

Politechnika  Białostocka

Wydział Elektryczny

Katedra Telekomunikacji i Aparatury Elektronicznej

**Instrukcja
do pracowni specjalistycznej z przedmiotu**

Obiektowe programowanie aplikacji

Kod przedmiotu: TS1C410201

**Opracował
dr inż. Adam Nikołajew**

Białystok 2017 r.

1. Cel zajęć

Przypomnienie i rozszerzenie umiejętności programowania w języku C++. Opanowanie podstawowych pojęć z zakresu programowania obiektowego. Umiejętność analizy programów w języku C++.

Temat 1: Typ wyliczeniowy, wskaźniki (2 godz.)

1. Napisać program w języku C++ z wykorzystaniem typu wyliczeniowego. Sprawdzić różne mechanizmy nadawania wartości elementom listy wyliczeniowej. Utworzyć tablicę elementów tego typu. Sprawdzić możliwości wykonywania działań na elementach typu wyliczeniowego.
2. W programie lub sekwencji krótkich programów napisanych w C++ sprawdzić działanie wskaźników. W programach należy przeanalizować:
 - a) mechanikę ruchu wskaźnika (np. wykorzystując elementy tablicy odpowiedniego typu: char, int, double)
 - b) działania, które można wykonywać na wskaźnikach
 - c) rozmiary wskaźników do różnych typów danych
 - d) działanie modyfikatora const w odniesieniu do wskaźników
 - e) rzutowanie const_cast w odniesieniu do wskaźników
 - f) zastosowanie wskaźników do tablicy łańcuchów (najlepiej na tablicy typu char).

Temat 2: Funkcje. Przekazywanie argumentów. Przeładowanie funkcji. Szablony funkcji (2 godz.)

1. Napisać program w języku C++ realizujący prostą funkcję. Sprawdzić różne mechanizmy przekazywania argumentów oraz zwracania wartości funkcji (wartość, wskaźnik, referencja).
2. Napisać program w C++, w którym zostanie zastosowane przeładowanie funkcji.
3. Zmodyfikować program z punktu 2. w taki sposób, aby zastosować szablon funkcji.
4. Zrealizować prosty program z wykorzystaniem makro.

Temat 3: Struktura, klasa (2 godz.)

1. Napisać program w języku C++ wykorzystujący prostą strukturę. Zdefiniować składowe oraz funkcje składowe struktury. Sprawdzić różne mechanizmy nadawania wartości składowych oraz dostęp do składowych struktury. Sprawdzić sposoby definiowania funkcji składowych wewnątrz i poza ciałem badanej struktury.
2. Zmodyfikować program z punktu 1, zamieniając strukturę na klasę.

Temat 4: Konstruktor obiektu klasy. Składowe statyczne klasy. Działanie funkcji składowych klasy i zaprzyjaźnionych (4 godz.)

Napisać program w języku C++ w którym zostanie stworzona klasa Samochod, opisująca zawartość komis samochodowego.

Każdy obiekt klasy powinien posiadać przynajmniej 5 składowych o różnym poziomie dostępu (public lub private). (np. nr VIN, marka, model. Przebieg rok produkcji itp.)

Stworzyć konstruktory (również konstruktor domyślny), tworzące poszczególne obiekty, stosując wszystkie podane na wykładzie konstrukcje. Zdefiniować funkcje wewnętrzne klasy oraz globalne, odczytujące lub wczytujące poszczególne składowe

obiektów klasy. Argumentami funkcji składowych powinny być: obiekt klasy, wskaźnik oraz referencja do obiektu klasy. Należy również sprawdzić działanie wskaźnika this.

Przeanalizować działanie destruktora obiektu klasy.

Wprowadzić składową statyczną klasy i funkcję statyczną.

Sprawdzić bezpośrednio tworzenie obiektów klasy oraz tworzenie ich za pomocą tablicy wskaźników do obiektów. Przy zastosowaniu wskaźników przeanalizować metodę (new-delete).

Wprowadzić funkcję zaprzyjaźnioną klasy i przeanalizować jej działanie.

Temat 5: Przeciążenie funkcji i operatorów. Szablony funkcji (2 godz.)

Napisać program w języku C++ w którym zostanie stworzona klasa Samochod, opisująca zawartość komisu samochodowego.

Każdy obiekt klasy powinien posiadać przynajmniej 5 składowych o różnym poziomie dostępu (public lub private). (np. nr VIN, marka, model, przebieg, cena, rok produkcji itp.)

Stworzyć konstruktory (również konstruktor domyślny), tworzące poszczególne obiekty. Zdefiniować przeładowane funkcje wewnętrzne klasy oraz globalne, działające na poszczególnych składowych obiektów klasy. Argumentami funkcji składowych powinny być: obiekt klasy, wskaźnik oraz referencja do obiektu klasy.

Stworzyć przeładowane operatory inkrementacji i dekrementacji, zmieniające cenę samochodu.

Stworzyć szablon funkcji zewnętrznej, działającej na wybraną składową obiektu klasy.

Temat 6: Dziedziczenie (2 godz.)

Napisać program w języku C++ w którym zostanie stworzona klasa bazowa oraz 2 klasy pochodne.

Obiekt klasy bazowej powinien posiadać przynajmniej 3 składowe o różnym poziomie dostępu (public, protected lub private). Obiekty klas pochodnych powinny posiadać co najmniej 3 składowe, niebędące składowymi klasy bazowej. Dostęp do nich powinien być podobnie zróżnicowany, jak w klasie bazowej. Klasy należy wyposażyć w odpowiednie funkcje składowe, pozwalające na dostęp do składowych z poziomu funkcji main() oraz obiektów klasy bazowej i klas potomnych. W procesie tworzenia obiektów poszczególnych klas najwygodniej jest użyć konstruktorów i destruktorów domyślnych.

Sprawdzić kolejność działania konstruktorów i destruktorów przy tworzeniu i likwidacji obiektów klas pochodnych.

Sprawdzić dostęp do poszczególnych składowych i funkcji klasy bazowej oraz klas potomnych w zależności od rodzaju dziedziczenia (public, protected, private).

Stworzyć wskaźniki do obiektów poszczególnych klas, a następnie sprawdzić możliwości tworzenia obiektów klas potomnych jako wskaźników do klasy bazowej i odwrotnie. Należy zwrócić szczególną uwagę na rozmiary poszczególnych obiektów.

Temat 7: Przesłanianie funkcji w klasie potomnej. Rzutowanie(4 godz.)

Napisać program w języku C++ w którym zostanie stworzona klasa bazowa oraz 2 klasy pochodne.

Obiekt klasy bazowej powinien posiadać przynajmniej 2 funkcje składowe o dostępie publicznym. Obiekty klas pochodnych powinny posiadać co najmniej 2 funkcje składowe o tych samych nazwach i dostępie publicznym. W procesie tworzenia obiektów poszczególnych klas najwygodniej jest użyć konstruktorów i destruktorów domyślnych.

Zbadać mechanizm przesłaniania funkcji klasy bazowej w klasie potomnej w następujących kombinacjach:

- a) tworzone są obiekty klasy bazowej i pochodnej
- b) obiekty klasy bazowej i potomnej są tworzone jako wskaźniki do klasy bazowej
- c) obiekty klasy bazowej są tworzone jako wskaźniki do klasy bazowej, zaś obiekty klasy potomnej – jako wskaźniki do klasy potomnej.

Powtórzyć punkt poprzedni w przypadkach, gdy funkcje są funkcjami wirtualnymi. Należy rozpatrzyć następujące przypadki:

- a) funkcja wirtualna w klasie bazowej
- b) funkcja wirtualna w klasie potomnej
- c) funkcje wirtualne w obu klasach.

Stworzyć wskaźniki do obiektów klasy bazowej jako wskaźniki do klasy potomnej. Zastosować rzutowanie `static_cast`, `dynamic_cast` oraz `reinterpret_cast`. Sprawdzić, jak są przesłaniane funkcje składowe klasy bazowej w tych przypadkach.

Sprawdzić działanie czystej funkcji wirtualnej.

Temat 8: Dziedziczenie wielokrotne. Dziedziczenie wirtualne. Klasy abstrakcyjne (4 godz.)

Napisać program w języku C++ w którym zostaną stworzone 2 klasy bazowe oraz klasa pochodna.

Obiekty klas bazowych powinny posiadać przynajmniej 2 funkcje składowe o dostępie publicznym, przy czym jedna z nich powinna być identyczna w obu klasach. Obiekty klasy pochodnej powinny posiadać co najmniej 1 funkcję składową niewystępującą w klasach bazowych dostępie publicznym. Identyczne warunki powinny spełniać również składowe klas bazowych i klasy pochodnej.

Zbadać mechanizm tworzenia i likwidacji obiektów klasy pochodnej (kolejność działania konstruktorów i destruktorów). Przeanalizować sposób tworzenia konstruktorów klasy pochodnej i zasady wywoływania w nich konstruktorów klas bazowych. Klasa pochodna powinna posiadać co najmniej 2 różne konstruktory i konstruktor domyślny.

Zbadać mechanizm dostępu do funkcji i składowych klasy potomnej, dziedziczonych z klas bazowych w następujących kombinacjach:

- d) tworzone są obiekty klasy pochodnej
- e) obiekty klasy potomnej są tworzone jako wskaźniki do jednej z klas bazowych
- f) obiekty klasy potomnej tworzone jako wskaźniki do klasy potomnej.

Przeanalizować działanie funkcji wirtualnych oraz jawne wywołanie składowej lub funkcji składowej.

Do stworzonego poprzednio programu dodać wspólną klasę bazową, z której będą dziedziczyły klasy bazowe z poprzedniej części ćwiczenia. Klasa ta powinna mieć co najmniej 1 składową i 1 funkcję składową o dostępie publicznym.

Zbadać mechanizm tworzenia i likwidacji obiektów klasy pochodnej (kolejność działania konstruktorów i destruktorów) w 2 przypadkach:

- a) dziedziczenie z głównej klasy bazowej nie jest wirtualne
- b) dziedziczenie z głównej klasy bazowej jest wirtualne.

W tym przypadku najwygodniej jest użyć konstruktorów domyślnych.

Zbadać mechanizm dostępu do funkcji i składowych klasy potomnej, dziedziczonych z klas bazowych. Przeanalizować działanie funkcji wirtualnych oraz jawne wywołanie składowej lub funkcji składowej.

Powtórzyć punkt poprzedni w sytuacji, gdy główna klasa bazowa jest klasą abstrakcyjną (posiada co najmniej 1 czystą funkcję wirtualną).

Temat 9: Standardowa biblioteka wzorców STL (4 godz.)

Napisać serię programów w języku C++ w których zostaną stworzone obiekty następujących klas:

1. `std::string`
2. `std::vector`
3. `std::list`
4. `std::set`
5. `std::multiset`
6. `std::map`
7. `std::multimap`

Zbadać mechanizmy tworzenia i dopisywania elementów do poszczególnych kontenerów (obiektów klasy), stosując metody przedstawione na wykładzie.

Przeanalizować działanie iteratorów w obiektach analizowanych klas.

Zbadać działanie wybranych algorytmów STL (`find`, `transform`).

Sprawozdania z poszczególnych tematów powinny zawierać:

- a) wydruki programów wraz ze zrzutami z wiersza poleceń lub okna wyjściowego w przypadku zamierzonego błędu w programie
- b) wnioski z napisanych programów, dotyczące zagadnień, ćwiczonych na zajęciach.

Wymagania BHP

Zgodnie z podanymi na pierwszych zajęciach i potwierdzonymi przez studentów zasadami obowiązującymi w pomieszczeniu, w którym odbywają się ćwiczenia. Stosowny regulamin BHP jest też wywieszony w pomieszczeniu laboratorium.

Literatura

- [1] Grębosz J., *Symfonia C++ standard : programowanie w języku C++ orientowane obiektowo t. 1, 2*, Kraków : Wydaw. Edition 2000, 2008.
- [2] Stroustrup B., *Programowanie : teoria i praktyka z wykorzystaniem C++*, Gliwice : Helion, 2013.
- [3] Prata S., *Język C++. Szkoła programowania*, Gliwice, Helion, 2013.
- [4] Grębosz J., *Opus magnum C++11. Programowanie w języku C++*. Gliwice, Helion, 2017.
- [5] Liberty J., Siddhartha R., Jones B. L., *C++ dla każdego*, Wyd. II, Gliwice, Helion, 2009