

ניתוח ותיכון מונחה עצמים - תרגיל מס' 1

מצביעים חכמים ועבודה עם STL's:

חלק א: מימוש מצביעים חכמים:

בחלק זה עליכם להשלים את המימוש של הפוינטרים החכמים שלמדנו:
לרשותכם הקבצים pointers.h ו-main.c
עליכם לוודא את תקינות המימוש של מחלקות:
1. Unique Pointer
2. Shared Pointer
3. Weak Pointer

יש לוודא שה- Shared וה- Weak עובדים בצורה מסונכרנת ביחד
ומכילים counter משותף כמו שראינו בתרגול.

חלק ב' מימוש המחלקה התבניתית:

CacheMemory<K,V>

אשר בהינתן שני פרמטרים K(מפתח) ו-V(ערך) מחוללת מחלקה אשר מכילה
שתי מפות (std::map):
1. מפה המכילה את כלל האיברים אשר הערך שלה הוא תמיד מצביע חכם משותף
std::shared_ptr<V> לערך V.
2. מפה המשמשת כזיכרון מטמון (Cache) אשר תכיל רק את האיברים שניגשו אליהם
לאחרונה, והערך שלה הינו מצביע חכם חלש std::weak_ptr<V> לערך V.

ממשו את המתודות הבאות בתוך המחלקה:

void add (const K& key, const V& value);

בהינתן מפתח key וערך value על המתודה להוסיף איבר למפה (לא ב-cache)
במידה והמפתח כבר קיים יש לזרוק חריגה מסוג "DuplicateKeyException".

void erase(const K& key);

בהינתן מפתח key, המתודה מחפשת במפה (לא ב-cache) איבר אשר לו מפתח זה ומוחקת
אותו. במידה ואיבר זה אינו קיים, יש לזרוק חריגה מסוג "ObjectNotExistException".

std::shared_ptr<V> get(const K &key);

בהינתן מפתח key, המתודה תחילת מחפשת ב cache שמא קיים ערך למפתח זה. במידה וכן, מבחינה המתודה בין שני מקרים:

1. הערך שנמצא ב cache תקין - במקרה זה, המתודה מחזירה מצביע משותף (shared_ptr) לאובייקט זה.
2. הערך שנמצא ב cache אינו תקין - במקרה זה, המתודה מוחקת האיבר מה cache.

במידה והאיבר **אינו נמצא** ב cache, מחפשת המתודה האיבר במפה. במידה ונמצא, תחילה **מוסיפה אותו ל- cache** ולאחר מכן מחזירה מצביע משותף (ptr_shared) לאובייקט זה. במידה ולא נמצא, המתודה מחזירה nullptr.

std::vector<V> getCacheValues();

המתודה תחזיר וקטור אשר מכיל את כל איברי מפת ה- Cache. אם המפה ריקה, יש להחזיר וקטור ריק.

חלק ג' - (בנוסף) קריאה מקבצים ושימוש במחלקה שמימשתם:

עליכם לייצר את מחלקת Employee בה לכל עובד השדות הבאים:

1. id (מחרוזת)
2. salary (מספר שלם - int)
3. seniority (מספר שלם - int)

מה הדבר החשוב ביותר בעבודה עם קבצים?
להכיר את הפורמט!
פורמט הקובץ נראה כך:

[ID1-Length][ID1][Salary1][Seniority1][ID2-Length][ID2][Salary2][Seniority2]...
[IDN-Length][IDN][SalaryN][SeniorityN]

ממשו את המתודות save & load לצורך שמירה וטעינה של עובדים מהקובץ (יש ליצור את emps.bin). בדקו נכונות מימוש חלק ב' ע"י הכנסת העובדים למבנה הנתונים שמימשתם בחלק ב' וביצוע פעולות עליו.

יש להדגים את העבודה עם הקבצים ועם מבנה הנתונים הגנרי בקובץ ה- main.cpp.

דגשים:

חלק א' - עליכם להגיש שני קבצים
main.cpp + pointers.h

חלק ב':

- את מחלקות החריגות ניתן לממש כמחלקות ריקות בתוך קובץ ה- h של המחלקה התבניתית
- השתמשו ב- STL מתוך std
- עליכם להגיש קובץ אחד - cached_map.h

חלק ג' - עליכם להגיש שלושה קבצים employee.h, employee.cpp ו- main.cpp המפעיל את הטעינה והמחלקות. (ניתן להשתמש באותו קובץ main.cpp מסעיף א')

יש להגיש את כלל הקבצים בתוך קובץ מכוון אחד לתיבת ההגשה במודל.

בהצלחה!
צוות ניתוח ותיכון