***כותרת:***

Matrix calculator.

===================================================

**פרטי סטודנטים:**

**שמות**

* יעקב בונדר, ת"ז: 322336488
* שלי צ'אוסובסקי, ת"ז: 323998526

**הסבר כללי על התרגיל:**

בתרגיל זה אנו יצרנו מחשבון מטריצות אשר בו ניתן להוסיף פעולות חדשות ולחשב מטריצות בהתאם לבקשת המשתמש. הפעולות אותן ניתן לעשות ובהתאם להן ליצור תרגילים חדשים הם:

* Id : מחזירה את המטריצה עצמה.
* Tran : מחזירה את המטריצה לאחר ביצועה פעולת transpose.
* Scal x : הכפלה של המטריצה כולה ב – x, כאשר x הוא מספר ממשי.

ניתן לבצע בין פונקציות אלה פעולות חיבור, חיסור והרכבה ( כאשר זוהי פעולה בה לא נבקש מטריצה חדשה אלא נשתמש במטריצה מהפעולה הקודמת).

**רשימה של קבצים שיצרנו:**

Function.h – מחלקה וירטואלית אשר ממנה יורשות המחלקות- Id, Tran , Scal

Scal.h –מכילה את הגדרת המחלקה Scal

Scal.cpp – מכילה את המימוש של המחלקה.

Tran.h –מכילה את הגדרת המחלקה Tran

Tran.cpp – מכילה את המימוש של המחלקה.

Id.h –מכילה את הגדרת המחלקה Id

Id.cpp – מכילה את המימוש של המחלקה.

Operation.h - מחלקה וירטואלית אשר ממנה יורשות המחלקות- ComplexOperation וה- BasicOperation

ComplexOperation.h –מכילה את הגדרת המחלקה ComplexOperation

ComplexOperation.cpp – מכילה את המימוש של המחלקה.

BasicOperation.h –מכילה את הגדרת המחלקה BasicOperation

BasicOperation.cpp – מכילה את המימוש של המחלקה.

Calculator.h- מכילה את הגדרת המחלקה Calculator

Calculator.cpp- מכילה את המימוש של המחלקה.

Action.h – מחלקה וירטואלית אשר ממנה יורשות המחלקות – Add, Sub, Comp.

Add.h –מכילה את הגדרת המחלקה Add.

Add.cpp – מכילה את המימוש של המחלקה.

Sub.h –מכילה את הגדרת המחלקה Sub.

Sub.cpp – מכילה את המימוש של המחלקה.

Comp.h –מכילה את הגדרת המחלקה Comp.

Comp.cpp – מכילה את המימוש של המחלקה.

Help.h- מכילה את הגדרת המחלקה Help

Help.cpp- מכילה את המימוש של המחלקה.

**מבני נתונים עיקריים ותפקידם:**

במחלקת מטריצה ישנו וקטור של ווקטורים של float בו אנו נזכור את ערכי המטריצה בהתאם לגודל שנקבל. בנוסף במחלקת Calculator ישנם 2 ווקטורים מסוג shared ptr. האחד מסוג Operation המחזיק כל שורה במחשבון שיצרנו (עם פעולתו) והשני מסוג Function המחזיק מצביע לכל פונקציה שציינו בהתחלה.

**אלגוריתמים הראויים לציון:**

**בתוכנית זו השתמשנו במעין רקורסיה. זאת אומרת כאשר ברצוננו לחשב את תוצאת השורה אותה המשתמש הזין אזי התוכנית ניגשת לשורה זו במחשבון. בשורה זו יש מצביעים או לפונקציה שיש בה (אם בשורה זו לא מצוי תרגיל) או ניגשת לשני האגפים האחרונים שבהם בוצא התרגיל (+ - או <-), בעצם שמורים לנו שני מצביעים המצביעים על שני אגפים אלו. אנו ניכנס למימוש שלהם ולחישוב ובעצם מעין רקורסיבית נקרא לשני התרגילים שמהם האגף הזה מורכב כך נמשיך עד שנגיע לפונציה בודדה ממנה נחזיר את המטריצת תשובה של כל אגף אליו הגענו.**

**תיכון (design)**

**המחלקה** Calculator **רצה ראשונה. לאחר מכן היא מחכה לקלט המשתמש, בו המשתמש יכול לבחור או להשתמש בפעולה קיימת או להוסיף פקודה חדשה. בהינתן הוא מוסיף פעולה שהינה סקאל נוסיף אותה לרשימת הפעולות הקיימות פעולה בסיסית.**

**בהינתן הוספת הפעולה הינה חיבור חיסור או** comp, **נוסיף אותה לרשימת הפעולות הקיימות כפעולה מורכבת (המחברת בין שתי פעולות קיימות).**

**אם המשתמש מחליט להשתמש בפעולה קיימת אז אנחנו נקבל ממנו את כמות המטריצות הרלוונטיות או את התרגיל בכך שהוא יזין אותם כקלט. החישוב של הפעולה מתבצע על ידי קריאות רקורסיביות אשר כל קריאה אנחנו יורדים לפעולה קטנה יותר עד שנגיע לפעולות הבסיסיות. בפעולות הבסיסיות נחשב את הפתרון של המטריצה ואז נחזיר אותה לפעולה המורכבת יותר שקראה לה, עד אשר נגיע לפעולה הראשונה שהתחילה את כל הקריאות וכן תחזיר את המטריצה שהתקבלה בחישוב, ונציג את מטריצת הפיתרון שיצאה למשתמש.**

**באגים ידועים:**

**הערות נוספות:**