

1. Cel ćwiczenia

Nauka

2. Pojęcia

Polimorfizm statyczny – rodzaj polimorfizmu, w którym decyzje o wyborze fragmentu kodu do wykonania, są podejmowane w trakcie kompilacji programu, np.: przeciążanie operatorów i szablony.

RAII (Resource Acquisition Is Initialization) – technika programowania w której łączy się przejęcie i zwolnienie zasobu z inicjowaniem i usuwaniem zmiennych.

Inteligentny wskaźnik – typ danych, który symuluje wskaźnik, ale dodatkowo zapewnia funkcje odśmiecania lub sprawdzania zakresu.

Do zapoznania przed laboratorium:

<http://www.cplusplus.com/doc/oldtutorial/templates/>

3. Instrukcja

Napisz klasę szablonową SmartPointer<class T>, która implementuje wzorzec projektowy inteligentny wskaźnik. Klasa SmartPointer:

- a. Posiada pole ptr, które jest wskaźnikiem na typ podany w szablonie
- b. Posiada pole counter, które jest wskaźnikiem na typ ReferenceCounter
- c. Typ ReferenceCounter jest klasą, która
 - i. Posiada pole counter typu int
 - ii. Implementuje konstruktor domyślny, inicjalizujący pole counter wartością zero.
 - iii. Implementuje konstruktor, który przyjmuje parametr typu int, który zostaje przypisany polu counter.
 - iv. Implementuje metody:
 - AddRef() – inkrementuje counter,
 - DelRef() – dekrementuje counter,
 - Count() – zwraca wartość zmiennej counter.
- d. Implementuje:
 - i. Konstruktor domyślny – konstruktor, który inicjalizuje pola obiektu wartościami nullptr.
 - ii. Konstruktor przyjmujący wskaźnik na typ podany w szablonie – konstruktor, który przypisuje do pola ptr wskaźnik podany jako argument.

- iii. Konstruktor kopiujący – przepisuje pole ptr i inkrementuje licznik referencji.
- iv. Destraktor – dekrementuje licznik referencji i usuwa obiekt, jeśli był to ostatni obiekt typu SmartPointer.
- v. Operator przypisania – przepisuje pole ptr i inkrementuje licznik referencji.
- vi. Operator -> - zwraca pole ptr.
- vii. Operator * - zwraca pole ptr.
- viii. Operator bool - sprawdza czy pole ptr jest równe pustemu wskaźnikowi.
- ix. Metodę reset() – zwalnia pole ptr.
- x. Metodę reset(T* ptr) – przypisuje do pola ptr podany argument.
- xi. Metodę swap(SmartPointer& r) – zamienia zawartość obiektów *this i r.
- e. Poza klasą zaimplementować funkcję szablonową make_smart_pointer(), która stworzy obiekt typu podanego w szablonie i zwróci obiekt klasy SmartPointer, wskazujący na stworzony obiekt.

4. Zaliczenie

Wysłać na platformę pliki z zaimplementowaną klasą SmartPointer.