Przydatne metody: Console.Write(), Console.WriteLine(), Console.ReadLine(), Console.Readkey().

1. Otwórz program Visual Studio.
2. Utwórz nowy projekt - aplikacja konsolowa w języku C#.
3. Napisz program wyświetlający zdanie “Hello world.”.
4. Stwórz zmienne: “bob” i “eva” i przypisz im odpowiednie przywitania (np. “Hello Bob”).
5. Stwórz zmienną “userName”, a następnie spraw, aby program pytał użytkownika o jego imię. Zmiena userName ma tworzyć ciąg “Hello X.”, gdzie X to podane imię. Do tego celu użyj operatora **+** do łączenia fragmentów ciągu. Na koniec wyświetl zmienną userName na ekranie.
6. Stwórz komentarze opisujące fragmenty kodu, a także dodaj komentarz zawierający dane autora.
7. Przetestuj działanie odwołania #region.
8. Umieść w komentarzu kod odpowiadający za wyświetlanie powitań wykorzystując **/\*** i **\*/** lub skrótu **ctrl + k**, **c** (odwrotność to **ctrl +k**, **u**)
9. Stwórz kilka zmiennych liczbowych i sprawdź działanie operatorów matematycznych (**+**, **-**, **\***, **/**,).
10. Stwórz zmienną logiczną oraz nadaj jej logiczną wartość.
11. Sprawdź działanie operatora **!**.
12. Przetestuj działanie następujących operatorów (**==**, **<**, **>**, **!=, &, I**).
13. Przeanalizuj program z pliku program103.cs. Następnie skompiluj i uruchom go – sprawdź przewidywane wyniki.
14. W Program103.cs ustaw breakpointy.
15. Sprawdź co możesz zrobić w trybie debugowania.
16. Stwórz nowy program, dodaj zmienna liczbową a następnie inkrementuj ją kilka razy.
17. Postaw breakpoint w linii w której została nadana wartość zmiennej.
18. W trybie debugowania dodaj zmienną do listy obserwowanych.
19. Przechodząc krok po kroku obserwuj zmianę wartości. Czy możesz ją zmienić ręcznie? Czemu może to służyć?
20. Przeanalizuj kod z pliku program101.cs. Czemu pokazuje błąd przy kompilacji? Popraw go tak, aby program działał poprawnie.
21. Utwórz program, w którym będą zadeklarowane 3 zmienne a,b,c typu int (bez wspólnych dzielników innych niż 1, np. 5,9,11). Następnie wykonaj i wyświetl w aplikacji wyniki operacji: a/b, a/c,b/c,b/a,c/a,c/b.
22. Przeanalizuj kod z pliku program102.cs. Nie uruchomiaj programu. Jakie wartości otrzymasz na poszczególnych krokach? Sprawdź to po kompilacji.
23. Zadeklaruj zmienną typu sbyte i przypisz jej wartość 127. Wykonaj na niej operację inkrementacji. Wynik wyświetl na ekranie.
24. Na pliku program102.cs (lub innym) poćwicz zmianę nazw zmiennych. Co jest lepsze: **ctrl+h** (find and replace) czy **ctrl+r, ctrl+r** (rename)?

Dodatkowe zadania:

* Utwórz program w którym zadeklarujesz tablicę liczbową. Z użyciem pętli **for** wypisz wszystkie wartości z tej tablicy.
  + Po przecinku (np. 1,2,3…)
  + Każdy element w nowej linii
  + Od ostatniego elementu do pierwszego (odwrotna kolejność)
* Stwórz program, który będzie służył do porównywania liczb. Zadeklaruj kilka zmiennych liczbowych, a następnie wykorzystując instrukcję warunkową **if...else** porównuj je ze sobą i wyświetl odpowiedni komunikat.
  + Przerób program tak aby przyjmował z klawiatury 3 liczby, a następnie sprawdzał warunek istnienia trójkąta.
* Stwórz program, który będzie wczytywał liczbę podaną przez użytkownika (ograniczoną “z góry” do 15 i “z dołu” do 5), a następnie odejmował od niej 1 i wypisywał wynik, aż do osiągnięcia 0. Użyj do tego pętli **while**.
* Stwórz program, który będzie wczytywał liczbę podaną przez użytkownika. Następnie sprawdź czy liczba spełnia podane wymagania (np. >4, <7, !=10 itp.). Jeśli liczba tak wyświetl komunikat kończący, jeśli nie, wyświetl prośbę o wczytanie liczby. Użyj do tego pętli **do...while**.