

Zadanie składa się z dwóch części:

1. implementacja szablonu kolekcji o dostępie sekwencyjnym (listy),
2. implementacja prostej hierarchii klas, których obiekty będą przechowywane w odpowiednio skonkretyzowanej kolekcji własnego autorstwa.

Część pierwsza zadania ma dwa warianty:

Lista jednokierunkowa

Do zaimplementowania jest lista jednokierunkowa. W każdym węźle takiej listy, oprócz przechowywanego elementu jest wskazanie na kolejny element listy (nullptr w przypadku elementu ostatniego). W liście należy zaimplementować także iterator pozwalający na przechodzenie elementów w przód.

Lista dwukierunkowa

Do zaimplementowania jest lista dwukierunkowa. W każdym węźle takiej listy, oprócz przechowywanego elementu są wskazania na następny i poprzedni element listy. W liście należy zaimplementować także iterator pozwalający na przechodzenie elementów w obu kierunkach. Ciekawym pytaniem w tym wariancie jest: czy używać strażnika (sentinel)? Strażnik, to węzeł, który nie jest elementem kolekcji (jest „pusty”), ale może ułatwić implementację, jako węzeł „przed pierwszym” i „za ostatnim” elementem kolekcji.

Zostawiamy dowolność w wyborze wariantu, ale sensownie byłoby, żeby w każdym zespole projektowym były realizowane oba warianty, żeby porównać szczegóły implementacji, ilość kodu, testów, etc.

Proszę zaimplementować typowe składowe kolekcji, w szczególności wstawianie i usuwanie elementów z/do kolekcji na pozycji wskazywanej przez iterator. Potrzebne będą też implementacje konstruktora kopiującego i operatora podstawienia (także w wersji przemieszczającej). Zwróćcie szczególną uwagę na zarządzanie pamięcią.

Warto obejrzeć wystąpienie Andre Kostura *The Rule of Five* z ostatniej konferencji CppCon z serii Back to Basics (<https://www.youtube.com/watch?v=juAZDfsaMvY>)

W części drugiej jest do zaimplementowania jedna z trzech hierarchii klas:

Kształty

Ograniczmy się do prostokąta, łamanej i koła. Klasy reprezentujące te kształty dziedziczą po wspólnej klasie bazowej (Shape) i mają:

1. Metodę do liczenia powierzchni kształtu.
2. Metodę do liczenia obwodu kształtu.
3. Metodę wyznaczającą prostokąt ograniczającego na płaszczyźnie (minimalnego prostokąta, w który jest wpisany kształt).

Dziennik

Dziennik (Logger) rejestruje komunikaty różnych typów obsługujących wspólny interfejs abstrakcyjny IMessage. W interfejsie są zdefiniowane przynajmniej:

1. Metoda write przekazująca treść wiadomości na wskazany strumień wyjściowy;
2. Metoda severity, której wartością jest informacja o istotności wiadomości (np. Trace, Debug, Info, Warning, Error, Fatal)

Element interfejsu (Widget)

Widget jest klasą bazową dla wielu elementów interfejsu graficznego użytkownika. Interfejs klasy definiuje przynajmniej:

1. Metodę pobierania współrzędnych (x, y) elementu interfejsu, a dokładniej jego lewego górnego rogu, w oknie.
2. Metody pobierania szerokości i wysokości elementu w pikselach.
3. Metodę pobierania etykiety elementu.

Klasy pochodne po Widget to: przycisk (Button), InBox (edytor to wpisywania wartości tekstowej), Slider (czyli ślizgać, czyli suwak pozwalający wybrać wartość między minimum i maksimum).

Ponownie, nie wskazujemy, które zadanie należy wykonać, ale zadбайcie o to, żeby wspólnicy projektowi realizowali różne zadania.