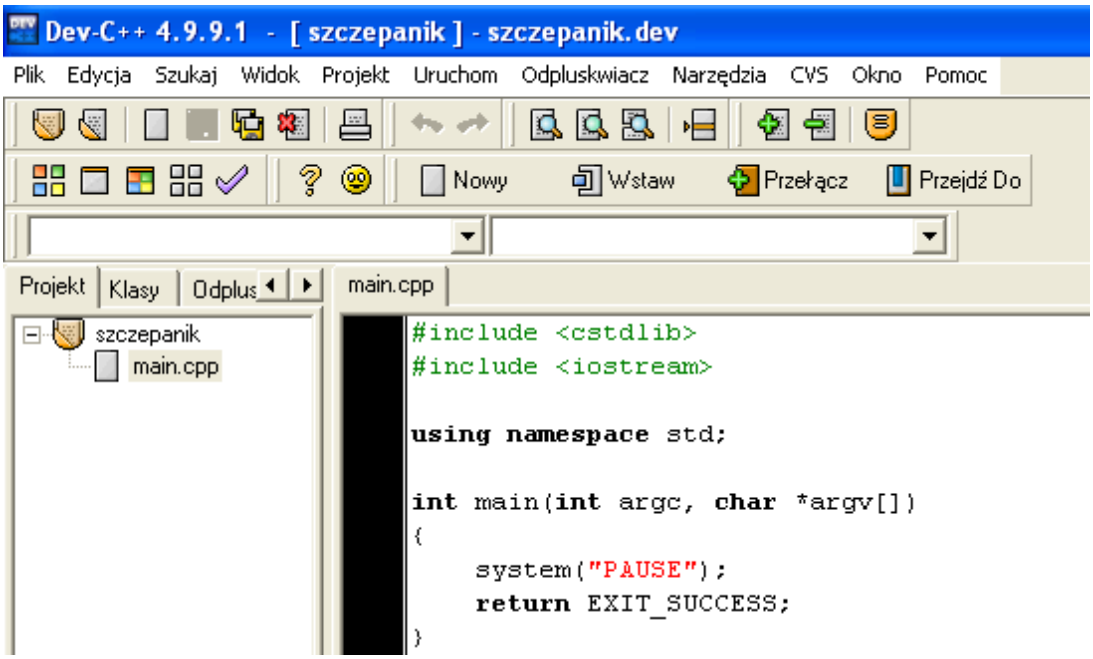
Część IV

Ćwiczenie 1

1.Tworzymy na pulpicie folder o nazwie nazwisko3

Tworzymy nowy projekt w Dev C++ o nazwie nazwisko3

Pierwsze trzy, wygenerowane przez Dev-C++, polecenia powinny znajdować się w każdym naszym kodzie.



Dwie pierwsze linijki naszego kodu źródłowego zaczynają się od znaku #, po którym występuje słowo **include** (z ang. dołącz).
W języku C++ zapis **#include** to tak zwana **dyrektywa preprocesora** - informacja dla preprocesora, aby w tym miejscu dołączył do naszego programu zawartość wskazanego pliku (czyli najpierw pliku o nazwie **cstdlib** , a później pliku **iostream**)

Preprocesor jest specjalnym programem, który przygotowuje kod do kompilacji, czyli zajmuje się nim przed przekazaniem kodu do kompilatora.

Plik cstdlib – tu znajdują się definicje różnych użytecznych funkcji, które możemy wykorzystać w programie

Plik iostream – (skrót od input-output stream – strumień wejścia-wyjścia) zawiera definicje poleceń pozwalających na wyświetlenie informacji na ekranie i wczytywanie do programu danych z klawiatury. Dzięki iostream wyświetlimy na ekranie dowolny komunikat.

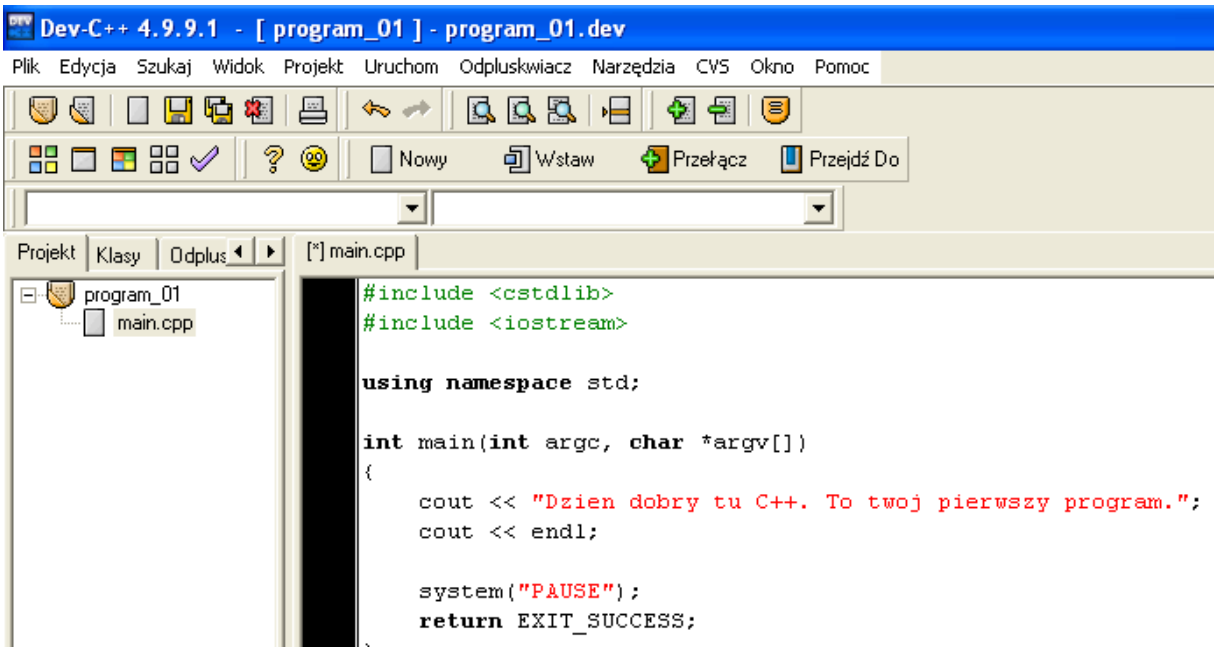
UWAGA

W wielu złożonych programach możemy spotkać się z nieco odmiennym zapisem dyrektywy **include** , w której nazwa dołączanego pliku umieszczana jest w cudzysłowach, a nie w nawiasach ostrych. Różnica w działaniu obu dyrektyw jest taka, że w wypadku nawiasów ostrych dołączany plik poszukiwany jest w katalogu, w którym kompilator przechowuje różne tego typu pliki (w wypadku aplikacji Dev-C+ + jest to katalog C:\Dev-Cpp\include). Jeśli nazwę zapiszemy w cudzysłowach, plik będzie poszukiwany w folderze bieżącym (tam gdzie zapisany jest plik z kodem źródłowym naszego programu, na przykład C:\biblioteczka\program_01)

Ćwiczenie nr 2.

1.Wewnątrz funkcji głównej umieszczamy dwa polecenia

```
cout << "Dzien dobry tu C++. To twoj pierwszy program.";
cout << endl;
```



2. Z menu Uruchom wybieramy Kompiluj i uruchom – widzimy efekt naszej pracy

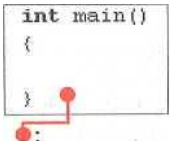
Ćwiczenie wykonujemy od początku pamiętając o utworzeniu folderów nazwisko4 i nazwisko5 dla dowolnych tekstów

Ćwiczenie 3

1.Do tej pory poznaliśmy znaczenie prawie wszystkich poleceń ze szkieletu naszego programu C+ +. Pozostała tylko jedna instrukcja **using namespace std;**
Rozwiązuje ona problem dublujących się nazw różnych funkcji i poleceń.
Gdybyśmy nie wpisali instrukcji **using" namespace std;** przy wykorzystywaniu poleceń (na przykład **cout** czy **endl**) musielibyśmy wskazywać kompilatorowi, skąd one pochodzą (czyli zamiast **cout** musielibyśmy napisac **std::cout** , a zamiast endl napisać **std: :endl**.

std - oznacza bibliotekę standardową, w której znajdują się definicje wszystkich najważniejszych symboli oraz poleceń i funkcji. Będziemy z nich często korzystać, więc warto za pomocą polecenia **using namespace** ułatwić sobie pracę.

2.Każdy program musi zawierać funkcję główną o nazwie **main**. Najprostsza definicja funkcji głównej ma postać

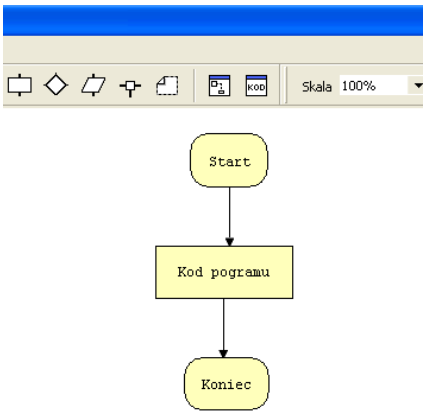


►- main jest nazwą funkcji głównej,

►- nawiasy okrągłe oznaczają, że mamy do czynienia właśnie z funkcją,
►- nawiasy klamrowe wyznaczają ciało funkcji; pomiędzy nimi umieszczamy instrukcje, które funkcja ma wykonać,
[int] oznacza typ wyniku zwracanego przez funkcję; [int] oznacza liczbę całkowitą

3. Funkcja główna na schemacie blokowym
Wykonywanie programu rozpoczyna się od funkcji **main**, a kończy wraz z ostatnim poleceniem występującym w ciele funkcji głównej. Dlatego na schemacie blokowym funkcja **main** reprezentowana jest przez dwa prostokąty z zaokrąglonymi bokami , które rozpoczynają i kończą działanie algorytmu programu.
Funkcja **main** po zakończeniu swojego działania zwraca liczbę całkowitą (określa to słowo **int** umieszczone przed nazwą funkcji). Dlatego ważne jest umieszczenie w ciele funkcji głównej, na samym jej końcu, polecenia return, po którym znajduje się zapis **return EXIT_SUCCESS;** . Jest on równoważny liczbie 0.
Jeśli funkcja **main** zostanie wykonana do końca, zwróci liczbę 0, co oznacza, że zakończyła się sukcesem. Jeśli działanie funkcji zostanie przerwane wcześniej, zwrócona wartość będzie większa od 0 - taka sytuacja oznacza dla systemu informację, że w funkcji głównej znajdują się błędy.

4. Narysuj za pomocą **Magicznych bloczków** schemat i zapisz go pod nazwą Schemat funkcji main w swoim folderze **nazwiskocplusplus**



.

5. Programowanie wymaga niesamowitej precyzji i brak lub złe zastosowanie nawet jednego znaku, może podczas kompilacji wywołać lawinę błędów.

Znak #występuje przed dyrektywami preprocesora, czyli informacjami dla programu, który przygotowuje kod źródłowy dla kompilatora. W Dev-C++ **dyrektywy preprocesora są oznaczane w kodzie kolorem zielonym**
Średnik jest znakiem szczególnym. Występuje on najczęściej na końcu linii, a jego obecność **powoduje, że polecenie znajdujące się przed nim staje się instrukcją.** Jego brak jest częstą przyczyną błędów w kodzie.

6. Kod może być tak długi i skomplikowany, że trudno nam będzie przypomnieć sobie, jakie zadanie miał spełniać każdy z jego fragmentów..
Wyobraźmy sobie, że stworzyliśmy kilkusetlinijkowy kod i po miesiącu chcemy go udoskonalić. Niemal pewne jest, że będziemy musieli od nowa go przeanalizować, aby przypomnieć sobie, jakie zadania spełniały jego poszczególne fragmenty (sytuacja może być jeszcze trudniejsza, gdy przyjdzie nam poprawiać kod innego programisty). Dlatego należy

komentować kod źródłowy. Komentarz widoczny dla kompilatora i w żaden sposób nie wpłynie na działanie i wielkość naszego programu.
Pierwszym sposobem skomentowania fragmentu kodu źródłowego jest umieszczenie komentarza pomiędzy znakiem otwarcia komentarza **/* a znakiem jego zakończenia */**. Taki komentarz może obejmować kilka linijek kodu. Dev-C + + komentarz koloruje na niebiesko.

7. Na pulpicie tworzymy folder o nazwie nazwisko6, tworzymy nowy projekt o nazwie nazwisko6 i umieszczamy przykładowy komentarz.

.cpp

```
/*Ten program bedzie losował numery totolotka.
Jak wygram zostane bogaty.
Milion oddam na cele charytatywne.
A za drugi milion kupię sobie Bugatti.*/
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    cout<<"Prosciej było skopiowac usunac exe, skompilowac i uruchomic jako 5. ";
    /*To na razie jest napis ale będzie za chwile program*/
    cout<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

8Tworzymy na pulpicie folderu nazwisko7 i nazwisko8 w których umieszczamy w podany wyżej sposób po dwa komentarze, pamiętając by za każdym razem z Menu Uruchom wybrać Kompiluj i uruchom
9. Wszystkie pliki z nazwiskiem i kolejnym numerem **umieszczamy w swoim folderze nazwiskocplusplus na serwerze.**