הנדסת תוכנה 094129

תרגיל בית 4

תכנות גינרי ותבניות עיצוב – Templates

תאריך אחרון להגשה : 30\06\2018 עד השעה 23:55

נושא התרגיל: מימוש מערכת לניהול טיסות.

מטרת התרגיל:

בשדה תעופה ממריאים ונוחתים מטוסים, כל מטוס נושא עמו מטען מסוג מסוים. ייתכנו מספר גדול של סוגי מטענים (לפעמים סוג המטען לא ידוע מראש).

בשדה התעופה ישנם טרמינלים שמהם ממריאים או אליהם נוחתים מטוסים, כלומר בטרמינל בודד תיתכן תעבורה של מספר רב של מטוסים. עקב מספר גדול של סוגי מטען הופיע צורך בקוד גמיש ,שיוכל להתאים לתנאים בזמן ריצה.

בגלל הדרישות לעיל יש לממש תבניות (Templates) עבור המחלקות הבאות :

- מחלקה שמייצגת מטוס Airplane
- מחלקה שמייצגת טרמינל Terminal .

מטרת התבניות :

1. להקל על הוספת סוגי מטענים חדשים למערכת
2. לאפשר לכל אחד מהטרמינלים לטפל רק בסוג מסוים של מטוסים (סוג מסוים של מטוס נקבע לפי סוג המטען שהוא נושא)

- **תיאור הדרישות:**

בתרגיל בית יש לממש את המתודות הבאות, בהתאם לממשק המתאור להלן :

Airplane	
Class members (private):	
שם של המטוס	const string_name
גודל של מטען שמטוס מביא\לוקח (טיפוס T של המשתנה לא ידוע מראש)	T_load
Class methods (public):	
קונסטרקטור	**Airplane** (const string& name, const T&load)
קופי-קונסטרקטור	**Airplane** (const Airplane &otherAirplane)
דסטרקטור – לממש רק במידת הצורך	**~Airplane()**
המתודה מחזירה שם של המטוס	string **getName()** const
המתודה מחזירה גודל של מטען שמטוס מביא\לוקח	T **getLoad()** const
המתודות משוות שני מטוסים לפי שמות המטוסים ומחזירה true אם השמות זהים, false אחרת.	bool **isEqual** (const string& name) const
	bool **isEqual** (const Airplane& otherAirplane) const

Terminal	
Class members (private):	
סוג של המטען שהטרמינל עובד איתו (אצלנו CARGO TERMINAL או PASSENGER TERMINAL)	string _termnType
נפח מקסימלי של מטען בטרמינל בהתאם לסוג מטען שלו	const T _maxTermnLoad
וקטור של המטוסים הנמצאים בטרמינל	vector<Airplane<T>* > _vecAirplanes;
כמות המטען הנוכחי בטרמינל (או, לדוגמה, מספר הנוסעים)	T_terminalLoad;
Class methods (public):	
קונסטרקטור	**Terminal** (const string &termnType, T maxTermnLoad);
קופי-קונסטרקטור	**Terminal** (const Terminal &otherTerminal);
דסטרקטור – לממש במידת הצורך	**~Terminal()**;
מתודה שמנהלת את טרמינל, הרחבה בהמשך	void **beginSession()**;
להוסיף מטוס ולהחזיר true אם הצלחנו, false אחרת	bool **addPlane** (Airplane<T>* newAirplane);
להוריד מטוס ולהחזיר true אם הצלחנו, false אחרת	bool **delPlane** (Airplane<T>* newAirplane);
האם המטוס קיים בטרמינל	bool **hasPlane** (const string &name) const;
להדפיס שמות של המטוסים הנמצאים בטרמינל	void **print()** const;

לצורך התרגיל אפשר להניח שכרגע תרמינל עובד עם שני סוגים של המטוסים :

1. **מטוסי מטען כללי** : טיפוס של T (ב- Terminal <T>) במקרה הזה מיוצג ע"י **double** הקיבולת המקסימלית של טרמינל המטען הכללי הוא **50** טון
2. **מטוסי נוסעים** : טיפוס של T (ב- Terminal <T>) במקרה הזה מיוצג ע"י **unsigned** הקיבולת המקסימלית של טרמינל נוסעים הוא **2000** אנשים

כמובן שניתן ליצור מטוסים עם מטענים מסוגים נוספים. מטען נוסף יכול להיות מוגדר כמחלקה. ב- main שקיבלתם יש שתי דוגמאות, בבדיקה שלנו נרחיב את הטיפוסים האפשריים ל T .
מותר להניח כי לטיפוס T יש (אין להניח שום דבר נוסף בנוגע לטיפוס T פרט לרשימה לעיל) :

1. בנאי ברירת מחדל default constructor, בנאי העתקה copy constructor, בנאי המקבל פרמטר double- constructor that receives double, הורס destructor
2. העמסות אופרטורים : אופרטור + , אופרטור - ,אופרטור =, אופרטור =< והעמסת אופרטור קלט (<<)

שימו לב אינכם נדרשים לממש זאת, אלה הנחות שאתם יכולים להניח במימוש.

void beginSession() הרחבה על :

זוהי מתודה שבעצם מנהלת את הטרמינל. לאחר שעיבדנו ב main את הקלט הראשוני שמתייחס לסוג המטען שהטרמינל עובד איתו (אצלנו CARGO TERMINAL או PASSENGER TERMINAL), בפונקציה הזו אנחנו מתעסקים בקליטת נתונים לגבי המטוסים והמטען שלהם.

יש לקלוט את שורת הפקודה כפי שמוסברת בחלק 2 (הקלט למערכת) (למשל באמצעות istringstream)

למשתנה בודד וממנו לפרסר למשתנים הבאים את שורת הפקודה :

string line : std:: - משתנה שיכיל את שורת הפקודה (A LY1 10) , המשתנה שממנו נפרסר

string mode : std:: – יכיל את תיאור הפקודה (PRINT/D/A)

string name : std:: – יכיל את השם של מטוס (למשל כאן LY1)

T load - יכיל את גודל המטען שמטוס לוקח/מביא (למשל כאן 10)

שימו לב עבור הפקודה PRINT אין משמעות למשתנים name ו- load , עבור פקודה זו יש להדפיס את שמות המטוסים הקיימים בטרמינל .

- **הקלט למערכת:**

ניתן לראות שתי דוגמאות לקלטי ופלטי המערכת בהתאם בתיקיית תרגיל הבית שלכם.
המערכת קולטת את נתוני הפקודות מערוץ הקלט הסטנדרטי (cin). הקלט מחולק לפקודות, וכל פקודה מופיעה בשורה נפרדת .סדר ביצוע הפקודות חייב להיות לפי סדר הופעתן בקובץ. המילה הראשונה בפקודה הינה תמיד השם שלה, ואח"כ באים פרמטרים לפקודה, מופרדים ברווחים. אפשר להניח שקלט תקין ואין בו טעויות. הפקודות האפשריות הן (ישנה חשיבות לגודל האותיות של הפקודות- חשוב לכתוב אותן באותיות גדולות בלבד):

מס	תיאור פקודה	מבנה שורת פקודה	פרמטרים
1	נחיתה	***A*** *Name Cargo_Size*	*A* – סימן של נחיתה *Name* – שם של המטוס *Cargo_Size* – גודל המטען (או מס' הנוסעים)
2	המראה	***D*** *Name Cargo_Size*	*D* – סימן של המראה *Name* – שם של המטוס *Cargo_Size* – גודל המטען (או מס' הנוסעים)
3	הדפס	***PRINT***	הדפסה של המטוסים שנמצאים בטרמינל

- **התנהגות התוכנה:**

1. נחיתה (***A***):
   a. במידה ומטוס כבר קיים בטרמינל – יש להתעלם מהפקודה (מנסים להנחית מטוס שכבר נחת ונמצא כרגע בטרמינל) ולהדפיס את שורה הבאה:
   The plane is already registered: *Name*
   Name – שם של המטוס הנוחת.
   b. במידה ואין מספיק מקום בטרמינל, כלומר עברנו את הנפח המקסימלי של מטען שהטרמינל יכול להכיל – יש להתעלם מהפקודה ולהדפיס את שורה הבאה:
   Not enough free space for: *Name*
   Name – שם של המטוס הנוחת
   c. במידה ויש מקום פנוי בטרמינל – יש להוסיף מטוס (נוכל להוסיף מטוס לטרמינל רק אם בתוספת המטען של המטוס אנו לא חורגים מהנפח המקסימלי של מטען שטרמינל יכול להכיל) ולעדכן את כמות המטען הנוכחית בטמרינל.
   - שימו לב, את הבדיקות צריך לבצע בסדר המתואר

2. המראה (***D***):
   a. במידה ומטוס לא קיים בטרמינל (מנסים להמריא מטוס שלא נמצא בכלל בטרמינל) – יש להתעלם מהפקודה ולהדפים את שורה הבאה:
   The plane was not registered: *Name*
   b. במידה ואין מספיק מטען בטרמינל (לא תיתכן כמות מטען שלילית בטרמינל , כמות המטען המינימאלית שיכולה להיות בכל טרמינל היא אפס) – יש להדפיס את שורה הבאה:
   Not enough allocated objects for: *Name*

c. במידה ויש מספיק מטען בטרמינל ומטוס קיים במערכת – יש למחוק את המטוס ולהוריד את הכמות של המטען של המטוס שהמריא
d. מטוס יכול להמריא עם כמות מטען יותר גדולה מהכמות שאיתה הוא נחת לטרמינל בשדה.

- שימו לב, את הבדיקות צריך לבצע בסדר המתואר

3. הדפסה (***PRINT***):
יש להדפיס שמות של המטוסים הקיימים בטרמינל: כל שם בשורה חדשה, במידה ואין מטוסים בטרמינל לא יודפס דבר.

- **הבנת הקוד הניתן:**

הפונקציה main שניתנה עם התרגיל, מממשת את אופן העבודה הבא:
אנחנו מקבלים מהקלט את טיפוס של המטען הרצוי בטרמינל ויוצרים אובייקט ממנו. מאותה נקודה עובדים עם המטען הנ"ל בלבד עד שמתקבלת הפקודה Ctrl+Z (או סוף קובץ) ואז התוכנית מסתיימת.
טיפוסים של מטען מוגדרים ע"י פקודות הבאות:

- CARGO TERMINAL
- PASSENGER TERMINAL

- **מימוש התבניות:**

עליכם לממש את התבניות לפי התיאור לעיל ובהתאם לשימוש שנעשה בה ב main. **אסור לשנות את הקובץ main.cpp וממשק של המחלקות המתוארות לעיל .** על שמות הפונקציות והתבנית להיות כפי שהם מופיעים בתיאור הדרישות.
הבהרות נוספות:

- מותר לממש **מתודות/אופרטורים** נוספים שאינם חלק מהממשק, אלא משמשות לעזר.
- התבניות צריכות לדעת לעבוד עם כל סוגי המטען (טיפוס כלשהו).
- מותר להניח תקינות קלט.
- יש לשמור על כללי תכנות נאות ו- const correctness .
- המימוש צריך להכיל רק את הקבצים הבאים: main.cpp, Airplane.h, Terminal.h
- ההגשה חייבת להכיל קובץ zip יחיד (לא rar. וכדומה)
- שם קובץ ה-zip חייב להיות: hw4_XXXXXXXXX_YYYYYYYYY.zip

כאשר XXXXXXXXX הינם מספרי ת"ז של השותפים ,כולל ספרת בקורת.

בהצלחה !