

תרגיל בית מספר 5: מטלת סיכום

נושא: מחשבון שכיחויות

דדליין: מוצ"ש, 26/06/2021, 23:59

הגשה בזוגות

בהצלחה רבה!

תיאור התרגיל

בתרגיל זה במערכת המאפשרת לספור שכיחויות של מספרים ומחרוזות בקלטים שונים. מטרת המטרת התרגיל היא להתנסות בעבודה עם אובייקטי תבנית, פקודות של מערכת הקבצים, ומנגנון החריגות המובנה ב-C++.

חלק א': מימוש מונה מוכלל

בחלק זה של התרגיל עליכם לממש את מחלקת התבנית Counter, כך ש-Counter<T> תהיה מחלקה המאפשרת ספירת מספר המופעים של ערכים מטיפוס T, בדומה למחלקה collections.Counter בפייתון. דרישות המימוש הן:

- [1] לאחר בניית האובקט, או הפעלת המתודה clear(), המונה ריק.
- [2] מתודה add, המקבלת ערך מטיפוס T ומגדילה את מספר המופעים שלו באחד. אם זה המופע הראשון של הערך, אז מספר המופעים שלו יהיו 1.
- [3] אופרטור [] למציאת מספר המופעים של ערך T. אם אין אף מופע של הערך יש להחזיר 0.
- [4] פונקציה most_common, המחזירה את הערך השכיח ביותר. אם יש כמה ערכים שמספר מופעיהם הגבוה ביותר, יש להחזיר את זה שערכו הקטן ביותר. אם המונה ריק, יש לזרוק שגיאה.
- [5] פונקציה add_from_stream המקבלת istream. הפונקציה קוראת ערכים מטיפוס T עד סוף הקלט על ידי שימוש באופרטור <<, ומעדכנת את המונה על ידי add. אם ארעה שגיאה במהלך קריאת הקלט, יש לזרוק שגיאה.
- [6] פונקציה print_to_stream המקבלת ostream, ומדפיסה אליו את הסטטיסטיקה. למשל, עבור Counter<string>, פלט אפשרי יהיו: { a:1 abc:3 }, ומשמעו שהמחרוזת a מופיעה פעם אחת, והמחרוזת abc מופיעה 3 פעמים. סדר ההדפסה לפי יחס הסדר המוגדר על T.
- [7] פונקציה print_most_common המקבלת ostream, ומדפיסה אליו את האיבר השכיח ביותר. אם המונה ריק, יש לזרוק שגיאה. למשל, עבור הדוגמה הקודמת, היא תדפיס abc:3.

מעבר לדרישות אלו, אתם חופשיים במימוש טיפוס זה כל עוד הפונקציונליות שלו נשמרת.

חלק ב': מנהל מונים

בחלק זה של התרגיל עליכם לממש מפרש (parser) המשתמש במחלקה שהוגדרה בסעיף א למימוש.

הפקודות הנתמכות הן:

1. הוספת מונה של שלמים או מחרוזות – int_counter <counter_name> | string_counter <counter_name>.
הפקודה מוסיפה למאגר מונה שלמים או מונה מחרוזות בשם הנתון.
אם כבר יש מונה בשם זה יש לפלוט הודעת שגיאה Counter name already exists, ולסיים את התוכנית.
2. הוספת ערכים מקובץ למונה -- add_from_file <counter_name> <file_name>.
קריאה הקובץ כערכים מטיפוס המונה, והוספת הערכים למונה.
אם המונה לא קיים יש להדפיס No counter named XXX כאשר XXX שם המונה.
אם הקובץ לא קיים יש להדפיס Failed to open file
אם אירעה שגיאה במהלך קריאה הקובץ יש להדפיס Read failed.
3. הוספת ערכים משורת הפקודה -- add_from_string <counter_name> <valud1> ... <valuen>.
הפקודה קוראת אפס או יותר ערכים משורת הפקודה ומוסיפה אותם למונה.

תכנות בשפת ++C, אביב 2021

אם המונה לא קיים יש להדפיס No counter named XXX כאשר XXX שם המונה.

אם אירעה שגיאה במהלך קריאת הערכים יש להדפיס Read failed

4. הדפסת הסטטיסטיקה למונה – stats <counter_name>

הפקודה מדפיסה את סטטיסטיקת המונה לפלט בפורמט של print_to_stream מחלק א.

אם המונה לא קיים יש להדפיס No counter named XXX כאשר XXX שם המונה.

5. הדפסת האיבר השכיח למונה – most_common <counter_name>

הפקודה מדפיסה את האיבר השכיח של המונה לפלט בפורמט של print_most_common מחלק א.

אם המונה לא קיים יש להדפיס No counter named XXX כאשר XXX שם המונה.

אם המונה ריק יש להדפיס Counter is empty.

הערות כלליות ל-parser:

i. כל פקודה נמצאת בשורה נפרדת.

ii. יש להתעלם משורות ריקות.

iii. בכל מקרה של שגיאה, יש להדפיס ל-stderr, ולסיים את התוכנית עם סטטוס שונה מאפס.

iv. במקרה שחסר ארגימנט לפקודה, יש להדפיס את אחת השגיאות הבאות ולסיים את התוכנית:

- Missing counter name

- Missing file name

v. במקרה של פקודה שאינה ברשימה למעלה, יש להדפיס שגיאה

Unknown command CCC כאשר CCC הפקודה הלא חוקית.

דוגמא

הרצת התוכנית יכולה להראות כך:

```
[input]    string_counter sc1
[input]    stats sc1
[output]   counter sc1 = { }
[input]    add_from_string sc1 abc abc a
[output]   counter sc1 = { a:1 abc:2 }
[input]    string_counter sc2
[input]    add_from_file sc2 poem.txt
[input]    stats sc2
[output]   counter sc2 = { And:3 But:1 By:1 I:1 May,:1 Nor:2 }
[input]    most_common sc2
[output]   most common element for counter sc2 is And:3
[input]    add_from_string sc1 abc
[input]    stats sc1
[output]   counter sc1 = { a:1 abc:3 }
[input]    int_counter ic
[input]    add_from_string ic 1 2 3 99 3 2 1
[input]    stats ic
[output]   counter ic = { 1:2 2:2 3:2 99:1 }
[input]    most_common ic
[output]   most common element for counter ic is 1:2
```

דגשים

- יש להשתמש במנגנון החריגות למימוש השגיאות (exceptions).
- יש לממש את המחלקה Counter כמחלקת תבנית (template).
- אתם רשאים להשתמש בספריית STL בתרגיל זה.

תכנות בשפת C++, אביב 2021

- אפשר, אבל אין חובה להשתמש בפולימורפיזם.
- עליכם לוודא כי התכנית עוברת קומפילציית g++ התואמת את הקומפיילר שעל שרת המכללה ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה.
- עליכם לתעד את הקוד באמצעות הערות המתארות בקצרה את הפונקציות השונות.
- יש להריץ את הבודק האוטומטי על שרת החוג בטרם ההגשה בכדי לוודא תאימות ונכונות של ההגשה: התחברו לשרת החוג והריצו **hwcheck** על הארכיב שלכם, או לחילופין העלו את הארכיב תוך שימוש בפרוטוקול HTML בקישור <https://cs.telhai.ac.il/homework>

הגשה

- עליכם להגיש במערכת Moodle קובץ ארכיב מטיפוס zip בלבד, ששמו כולל את קוד הקורס ('32'), שם התרגיל ('ex5') ותעודת הזהות של הסטודנט/ית המגיש/ה, מופרדים בקו תחתי בפורמט הבא:
32_ex5_studID.zip
- ניתן, אך לא חייב, להגיש בזוגות. במקרה כזה רק אחד מהסטודנטים מגיש ושם קובץ ההגשה יהיה בפורמט הבא:
32_ex5_firstID_secondID.zip
- על ארכיב zip זה להכיל את כל קבצי המקור (ממשק/מימוש) הנדרשים לקומפילציה, והוא רשאי להכיל תיעוד טקסטואלי; מבחינת טיפוס קבצים, עליו לכלול רק קבצים עם סיומות ***.txt *.h *.cpp**
- לדוגמה: על סטודנט שמספר הזיהוי שלו הינו 012345678 להגיש ארכיב בשם **32_ex5_012345678.zip** הכולל את כל קבצי המקור של הפרוייקט, ללא תיקיות כלשהן, ורשאי להכיל קובץ טקסטואלי לתיעוד.

אי-הקפדה על ההנחיות, כולל פורמט ההגשה הדיגיטלי, תגרור הורדה בציון התרגיל.
לא תתקבלנה הגשות באיחור!