Menu

Marta Pacuszka

Spis treści

Rozdział 1

Informacje o projekcie

Polecenie

Menu programu okienkowego jest kolekcją wyborów. Każdy z wyborów jest albo jednoznaczny albo wskazuje na inne menu. Oprogramować bibliotekę klas reprezentujących menu. Biblioteka ma umożliwiać następujące czynności:

- Dodawanie/usuwanie wyborów jednoznacznych do/z (pod)menu.
- Dodawanie/usuwanie podmenu do/z (pod)menu.
- Przypisywanie wyborów jednoznacznych do pewnych funkcji.
- Rozwijanie/zwijanie podmenu.
- Dokonywanie wyborów jednoznacznych. Napisać program, który wszystkie powyższe funkcjonalności udostępnia poprzez polecenia wydawane z klawiatury. Przyjąć, że za wyborami jednoznacznymi może kryć się stały zbiór funkcji wypisujący na konsoli komunikaty typu “Zadziałała funkcja nr 14”.

Założenia

1. Menu zawsze musi się składać z conajmniej jednego Wyboru typu Podmenu. Nie można usunąć takiego Podmenu.
2. Usunięcie Podmenu oznacza usunięcie razem z nim wszystkich zagnieżdżonych w nim Wyborów.
3. Po usunięciu elementu (elementów) Menu kursor wskazuje na poprzedni obiekt.
4. Po dodaniu nowego zagnieżdżonego Wyboru, kursos jest ustawiony na tym nowododanym Wyborze.
5. Jeżeli Podmenu jest zwinięte, nie ma możliwości przejścia kursorem do któregokolwiek z zagnieżdżonych w nim Wyborów. W tym celu należy je najpierw rozwinąć.

Struktura klas reprezentujących Menu

Klasy użyte do zaprojektowania Menu:

1. Wybor
    - Jednoznaczny
    - Podmenu
2. Kolekcja

3. Obsluga
4. Interfejs

Metody wykorzystywane w interfejsie użytkownika

Użytkownik dysponuje wieloma metodami, pozwalającymi na swobodne poruszanie się po Menu i zarządzanie nim. Są to metody publiczne klasy Obsluga - w sekcji tej klasy szczegółowo opisano działanie i parametry każdej z metod.

Sposób testowania

Za obsługę interfejsu tekstowego służącego do testowania programu odpowiada klasa Interfejs - szczegółowy opis klasy wraz z opisem jej metod publicznych znajduje się w osobnej sekcji poświęconej tej klasie.

Sposób 1 - od zera

Po uruchomieniu programu użytkownik zostaje poproszony o podanie nazwy Menu - zostanie utworzone pierwsze Podmenu o stopniu zagnieżdżenia 0. Następnie uzytkownik może poruszać się po Menu za pomocą klawiszy W (góra) i S (dół). Dodatkowe opcje są zależne od tego, czy kursos pokazuje na Podmenu czy Wybór Jednoznaczny i wszystkie możliwe do użycia skróty wraz z opisami są wypisane na ekranie bezpośrednio pod Menu. Taki sposób testowania umożliwia metoda publiczna klasy Interfejs Start().

Sposób 2 - bazując na stworzonym Menu

Program można również uruchomić dla stworzonego wcześniej w interfejsie użytkownika Menu. Wówczas użytkownik nie musi wprowadzać nazwy Menu - od razu można nawigować i zarządzać Menu w taki sam sposób, jak opisano dla sposobu 1. Taki sposób testowania umozliwia metoda publiczna klasy Interfejs Program(Obsluga &M), gdzie obiekt M został wcześniej stworzony (i modyfikowany).

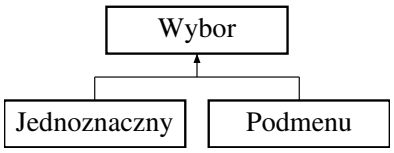
Rozdział 2

Dokumentacja klas

2.1 Dokumentacja klasy Wybor

`#include <wybor.h>`

Diagram dziedziczenia dla Wybor



Metody publiczne

- **Wybor** (std::string s)

Metody chronione

- int **JakiStopienZagniezdzenia** ()
- void **Ukryj** ()
- void **Pokaz** ()
- bool **hidden** ()
- void **NadajStopienZagniezdzenia** (int n)
- virtual void **Wypisz** (std::ostream &ekran)

Atrybuty chronione

- std::string **nazwa**
- int **stopienZagniezdzenia**
- bool **czyUkryty**

Przyjaciele

- class **Kolekcja**
- class **Obsluga**
- std::ostream & **operator**<< (std::ostream &ekran, Wybor &w)

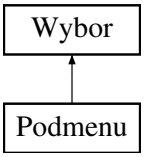
2.1.1 Opis szczegółowy

Klasa reprezentująca pojedynczy wybór - jest klasą bazową dla klas reprezentujących wybory jednoznaczne i podmenu . Z jej obiektów korzysta zaprzyjaźniona klasa Kolekcja. Klasa przechowuje takie informacje o wyborze, jak jego nazwa czy stopień zagnieżdżenia.

2.2 Dokumentacja klasy Podmenu

`#include <wybor.h>`

Diagram dziedziczenia dla Podmenu



Metody publiczne

- **Podmenu** (std::string s)

Przyjaciele

- class **Kolekcja**
- class **Obsluga**

Dodatkowe Dziedziczone Składowe

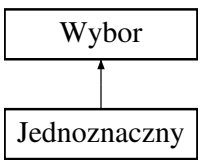
2.2.1 Opis szczegółowy

Klasa dziedzicząca z klasy Wybór. Obiekty tej klasy reprezentują Podmenu - Wybór, w którym można zagnieżdżać inne Wybory (o wyższym stopniu zagnieżdżenia).

2.3 Dokumentacja klasy Jednoznaczny

`#include <wybor.h>`

Diagram dziedziczenia dla Jednoznaczny



Metody publiczne

- **Jednoznaczny** (std::string s)

Przyjaciele

- class **Kolekcja**
- class **Obsluga**

Dodatkowe Dziedziczone Składowe

2.3.1 Opis szczegółowy

Klasa dziedzicząca z klasy Wybór. Obiekty tej klasy reprezentują Wybór Jednoznaczny. Poza polami i metodami dziedziczonymi, posiada swoje własne, charakterystyczne dla siebie metody i pola związane z przypisywaniem do wyborów jednoznacznych konkretnych funkcji.

2.4 Dokumentacja klasy Kolekcja

`#include <kolekcja.h>`

Metody publiczne

- **Kolekcja** (std::string s)

Przyjaciele

- class **Obsluga**

2.4.1 Opis szczegółowy

Klasa zaprzyjaźniona z klasą Wybór i klasami pochodnymi, obsługująca obiekty tej klasy. Wybory są przechowywane w liście ZbiorWyborow.

Zgodnie z założeniem, że każda Kolekcja musi posiadać jeden obiekt klasy Podmenu, jej konstruktor tworzy nowy obiekt klasy Podmenu o stopniu zagnieżdżenia 0 i dodaje go na początek listy ZbiórWyborow. Nazwa tego obiektu klasy Podmenu jest jednocześnie nazwą całej kolekcji.

2.5 Dokumentacja klasy Obsluga

`#include <obsluga.h>`

Metody publiczne

- **Obsluga** (std::string s)
- void DodajMenu (std::string s)
- void DodajWyborJedn (std::string s)
- void PrzypiszFunkcje (int n)
- void Usun ()
- void Dalej ()
- void Wstecz ()
- void Zwin ()
- void Rozwin ()
- void Wykonaj ()

Przyjaciele

- class **Interfejs**
- std::ostream & **operator**<< (std::ostream &ekran, Obsluga &m)

2.5.1 Opis szczegółowy

Klasa obsługująca obiekt klasy Kolekcja. Potrafi modyfikować obiekty klasy Wybór, przechowywane przez atrybut klasy Kolekcja. Umożliwia wykonywanie operacji na Kolekcji - dodawanie Wyborów, zwijanie i rozwijanie Podmenu oraz swobodne poruszanie się po kolejnych Wyborach, z uwzględnieniem omijania Wyborów ukrytych.

2.5.2 Dokumentacja funkcji składowych

2.5.2.1 void Obsluga::Dalej ( )

Przesuwa kursor na kolejny element listy ZbiorWyborow. Uwzględnia fakt, że część elementów może być ukryta i pomija je. Gdy Kursor pokazuje na ostatni element, przesuwa go na początek.

2.5.2.2 void Obsluga::DodajMenu (std::string *s*)

Metoda tworząca nowy obiekt typu Podmenu i dodająca go do listy ZbiorWyborow zaraz za elementej tej listy pokazywanym przez atrybut Kursor.

Parametry

s	nazwa obiektu typu Podmenu

2.5.2.3 void Obsluga::DodajWyborJedn (std::string *s*)

Metoda tworząca nowy obiekt typu Jednoznaczny i dodająca go do listy ZbiorWyborow zaraz za elementej tej listy pokazywanym przez atrybut Kursor.

Parametry

s	nazwa obiektu typu Jednoznaczny

2.5.2.4 void Obsluga::PrzypiszFunkcje (int *n*)

Przypisuje do odpowiedniego pola obiektu typu Jednoznaczny wskaźnik na funkcję, którą ma wywoływać ten Wybór.

Parametry

n	numer funkcji, która ma zostać przypisana Wyborowi

2.5.2.5 void Obsluga::Rozwin ( )

Metoda pokazuje wszystkie elementy Kolekcji zagnieżdżone w Podmenu (jeżeli były wcześniej ukryte). Jeżeli Podmenu nie jest zwinięte, zgłasza wyjątek. Jeżeli wywołane dla typu innego niż Podmenu, zgłasza wyjątek.

2.5.2.6 void Obsluga::Usun ( )

Usuwa Wybór, na który aktualnie wskazuje Kursor wraz ze wszystkimi Wyborami zagnieżdżonymi. Kursor na koniec jest ustawiony na poprzednim obiekcie.

2.5.2.7 void Obsluga::Wstecz ( )

Przesuwa kursor na poprzedni element listy ZbiorWyborow. Uwzględnia fakt, że część elementów może być ukryta i pomija je. Gdy Kursor pokazuje na pierwszy element, przesuwa go na koniec.

2.5.2.8 void Obsluga::Wykonaj ( )

Metoda wywoływana dla obiektu klasy Jednoznaczny (dla każdego innego zgłasza wyjątek). Wywołuje funkcję przypisaną do Wyboru Jednoznacznego pokazywanego przez Kursor. Jeżeli funkcja nie została jeszcze przypisana, zgłasza wyjątek.

2.5.2.9 void Obsluga::Zwin ( )

Metoda ukrywa wszystkie elementy Kolekcji zagnieżdżone w Podmenu. Jeżeli wywołane dla typu innego niż Podmenu, zgłasza wyjątek.

2.6 Dokumentacja klasy Interfejs

`#include <interfejs.h>`

Metody publiczne

- void Start ()
- void Program (Obsluga &M)

2.6.1 Opis szczegółowy

Klasa reprezentująca interfejs użytkownika. Menu wyświetlane jest w terminalu w następujący umówny sposób:

- Poziom zagnieżdżenia reprezentowany jest odpowiednim wcięciem.
- Pozycja Kursora pokazywana jest symbolem $***$.
- Lista dostępnych poleceń wyświetlana jest bezpośrednio pod Menu.
- Podmenu reprezentowane jest zapisem: | **Nazwa podmenu** , gdy menu nie posiada zagnieżdżonych obiektów lub | **Nazwa podmenu** |, gdy posiada zagnieżdżone obiekty (w szczególności można je rozwinąć).
- Wybór jednoznaczny reprezentowany jest zapisem: **->** **Nazwa wyboru**. Jeżeli do wyboru nie została jeszcze przypisana żadna funkcja, po dwukropku znajduje się dopisek **funkcja nieaktywna**.

2.6.2 Dokumentacja funkcji składowych

2.6.2.1 void Interfejs::Program (Obsluga & *M*)

Metoda przetwarzająca polecenia uzytkownika. Może być wywołana dla stworzonego wcześniej obiektu typu Obsluga.

2.6.2.2 void Interfejs::Start ( )

Funkcja umożliwiająca użytkownikowi stworzenie nowego Menu od zera. Pobiera z klawiatury informacje o Podmenu o stopniu zagnieżdżenia 0, niezbędnym do utworzenia nowego obiektu typu Obsluga, a następnie wywołuje funkcję Program(Obsluga& M).