניס פקודה, יש לוודא שהמשתמש לא הכניס פקודה שגויה או לא קיימת. אם אין פקודה כזו, התכנית תתריע עם הודעת שגיאה מתאימה ותציג את רשימת הפונקציות מחדש.

3. עבור פקודה תקינה (הפקודות הקיימות מתרגיל 1 והפקודות הנוספות בהמשך), יש לוודא שהמשתמש לא הזין נתון לא תקין (כמו אותיות או מספרים שליליים כשמצפים למספר חיובי, מספר פונקציה לא תקין, וכו'). כך גם אם הקלט לא מכיל מספיק ארגומנטים לפקודה או שיש יותר מידי ארגומנטים לפקודה הנוכחית (רלוונטי בקריאה מקובץ כפי שמוסבר בהמשך, אבל אולי גם בקריאה מהמקלדת, ראו בהערות בהמשך). בכל המקרים הללו התכנית תתריע עם הודעת שגיאה מתאימה ותציג את רשימת הפונקציות מחדש.
4. אורך מחרוזת, בין אם התקבלה כקלט או שנוצרה כתוצאה מחישוב ערך פונקציה על מחרוזת - לא יעלה על 20 תווים. גם כאן, במקרה שהאורך חורג מהמותר פעולת החישוב לא תתבצע, התכנית תתריע עם הודעת שגיאה מתאימה ותציג את רשימת הפונקציות מחדש.
5. נוסף לרשימת הפקודות פקודת read שתקבל כפרמטר נתיב לקובץ עם פקודות, ושדע לקרוא את הקובץ. הקובץ יכול להכיל את כל סוגי הפקודות, למעט הפקודה read עצמה שאין חובה לתמוך בה (ציינו ב-Readme האם היא נתמכת בקריאה מקובץ או לא).
6. אם פקודת read לא הצליחה לפתוח את הקובץ, היא תציג הודעת שגיאה. אם היא הצליחה לפתוח את הקובץ הנתון, היא תקרא פקודה אחרי פקודה ותבצע אותן, כל שורה היא פקודה נפרדת (כך שיתכן שיש יותר מידי או פחות מידי ארגומנטים עבור הפקודה המסוימת). אם ביצוע אחת השורות נכשל (מאחת הסיבות המוזכרות בסעיפים האחרים), נפסיק את פעולת הקריאה, נציג את השורה השגויה בתוספת הודעת שגיאה מתאימה, ונציע למשתמש לבחור האם להמשיך לקרוא ולבצע את המשך הקובץ, או להפסיק לקרוא ולחזור למצב הקודם.
7. כשנסיים את המעבר על הקובץ, בין אם בהצלחה ובין אם בגלל שגיאה (וכמוסבר בסעיף הקודם), נחזור לשלב הקודם. כלומר, אם לפני כן היינו במצב של קריאה מהמקלדת, נחזור להדפיס את רשימת הפונקציות ולקרוא פקודות מהמקלדת. אם היינו לפני כן בקריאה של קובץ אחר, ומימשנו תמיכה בפקודות read מתוך קובץ, נחזור לבצע את הקובץ הקודם.
8. בפתיחת התוכנית, לפני הצגת הפונקציות, נבקש מהמשתמש לקבוע מה מספר הפונקציות המקסימלי שניתן יהיה לאחסן בתכנית זו. נוודא שמספר הפונקציות המקסימלי לא גדול מ-100 ולא קטן מ-3. גם פה נוודא שהקלט תקין ואם הוא לא תקין (כלומר אם הקלט גדול מ-100, קטן מ-3, או שהוא כלל לא מספר, אלא אותיות וכדומה) נציג שגיאה ונשאל שוב. במהלך התכנית, נוודא שלא נוצרות יותר פונקציות ממה שנקבע מלכתחילה, כלומר אם המשתמש מנסה ליצור פונקציה חדשה (ישירות או כפקודה המגיעה מקריאת הקובץ) והמכסה התמלאה כבר, הפקודה צריכה להיכשל. כמו כן, בהדפסת רשימת הפונקציות, יש להציג את המידע על מספר הפונקציות המותר. שימו לב שהמגבלה איננה על מספר הפונקציות שנוצרות במהלך כל ריצת התוכנית, אלא על מספר הפונקציות הנוכחי, כלומר אם המשתמש מוחק פונקציות, הן כבר לא נספרות במכסת הפונקציות המותרת. כאמור, גם במהלך קריאת הקובץ צריך להתחשב במגבלת המקום ולהימנע מביצוע פקודה שתגרום להוספת פונקציה כשהרשימה מלאה (ואז נפעל כמתואר בסעיף 5, נשאל את המשתמש אם להמשיך או להפסיק את ביצוע הקובץ).

9. פקודה נוספת שתתווסף לרשימת הפקודות היא `resize`, בה יוכל המשתמש לשנות את מכסת הפונקציות כרצונו. אותן בדיקות מסעיף 7 נדרשות גם פה. בנוסף, אם יקטין המשתמש את כמות הפונקציות מהקיימות - זו איננה שגיאה (ולכן לא נדרוש שימוש בחריגה), והוא יקבל הודעה מתאימה שתציע לו ביטול של הפעולה או מחיקת הפונקציות שבסוף הרשימה עד למכסה המותרת.

הערות לתיכון התוכנית:

- ברור שעליכם לחשוב על תכנון ראוי שיתאים לקוד הקיים כדי לאפשר לו להתקמפל ולרוץ באופן תקין, אולם חשוב שלא תצטמצמו רק לשילוב הדרישות החדשות בקוד הקיים אלא גם תחשבו מחדש על מבנה הקוד, איחוד או פיצול של מתודות מסוימות, וכדומה. אולי כדאי לחשוב על מחלקות חדשות שיעטפו אזורים בקוד ויעשו את הטיפול בשגיאות קל יותר.
- שקלו את האפשרות להשתמש בחריגות המובנות של ה-`streams` למיניהם.
- כדאי לשנות את הקוד בשלבים (קטנים ככל האפשר) כך שבכל שלב אפשר לוודא שהקוד עדיין מתקפל והשגיאות מתקבלות בהצלחה כמצופה.
- כדי לעבוד על כל שורה (מהקובץ) בנפרד, זכרו את הכלים `std::getline` ו-`stringstream`.
- בידכם להחליט האם הקריאה מהמקלדת גם היא תיעשה על כל שורה בנפרד. אם כן, צריך לטפל גם פה במקרים של יותר מידי או פחות מידי ארגומנטים לפקודה. אם לא – אין כזה דבר יותר מידי או פחות מידי ארגומנטים, נקרא מהמקלדת ארגומנטים עד שיש לנו מספיק עבור הפקודה הנוכחית וארגומנטים עודפים יישארו ב-`buffer` עד הפעם הבאה שננסה לקרוא מהמקלדת. בכל מקרה, ציינו את החלטתכם בקובץ ה-`Readme`.
- שימו לב שכשמתקבל קלט לא תקין מהמקלדת, ייתכן שהמימוש שלנו גורם לקבלת הקלט להפסיק ושאר הקלט יחכה לפעם הבאה וייתכן שקודם נקרא את המידע ורק אז נוציא את השגיאה וממילא כל הקלט עבור הפקודה הנוכחית "נפסל". למשל, אם הפקודה מצפה לקבל שני מספרי פונקציות, והראשון מביניהם הוא מספר לא חוקי, ייתכן שנעצור מיד ולא נקרא בכלל את מספר הפונקציה השניה, וייתכן שנקרא את מספר הפונקציה השניה ורק אחר כך נבדוק ונגלה שהמספר הראשון איננו חוקי. שני המימושים האלה לגיטימיים, אין צורך להתאמץ במיוחד לממש כך או כך, ועדיף קוד קצר, מסודר וקריא מאשר קוד מורכב ומסובך רק כדי להשיג התנהגות מסוימת. בכל מקרה, ציינו את ההתנהגות שבחרתם בקובץ ה-`Readme`. נעיר כי בקריאה מהקובץ, אנחנו מחוייבים להתייחס לכל שורה בנפרד, ולכן בכל מקרה כל השורה "נפסלת" כשחלק מהמידע בה שגוי.

- חלק מהקלטים השגויים כבר מטופלים בקוד שתקבלו, כי אלה היו חלק מהדרישות המקוריות של תרגיל 1. למרות זאת, כעת תצטרכו להמיר את הזיהוי והטיפול הזה לשימוש בחריגות, כאמור. שקלו אם יש חלקי קוד שניתן לשפר בעקבות המעבר לשימוש בחריגות.

קובץ ה-README:

יש לכלול קובץ README שיקרא README.doc, README.docx או README.txt (ולא בשם אחר). הקובץ יכול להיכתב בעברית ובלבד שיכיל את הסעיפים הנדרשים.

קובץ זה יכיל לכל הפחות:

1. כותרת.
 2. פרטי הסטודנט: שם מלא כפי שהוא מופיע ברשימות המכללה, ת"ז.
 3. הסבר כללי של התרגיל.
 4. רשימה של הקבצים שיצרנו, עם הסבר קצר (לרוב לא יותר משורה או שתיים) לגבי תפקיד הקובץ.
 5. מבני נתונים עיקריים ותפקידיהם.
 6. אלגוריתמים הראויים לציון.
 7. באגים ידועים.
 8. הערות אחרות.
- יש לתמצת ככל שניתן אך לא לוותר על אף חלק. אם אין מה להגיד בנושא מסוים יש להשאיר את הכותרת ומתחתיו פסקה ריקה. נכתוב ב-README כל דבר שרצוי שהבודק ידע כשהוא בודק את התרגיל.

אופן ההגשה:

הקובץ להגשה: ניתן ליצור בקלות קובץ zip המותאם להגדרות ההגשה המפורטות להלן ישירות מתוך VS, כפי המוסבר תחת הכותרת "יצירת קובץ ZIP להגשה או לגיבוי" בקובץ "הנחיות לשימוש ב-Visual Studio 2022". אנא השתמשו בדרך זו (אחרי שהגדרתם כראוי את שמות הצוות ב-MY_AUTHORS: **נשים את שמות המגישים בתוך המרכאות, נקפיד להפריד בין השם הפרטי ושם המשפחה בעזרת קו תחתית ואם יש יותר ממגיש אחד, נפריד בין השמות השונים בעזרת מקף (מינוס '-')**) וכך תקבלו אוטומטית קובץ zip המותאם להוראות, בלי טעויות שיגררו אחר כך בעיות בבדיקה.

באופן כללי, הדרישה היא ליצור קובץ zip בשם exN-firstname_lastname.zip (או במקרה של הגשה בזוג – exN-firstname1_lastname1-firstname2_lastname2.zip), כשהקובץ כולל את כל קובצי הפרויקט, למעט תיקיות out ו-vs. כל הקבצים יהיו בתוך תיקייה ראשית אחת.

את הקובץ יש להעלות ל-Moodle של הקורס למשימה המתאימה. בכל מקרה, **רק אחד** מהמגישים יגיש את הקובץ ולא שניהם.

הגשה חוזרת: אם מסיבה כלשהי סטודנט מחליט להגיש הגשה חוזרת יש לוודא ששם הקובץ זהה לחלוטין לשם הקובץ המקורי. אחרת, אין הבודק אחראי לבדוק את הקובץ האחרון שיוגש.

כל שינוי ממה שמוגדר פה לגבי צורת ההגשה ומבנה ה-README עלול לגרור הורדת נקודות בציון.

מספר הערות:

1. נשים לב לשם הקובץ שאכן יכול את שמות המגישים.
 2. נשים לב לשלוח את תיקיית הפרויקט כולה, לא רק את קובצי הקוד שהוספנו. תרגיל שלא יכול את כל הקבצים הנדרשים, לא יתקבל וידרוש הגשה חוזרת (עם כללי האיחור הרגילים).
- המלצה כללית: אחרי הכנת הקובץ להגשה, נעתיק אותו לתיקייה חדשה, נחלץ את הקבצים שבתוכו ונבדוק אם ניתן לפתוח את התיקייה הזו ולקמפל את הקוד. הרבה טעויות של שכחת קבצים יכולות להימנע על ידי בדיקה כזו.

בהצלחה!