

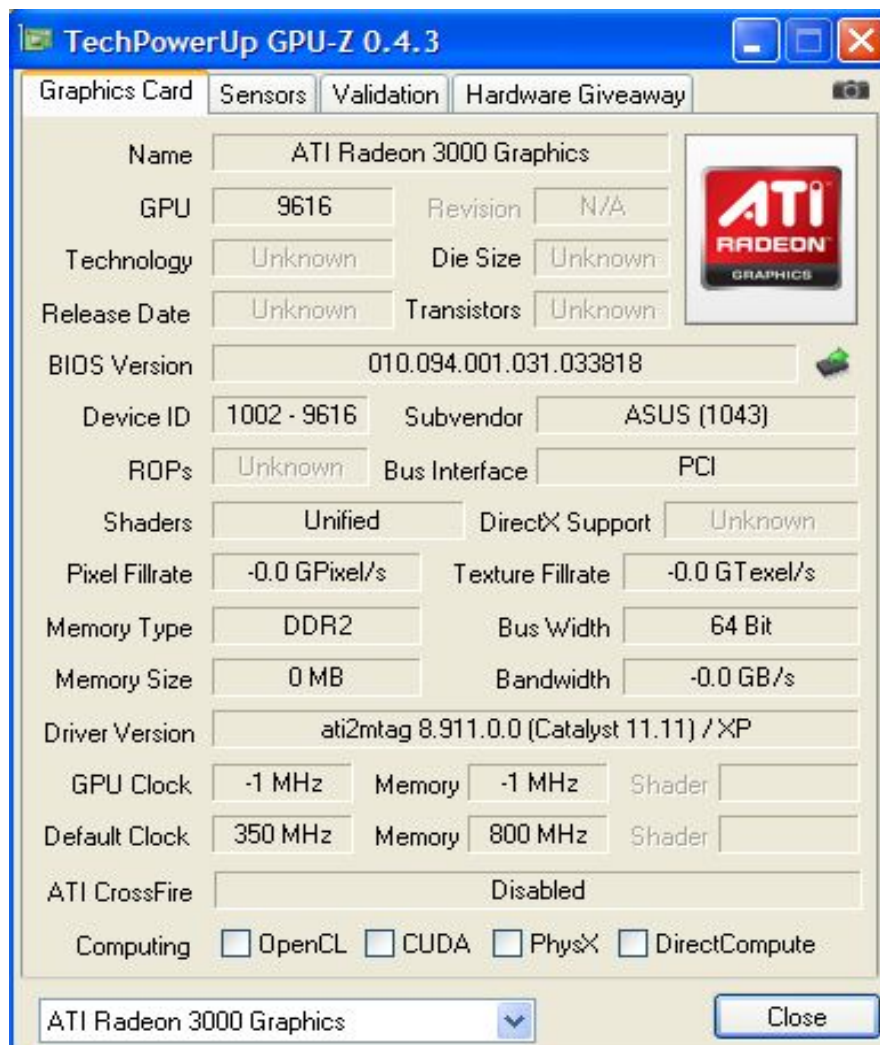
# РАБОТА С CUDA ТЕХНОЛОГИЕЙ

Необходимые системы.

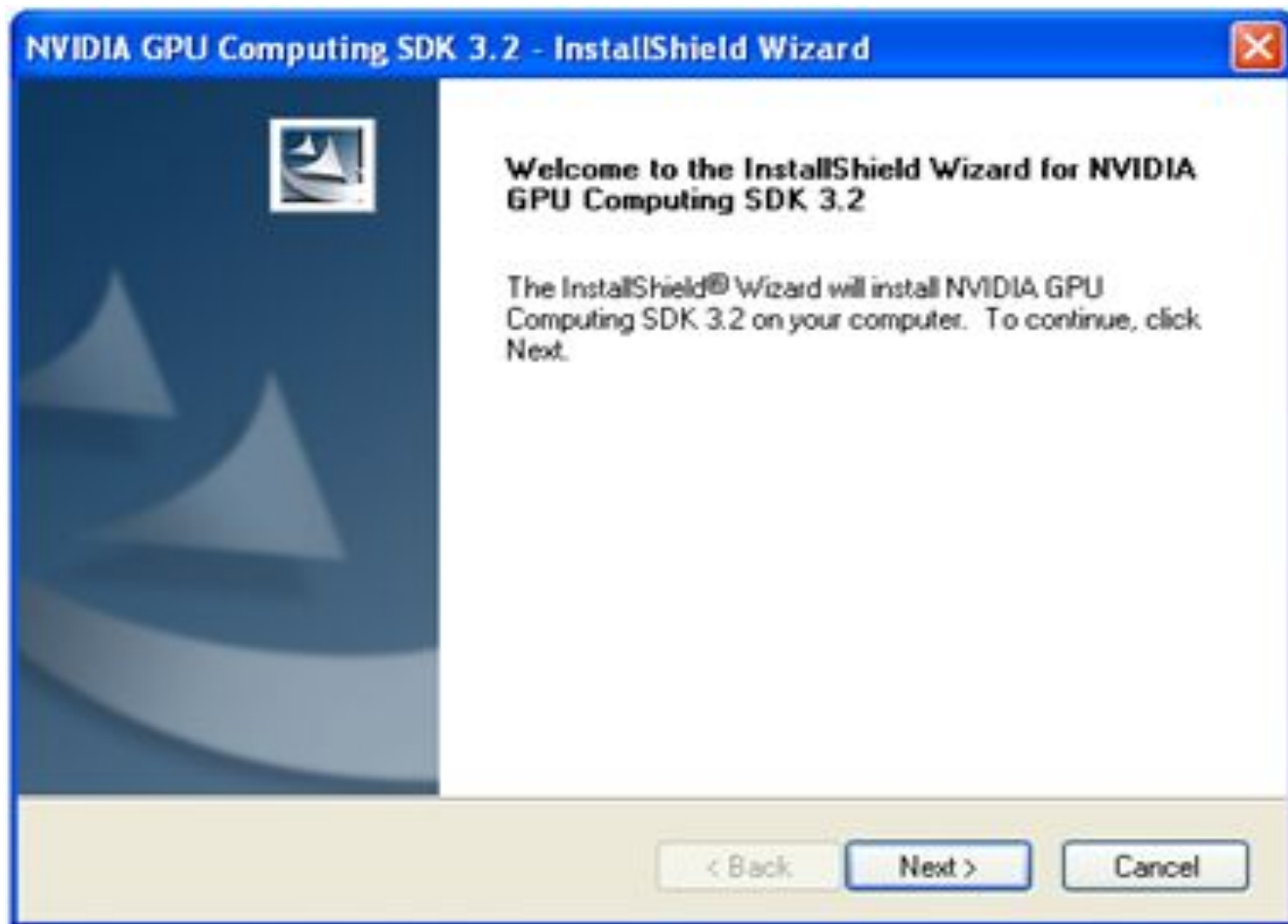
1. GPU-Z Проверка видеокарты
2. Microsoft Visual Studio 2008
3. NVIDIA CUDA SDK 2.3
4. NVIDIA CUDA Toolkit 2.3
5. CUDA\_VS\_Wizard - засіб налаштування Visual Studio;
6. NVIDIA Developer Drivers – драйвери відеокарти;

# Проверка Видеокарты

- Запускаем программу GPU-Z.exe



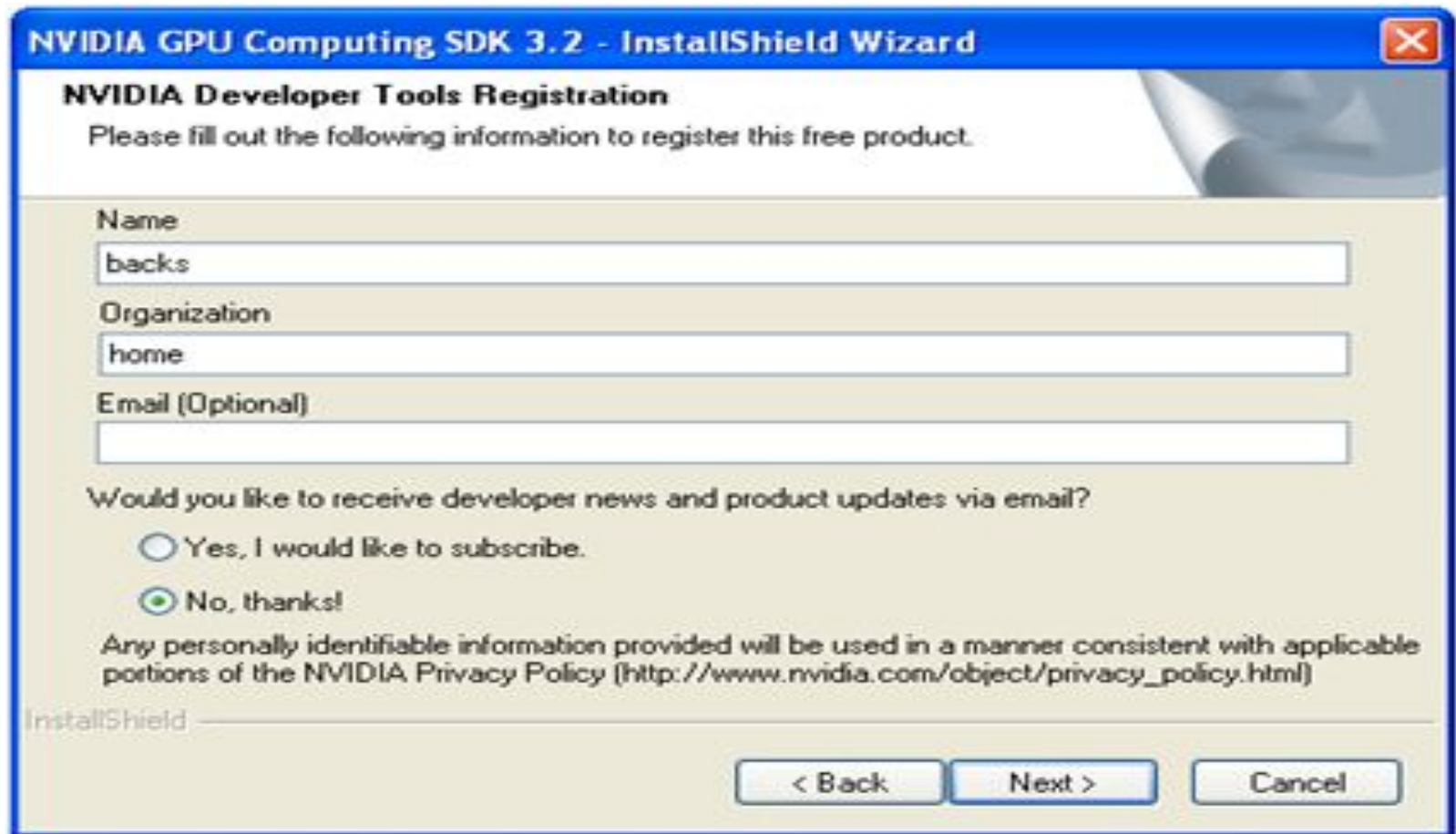
# Порядок встановлення NVIDIA CUDA SDK



# Порядок встановлення NVIDIA CUDA SDK



# Порядок встановлення NVIDIA CUDA SDK



**NVIDIA GPU Computing SDK 3.2 - InstallShield Wizard**

**NVIDIA Developer Tools Registration**

Please fill out the following information to register this free product.

Name  
backs

Organization  
home

Email (Optional)

Would you like to receive developer news and product updates via email?

Yes, I would like to subscribe.

No, thanks!

Any personally identifiable information provided will be used in a manner consistent with applicable portions of the NVIDIA Privacy Policy ([http://www.nvidia.com/object/privacy\\_policy.html](http://www.nvidia.com/object/privacy_policy.html))

InstallShield

< Back   Next >   Cancel

# Порядок встановлення NVIDIA CUDA SDK

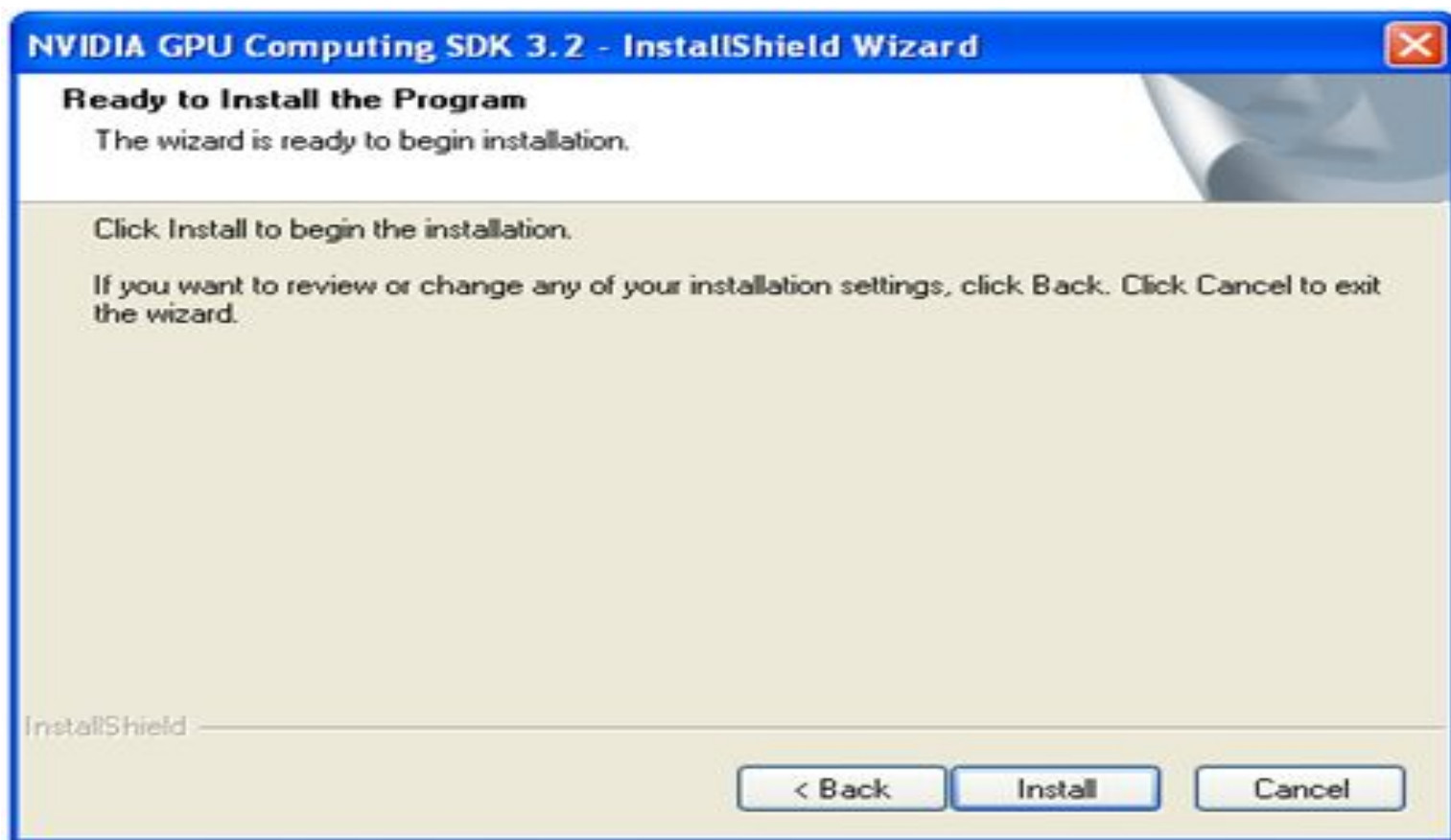


# Порядок встановлення NVIDIA CUDA SDK





# Порядок встановлення NVIDIA CUDA SDK

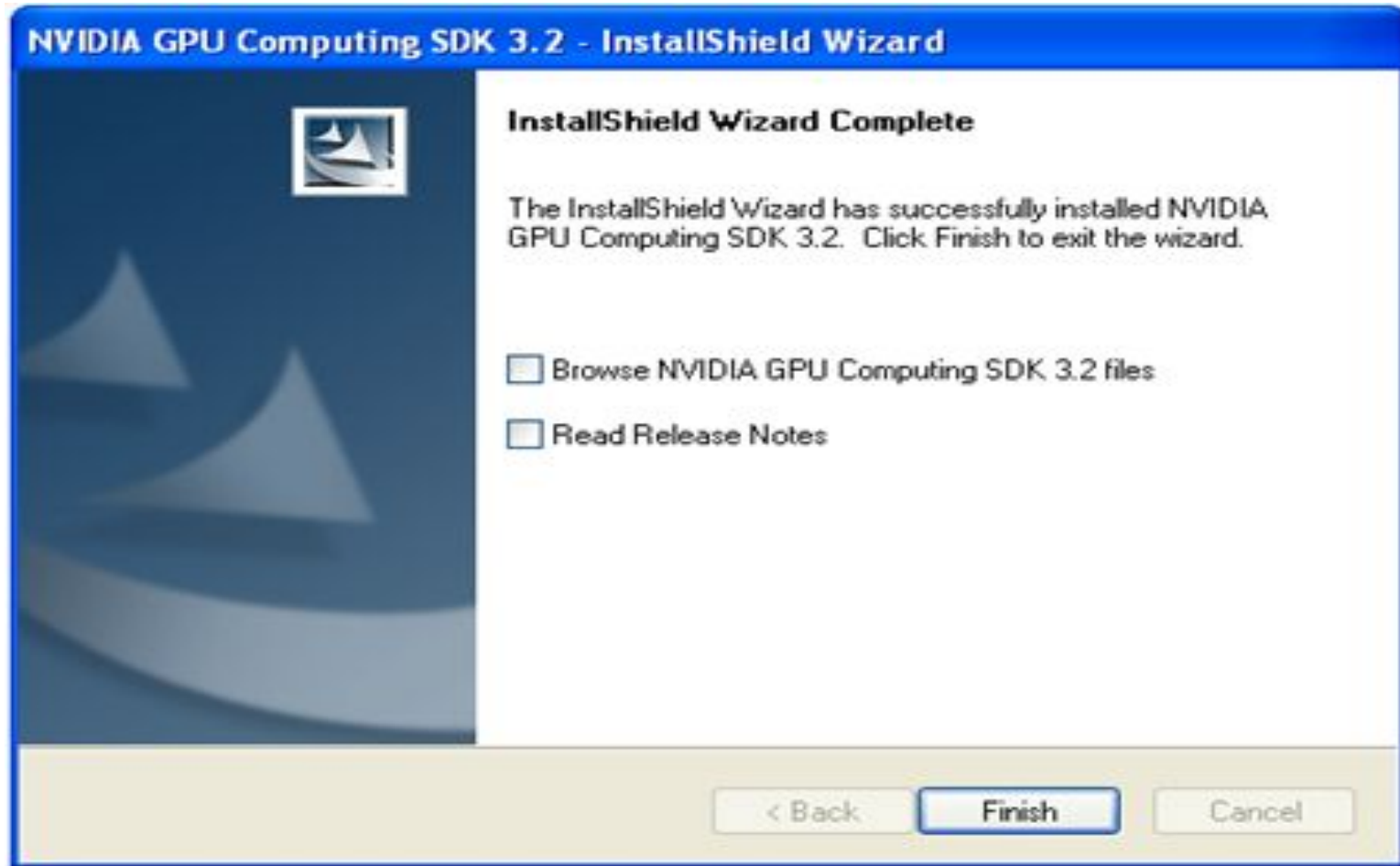




# Порядок встановлення NVIDIA CUDA SDK



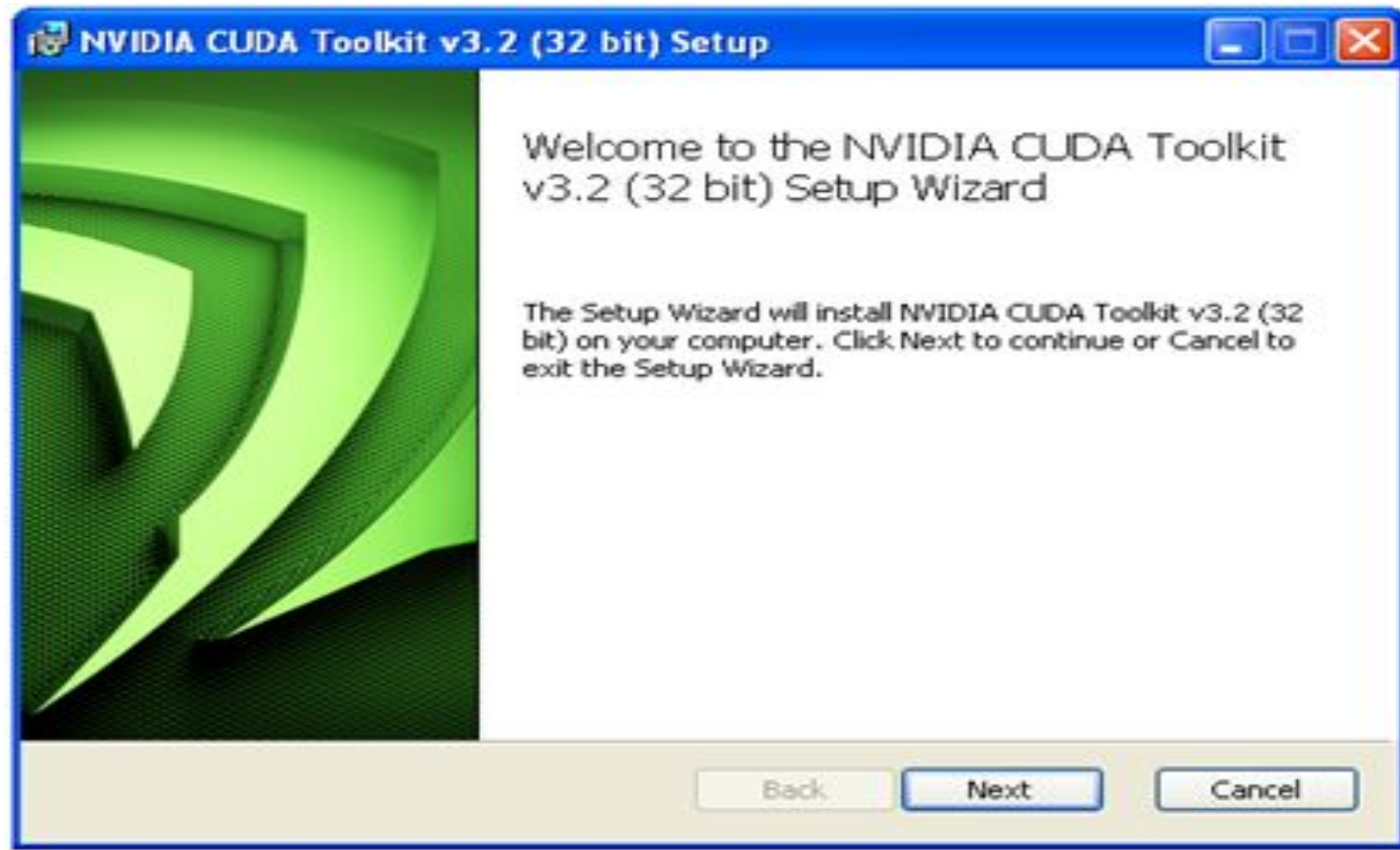
# Порядок встановлення NVIDIA CUDA SDK



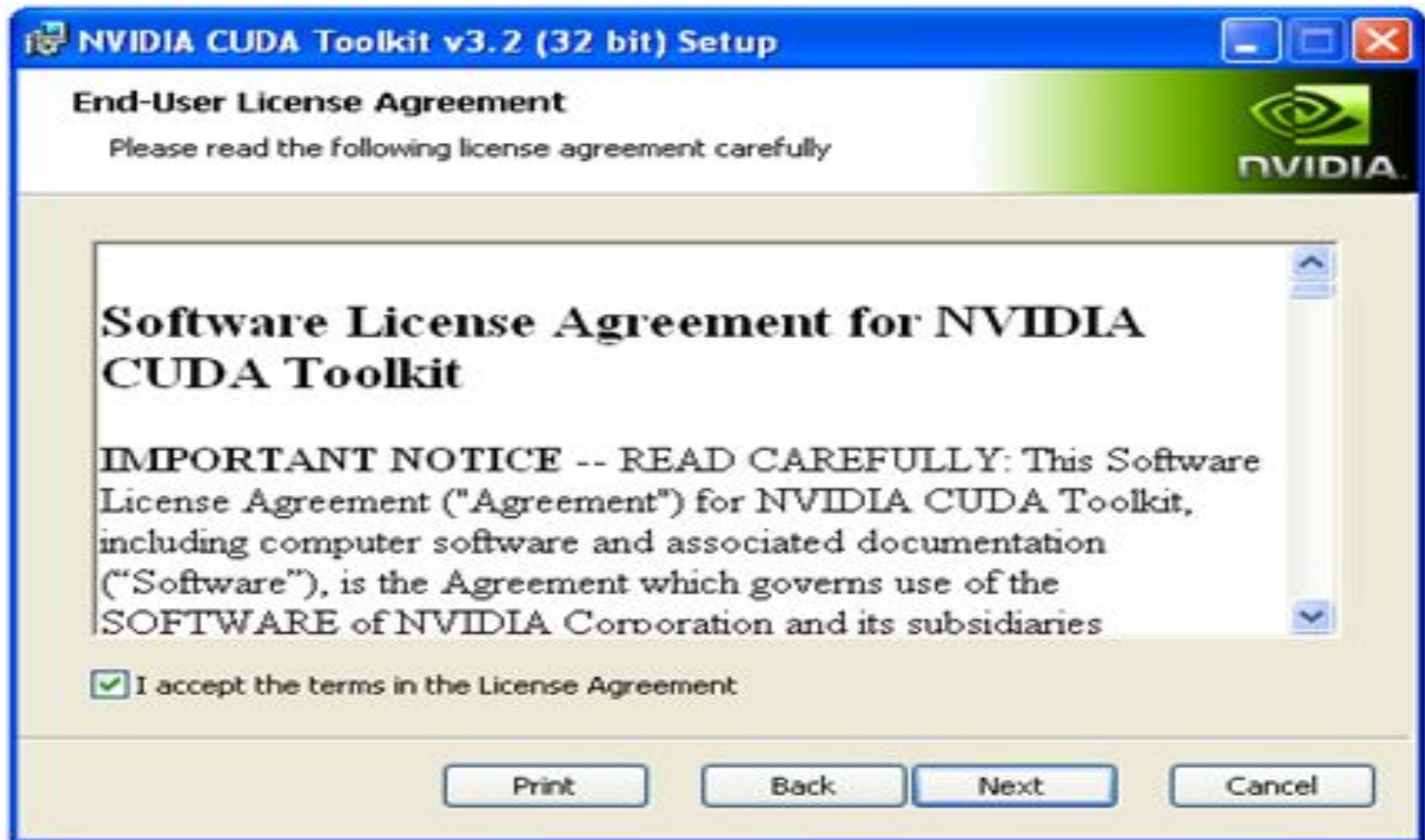
# Проверяем установлен ли компилятор nvcc

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Виктор>nvcc -U
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2009 NVIDIA Corporation
Built on Tue_Feb_23_16:37:32_PST_2010
Cuda compilation tools, release 3.0, V0.2.1221
C:\Documents and Settings\Виктор>
```

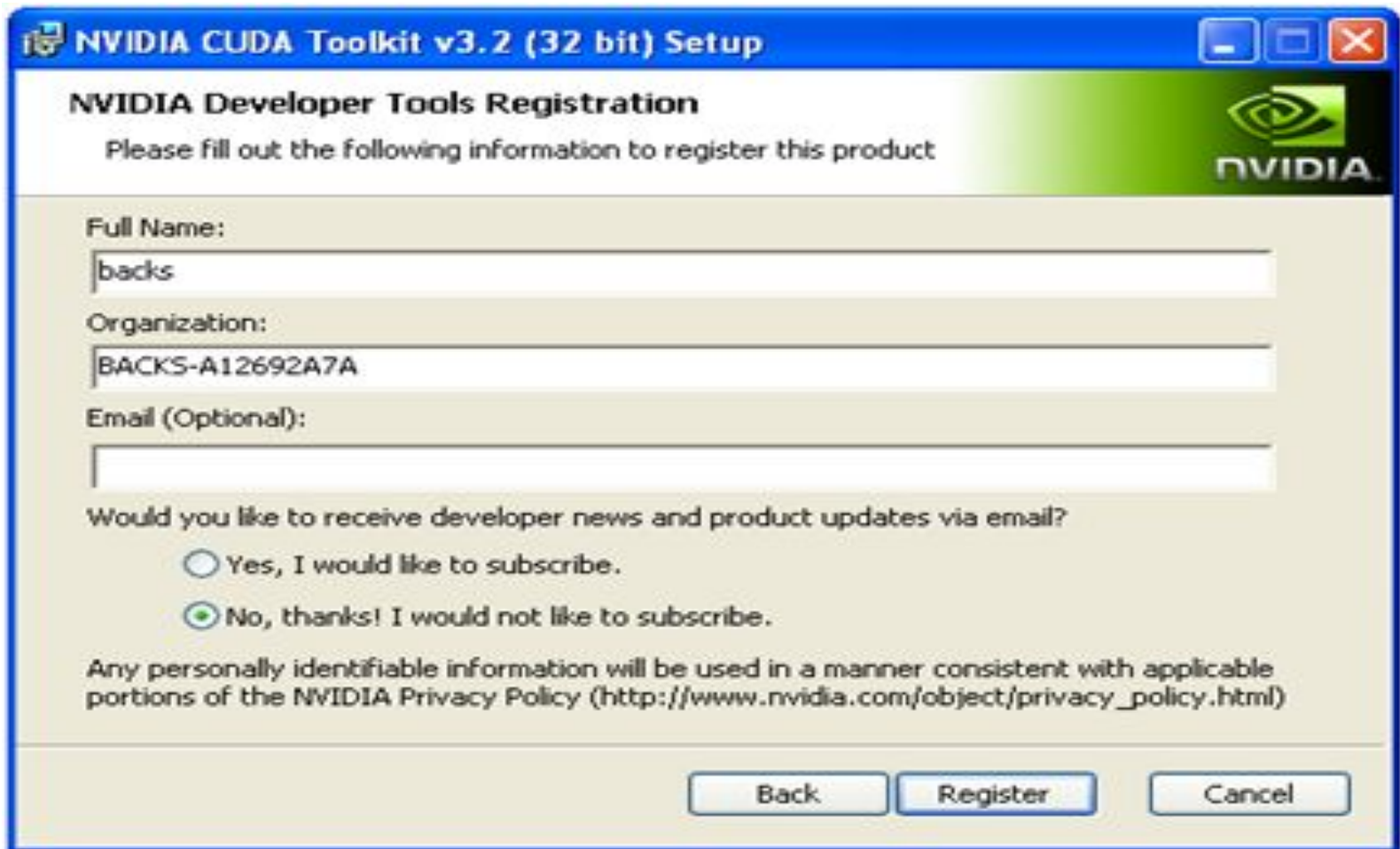
# Встановлення і налаштування NVIDIA CUDA Toolkit



# Встановлення і налаштування NVIDIA CUDA Toolkit



# Встановлення і налаштування NVIDIA CUDA Toolkit



The screenshot shows a Windows installation window titled "NVIDIA CUDA Toolkit v3.2 (32 bit) Setup". The window has a blue title bar with standard Windows window controls (minimize, maximize, close) on the right. The main content area has a green header with the NVIDIA logo and the text "NVIDIA Developer Tools Registration". Below the header, there is a prompt: "Please fill out the following information to register this product".

The registration form contains the following fields and options:

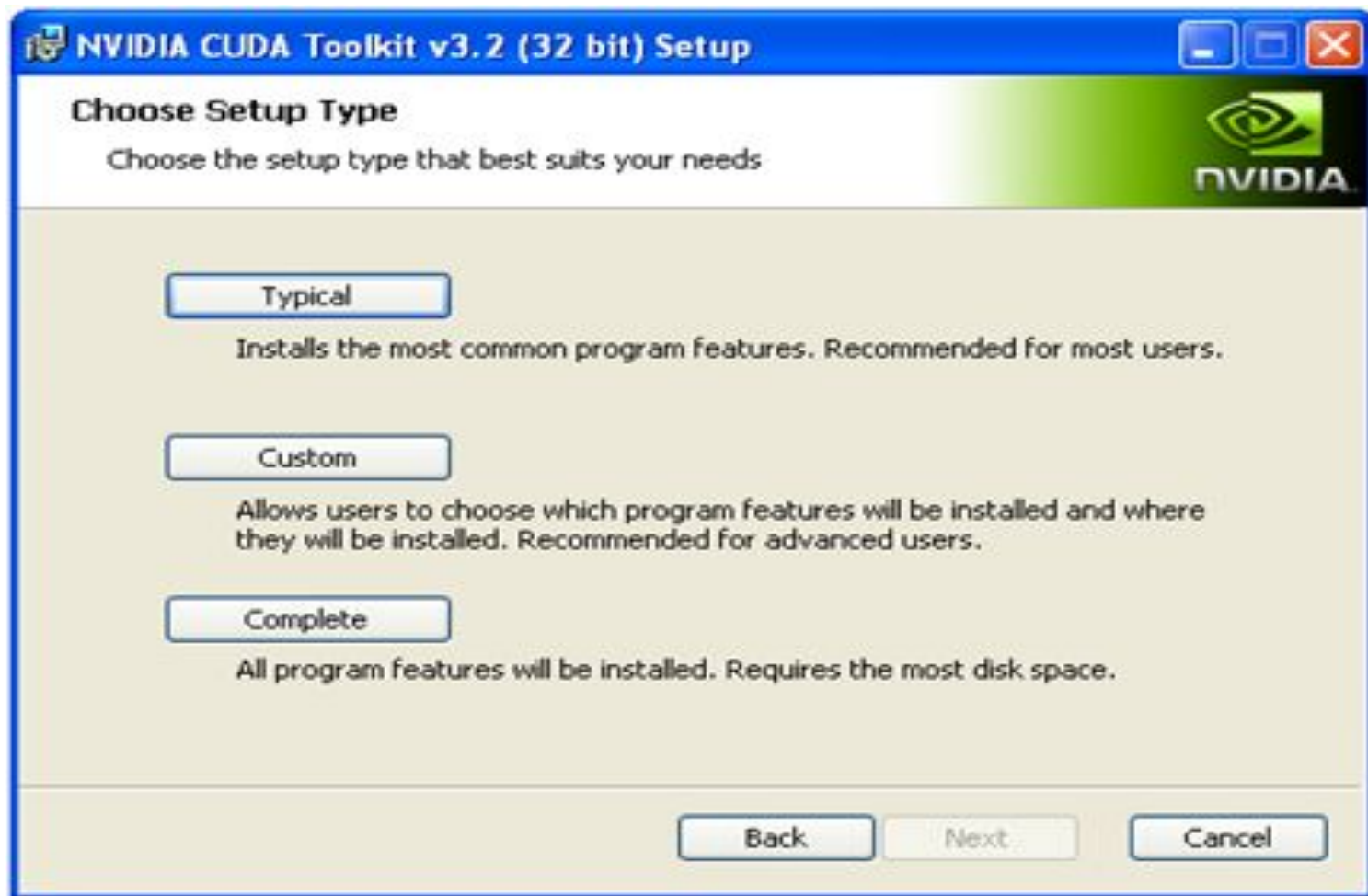
- Full Name:** A text input field containing the text "backs".
- Organization:** A text input field containing the text "BACKS-A12692A7A".
- Email (Optional):** An empty text input field.
- Subscription:** A question "Would you like to receive developer news and product updates via email?" with two radio button options:
  - Yes, I would like to subscribe.
  - No, thanks! I would not like to subscribe.

At the bottom of the window, there is a disclaimer: "Any personally identifiable information will be used in a manner consistent with applicable portions of the NVIDIA Privacy Policy ([http://www.nvidia.com/object/privacy\\_policy.html](http://www.nvidia.com/object/privacy_policy.html))".

At the bottom right of the window, there are three buttons: "Back", "Register", and "Cancel".

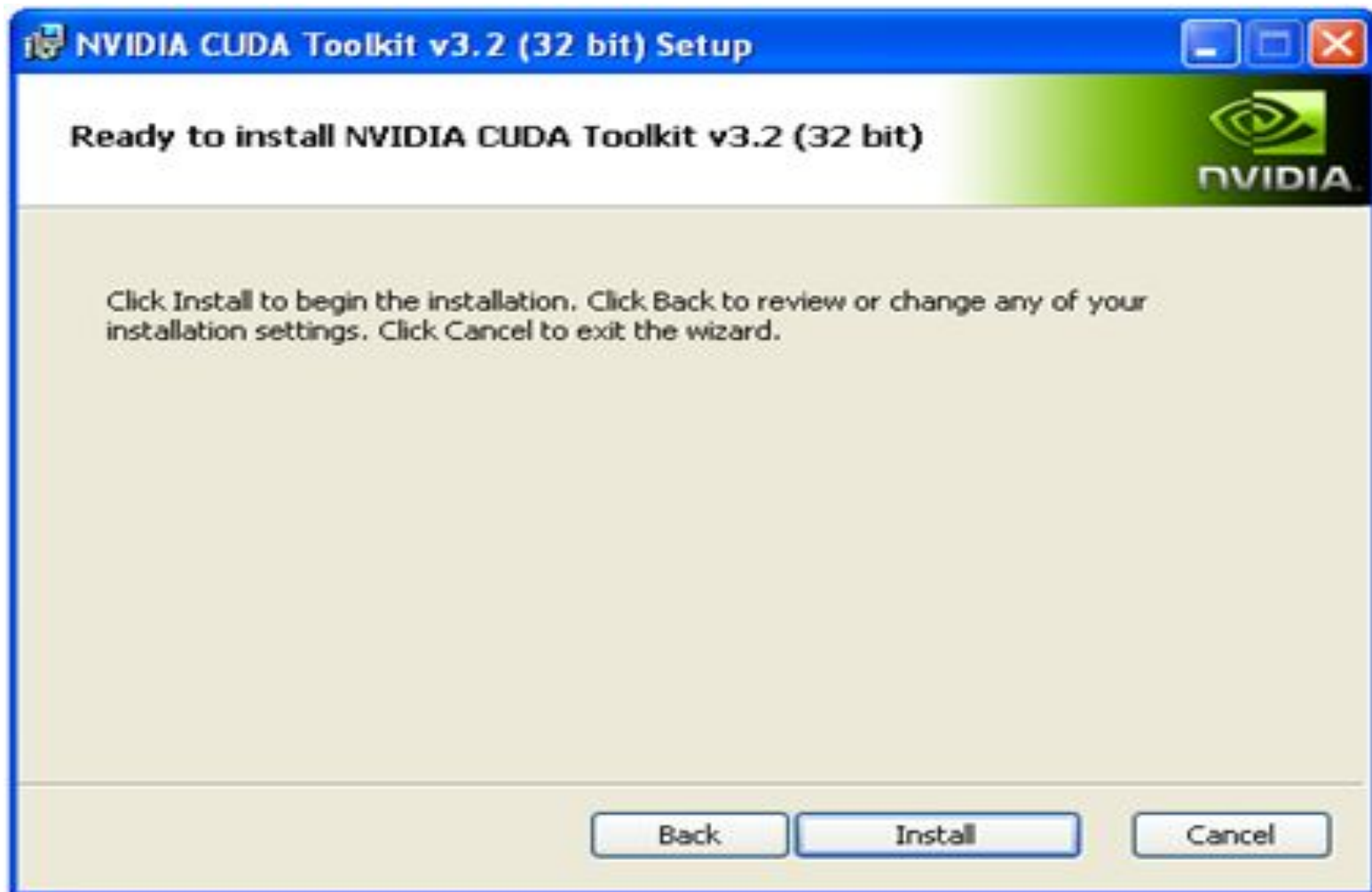


# Встановлення і налаштування NVIDIA CUDA Toolkit

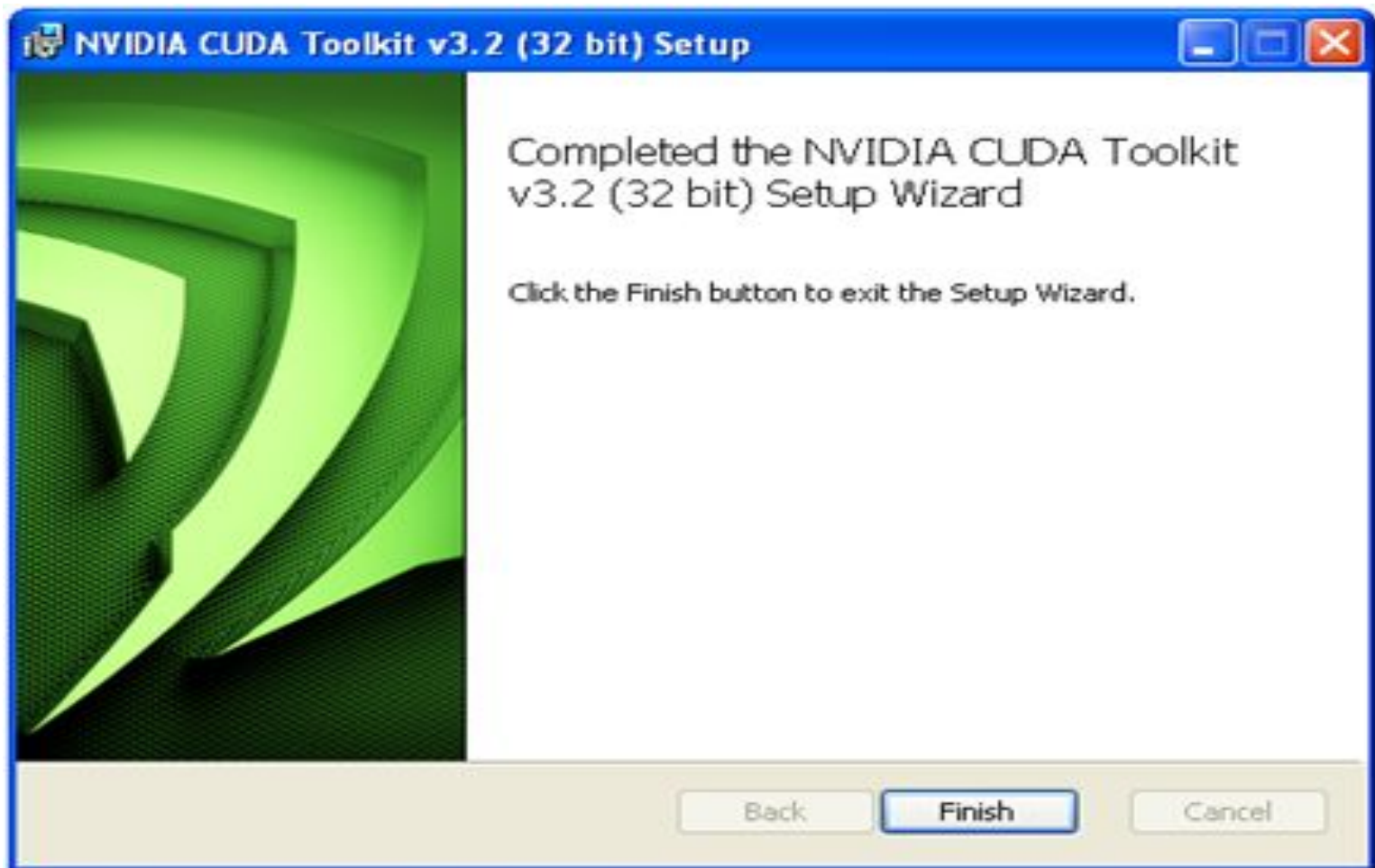




# Встановлення і налаштування NVIDIA CUDA Toolkit

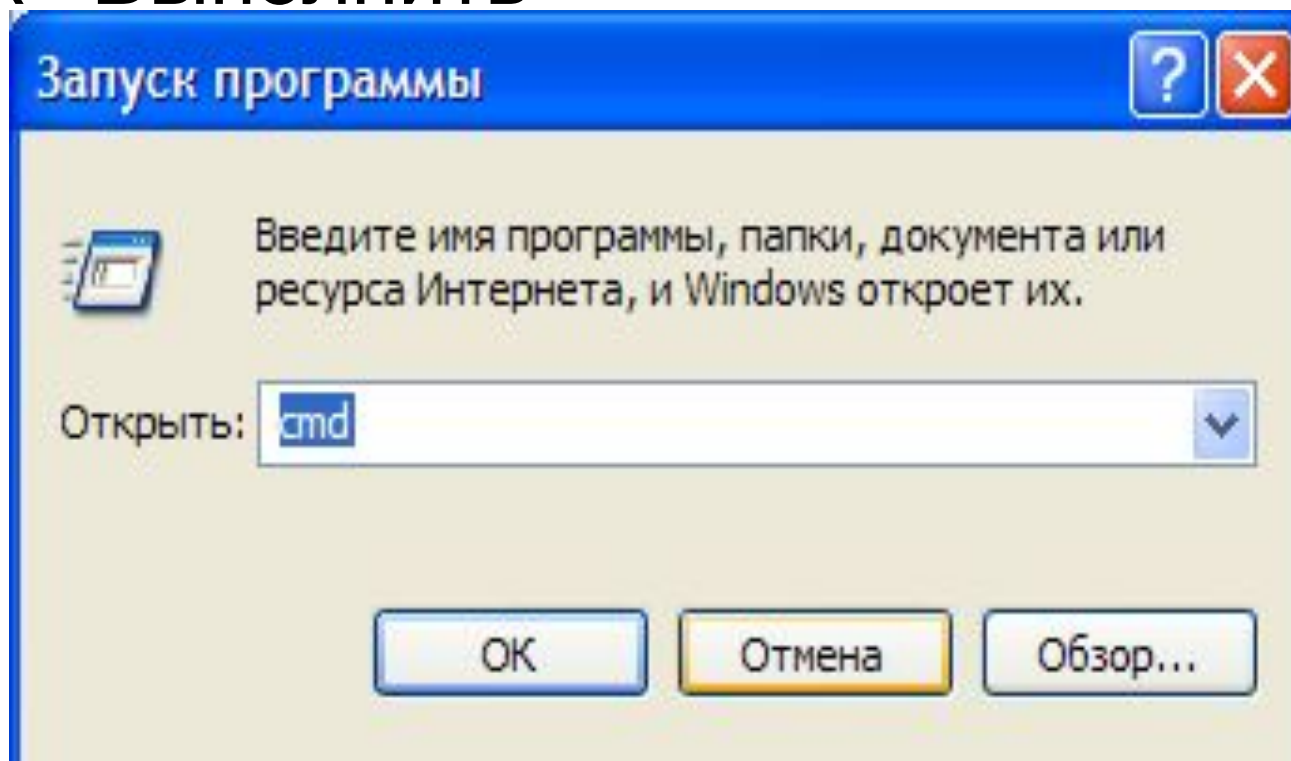


# Встановлення і налаштування NVIDIA CUDA Toolkit

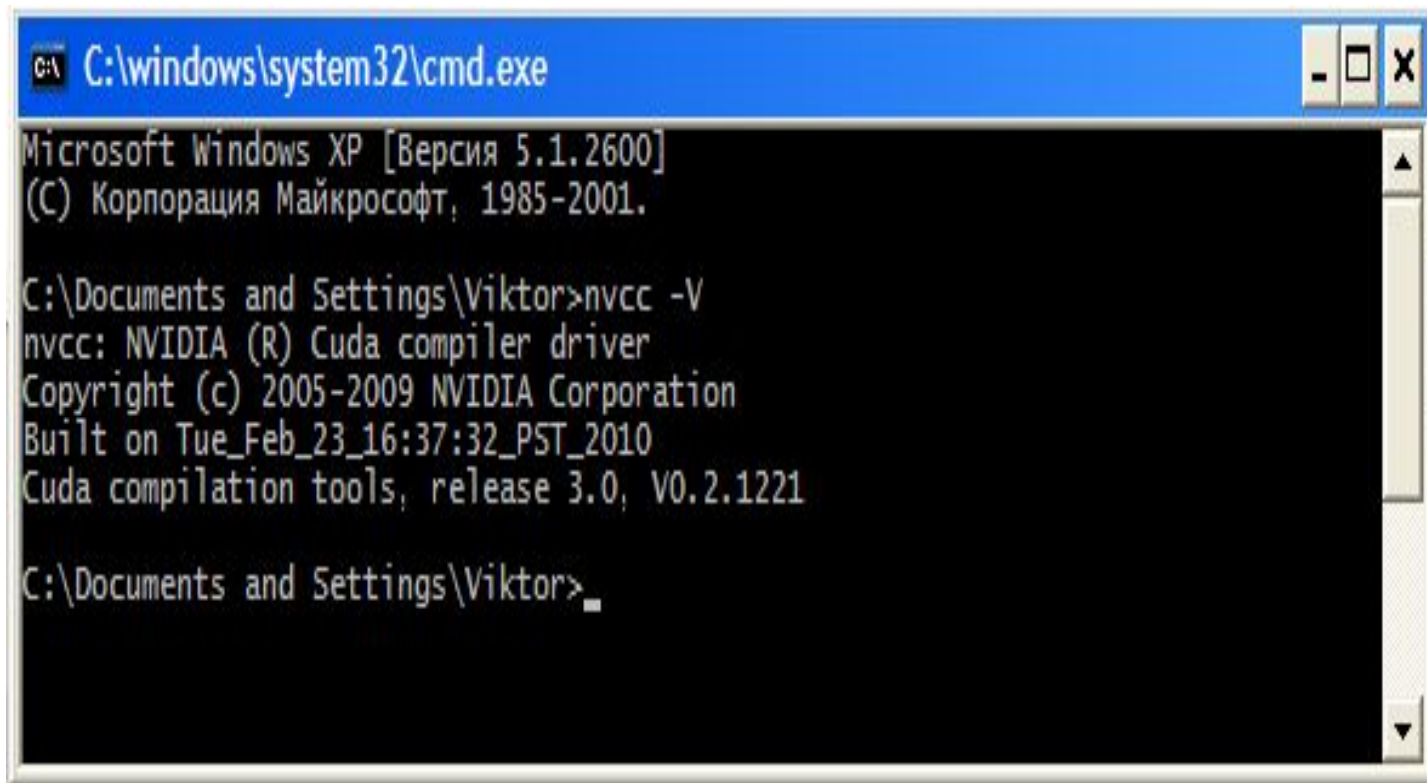


# Проверка компилятора CUDA

- Пуск->Выполнить



# Проверка работы CUDA-компилятора



```
C:\windows\system32\cmd.exe
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\Viktor>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2009 NVIDIA Corporation
Built on Tue_Feb_23_16:37:32_PST_2010
Cuda compilation tools, release 3.0, V0.2.1221

C:\Documents and Settings\Viktor>
```

# Подключение библиотек

- Зайти в **Tools** → **Options** → **Projects and Solutions** → **VC++ Directories**
- Открыть вкладка **Executable files**  
добавить новый параметр и ввести
- C:\Documents and Settings\All Users\Application Data\NVIDIA Corporation\NVIDIA GPU Computing SDK\C\bin
- либо \$(CUDA\_BIN\_PATH)

# Подключение библиотек

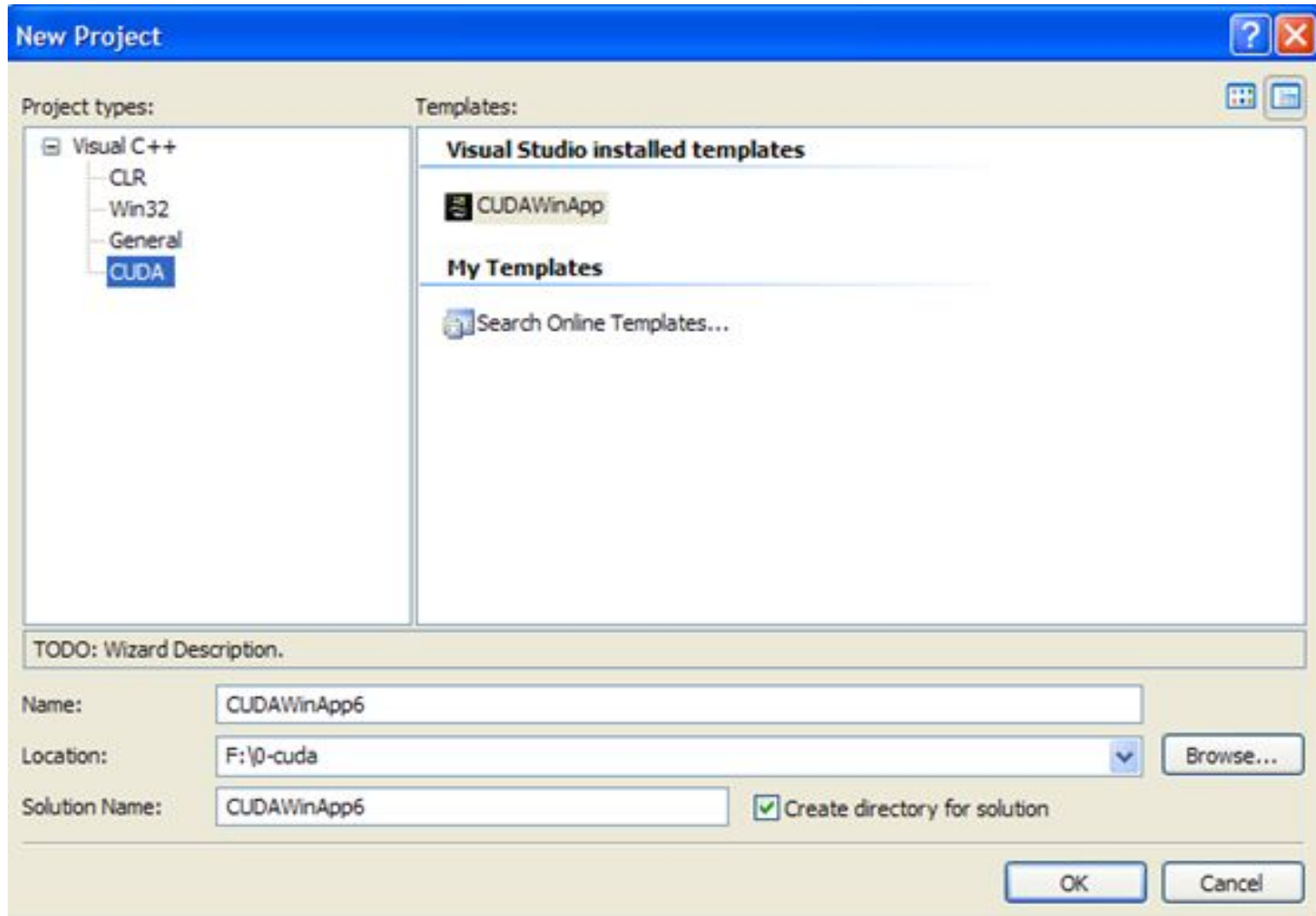
- Открыть вкладку **Include files**  
добавить новый параметр и ввести C:\  
CUDA\include, либо \$(CUDA\_INC\_PATH)  
добавить новый параметр и ввести  
C:\Documents and Settings\All  
Users\Application Data\NVIDIA  
Corporation\NVIDIA GPU Computing  
SDK\C\common\inc, либо  
\$(NVSDKCUDA\_ROOT)\common\inc

# Подключение библиотек

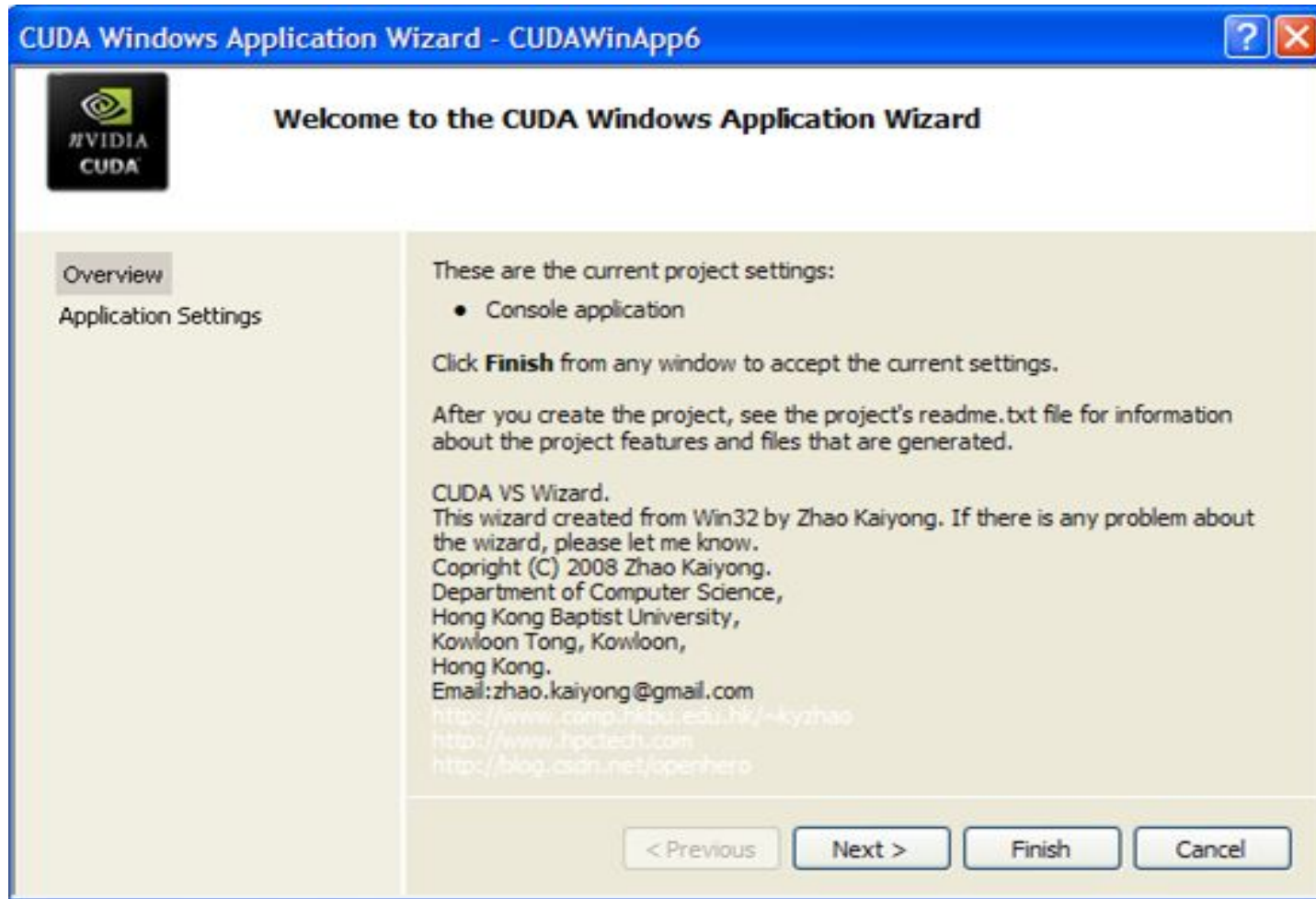
- Открыть вкладку **Library files**
- Добавить новый параметр и ввести `C:\CUDA\lib`, либо `$(CUDA_LIB_PATH)`  
Добавить новый параметр и ввести `C:\Documents and Settings\All Users\Application Data\NVIDIA Corporation\NVIDIA GPU Computing SDK\C\common\lib`, либо `$(NVSDKCUDA_ROOT)\common\lib`



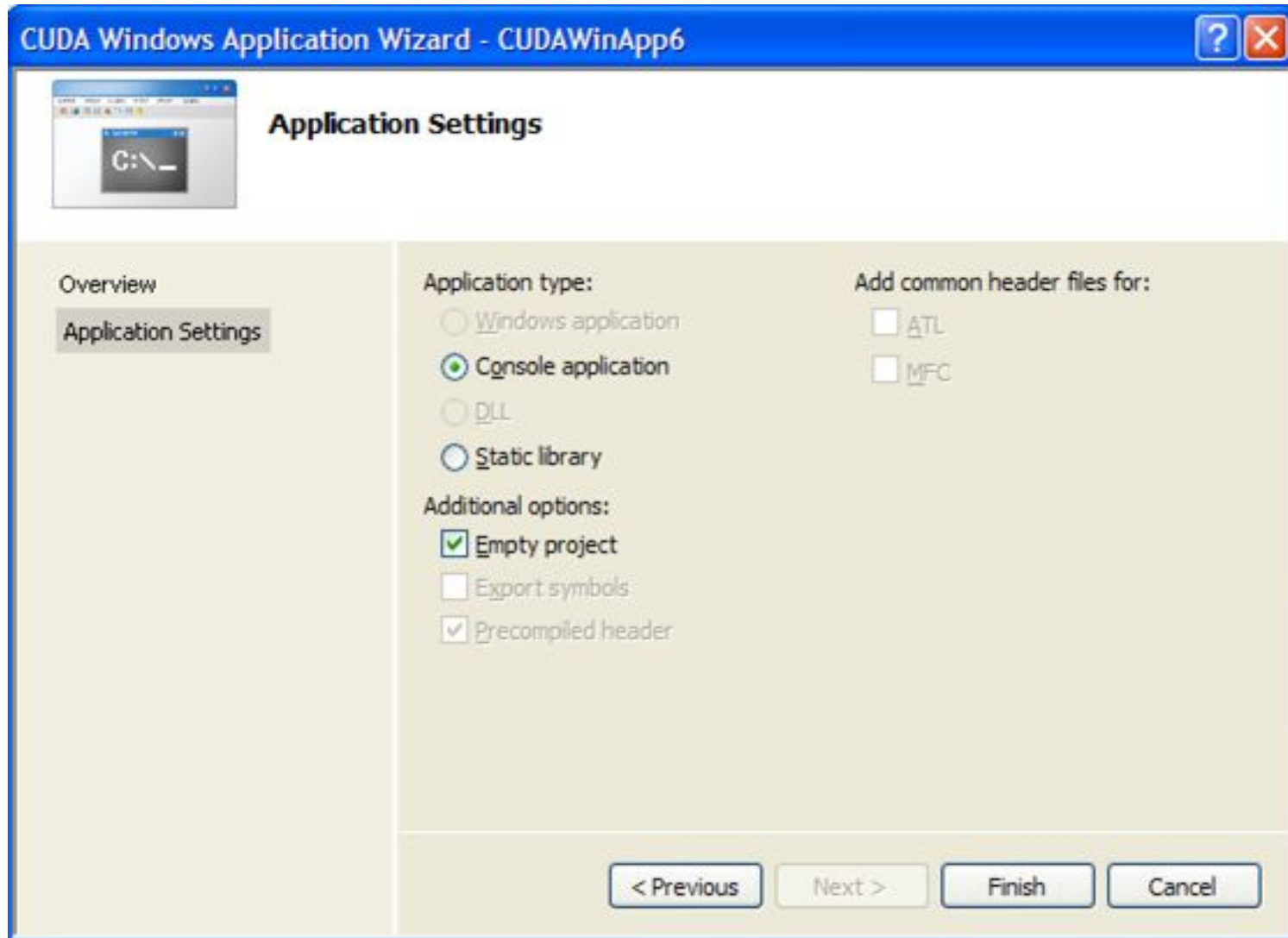
# File → New → Project



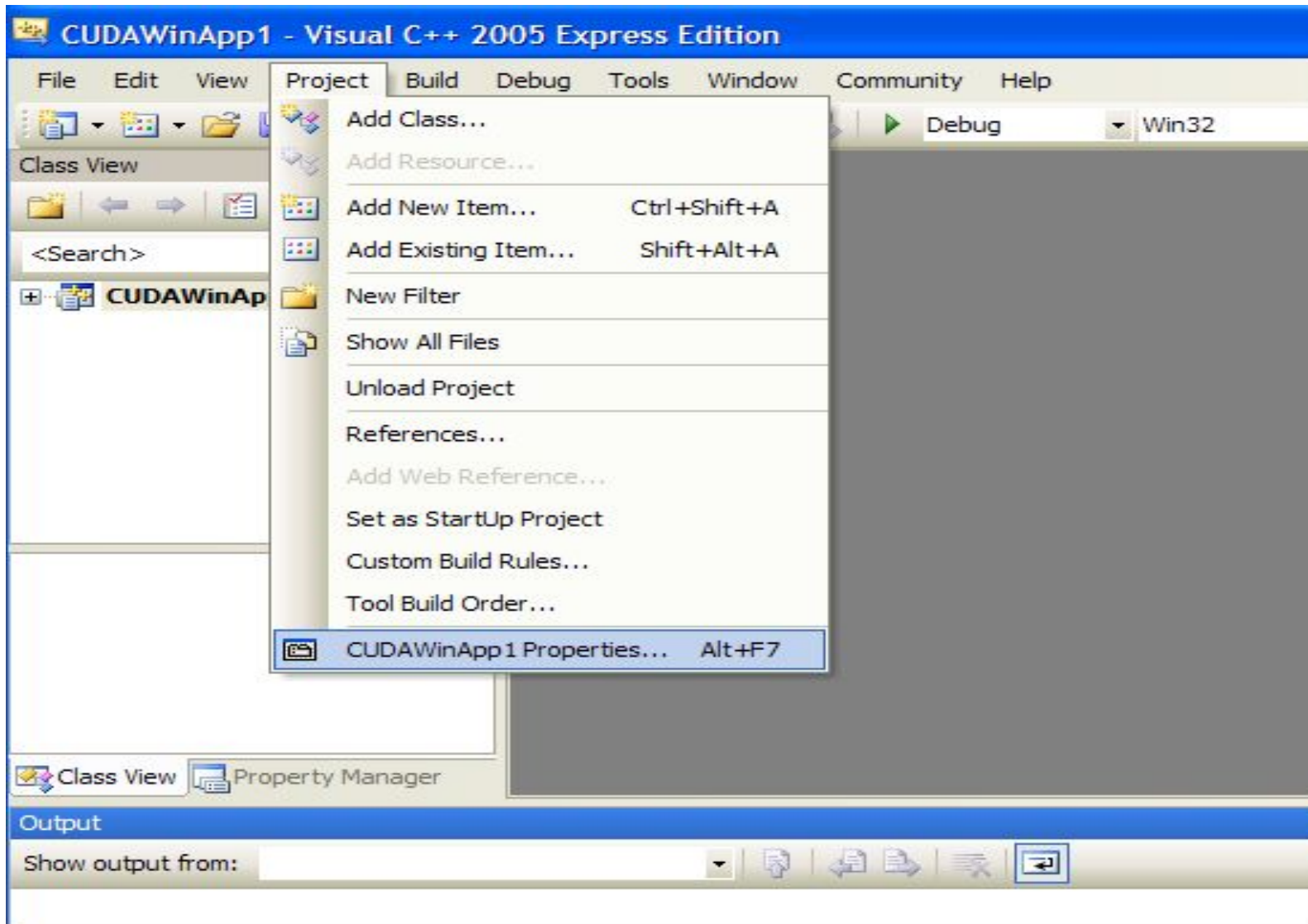
# Наш проект - консольное приложение



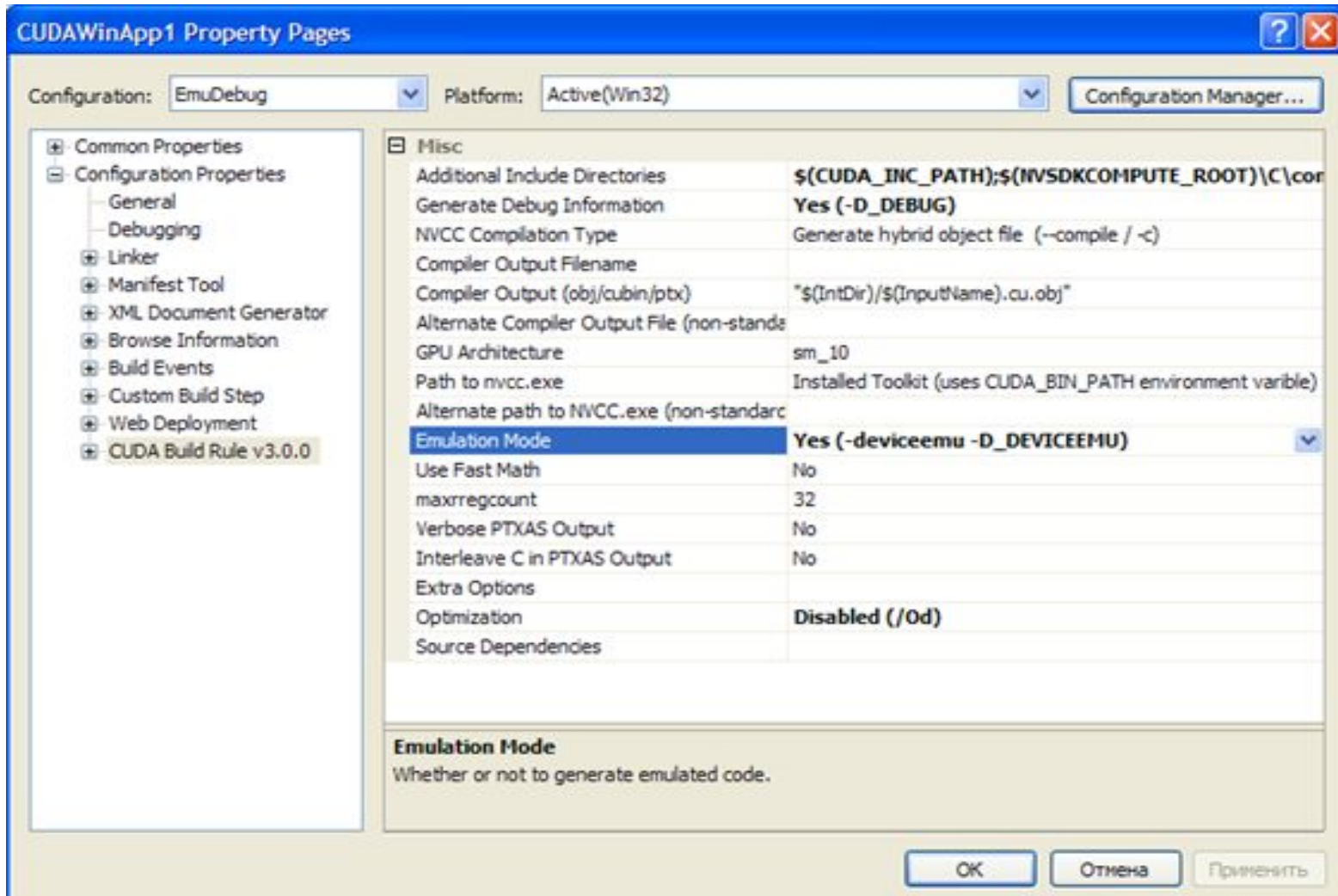
# Пустой проект



# Работа с CUDA в режиме ЭМУЛЯЦИИ



# Работа с CUDA в режиме ЭМУЛЯЦИИ

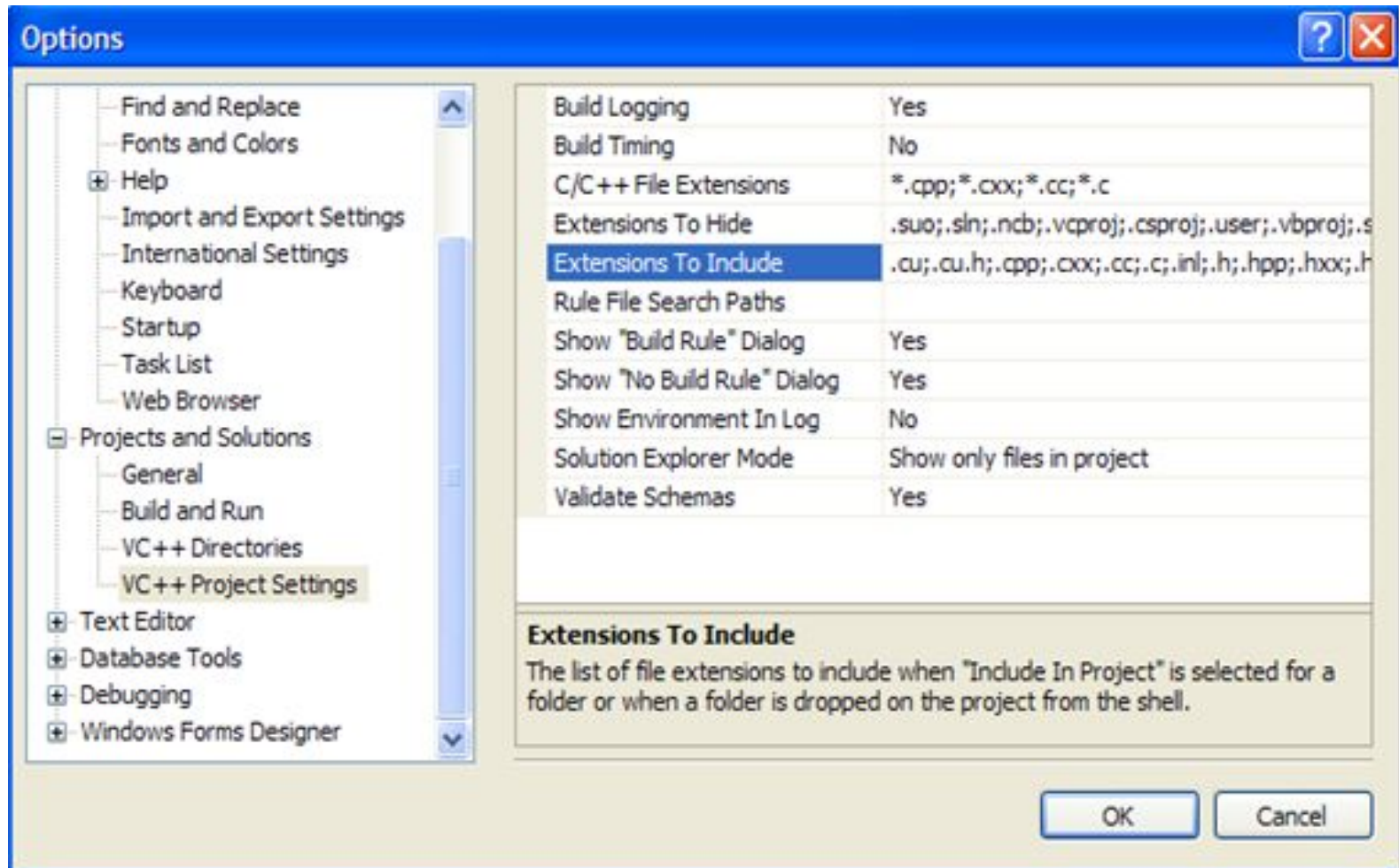


# Интеграция NVidia CUDA с Microsoft Visual Studio

- Для того чтобы файлы CUDA (.cu) корректно включались в наши будущие проекты, в Visual Studio 2010 идём в “Сервис -> Параметры” (Tools → Options), выбираем “Проекты и решения → Параметры проекта VC++” (Projects and Solutions → VC++ Project Settings), ищем строчку “включаемые расширения” (Extensions To Include) и добавляем туда расширения “.cu” и “.cu.h”



# Интеграция NVidia CUDA с Microsoft Visual Studio

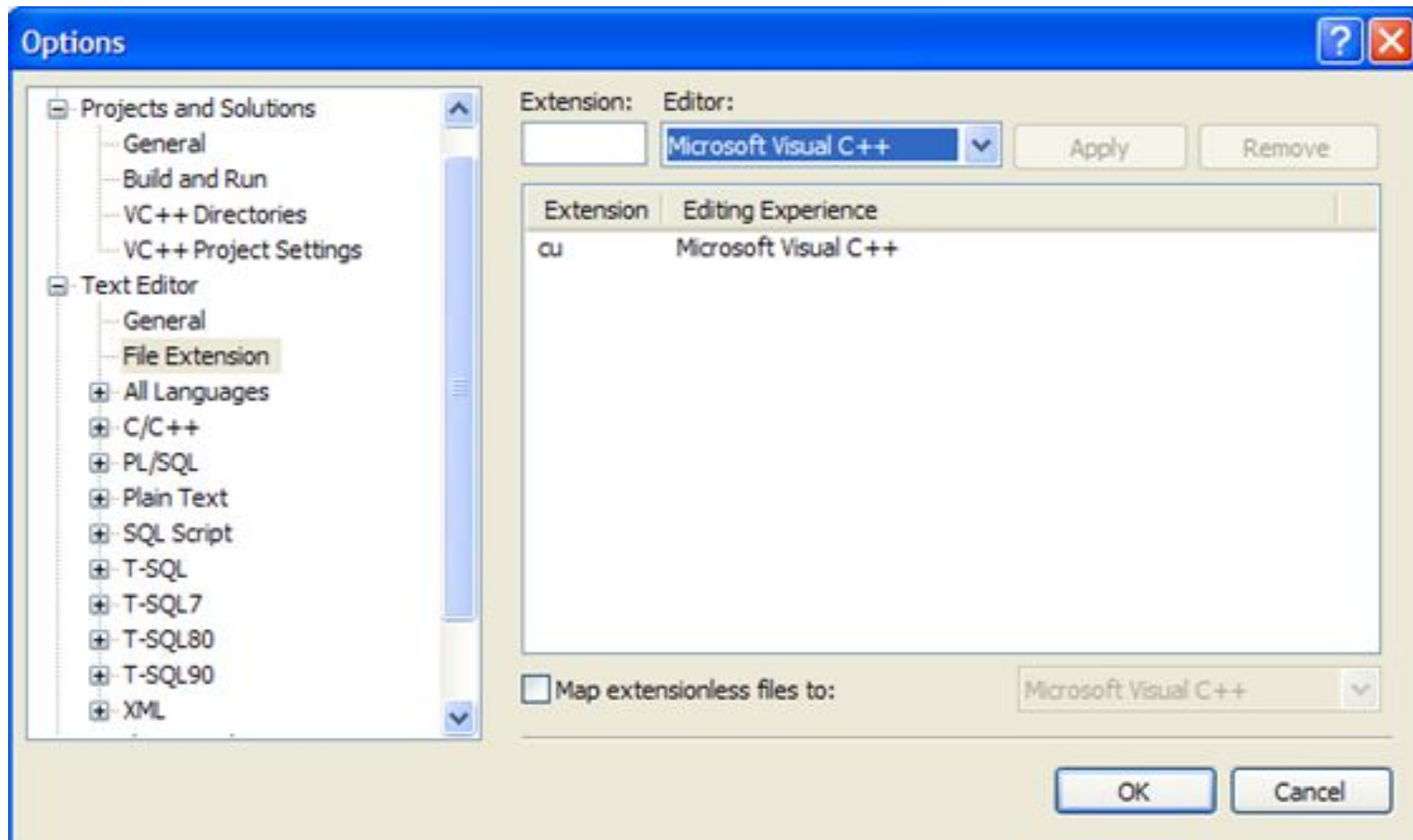




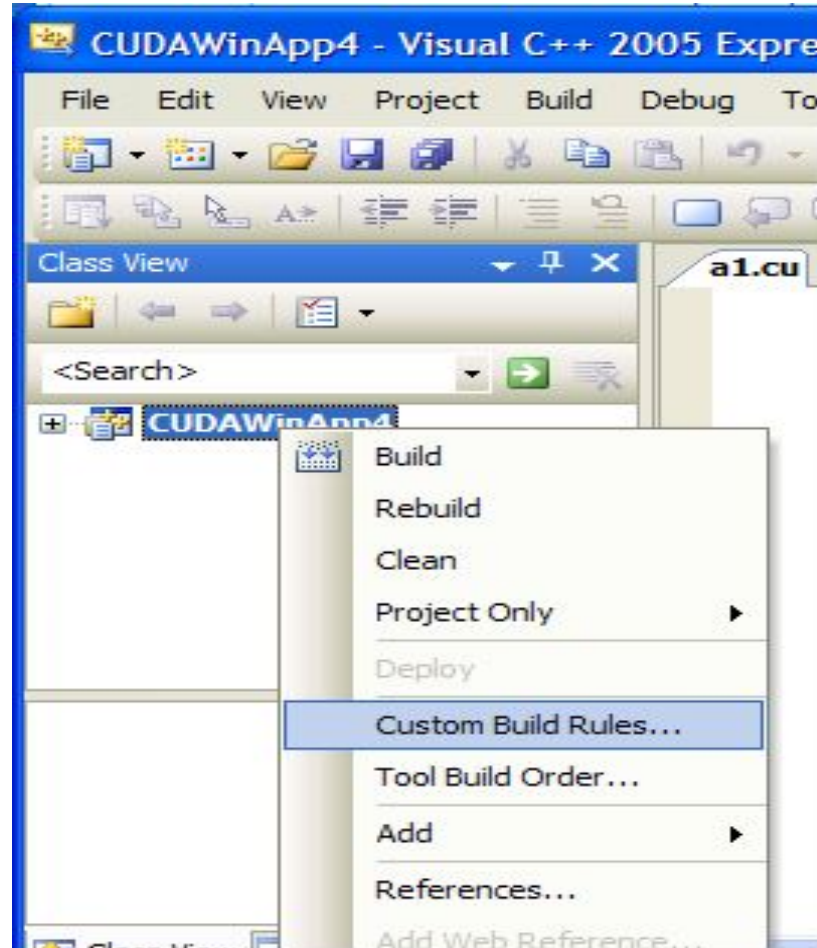
# Интеграция NVidia CUDA с Microsoft Visual Studio

- Сервис → Параметры → Текстовый редактор → “Файловые расширения” (Tools→Options→Text Editor→File Extension), добавить новое расширение “cu” и выбрать Microsoft Visual C++ в качестве редактора кода. Это позволит включить C++-подсветку синтаксиса в .cu-файлах;

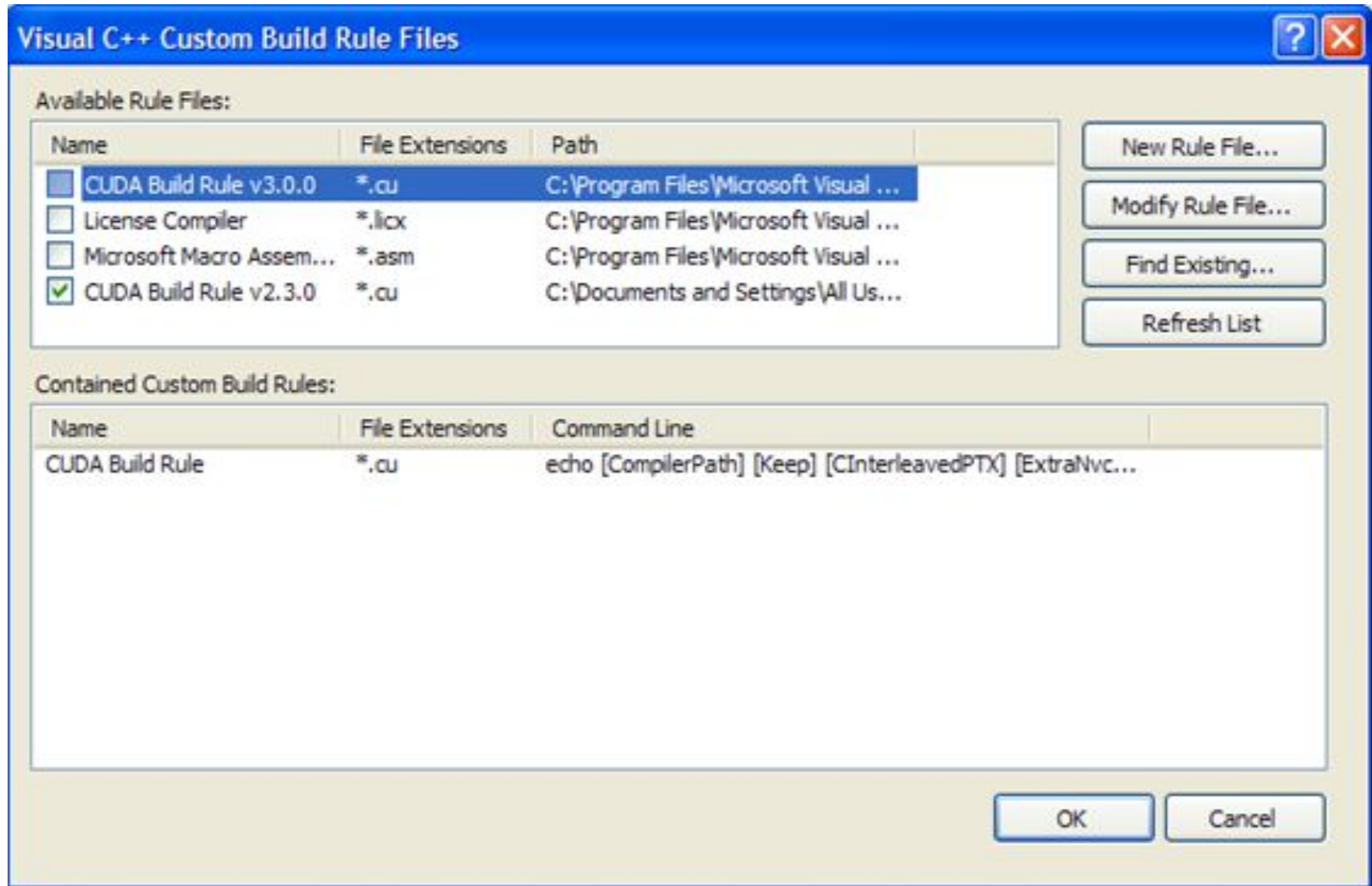
# Интеграция NVidia CUDA с Microsoft Visual Studio



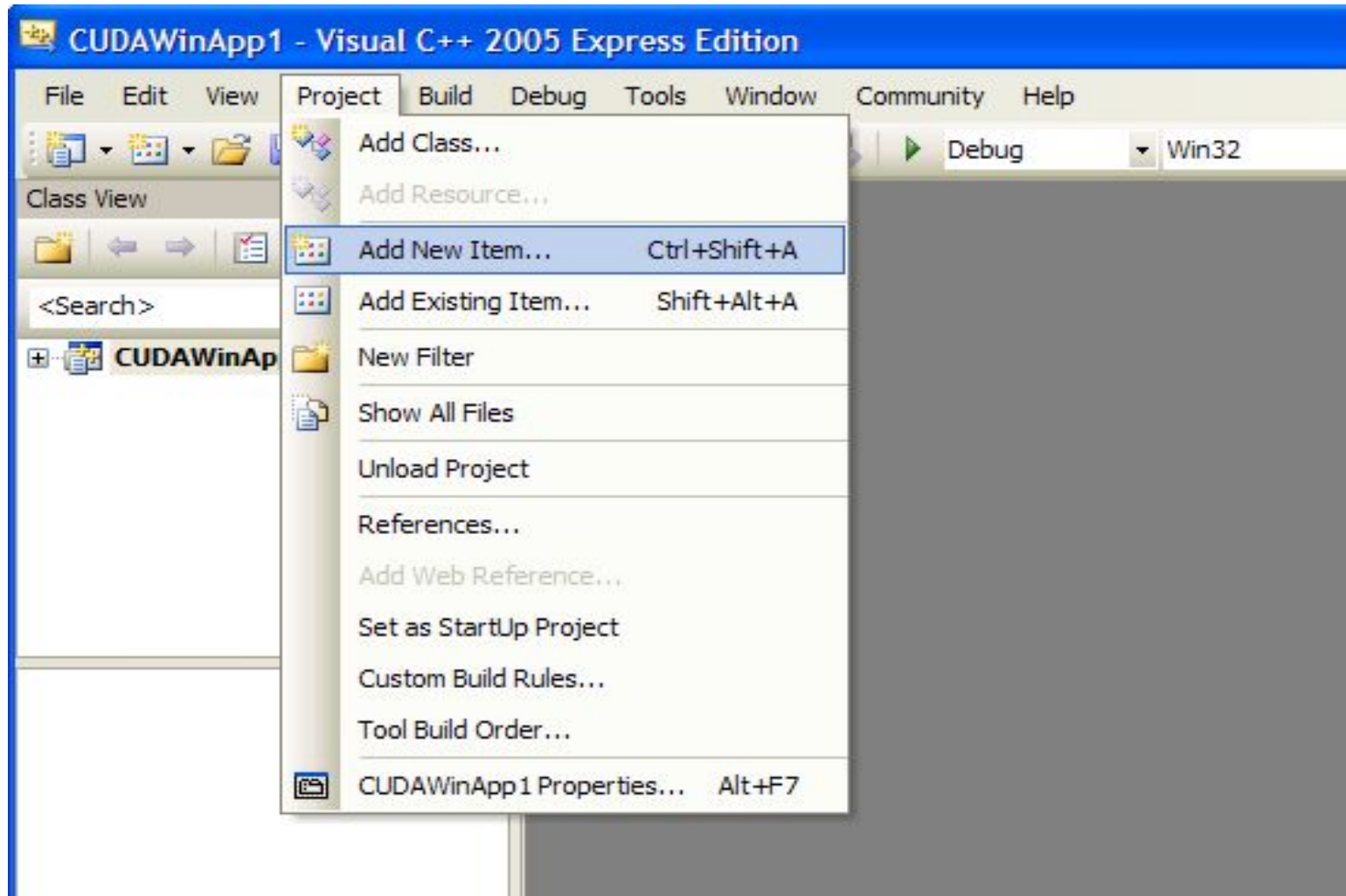
# Выбираем правила построения проекта



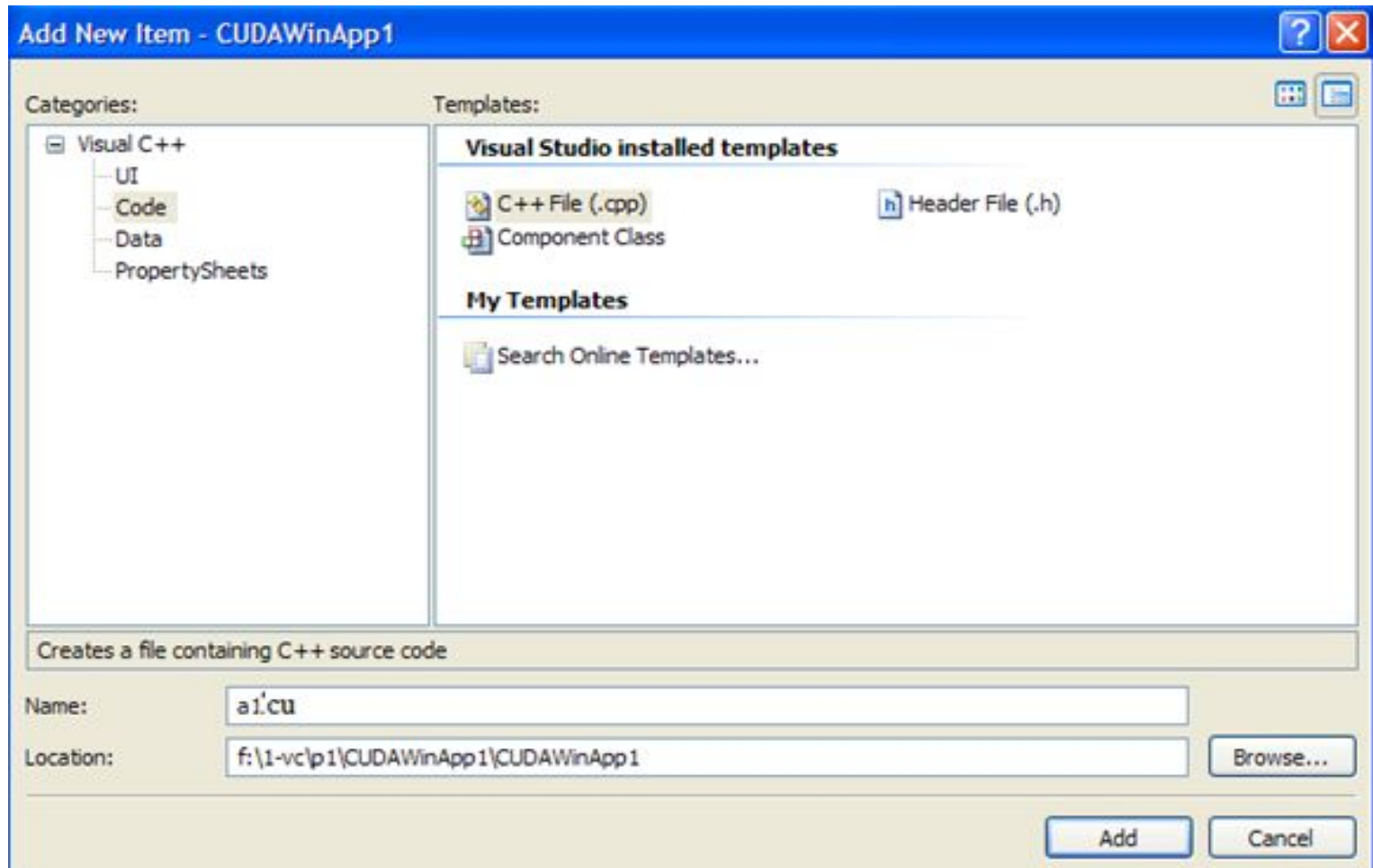
# Устанавливаем CUDA Build Rule



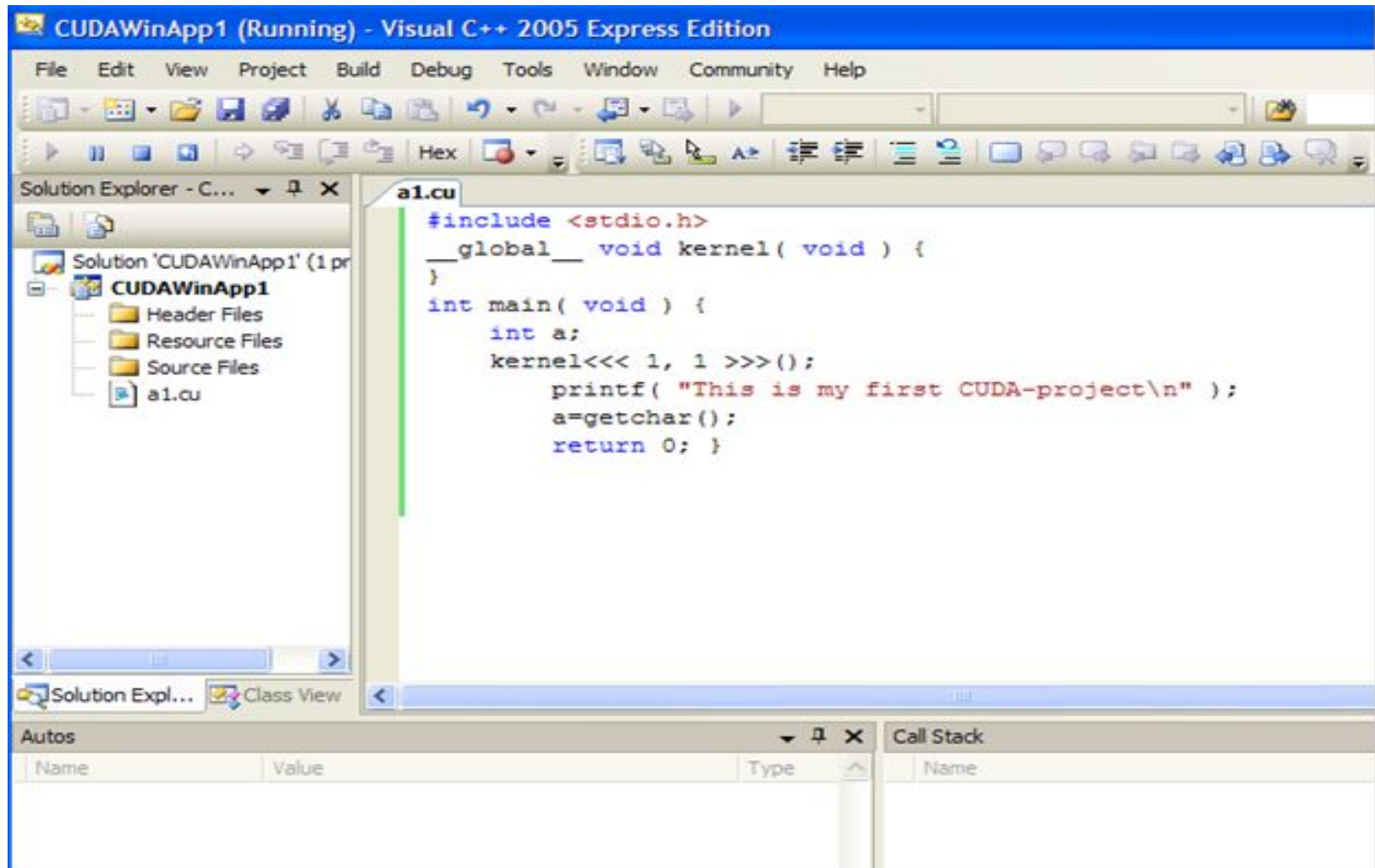
# Ввод программы проекта и его запуск



# Ввод программы проекта

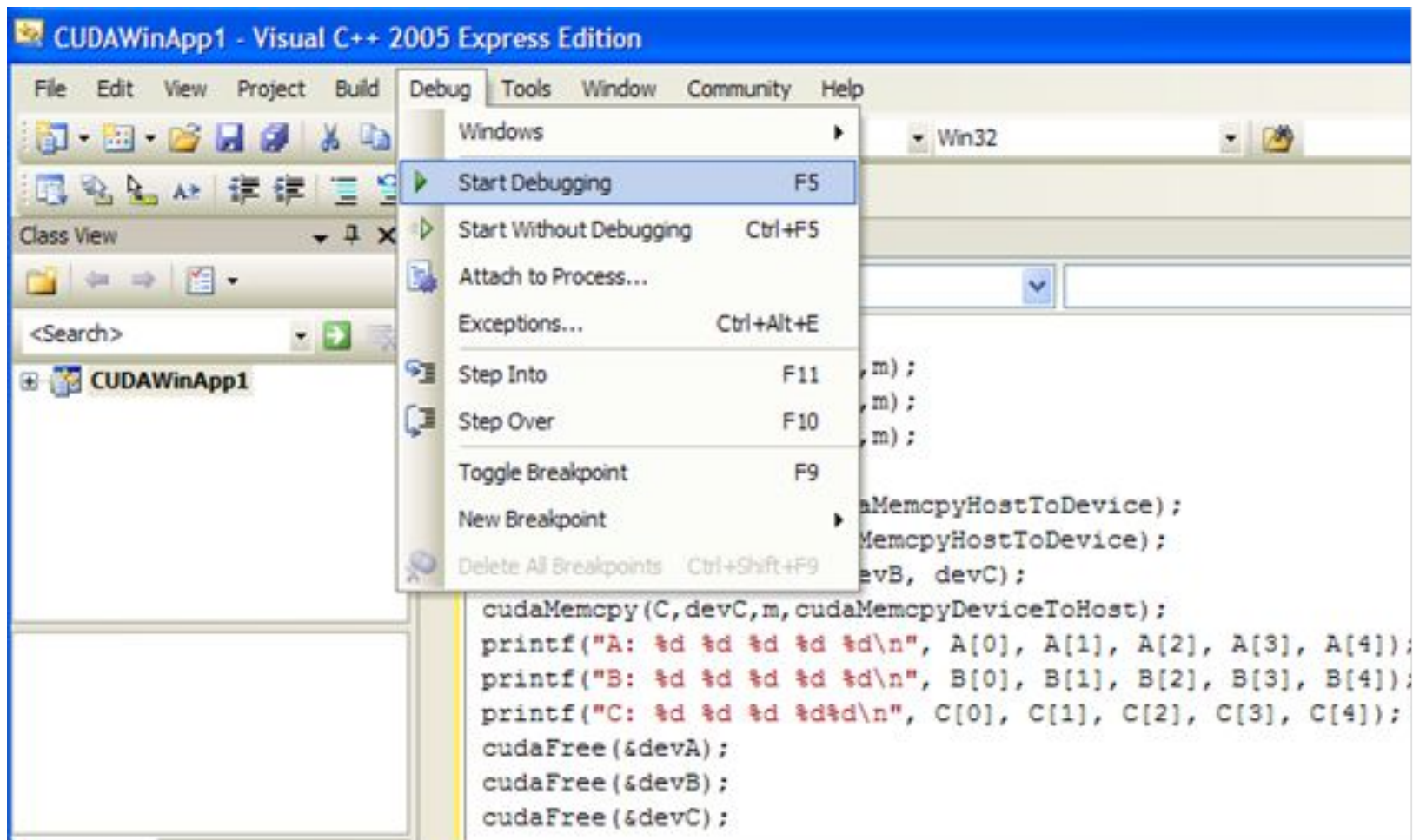


# Ввод программы проекта





# Запуск проекта





# Результаты работы проекта



A screenshot of a Windows command prompt window. The title bar is blue and contains the text "f:\0-cuda\cudawinapp1\debug\CUDAWinApp1.exe" and standard window control buttons (minimize, maximize, close). The main area is black with white text that reads "This is my first CUDA-project".

```
f:\0-cuda\cudawinapp1\debug\CUDAWinApp1.exe
This is my first CUDA-project
```

# CPU передает величины a (2) и b(7) в GPU для их сложения.

- `#include <stdio.h>`
- `__global__ void add( int *a, int *b, int *c ) {`
- `*c = *a + *b;`
- `}`
- 
- `int main( void ) {`
- `int a, b, c;                  // host копии a, b, c`
- `int *dev_a, *dev_b, *dev_c;  // device копии of a, b, c`
- `int size = sizeof( int );`
- `//выделяем память для device копий для a, b, c`
- `cudaMalloc( (void**)&dev_a, size );`
- `cudaMalloc( (void**)&dev_b, size );`
- `cudaMalloc( (void**)&dev_c, size );`
- `a = 2;`
- `b = 7;`

# CPU передает величины a (2) и b(7) в GPU для их сложения

- `// копируем ввод на device`
- `cudaMemcpy( dev_a, &a, size, cudaMemcpyHostToDevice );`
- `cudaMemcpy( dev_b, &b, size, cudaMemcpyHostToDevice );`
- `// запускаем add() kernel на GPU, передавая параметры`
- `add<<< 1, 1 >>>( dev_a, dev_b, dev_c );`
- `// copy device result back to host copy of c`
- `cudaMemcpy( &c, dev_c, size, cudaMemcpyDeviceToHost );`
- `cudaFree( dev_a );`
- `cudaFree( dev_b );`
- `cudaFree( dev_c );`
- `printf("%d",c);`
- `a=getchar();`
- `return 0;`
- `}`

# Результаты работы



```
C:\ f:\0-cuda\CUDAWinApp4\debug\CUDAWinApp4.exe
9
```

The image shows a screenshot of a Windows command prompt window. The title bar is orange and contains the text "C:\ f:\0-cuda\CUDAWinApp4\debug\CUDAWinApp4.exe". The main area of the window is white and contains the number "9" on the first line. The window has a black border.

# Вычисление суммы элементов двух массивов (Один блок с N НИТЯМИ)

- `#include <stdio.h>`
- `#include <stdlib.h>`
- `//#include <cuda_runtime.h>`
- `//#include <cutil.h>`
- 
- `//#if __DEVICE_EMULATION__ bool InitCUDA(void)`
- `//{return true;}`
- `//#else`
- `bool InitCUDA(void)`
- `{ int count = 0; int i = 0;`
- `cudaGetDeviceCount(&count);`
- `if(count == 0) { fprintf(stderr, "There is no device.\n"); return false; }`
- `for(i = 0; i < count; i++)`
- `{ cudaDeviceProp prop;`
- `if(cudaGetDeviceProperties(&prop, i) == cudaSuccess)`
- `{ if(prop.major >= 1)`
- `{ break; }`
- `}`
- `}`
- `}`

# Вычисление суммы элементов двух массивов (Один блок с N НИТЯМИ)

- `if(i == count) {fprintf(stderr, "There is no device supporting CUDA.\n"); return false; }`
- `cudaSetDevice(i);`
- `printf("CUDA initialized.\n"); return true;`
- `}`
- `//#endif`
- `// Определение ядра`
- `__global__ void VecAdd(int *A, int *B, int *C)`
- `{ int i = threadIdx.x; C[i] = A[i] + B[i]; }`
- 
- `int main(int argc, char* argv[])`
- `{ // if(!InitCUDA()) {return 0;}`
- `int N=5; int m=5*sizeof(int);`
- `int A[5]={1,2,3,4,5}, B[5]={6,7,8,9,10}, C[5];`
- `int *devA=NULL;`
- `int *devB=NULL;`
- `int *devC=NULL;`
- `cudaMalloc((void**)&devA,m);`
- `cudaMalloc((void**)&devB,m);`
- `cudaMalloc((void**)&devC,m);`

# Вычисление суммы элементов двух массивов (Один блок с N НИТЯМИ)

- //Вызов ядра N потоками
- cudaMemcpy(devA,A,m,cudaMemcpyHostToDevice);
- cudaMemcpy(devB,B,m,cudaMemcpyHostToDevice);
- VecAdd<<<1, N>>>(devA, devB, devC);
- cudaMemcpy(C,devC,m,cudaMemcpyDeviceToHost);
- printf("A: %d %d %d %d %d\n", A[0], A[1], A[2], A[3], A[4]);
- printf("B: %d %d %d %d %d\n", B[0], B[1], B[2], B[3], B[4]);
- printf("C: %d %d %d %d %d\n", C[0], C[1], C[2], C[3], C[4]);
- cudaFree(&devA);
- cudaFree(&devB);
- cudaFree(&devC);
- int c=getchar();
- return 0;}



# Результаты работы $C=A+B$

```
c:\ f:\0-cuda\cudawinapp5\emudebug\CUDAWinApp5.exe
A: 1 2 3 4 5
B: 6 7 8 9 10
C: 7 9 11 13 15
-
```

# Конец обзора

- Можно приступать к работе