Laboratorium – C++11

## rdzeń języka I ulepszenie funkcjonalności

### 6 kwietnia 2016

1. Wstęp

Podczas laboratorium zostanie sprawdzona nabyta wiedza zdobyta podczas seminarium o tej samej tematyce. Wszystkie zadania będą związane z nowymi funkcjonalnościami języka C++11.

1. Zadanie I
   1. Struktura zadania

|  |  |
| --- | --- |
| Main.cpp | Plik z zadaniem |
| Zad1.h | Plik z rowiązaniem |

* 1. Cel zadania

Student ma skonstruować klase *Join* przy użyciu **listy inicjalizacyjnej**. Klasa ta będzie potrafiłą łączyć wektory które zostaną jej podane wewnątrz listy.

* 1. Potrzebna wiedza
     + std::initializer\_list<T>

Student powinien potrafić posługiwać się tą klasą w celu użycia jej w konstruktorze. Musi znać takie metody jak **begin**(), **end**() i **size**() występujące w tej klasie.

* 1. Przykładowe rozwiązanie zadania

Na początku zostaje zdefiniowana klasa szablonowa Join której parametr szablonu posłuży do okreslenia typu przyjmowanych wartości przez wektory w liście inicjalizacyjnej. Join będzie zawierać jedną składową prywatną która będzie wektorem przechowywującym złączone wartości.

Najważniejszą częscią zadania jest napisanie konstruktora który będzie potrafił łączyć wszystkie elementy wielu wektorów.

Przykładowy konstruktor :

|  |
| --- |
| Join**(**std**::**initializer\_list**<**std**::**vector**<**T**>>** list**)**  **{**  int maxSize **=** 0**;**  **for** **(**auto vec **:** list**)**  **{**  maxSize **+=** vec**.**size**();**  **}**  joined**.**reserve**(**maxSize**);**  int s **=** 0**;**  **for** **(**auto vec **:** list**)**  **{**  joined**.**insert**(**joined**.**begin**()** **+** s**,** vec**.**begin**(),** vec**.**end**());**  s **+=** vec**.**size**();**  **}**  **}** |

Ostatnią częścią zadania jest zdefiniowanie operatora który będzie potrafił zwracać nasz wektor :

|  |
| --- |
| **operator** std**::**vector**<**T**>()**  **{**  **return** joined**;**  **}** |

1. Zadanie II
   1. Struktura zadania

|  |  |
| --- | --- |
| Main.cpp | Plik z zadaniem |
| Zad2.h | Plik z rowiązaniem |
| Defines.h | Plik ze stałą określająca czy powinny wystąpić błędy kompilacji |

* 1. Cel zadania

Student ma zdefiniować klase Base która posłuży jako klasa bazowe dla klasy Bar. Klasa Bar jak i funkcja main będa sprawdzać czy klasa Base została skonstruowana w taki sposób aby umożliwić programowi poprawną kompilacje.

Dodatkowo student ma za zadanie zdefiniować funkcje szablonową test która informuje o tym że rzuca wyjątki jedynie wtedy kiedy przyjmuje argumenty typu const char \*

* 1. Potrzebna wiedza
     + Słowo kluczowe default

Jest potrzebne do zdefiniowania chronionego (protected) operatora przypisania i domyślnego konstruktora.

* + - Słowo kluczowe delete

Posłuży do usunięcia definicji konstruktora kopiującego

* + - Słowo kluczowe noexcept

Musi zostać określone przy metodzie **fun**() wewnątrz klasy Base w celu poprawnego jej nadpisania przez klase pochodną

Dodatkowo jest wykorzystywane przez funkcje **test**() w połączeniu z std::is\_same<T1, T2> w celu sprawdzenia czy powinien zostać rzucony wyjątek gdy do funkcji zostanie przekazany typ const char \*

* + - Słowo kluczowe std::is\_same

Jego wykorzystanie zostało omówione we wcześniejszym paragrafie

* + - Słowo kluczowe override

Nie jest bezpośrednio wykorzystywane przez studenta. Lecz musi zostać zrozumiana jego rola w pisaniu klas pochodnych w celu poprawnego wykonania zadania

* + - Słowo kluczowe final

Jest wykorzystywane w metodzie **pun**(). Metoda ta musi zostać zadeklarowana jako finalna w klasie Base w celu poprawnej kompilacji programu.

* 1. Przykładowe rozwiązanie zadania

|  |
| --- |
| class Base  **{**  public **:**  Base**()** **=** **default;**  Base**(**const Base **&)** **=** **delete;**  virtual void fun**()** const noexcept **{}**  virtual void pun**()** final **{}**  protected **:**  Base**&** **operator** **=(**const Base **&)** **=** **default;**  **};**  template**<**typename T**>**  void test**(**T arg**)** noexcept**(!**std**::**is\_same**<**T**,** const char **\*>::**value**)** **{}** |