

Menu

Wygenerowano przez Doxygen 1.8.6

Pn, 16 sty 2017 22:51:10

Spis treści

1	Informacje o projekcie	1
2	Indeks hierarchiczny	3
2.1	Hierarchia klas	3
3	Indeks klas	5
3.1	Lista klas	5
4	Dokumentacja klas	7
4.1	Dokumentacja klasy Interfejs	7
4.1.1	Opis szczegółowy	7
4.1.2	Dokumentacja funkcji składowych	7
4.1.2.1	Program	7
4.1.2.2	Start	7
4.2	Dokumentacja klasy Jednoznaczny	7
4.2.1	Opis szczegółowy	8
4.3	Dokumentacja klasy Kolekcja	8
4.3.1	Opis szczegółowy	8
4.3.2	Dokumentacja konstruktora i destruktora	8
4.3.2.1	Kolekcja	8
4.4	Dokumentacja klasy Obsługa	9
4.4.1	Opis szczegółowy	9
4.4.2	Dokumentacja funkcji składowych	9
4.4.2.1	Dalej	9
4.4.2.2	DodajMenu	9
4.4.2.3	DodajWyborJedn	10
4.4.2.4	PrzypiszFunkcje	10
4.4.2.5	Rozwin	10
4.4.2.6	Usun	10
4.4.2.7	Wstecz	10
4.4.2.8	Wykonaj	10
4.4.2.9	Wypisz	10

4.4.2.10	Zwin	10
4.5	Dokumentacja klasy Podmenu	11
4.5.1	Opis szczegółowy	11
4.6	Dokumentacja klasy Wybor	11
4.6.1	Opis szczegółowy	12
4.6.2	Dokumentacja atrybutów składowych	12
4.6.2.1	stopienZagniezdzenia	12

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 jednoznanymi może kryć się stały zbiór funkcji wypisujący na konsoli komunikaty typu "Zadziałała funkcja nr 14".

Założenia

1. Menu zawsze musi się składać z co najmniej 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 następny obiekt.
4. Jeżeli [Podmenu](#) jest zwinięte, nie ma możliwości przejścia kursorem do któregośkolwiek z zagnieżdżonych w nim Wyborów. W tym celu należy je najpierw rozwinąć.

Struktura klas reprezentujących Menu

Lista klas w kolejności nadrzędności, tzn. klasa wymieniona jako pierwsza nie korzysta z żadnej z innych klas, jest za to wykorzystywana przez klasy nadrzędne, w szczególności przez klasę bezpośrednio nadrzędną, tzn. klasę o numerze 2.

1. [Wybor](#)
 - [Jednoznaczny](#)
 - [Podmenu](#)
2. [Kolekcja](#)

- 3. [Obsługa](#)
- 4. [Interfejs](#)

Rozdział 2

Indeks hierarchiczny

2.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Interfejs	7
Kolekcja	8
Obsługa	9
Wybor	11
Jednoznaczny	7
Podmenu	11

Rozdział 3

Indeks klas

3.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Interfejs	7
Jednoznaczny	7
Kolekcja	8
Obsługa	9
Podmenu	11
Wybor	11

Rozdział 4

Dokumentacja klas

4.1 Dokumentacja klasy Interfejs

```
#include <interfejs.h>
```

Metody publiczne

- void [Start](#) ()
- void [Program](#) ([Obsługa](#) &M)

4.1.1 Opis szczegółowy

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

- Poziom zagnieżdżenia reprezentowany jest ilością spacji
- Pozycja Kursora pokazywana jest symbolem ***
- Lista dostępnych poleceń wyświetlana jest bezpośrednio pod Menu.

4.1.2 Dokumentacja funkcji składowych

4.1.2.1 void Interfejs::Program (Obsługa & M)

Metoda przetwarzająca polecenia użytkownika. Może być wywołana dla stworzonego wcześniej obiektu typu [Obsługa](#).

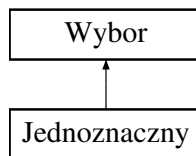
4.1.2.2 void Interfejs::Start ()

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

4.2 Dokumentacja klasy Jednoznaczny

```
#include <wybor.h>
```

Diagram dziedziczenia dla Jednoznaczny



Metody publiczne

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

Przyjaciele

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

Dodatkowe Dziedziczone Składowe

4.2.1 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.

4.3 Dokumentacja klasy Kolekcja

```
#include <kolekcja.h>
```

Metody publiczne

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

Przyjaciele

- class **Obsluga**

4.3.1 Opis szczegółowy

Klasa zaprzyjaźniona z klasą Wybór i klasami pochodnymi, obsługująca obiekty tej klasy.

4.3.2 Dokumentacja konstruktora i destruktora

4.3.2.1 Kolekcja::Kolekcja (std::string s)

Konstruktor obiektu typu **Kolekcja**. Tworzy nowy obiekt klasy **Podmenu**, zgodnie z założeniem, że każda **Kolekcja** musi posiadać jeden obiekt klasy **Podmenu**, o stopniu zagnieżdżenia 0, znajdujący się na początku listy Zbiór-Wyborow.

Parametry

s	Nazwa Kolekcji, będąca jednocześnie nazwą obiektu Podmenu o stopniu zagnieżdżenia 0
---	---

4.4 Dokumentacja klasy Obsluga

```
#include <obsługa.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 [Wypisz](#) (std::ostream &ekran)
- void [Zwin](#) ()
- void [Rozwin](#) ()
- void [Wykonaj](#) ()

Przyjaciele

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

4.4.1 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.

Klasa ma charakter uniwersalny, nie narzuca żadnego konkretnego interfejsu użytkownika. Z powodzeniem jej metody mogą być wykorzystywane przy tworzeniu różnych interfejsów.

4.4.2 Dokumentacja funkcji składowych

4.4.2.1 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, przesuw go na początek.

4.4.2.2 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

<i>s</i>	nazwa obiektu typu Podmenu
----------	--

4.4.2.3 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

<i>s</i>	nazwa obiektu typu Jednoznaczny
----------	---

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

Przypisuje do odpowiedniego pola obiektu typu [Jednoznaczny](#) wskaźnik na funkcję, którą ma wywoływać ten Wybór.

Parametry

<i>n</i>	numer funkcji, która ma zostać przypisana Wyborowi
----------	--

4.4.2.5 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.

4.4.2.6 void Obsluga::Usun ()

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

4.4.2.7 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, przesuw go na koniec.

4.4.2.8 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.

4.4.2.9 void Obsluga::Wypisz (std::ostream & *ekran*)

Metoda wykorzystywana przez przeładowany operator <<

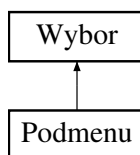
4.4.2.10 void Obsluga::Zwin ()

Metoda ukrywa wszystkie elementy Kolekcji zagnieżdżone w [Podmenu](#). Jeżeli wywołane dla typu innego niż [Podmenu](#), zgłasza wyjątek.

4.5 Dokumentacja klasy Podmenu

```
#include <wybor.h>
```

Diagram dziedziczenia dla Podmenu



Metody publiczne

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

Przyjaciele

- class **Kolekcja**
- class **Obsługa**

Dodatkowe Dziedziczone Składowe

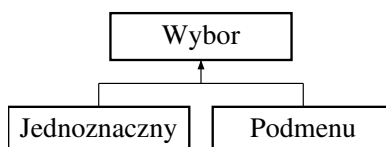
4.5.1 Opis szczegółowy

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

4.6 Dokumentacja klasy Wybor

```
#include <wybor.h>
```

Diagram dziedziczenia dla Wybor



Metody publiczne

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

Metody chronione

- int [JakiStopienZagniezdzenia](#) ()
Funkcja zwracająca wartość całkowitą, oznaczającą stopień zagnieżdżenia obiektu.
- void [Ukryj](#) ()
Zmienna logiczna czyUkryty zmienia wartość na "prawda".

- void [Pokaz](#) ()
Zmienna logiczna czyUkryty zmienia wartość na "fałsz".
- bool [hidden](#) ()
Metoda zwracająca wartość zmiennej czyUkryty.
- void [NadajStopienZagniezdzenia](#) (int n)
Metoda modyfikująca atrybut stopienZagniezdzenia.
- virtual void [Wypisz](#) (std::ostream &ekran)
Metoda używana przez zaprzyjaźnioną funkcję przeładującą operator<<.

Atrybuty chronione

- std::string **nazwa**
- int [stopienZagniezdzenia](#)
- bool [czyUkryty](#)
Zmienna logiczna: prawdziwa gdy wybór jest ukryty, np. na skutek zwinienia [Podmenu](#), w którym obiekt był zagnieżdżony.

Przyjaciele

- class **Kolekcja**
- class [Obsługa](#)
Klasa bezpośrednio obsługująca obiekty klasy Wybór.
- std::ostream & **operator**<< (std::ostream &ekran, [Wybor](#) &w)

4.6.1 Opis szczegółowy

Klasa reprezentująca pojedynczy wybór. Z jej obiektów korzysta zaprzyjaźniona klasa [Kolekcja](#). Jest to klasa najbardziej podrzędna w strukturze klas programu.

4.6.2 Dokumentacja atrybutów składowych

4.6.2.1 int Wybór::stopienZagniezdzenia [protected]

Informacja o stopniu zagnieżdżenia danego wyboru wewnątrz Kolekcji Wyborów. Zwracana przez metodę int [JakiStopienZagniezdzenia\(\)](#). Modyfikowana z zewnątrz klasy poprzez metodę void [NadajStopienZagniezdzenia\(int n\)](#).