[Szersze przedstawienie pojęcia zmiennej]

W pamięci komputera możemy przechowywać różnego rodzaju dane. Najczęściej spotykane to liczby naturalne, liczby całkowite, liczby wymierne, niewymierne, które razem tworzą zbiór liczb rzeczywistych. Oprócz liczb możemy także wyróżnić znaki alfanumeryczne, oraz napisy tworzony z tych znaków. Wszystkie te dane w zależności od potrzeby mogą być przechowywane w różnych miejscach. Chyba najbardziej znanym rodzajem pamięci komputera jest dysk twardy. Charakteryzuje się bardzo dużą pojemnością. Innym znanym rodzajem pamięci jest pamięć operacyjna, pamięć ram. Jest ona mniej pojemna lecz jest również znacznie szybsza. W komputerach możemy wyróżnić co najmniej jeszcze jeden rodzaj pamięci. Jest to pamięć rejestrów procesora. Jest to najszybszy rodzaj pamięci w komputerze. Oczywiście i ten rodzaj pamięci podlega prawidłowością świata informatyki. Ta pamięć jest najszybsza, ale jest też najmniej pojemna. No i oczywiście jest też najdroższa w przeliczeniu na jednostkę pamięci.

To właśnie dostępność i cena pamięci wymogła na programistach specjalne podejście do niej. Wydaje się, że nasze komputery dysponują gigabajtami pamięci, więc nie powinniśmy się za bardzo przejmować oszczędzaniem jej. Ten tok myślenia jest jednak błędny, gdyż tak naprawdę dysponujemy tylko niewielką ilością pamięci szybkiej. Dlatego dla komputera ogromną różnicą jest dodać dwie kilkudziesięciocyfrowę liczby, a dodać dwie liczby krótkie. Dla przykładu. Te dwie duże liczby są za duże, by w całości zmieścić się w pamięci rejestrów procesora. Przypomnę tutaj tylko, że rejestry procesora pośredniczą w niemalże każdej operacji wykonywanej na komputerze. By dodać duże liczby, komputer musi wczytywać je do rejestrów na raty, małymi częściami. Największym obciążeniem tutaj, nie jest to, że musi wczytywać kilka części tych liczb, lecz to, że wczytywanie odbywa się z pamięci znacznie wolniejszej. Drugie działanie, wykona się błyskawicznie, gdyż wszystkie potrzebne dane będą mogły zostać wczytane do rejestrów w całości za jednym razem.

Dane w programach są przechowywane w zmiennych. Spotkaliśmy się już z tym pojęciem pisząc poprzedni program. Z uwagi na wydajność obliczeń komputera, nie wszystkie zmienne są tej samej wielkości.

Jeżeli mamy jakąś liczbę, możemy zapisać ją do zmiennej odpowiedniego typu.

Jeżeli np. musimy zapisać mniejszą liczbę do jakiejś zmiennej, to dobierając rozmiar zmiennej powinniśmy wziąć pod uwagę rozmiar samej liczby. Oczywiście liczba 54 bez problemu zmieściłaby się w zmiennej w które obecnie jest nasza druga, dłuższa liczba. Możemy jednak zaoszczędzić trochę miejsca i zapisać tą liczbę w mniejszej zmiennej.

Mówimy, że zmienna do przechowywania liczb całkowitych jest typu całkowitoliczbowego. Jednakże macierzystym językiem infotmatyki jest oczywiście angielski, dlatego mówimy że taka zmienna jest typu int. Int to skrót od angielskiego tłumaczenia liczby całkowitej: integer. Podobnie sprawa ma się z naszą drugą zmienną. Podsumowując… zmienna a jest typu int, a zmienna b jest typu short int.

Jeżeli chcielibyśmy do zmiennej zapisać nową liczbę, to jej stara wartość zostanie bezpowrotnie stracona.

[Wprowadzenie pojęcia stałej]

Samo pojęcie zmiennej, pochodzi od słowa „zmiana”. Słowo „zmienna” tak silnie podkreśla zmienną naturę zmiennej, dlatego, że jej właściwości wcale nie są takie powszechne wśród wszystkich typów danych. Istnieją bowiem takie typy danych, których nie można zmieniać po zainicjalizowaniu daną wartością. By stworzyć taki typ danych podczas deklaracji, przed nazwą typu dopisujemy modyfikator const. Tak stworzony typ danych nazywamy stałą. Do stałej możemy przypisać wartość tylko raz. W późniejszym czasie, możemy sobie na jej wartość tylko popatrzeć. Później nie da się jej już zmodyfikować.

Do czego mogą nam się przydać stałe? W takiej stałej możemy np. zapisać sobie liczbę pi. Będziemy wówczas mieli pewność, że nawet przez nie uwagę nie zmienimy jej wartości. Kompilator po prostu na to nie pozwoli i zasygnalizuje błąd.

[Tabelka z typami]

[TYPY CAŁKOWITE - tabelka]

Poznaliśmy już dwa typy danych które mogą przechowywać liczby całkowite. Poznajmy kolejne dwa. Są nimi long int oraz long long int. Spójrzmy na tą **tabelkę**:

[Prezentacja, tabelka]

[TYPY RZECZYWISTE- tabelka]

Oczywiście nie z samych liczb całkowitych składa się świat matematyki. Dlatego do dyspozycji jest też parę typów reprezentujących liczby rzeczywiste.

Tymi typami są float, double, long double.

Pamiętaj jeszcze o jednym. Wpisując do kodu, oraz przez konsolę liczbę rzeczywistą, jako separatora używaj zawsze kropki, a nie przecinka. Czyli nie wpisuj 16 przecinek 5, lecz 16 kropka 5. Przyzwyczajenie może wielokrotnie spowodować, że tak jak każdy Polak będziesz chciał używać w takich sytuacjach przecinka, jednakże jak już wspomniałem języki programowania mają dużo wspólnego z krajami anglojęzycznymi, a w nich jako separatora w liczbach rzeczywistych używa się właśnie kropki.

[TYPY ZNAKOWE – tabelka]

Oprócz typów liczbowych mamy też typy znakowe. Jak się domyślasz, można w niej przechowywać znaki, takie jak litera A, znak dolara, czy znak gwiazdki. Lecz tak naprawdę w zmiennych znakowych nie są przechowywane żadne znaki. Komputer to przecież prosta maszyna, która wszystko przechowuje w postaci liczb. Dla przykładu, zmienna **char** służy do przechowywania liczb całkowitych, które reprezentują znaki z tablicy ASCII. Gdy będziemy chcieli zapisać tam znak małej litery a, to kompilator najpierw sprawdzi kod tego znaku, a następnie do zmiennej zapisze odpowiednią wartość. W przypadku małej litery a, będzie to 97. Gdy będziemy chcieli wypisać znak to nastąpi sytuacja odwrotna i kod najpierw zostanie dopasowany do odpowiedniego znaku i dopiero wówczas wyświetlony. Teoretycznie użytkownik mógłby o tym nie wiedzieć, jednakże fakt że typ znakowy możemy traktować jako jednobajtowy typ całkowitoliczbowy, stanowi dla nas ważną informację. W krótce zobaczymy, że to naprawdę dobra wiadomość ☺

Jest jeszcze jeden bardzo pożyteczny typ. Jest to typ logiczny bool. Zmienne tego typu mogą posiadać tylko dwa stany. Prawdę lub fałsz. Ten typ jest także najmniejszym typem, gdyż zajmuje tylko 1 bit.

[Literały znakowe]

Literałem znakowym jest np. liczba 5 wpisana jawnie w kodzie. Literałem znakowym jest także liczba 4.7, litera ‘a’ objęta apostrofami, słowem są to wszystkie stałe dosłowne pojawiające się w programach. Możemy oczywiście wyróżnić stałe dosłowne typu całkowitego, typu rzeczywistego, znakowego, czy logicznego.

Tak więc jeżeli w kodzie spotkamy

[TODO]

[limits.h]

Jeżeli używasz innego kompilatora, to może się zdarzyć, że zakres zmiennych będzie trochę się różnił od tego tutaj przedstawionego. Informacji na ten temat możesz szukać w pliku nagłówkowym limits.h, który powinien się znajdować w jednym z katalogów Twojego kompilatora.

[słowa kluczowe]

Pozostaje jeszcze tylko wyjaśnić, jak nie możemy nazywać zmiennych.

Nazwa zmiennej może się składać tylko i wyłącznie z ciągu liter, cyfr i znaków podkreślenia, z zastrzeżeniem, że może się zaczynać cyfrą. Oprócz tego nazwa zmiennej nie może być żadnym ze słów kluczowych w języku C++.

Oto lista tych słów kluczowych języka C++ - zakazanych nazw dla zmiennych

**asm auto bool break case catch char class const const\_cast continua default delete do double dunamic\_cast else enum explicite export extern false float for friend goto if inline int long mutable namespace new operator private protected public register reinterpreter\_cast return short signed sizeof static static\_cast struc switch template this throw true try typedef typeid typename union unsigned using virtual void volatile wchar\_t while**

**and and\_eq bitand bitorcompl not not\_eq or or\_eq xor xor\_eq**

[Poprawienie programu z poprzedniego rozdziału]

A teraz czas trochę po programować.

Teraz gdy już znamy więcej typów zmiennych możemy udoskonalić nasz poprzedni program pytający o wiek, by mógł pracować również na liczbach rzeczywistych.

Wystarczy tylko w jednym miejscu zmienić int na jeden z typów rzeczywistych, np. float… i już. Dzięki tej zmianie, teraz użytkownik może podać jako swój wiek liczbę 16.5. Jeszcze ze szkoły podstawowej wiemy, że zbiór liczb całkowitych należy do zbioru liczb rzeczywistych. Co za tym idzie, użytkownik nadal może podać jako swój wiek liczbę całkowitą. To jest zgodne z regułami matematyki. Jednakże, jeżeli cofniemy zmiany jakie wprowadziliśmy i podczas działania programu wpiszemy zamiast oczekiwanej liczby całkowitej liczbę rzeczywistą, to program zapiszę do zmiennej tylko jej część całkowitą, porzucając część ułamkową. Zwróć uwagę na to, że tutaj nie występuję żadne zaokrąglenie. Zarówno liczba 16.3 jak i 16.9 zostaną zapisane jako 16.

Żeby sobie to wszystko utrwalić napiszmy program w którym będziemy wykonywać różne operacje na zmiennych i stałych.

[Program – zmienne.cpp]

Mam nadzieję, że zdążyłeś już się nauczyć jak wygląda szkielet programu. Jeżeli nie, to mam nadzieję, że nie kopiujesz go za każdym razem . Przepisuj go ręcznie pisząc kolejne programy a nie wklejaj, wówczas bez problemu to zapamiętasz i będziesz już mógł napisać program całkowicie sam.

Najpierw przypomnijmy sobie jak się definiowało zmienne. Najpierw typ zmiennej, w tym przypadku int, następnie nazwa zmiennej.

Mam nadzieję, że pamiętasz że nowo stworzona zmienna ma w sobie zapisaną jakąś wartość, jakiegoś śmiecia. Żeby się go pozbyć, możemy w kolejnym kroku nadać tej zmiennej nową wartość. Czasami zdarza się jednak tak, że już podczas deklaracji wiemy jaką wartość chcemy nadać od zmiennej.

Wtedy nie musimy tego robić w dwóch krokach. Stwórzmy nową zmienną typu long double i od razu nadajmy jej wartość. Proste, prawda?

Nadanie wartości zmiennej w momencie jej narodzin nazywa się inicjalizacją zmiennej.

Znajomość tego pojęcia jest niezbędna jeżeli chcielibyśmy stworzyć stałą. Jak się to robiło? Wystarczyło na na samym początku deklaracji zmiennej dopisać modyfikator const. I dzięki naszemu modyfikatorowi nasza zmienna staje się stałą.

Czyli np. const char znak;

Może brzmi to dziwnie, ale tak właśnie się dzieje.

Skoro stałym nie da się zmienić wartości, to nie możliwa jest taka operacja poniżej

znak = ‘w’;

Może się wydawać, że jest wszystko porządku, bo przecież nadajemy tej stałej wartość dopiero pierwszy raz. A pamiętasz jak to było ze zmiennymi? Jeżeli nie nadałeś im wartości na początku, to był tam zawsze jakiś śmieć. Druga linijka nie nadawała wartości, lecz już ja zmieniała. A przecież żadne zmiany nie są na stałych dozwolone. Dlatego jedynym wyjściem jest inicjalizacja stałej w trakcie jej powstania. Jeżeli nie nadamy stałej żadnej wartości, mimo tego, że nawet później nigdzie nie będziemy próbować nic z nią zrobić, to kompilator i tak zwróci błąd. Nie pozwoli by tam był jakiś śmieć.

Podsumowując //każdej stałej musimy nadać wartość podczas inicjalizacji.

Wypisywanie wartości w przypadku wszystkich zmiennych odbywa się tak samo.

Możemy oczywiście modyfikować wszystkie zmienne. Jednake, nie możemy modyfikować stałych.

No i dla kontroli wypiszmy wszystko jeszcze raz :)

[deklaracja łączna]

Gdybyśmy chcieli mieć do dyspozycji np. 3 zmienne tego samego typu, to nie musimy tego robić w trzech osobnych instrukcjach. Możemy zastosować tutaj deklarację łączną.

Int a, b=4, c;

Jest równoważne

Int a; Int b=4; Int c;