

Zadanie 1

1. Czym jest kompilator i linker?

Kompilator jest programem który tłumaczy kod źródłowy napisany przez nas do języka assembler, a następnie do postaci zrozumiałej dla komputera (maszyny cyfrowej), czyli do postaci ciągu zer i jedynek które sterują pracą poszczególnych elementów komputera. Jest to program uruchamiany po etapie kompilacji jednego lub kilku plików źródłowych, po procesie kompilacji. Linker łączy wszystkie skompilowane pliki źródłowe i inne funkcje bibliotek które były użyte w naszym programie, a nie były zdefiniowane w naszych plikach źródłowych. Wywoływany jest on na ogół automatycznie przez kompilator, przez co nie musimy się martwić jego obsługą.

2. Co to jest "include guard" lub inaczej "header guard"?

Include guard to technika pozwalająca na upewnienie się, każdy plik zawierający definicję jakiegokolwiek elementu będzie przetwarzany tylko raz, tak, aby nie powodować błędu powtórzenia tego samego kodu.

3. Do czego służy #pragma once?

Jest to dyrektywa preprocesora o podobnej funkcji co „include guard” czyli zapobiegająca wielokrotnemu dołączaniu tego samego pliku nagłówkowego.

4. Do czego służy słowo kluczowe "auto"?

Słowo kluczowe auto służy do automatycznego określania typu zmiennej na podstawie przypisanej jej wartości.

5. Omów na czym polega programowanie obiektowe?

Programowanie obiektowe – paradygmat programowania, w którym programy definiuje się za pomocą obiektów – elementów łączących stan (czyli dane, nazywane najczęściej polami) i zachowanie (czyli procedury, tu: metody). Obiektowy program komputerowy wyrażony jest jako zbiór takich obiektów, komunikujących się pomiędzy sobą w celu wykonywania zadań.

Podejście to różni się od tradycyjnego programowania proceduralnego, gdzie dane i procedury nie są ze sobą bezpośrednio związane. Programowanie obiektowe ma ułatwić pisanie, konserwację i wielokrotne użycie programów lub ich fragmentów.

Największym atutem programowania, projektowania oraz analizy obiektowej jest zgodność takiego podejścia z rzeczywistością – mózg ludzki jest w naturalny sposób najlepiej przystosowany do takiego podejścia przy przetwarzaniu informacji.

6. Omów wpływ modyfikatorów dostępu użytych podczas dziedziczenia na widoczność pól i metod.

Modyfikatory dostępu do klas dzieli się na:

protected – zmienne/metody dostępne są z poziomu klasy w której się znajdują lub poziomu klas pochodnych

public – zmienne/metody dostępne są z każdego miejsca w kodzie

private – zmienne/metody dostępne są tylko z poziomu klasy w której się znajdują

7. Czym różni się metoda "virtual" od "pure virtual"?

Metoda „pure virtual” w przeciwieństwie do metody „virtual” kończy się zapisem „=0” oraz nie posiada ciała.

8. Co to jest "vtable"?

vtable jest tablicą przechowywującą wskaźniki do wszystkich metod wirtualnych danej klasy.

9. Opisz różnicę pomiędzy destruktorem a wirtualnym destruktorem. Jaki należy stosować częściej i dlaczego?

Wywołanie destruktora wirtualnego w przeciwieństwie do zwykłego powoduje również wywołanie destruktora klas pochodnych dzięki czemu można uniknąć problemów z wyciekami pamięci.

Zadanie 2

1. Czym jest stos (stack), czym jest sterta (heap)?

Stos: to struktura danych oparta na FIFO (First In First Out) , oznacza to że elementy przychodzące wcześniej, wcześniej też opuszczają tę strukturę.

Sterta: inaczej kopiec to obszar pamięci, udostępniony na wyłączność uruchomionemu programowi (procesowi). Przechowuje się tam zmienne dynamiczne.

2. Czym jest wskaźnik?

Wskaźnik jest to zmienna przechowująca adres innej zmiennej. Inaczej, jest to zmienna wskazująca na adres innej zmiennej. Skoro jest zmienną więc posiada także swój adres.

3. Czym jest wskaźnik na wskaźnik i jakie może mieć zastosowania?

Jest to zmienna przechowująca adres zmiennej która z kolei przechowuje adres jeszcze innej zmiennej, znajduje zastosowanie na przykład w implementacji tablic wielowymiarowych.

4. Na czym polega arytmetyka wskaźników?

Arytmetyka wskaźników wykorzystuje fakt że dodanie do wskaźnika liczby np. 2 spowoduje przesunięcie się wskaźnika o wartość $2 * \text{wielkość elementu na który wskazuje wskaźnik}$.

5. Omów konwersje typów.

Jest to konstrukcja programistyczna umożliwiająca traktowanie danej pewnego, konkretnego typu, jak daną innego typu, lub taką zmianę tej danej albo jej reprezentacji w pamięci operacyjnej, aby wartość tej danej, odpowiadała według przyjętych kryteriów odwzorowania, danej innego, wybranego typu.

6. Omów rzutowanie typów.

Rzutowanie to przeprowadzona w sposób jawny konwersja typu. Od niejawnej konwersji różni się tym, że stosujemy ją gdy może nastąpić utrata informacji, np. gdy rzutujemy liczbę zmiennoprzecinkową na całkowitą to część po przecinku zostanie utracona. Dlatego musimy zaznaczyć w programie, że chcemy dokonać konwersji i wiemy co robimy.

7. Wymień sposoby rzutowania typów w C++.

const_cast :

rzutowanie ze zmiennych z modyfikatorem const i volatile na zmienne bez tych modyfikatorów.

static_cast :

rzutowanie w którym typ obiektu musi być znany w momencie kompilacji.

dynamic_cast :

rzutowanie wskaźników na obiekty. Umożliwia sprawdzenie, czy można bezpiecznie przypisać adres obiektu do wskaźnika danego typu. Typ obiektu jest dynamicznie określany, w czasie wykonywania programu. Jest do tego używany mechanizm dynamicznej identyfikacji typu RTTI (ang. runtime type identification).

reinterpret_cast :

niebezpieczne rzutowania, które zmieniają zupełnie sens interpretacji bitów w zmiennych. Rzutowanie to nie pozwala na zdjęcie modyfikatora const.

8. Omów różnicę pomiędzy pętlą "while" a "do ... while".

Pętle te wykonują się dopóki spełniony jest dany warunek, różnicą jest to że pętla do ... while wykonuje blok instrukcji jeszcze przed sprawdzeniem warunku co oznacz że zawsze zostanie wykonana co najmniej raz.

9. Opisz różnicę pomiędzy statyczną i dynamiczną alokacją pamięci.

Alokacja pamięci to przydzielenie obiektowi pewnego obszaru pamięci oraz nadanie mu wartości początkowej. System operacyjny przydziela pulę wolnej pamięci, która może zostać wykorzystana przez program. Alokacja statyczna następuje przed rozpoczęciem wykonywania programu natomiast dynamiczna następuje w fazie wykonywania programu z czym wiąże się konieczność zwolnienia pamięci przez programistę.

Zadanie 3

1. Co to jest przestrzeń barw RGB?

Jest to model przestrzeni barw. Jego nazwa powstała ze złożenia pierwszych liter angielskich nazw barw: R – red (czerwonej), G – green (zielonej) i B – blue (niebieskiej), z których model ten się składa. Jest to model wynikający z właściwości odbiorczych ludzkiego oka, w którym wrażenie widzenia dowolnej barwy można wywołać przez zmieszanie w ustalonych proporcjach trzech wiązek światła o barwie czerwonej, zielonej i niebieskiej.

2. Co to jest przestrzeń barw HSV?

Jest to model opisu przestrzeni barw. Model nawiązuje do sposobu, w jakim widzi ludzki narząd wzroku, gdzie wszystkie barwy postrzegane są jako światło pochodzące z oświetlenia. Według tego modelu wszelkie barwy wywodzą się ze światła białego, gdzie część widma zostaje wchłonięta, a część odbita od oświetlanych przedmiotów. Symbole w nazwie modelu to pierwsze litery nazw angielskich dla składowych opisu barwy: H – odcień światła (ang. Hue) wyrażona kątem na kole barw przyjmująca wartości od 0° do 360° . Model jest rozpatrywany jako stożek, którego podstawą jest koło barw.

3. Co to jest filtr 2D?

Jest to filtr operujący na dwuwymiarowej tablicy którego działanie polega na modyfikacji poszczególnych elementów obrazu w zależności od stanu ich samych i stanu ich otoczenia.

4. Jak wygląda reprezentacja obrazu w pamięci?

W przypadku obrazu rastrowego obraz w pamięci jest reprezentowany jako bitmapa czyli dwuwymiarowa tablica pikseli, natomiast w przypadku grafiki wektorowej zapis obrazu oparty jest na formułach matematycznych.