**תרגיל 2 סמסטר ב' נושא התרגיל – Templates**

שם המגישים: מיכאל אורייב ישעיה וינסטוק

ת.ז. - michaelur 317458123

ת.ז. ishaiawi - 304939234

*הסבר כללי:*

בתרגיל זה קיבלנו קובץ main מוכן והיינו צריכים ליצור קוד גנרי המשלב templates כך שקובץ הmain יוכל לפעול ללא שום שינוי מצדינו.

*תיכון + רשימת הקבצים שנוצרו:*

בתוכנית יש את האובייקטים הבאים:

BaseField – מכיל פונקציות שקיימות בכל השדות, את השאלה שיש לשאול את המשתמש בעת הקלט, ודגל שאומר האם השדה תקין.

Field – מייצג שדה שביחס אליו יפרסו יתר המחלקות, בתוכו נקלוט כל שדה בצורה טמפלטית שיפרס ביחס לאותו משתנה T שצריך, בתוך השדה יש ווקטור T מסוג ולידטור שיפרס לפי הסוג שנקבל

BaseValidator – מכיל פונקציות וירטואליות שממומשות בשאר ה-validators

ClassNames / CurrencyNames / DestinationNames – כל אחת מכיל וקטור בעל השמות של המחלקות/מדינות/מטבעות ופונקציה שמחזירה אותם.

ComplexValidator – בכדי לבצע המרה בין מחלקה שמקבלת טמפלט אחד למחלקה שמקבלת 2 טמפלטים

DestinationToClassValidator / DestinationToCurrencyValidator – יורשים מ-ComplexValidator ובודקים האם ניתן לבחור מטבע/מחלקה מסוימים בהינתן מדינה מסוימת

Form - המקום בו ננהל את הקריאה מהטופס, כתיבה, הצגה ועוד

IdValidator - בודק תקינות של ספרות של תעודת זהות

NoDigitValidator – בודק תקינות של String

RangeValidator – בודק תקינות של קלט שהוא בטווח מסויים

ValueToNames – מקבלת מספר ומחזירה את הString הרלוונטי למספר שהתקבל

*התקשורת בין האובייקטים כדלקמן:*

כאשר יוצרים אובייקט כלשהו האובייקט נפרס ביחס למה שמבקשים ממנו בזמן היצירה, בדרך זו חוסכים צורך ביצירת מחלקות רבות, ואפשר להשתמש בכלים של הספריה הסטדנרטית של טמפלט.

כגון: הכנסה,הדפסה,שינוי, קליטת ערכים וכ'ו..

*רשימת הקבצים שנוצרו:*

BaseField.h

Field.h

BaseValidator.h

ClassNames .h

CurrencyNames .h

DestinationNames.h

ComplexValidator.h

DestinationToClassValidator.h

DestinationToCurrencyValidator.h

Form.h

IdValidator.h

NoDigitValidator.h

RangeValidator.h

.h ValueToNames

Form.cpp

Main.cpp

IdValidator.cpp

*מבני נתונים:*

וקטורים שמחזיקים string-ים לשמירת השאלות שמציגים למשתמש, מקומות טיסה וכ'ו

וקטור מהסוג vector<BaseValidator<T>\*> כדי לשמור רשימה של ולידטורים(במחלקת field)

וקטור מסוג vector<BaseField \*> כדי לשמור את הfield- השונים

וקטור מסוג vector<BaseValidator <class T>\*>

*אלגוריתמים מתוחכמים:*

/

*באגים:*

/

*הערות:*

/