

## Zajęcia 1. Wprowadzenie

### 1. Zapoznanie się ze środowiskiem pracy

#### Praca w sieci lokalnej

Laboratorium prowadzone jest na komputerach wyposażonych w system operacyjny **Linux** z Oracle VM VirtualBox. Ponadto katalogi robocze są umieszczone na serwerze plikowym pracującym pod kontrolą systemu **Linux**. Aby przystąpić do zajęć laboratoryjnych, należy uruchomić program Oracle VM VirtualBox Manager, a następnie wybrać maszynę wirtualną „**Lab\_A300\_VS\_Community**” i wcisnąć ikonę „**Start**”.

#### Sposób logowania

Aby się zalogować do systemu należy wykonać następujące czynności:

1. Wybrać: **Inny użytkownik** (w dolnym lewym rogu)
2. W miejscu **Nazwa użytkownika** należy podać domenę oraz identyfikator: **A300\TD1labvxyz**, gdzie **vx** oznacza numer grupy laboratoryjnej; **yz** oznacza numer zespołu w grupie (przykładowy identyfikator **TD1lab0205** – co oznacza grupę L02, stanowisko nr 5)
3. W miejsce **Hasło** wpisujemy: student

Uwaga! Identyfikatory **TD1labvxyz** zostaną nadane poszczególnym studentom przez prowadzącego zajęcia. Są ważne do końca semestru i powiązane z danym studentem, a nie z konkretnym komputerem.

#### Katalogi robocze

Pliki robocze (źródłowe i pliki projektu) powinny się znaleźć w katalogach umieszczonych na serwerze plikowym na dysku **h:**.

#### Praca w środowisku Visual Studio

Aby uruchomić środowisko programistyczne należy kliknąć na ikonę **Visual Studio 2015**, która znajduje się na pulpicie, pasku zadań lub w grupie programów **Visual Studio 2015**.

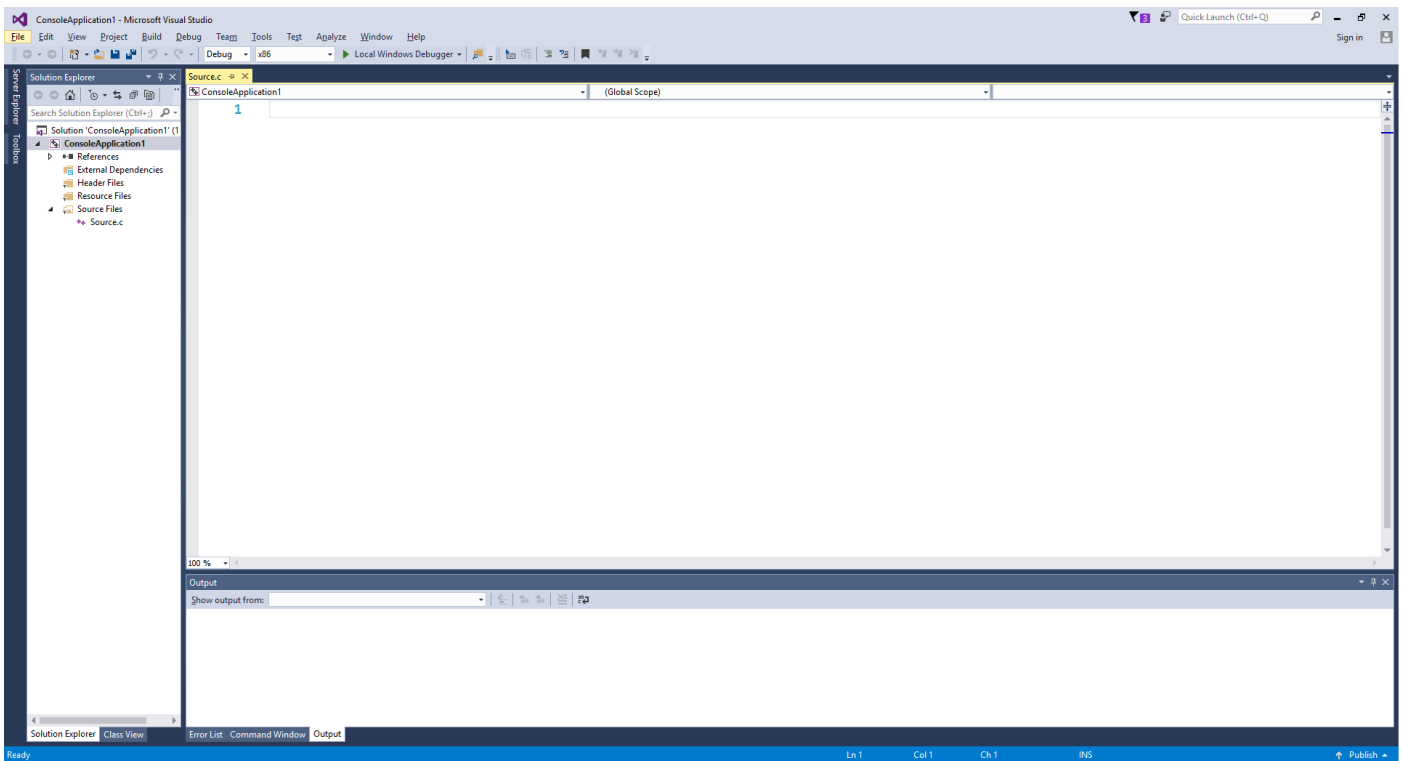
#### Tworzenie projektu

**Visual Studio 2015** (VS 2015) nie jest środowiskiem programistycznym bezpośrednio przeznaczonym do programowania w języku C. Bazuje ono na platformie **.NET**, która nie wspiera składni oraz zaplecza bibliotecznego tego języka. Twórcy VS 2015 udostępnili jednak możliwość programowania w języku C. Wykorzystanie języka C może być zrealizowane przy użyciu prostych programów w ramach projektów typu **Visual C++**, których efekt działania przedstawiony jest w oknie linii poleceń.

Aby utworzyć nowy projekt typu **Visual C++** należy w menu **File** wybrać opcję **New** i kolejno zakładkę **Project** (domyślny skrót klawiszowy **Ctrl+Shift+N**). Następnie w obrębie zainstalowanych szablonów (**Templates**) wybrać **Visual C++** i w prawym panelu zaznaczyć typ **Win32 Console Application**. Na sam koniec konieczne jest podanie nazwy projektu (pole tekstowe z etykietą **Name**) i wskazanie lokalizacji, w której ma być on utworzony (wybrać dysk sieciowy **h:\**). Opcje **Create directory for solution** oraz **Add to Source Control** mają pozostać niezaznaczone. Po poprawnym wprowadzeniu wszystkich w/w ustawień należy przyciskiem **OK** zatwierdzić pierwszy etap tworzenia projektu. Spowoduje to pojawienie się okna kreatora projektu z ogólną informacją. Tutaj wymagane jest jedynie kliknięcie przycisku **Next>**. Rezultatem będzie pojawienie się trzeciego, ostatniego okna kreatora ze szczegółowymi ustawieniami aplikacji. W tym oknie, wystarczy zaznaczyć opcję **Empty project** i kliknąć

przycisk **Finish**. W przypadku pomyłki w typie projektu zaleca się usunięcie projektu i utworzenie nowego wg podanych wyżej zasad.

Po utworzeniu nowego projektu z menu lokalnego **Project** należy wybrać opcję **Add New Item**. Ukaże się wówczas okno dialogowe służące do dodania do projektu pliku umożliwiającego pisanie kodu w języku C++. Kolejne kroki sprowadzają się do: 1 – zaznaczenia typu pliku – **C++ File (.cpp)**, 2 – podania nazwy pliku z rozszerzeniem **\*.c** w polu nazwy, 3 – wskazania lokalizacji (zawartość tego pola musi pozostać domyślna) oraz 4 – zatwierdzenia wszystkich operacji za pomocą przycisku **Add**. Uwaga: pozostawienie nazwy pliku z rozszerzeniem **\*.cpp** zablokuje możliwość tworzenia kodu w języku C. Rysunek 1.1 przedstawia okno projektu z dodanym plikiem **Source.c**.



Rysunek 1.1. Okno projektu typu **Win32 Console Application** z dodanym plikiem **Source.c**.

Pionowy panel po lewej stronie prezentuje strukturę projektu z poziomu plików (zakładka **Solution Explorer**) oraz z poziomu klas/funkcji (**Class View**). Duże okno edycyjne z zakładką **Source.c** służy do pisania kodu, natomiast podłużny panel **Output** u dołu daje m.in. możliwość przeglądania informacji wygenerowanych przez środowisko po etapie kompilacji (np. na temat błędów lub ostrzeżeń). Panel ten może być również wykorzystany do weryfikacji kodu w trakcie pracy krokowej.

## Kompilacja

Narzędzia umożliwiające kompilację i konsolidację znajdują się w menu **Build**. Poszczególne pliki źródłowe można kompilować z osobna poleceniem **Compile** (po uprzednim wskazaniu pliku) albo wybrać **Build Solution**, aby przeprowadzić proces kompilacji i konsolidacji całego projektu. Istnieje również możliwość wyboru opcji **Rebuild Solution**, aby skompilować ponownie wszystkie pliki źródłowe i przeprowadzić konsolidację. Wyniki kompilacji i konsolidacji są pokazywane w panelu **Output**. W przypadku pojawienia się błędu w kodzie, można dwukrotnie kliknąć na linii z komunikatem o błędzie, co spowoduje przeniesienie kursora w panelu edycyjnym do odpowiedniej linii w celu skorygowania błędu.

## Uruchamianie programu

Aby uruchomić stworzony program wystarczy z menu **Debug** wybrać polecenie **Start Without Debugging** (skrót klawiszowy **Ctrl+F5**) albo kliknąć pusty zielony trójkąt w paletce **Debug**. Rezultat działania widoczny jest w postaci „okienka konsolowego”. Jeżeli program zostanie uruchomiony za pomocą opcji **Start Without Debugging**, po zakończeniu działania aplikacji proces jest wstrzymywany i czeka na reakcję użytkownika.

## Praca krokowa

Praca krokowa polega na wstrzymywaniu działania programu po wykonaniu instrukcji zakodowanych w jednej linii. Rozpoczęcie tego trybu pracy odbywa się przez wybranie polecenia **Run to cursor** (skrót **Ctrl+F10**) albo **Step Into** (skrót **F11**) z menu **Debug**. Wykonanie pojedynczej linii programu odbywa się przez polecenie **Step Over** (skrót **F10**). Wygodniejsze jest jednak korzystanie z palety uruchomieniowej **Debug**. W czasie pracy krokowej dolny panel zawiera listę zmiennych (obiektów) tworzonych w trakcie działania programu umożliwiając śledzenie zmian ich wartości. Ponadto w panelu **Watch 1** można utworzyć własną listę zmiennych do podglądnięcia. Zakończenie pracy krokowej i powrót do trybu edycji odbywa się przez wybranie polecenia **Stop Debugging** z menu **Debug** albo z palety uruchomieniowej.

## 2. Opracowanie prostych programów w języku C

1. Dodaj do projektu plik **pierwszy.c**, zdefiniuj w nim funkcję **main**, a następnie napisz kod pozwalający wprowadzić dowolną liczbę całkowitą z klawiatury, a następnie wypisywał ją na ekran.
2. Dodaj do projektu plik **silnia.c**, zdefiniuj w nim funkcję **main**, a następnie opracuj algorytm obliczania silni z liczby, która ma być wczytywana z klawiatury.
3. Dodaj do projektu plik **fibo.c**, zdefiniuj w nim funkcję **main**, a następnie opracuj algorytm obliczania  $n$ -tej liczb Fibonacciego, która ma być zadawana (wczytywana) z klawiatury.
4. Dodaj do projektu plik **pierwiastki.c**, zdefiniuj w nim funkcję **main**, a następnie opracuj algorytm pozwalający rozwiązywać równanie kwadratowe  $ax^2+bx+c=0$ . Współczynniki mają być wprowadzane z klawiatury.

### Uwagi:

1. W projekcie wymagane jest dołączenie odpowiednich plików nagłówkowych.
2. W projekcie może znajdować się wyłącznie jeden plik z funkcją główną **main**. W związku z tym, definicje poszczególnych funkcji mogą być umieszczone w odrębnych plikach a ich deklaracje zapowiadające oraz wywołanie w pliku głównym.