

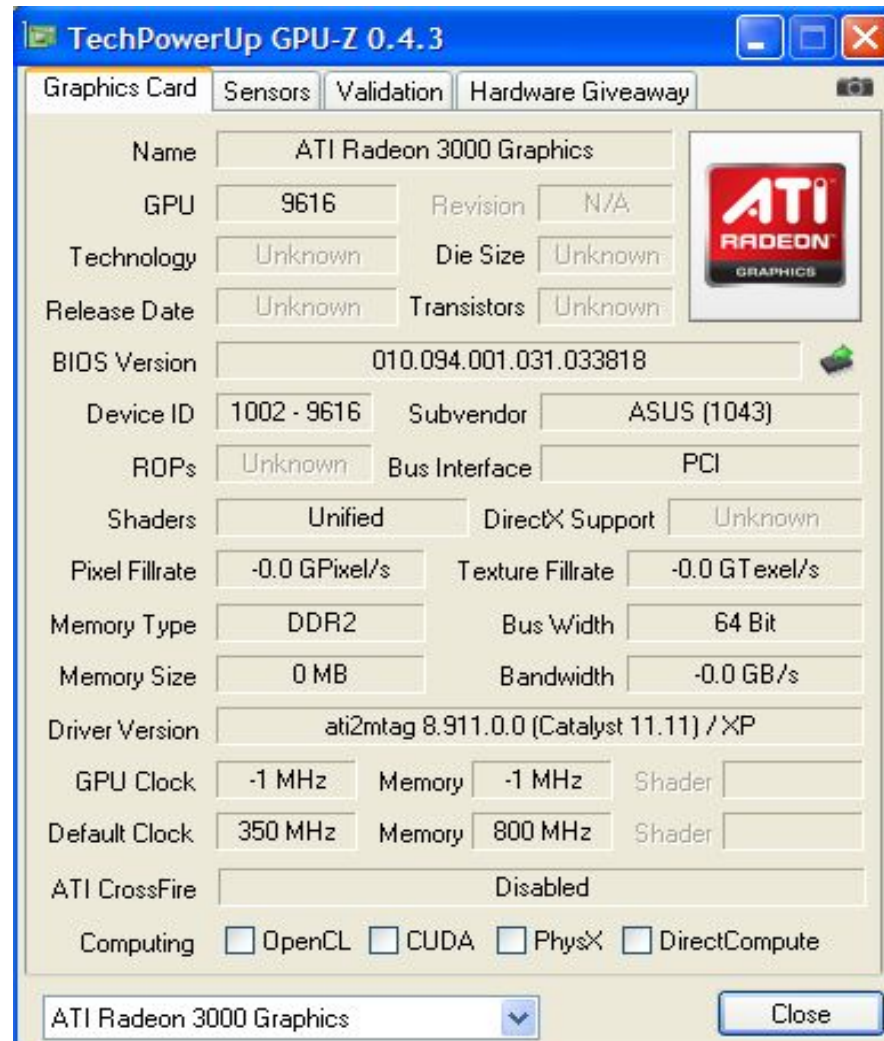
РАБОТА С CUDA ТЕХНОЛОГИЕЙ

Необходимые системы.

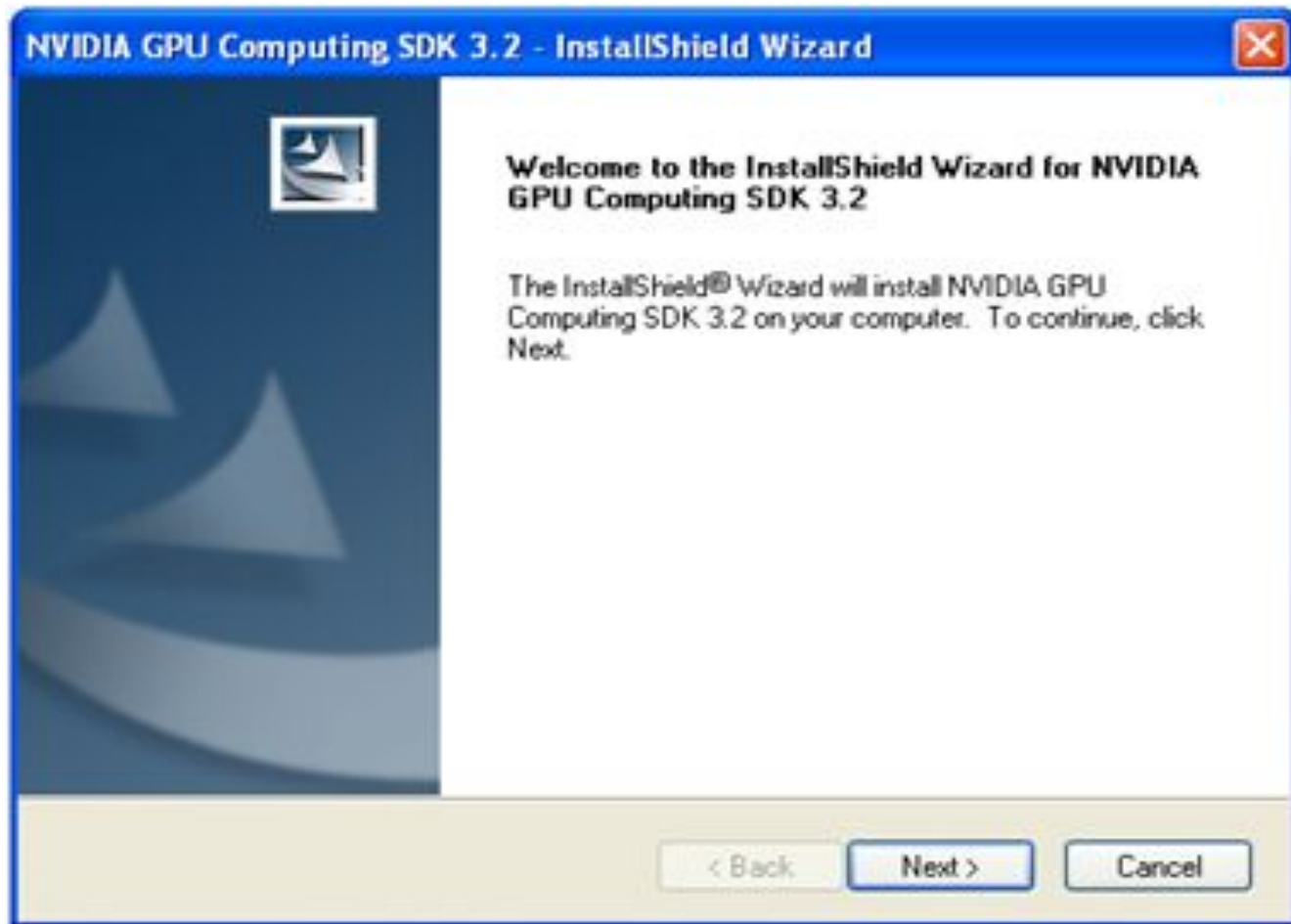
1. GPU-Z Проверка видеокарты
2. Microsoft Visual Studio 2008
3. NVIDIA CUDA SDK 2.3
4. NVIDIA CUDA Toolkit 2.3
5. CUDA_VS_Wizard - засіб налаштування Visual Studio;
6. NVIDIA Developer Drivers – драйвери відеокарти;

Проверка Видеокарты

- Запускаем программу GPU-Z.exe



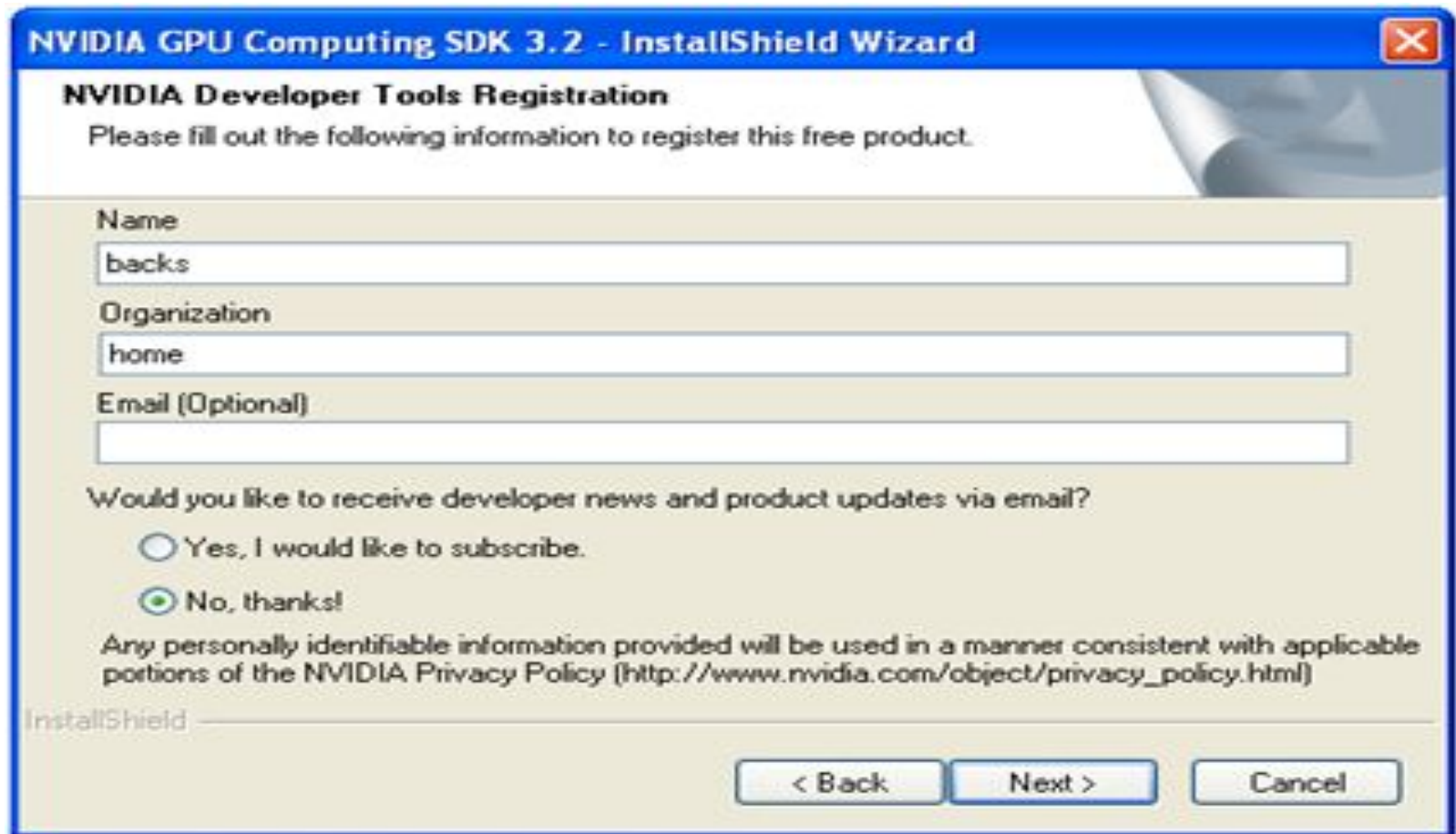
Порядок встановлення NVIDIA CUDA SDK



Порядок встановлення NVIDIA CUDA SDK



Порядок встановлення NVIDIA CUDA SDK



The screenshot shows a Windows-style dialog box titled "NVIDIA GPU Computing SDK 3.2 - InstallShield Wizard". The main heading is "NVIDIA Developer Tools Registration". Below this, it says "Please fill out the following information to register this free product." There are three text input fields: "Name" with the value "backs", "Organization" with the value "home", and "Email (Optional)" which is empty. Below the fields are two radio buttons for the question "Would you like to receive developer news and product updates via email?". The first option is "Yes, I would like to subscribe." and the second, selected option is "No, thanks!". At the bottom, there is a line of text: "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)". The bottom of the window features three buttons: "< Back", "Next >", and "Cancel". The "InstallShield" logo is visible in the bottom left corner.

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)

InstallShield

< Back Next > Cancel

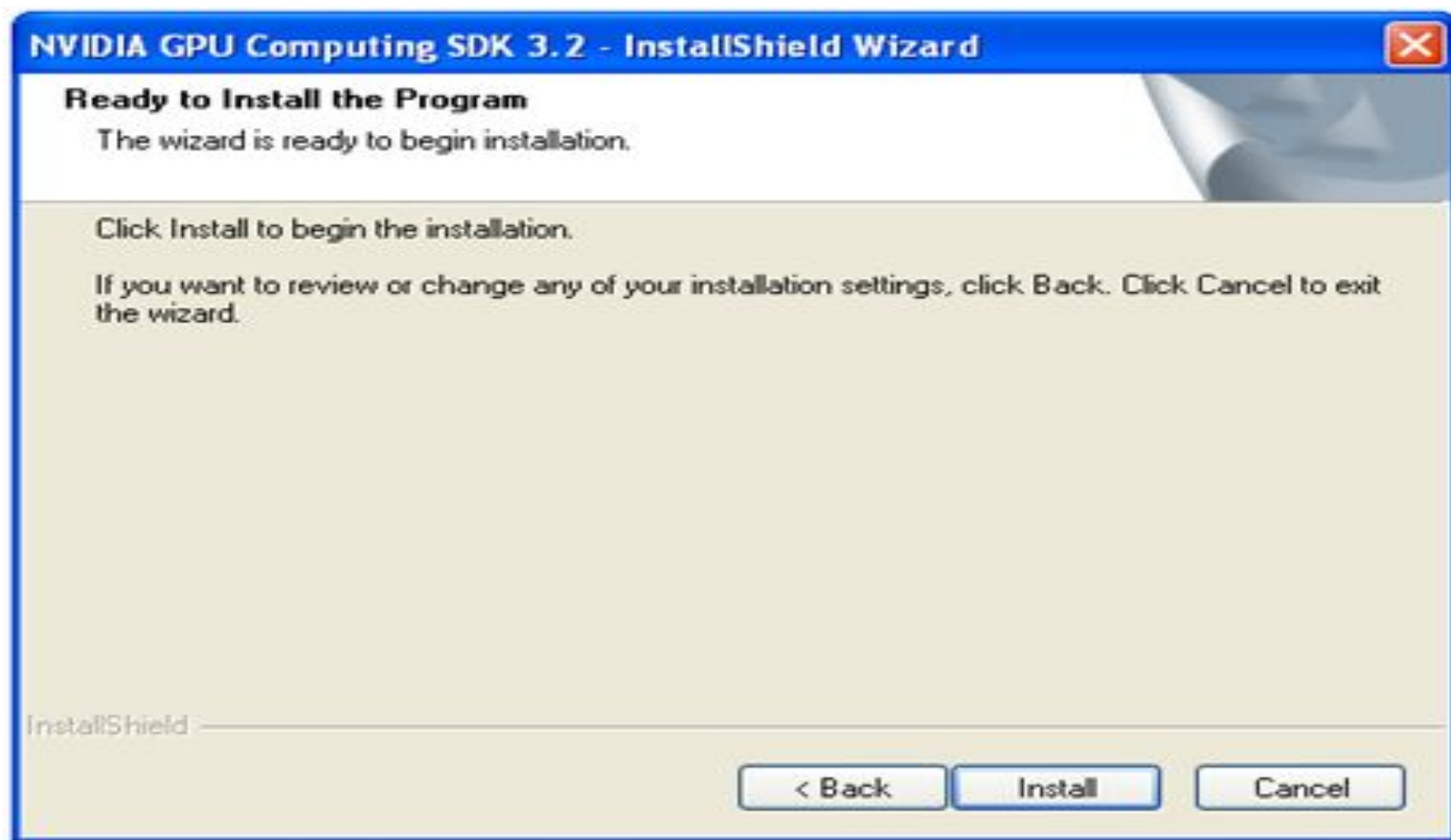
Порядок встановлення NVIDIA CUDA SDK



Порядок встановлення NVIDIA CUDA SDK



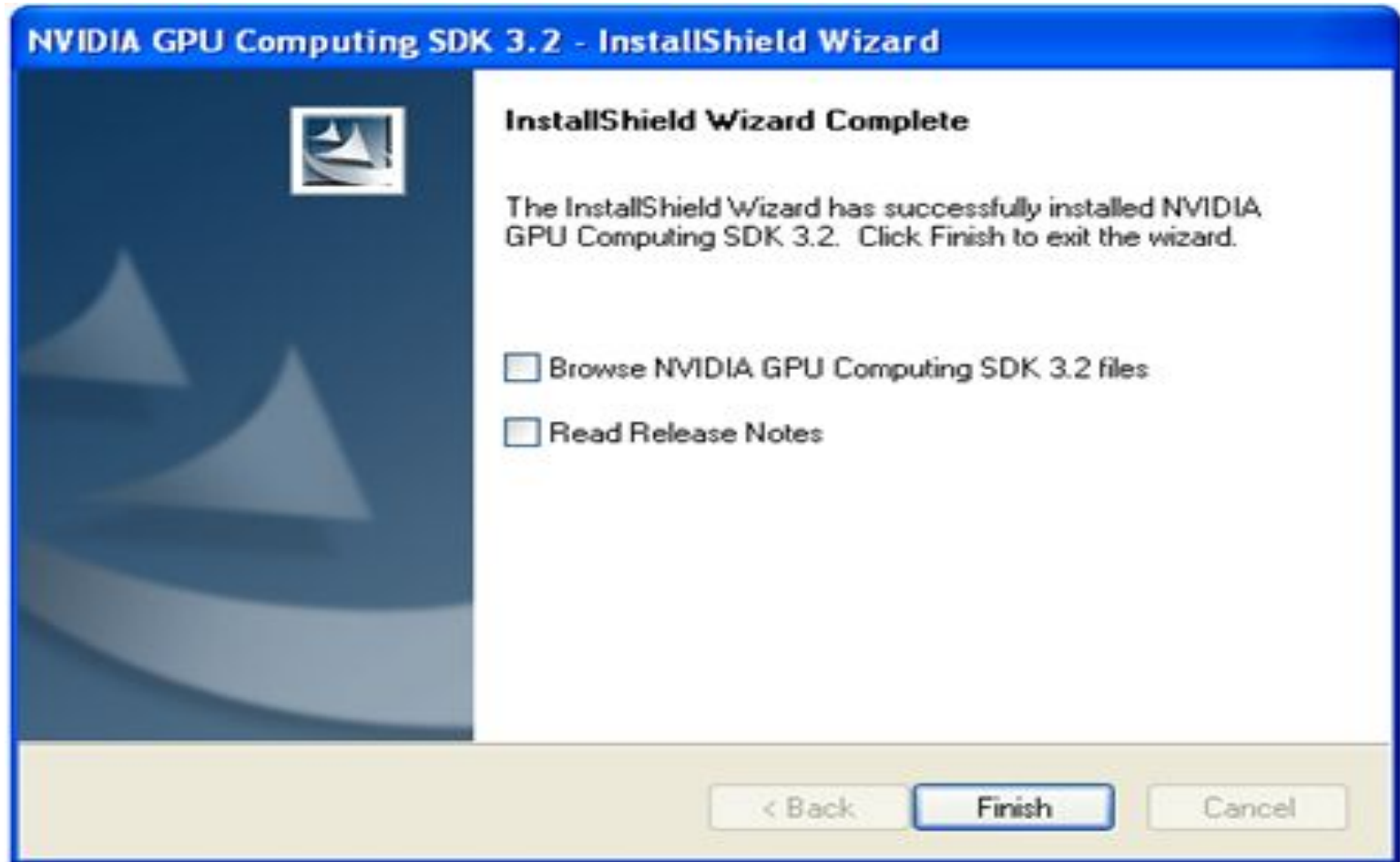
Порядок встановлення NVIDIA CUDA SDK



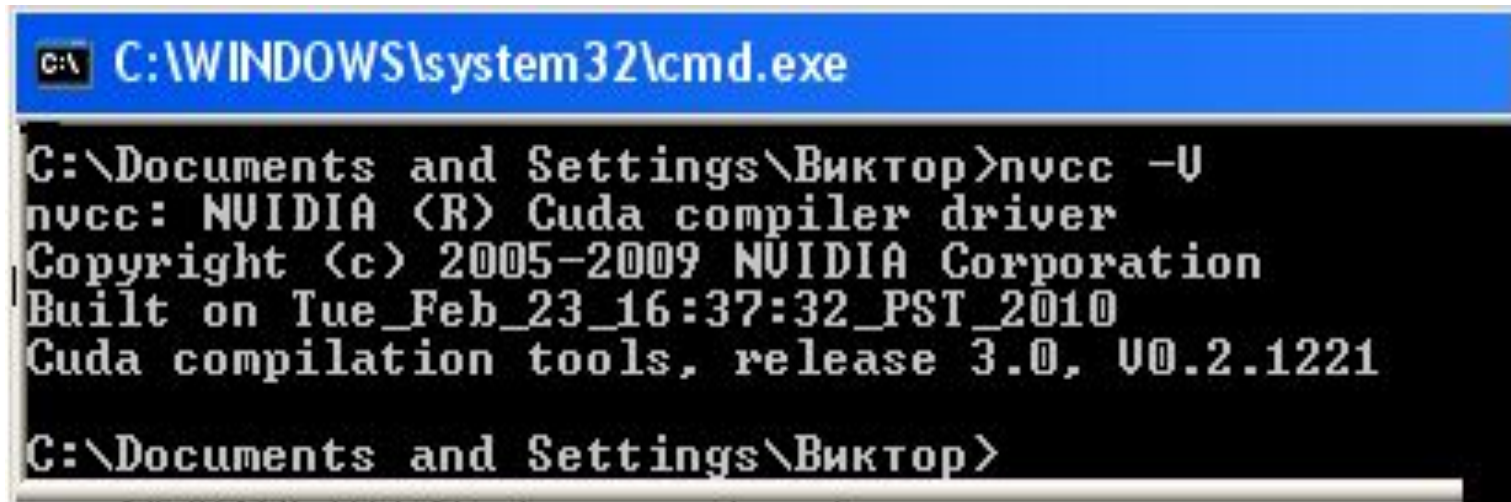
Порядок встановлення NVIDIA CUDA SDK



Порядок встановлення NVIDIA CUDA SDK



Проверяем установлен ли компилятор nvcc



```
C:\WINDOWS\system32\cmd.exe

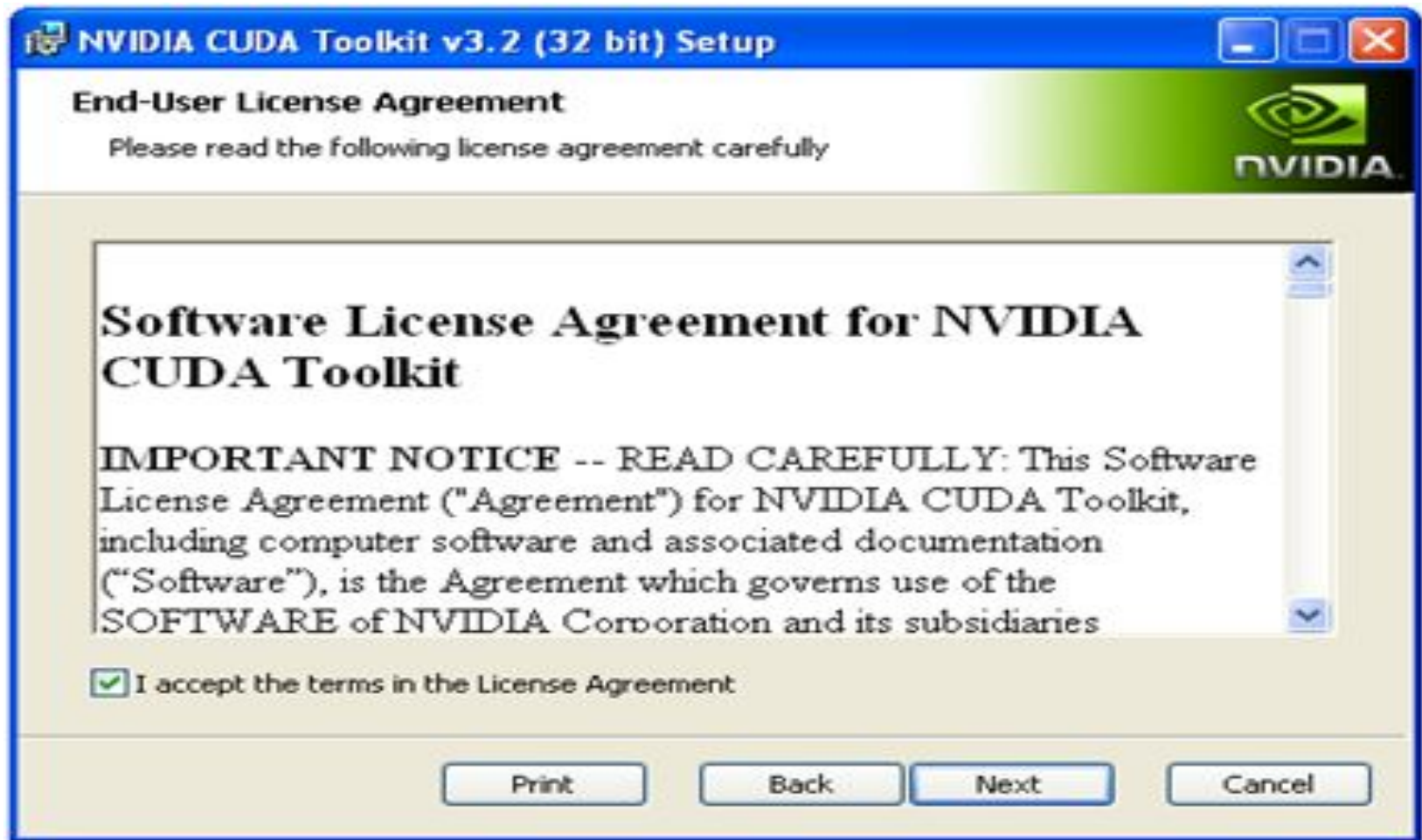
C:\Documents and Settings\Виктор>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2009 NVIDIA Corporation
Built on Tue_Feb_23_16:37:32_PST_2010
Cuda compilation tools, release 3.0, V0.2.1221

C:\Documents and Settings\Виктор>
```

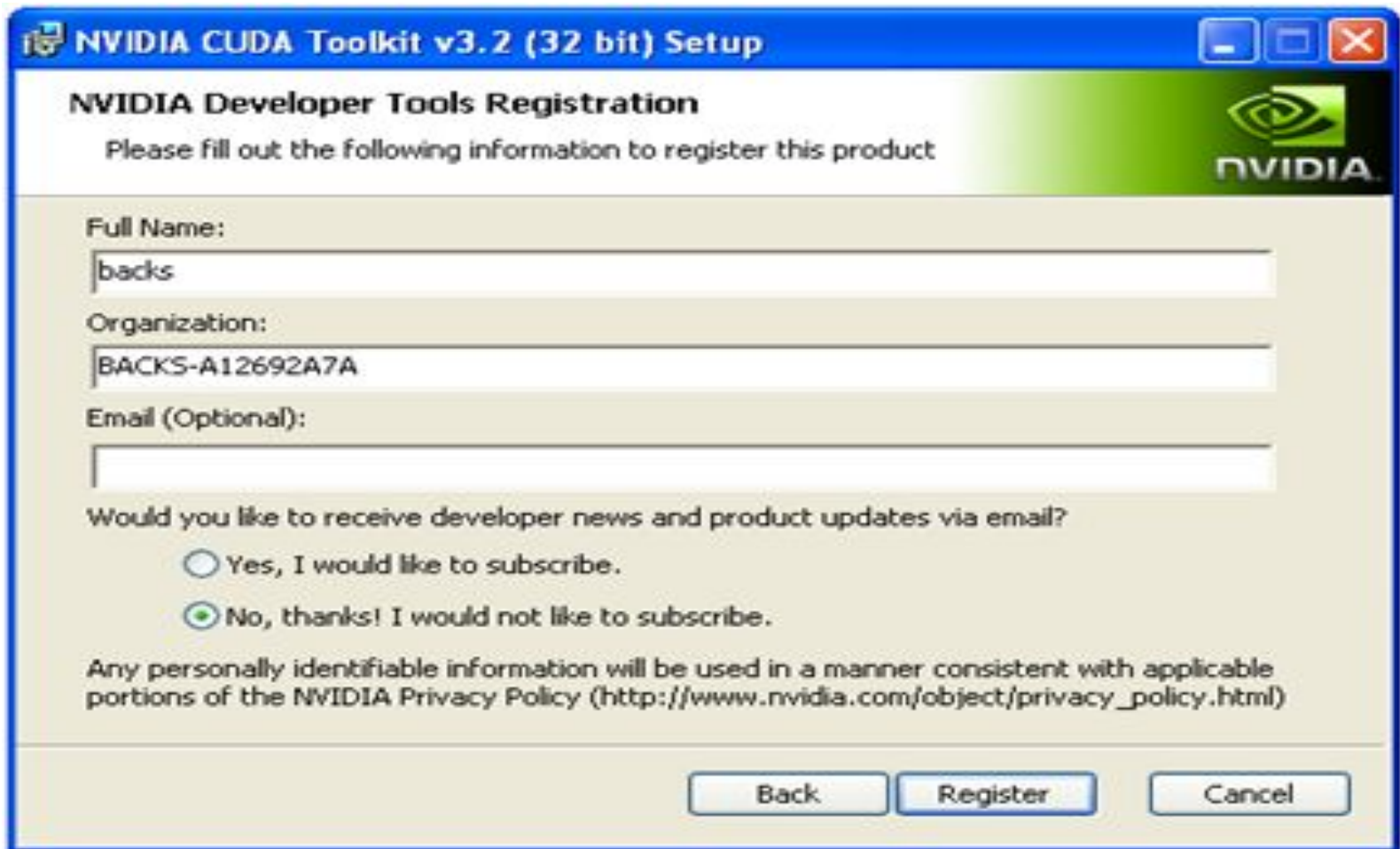
Встановлення і налаштування NVIDIA CUDA Toolkit



Встановлення і налаштування NVIDIA CUDA Toolkit



Встановлення і налаштування NVIDIA CUDA Toolkit



The screenshot shows a Windows-style installation window titled "NVIDIA CUDA Toolkit v3.2 (32 bit) Setup". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area has a light beige background. At the top, there's a green header bar with the NVIDIA logo on the right. Below the header, the text "NVIDIA Developer Tools Registration" is displayed, followed by the instruction "Please fill out the following information to register this product". There are three text input fields: "Full Name:" with the text "backs", "Organization:" with the text "BACKS-A12692A7A", and "Email (Optional):" which is empty. Below these fields is a question: "Would you like to receive developer news and product updates via email?". There are two radio button options: "Yes, I would like to subscribe." (which is unselected) and "No, thanks! I would not like to subscribe." (which is selected). At the bottom of the window, there is a line of text: "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)". At the very bottom, there are three buttons: "Back", "Register", and "Cancel".

NVIDIA CUDA Toolkit v3.2 (32 bit) Setup

NVIDIA Developer Tools Registration

Please fill out the following information to register this product

Full Name:
backs

Organization:
BACKS-A12692A7A

Email (Optional):

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

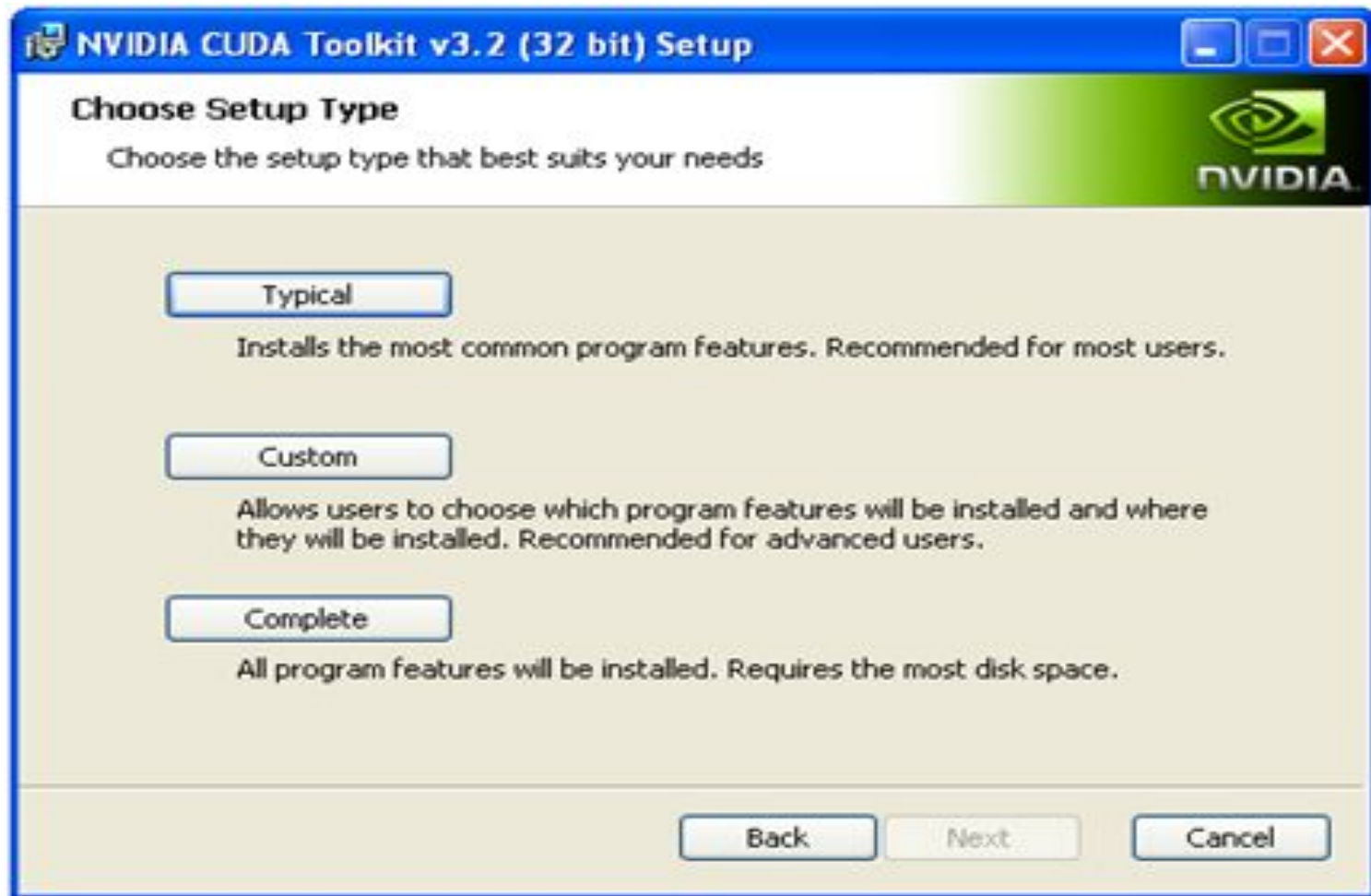
☐ Yes, I would like to subscribe.

☒ No, thanks! I would not like to subscribe.

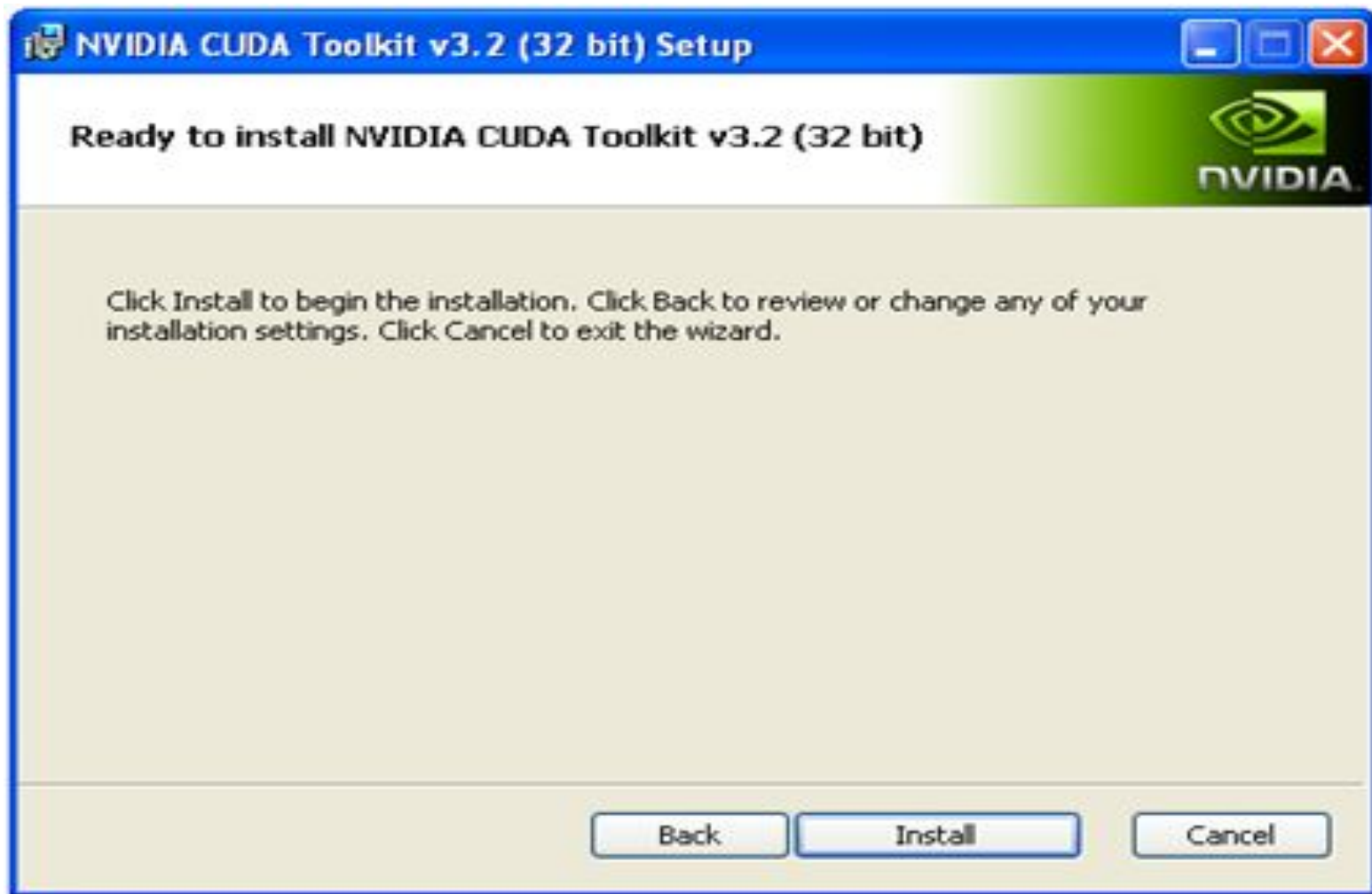
Any personally identifiable information will be used in a manner consistent with applicable portions of the NVIDIA Privacy Policy (http://www.nvidia.com/object/privacy_policy.html)

Back Register Cancel

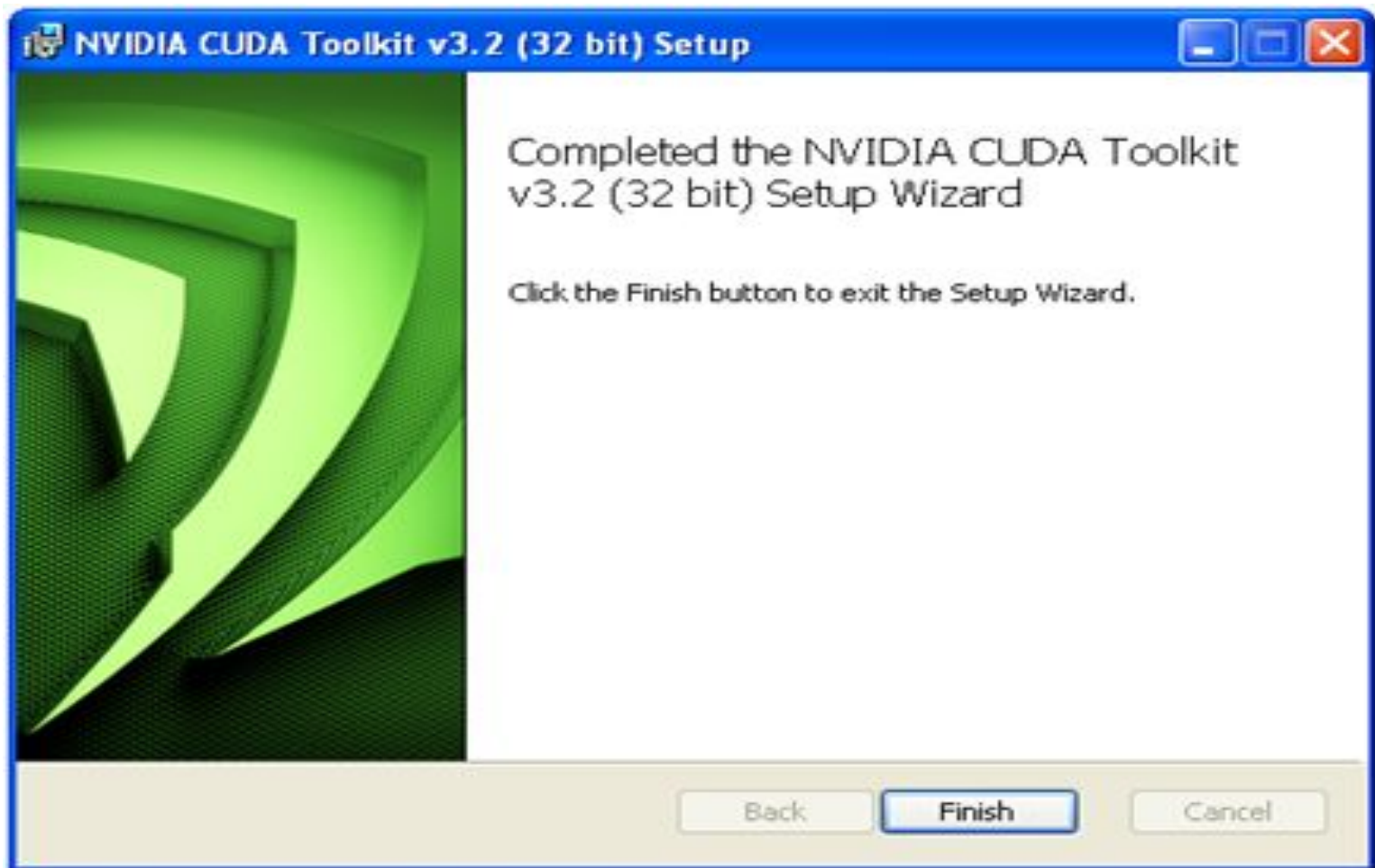
Встановлення і налаштування NVIDIA CUDA Toolkit



Встановлення і налаштування NVIDIA CUDA Toolkit

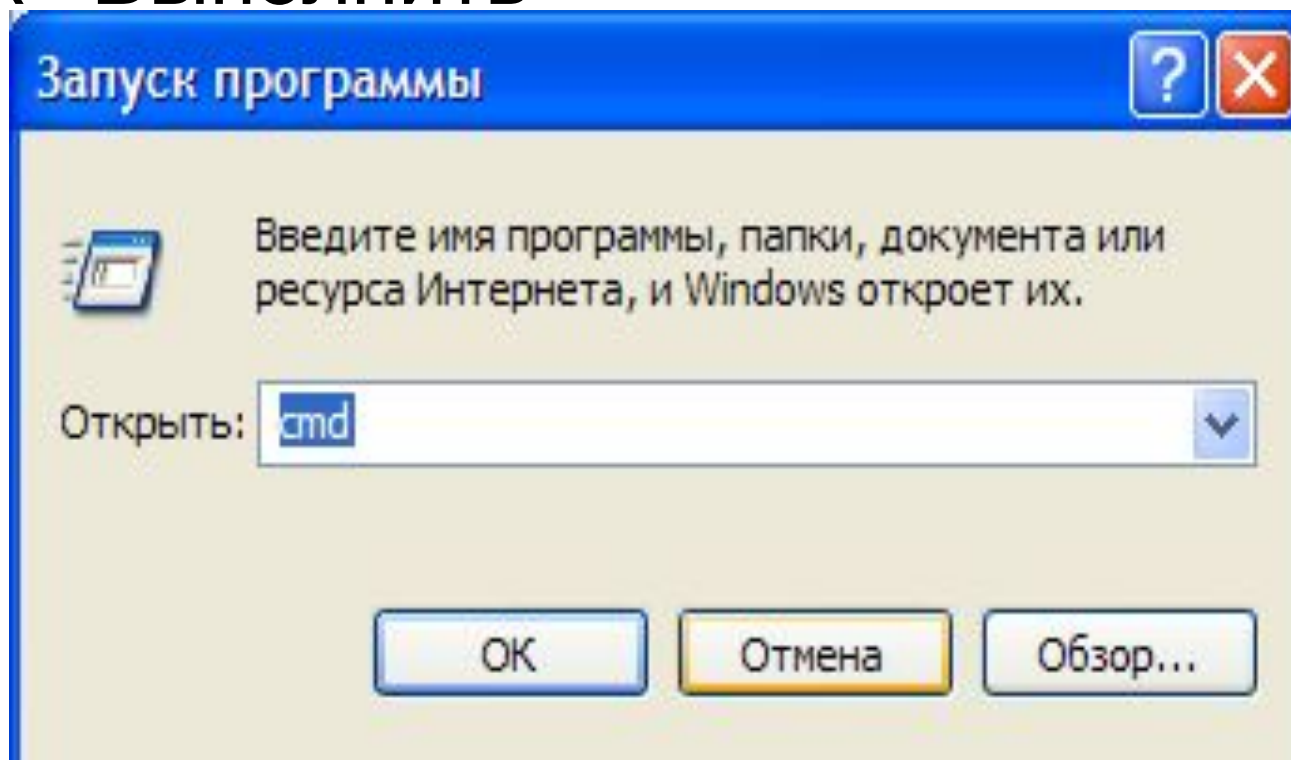


Встановлення і налаштування NVIDIA CUDA Toolkit

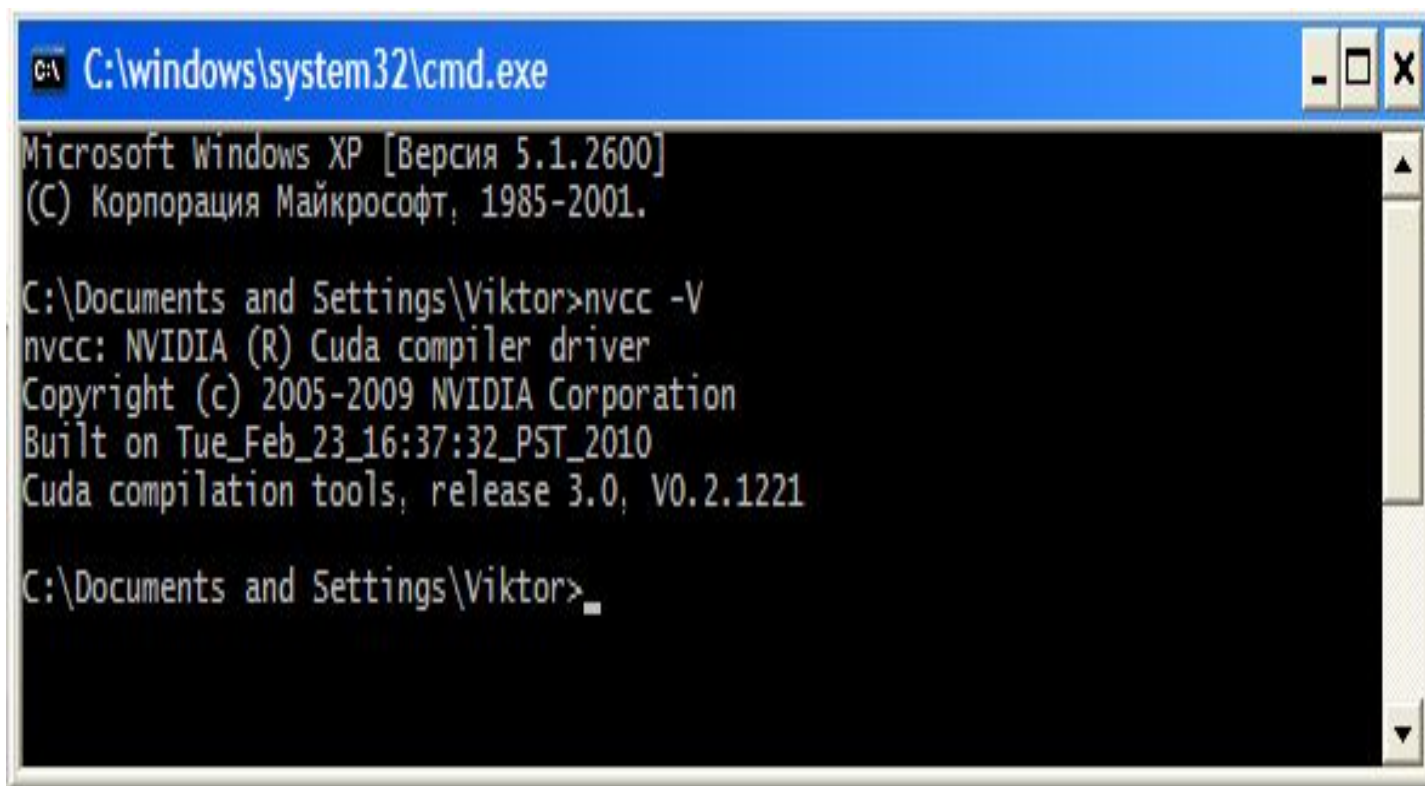


Проверка компилятора CUDA

- 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)

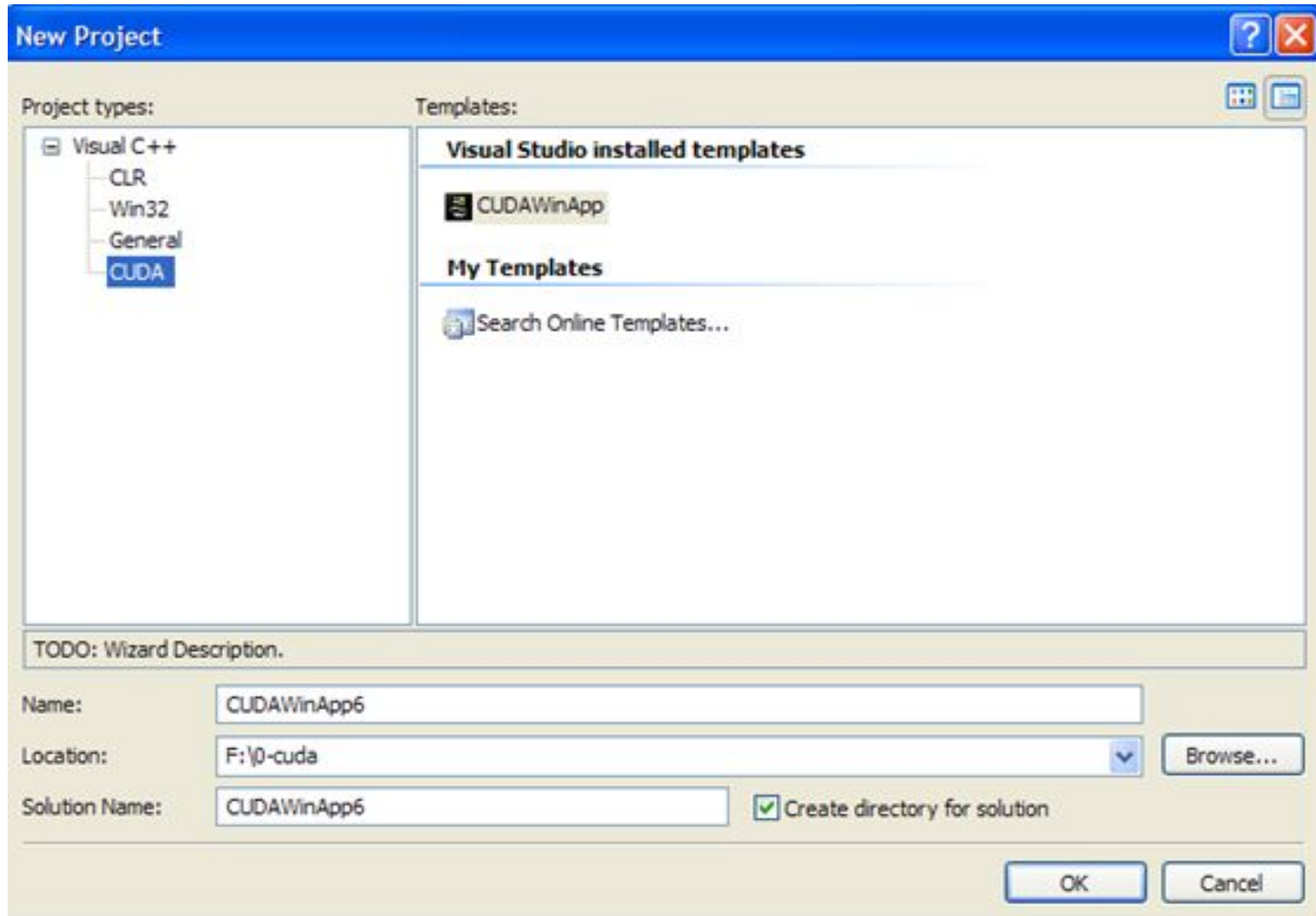
Подключение библиотек

- 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

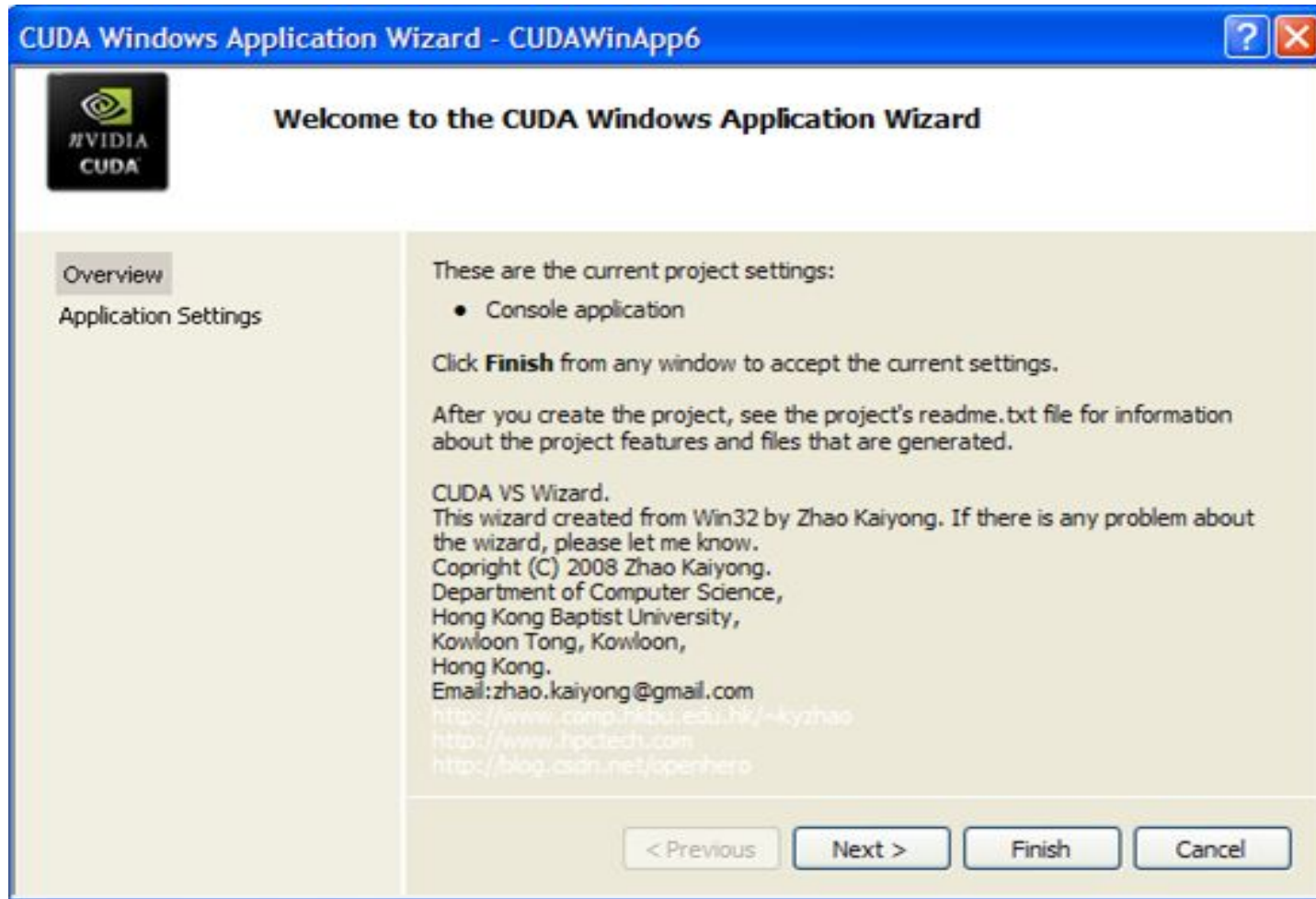
Подключение библиотек

- 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

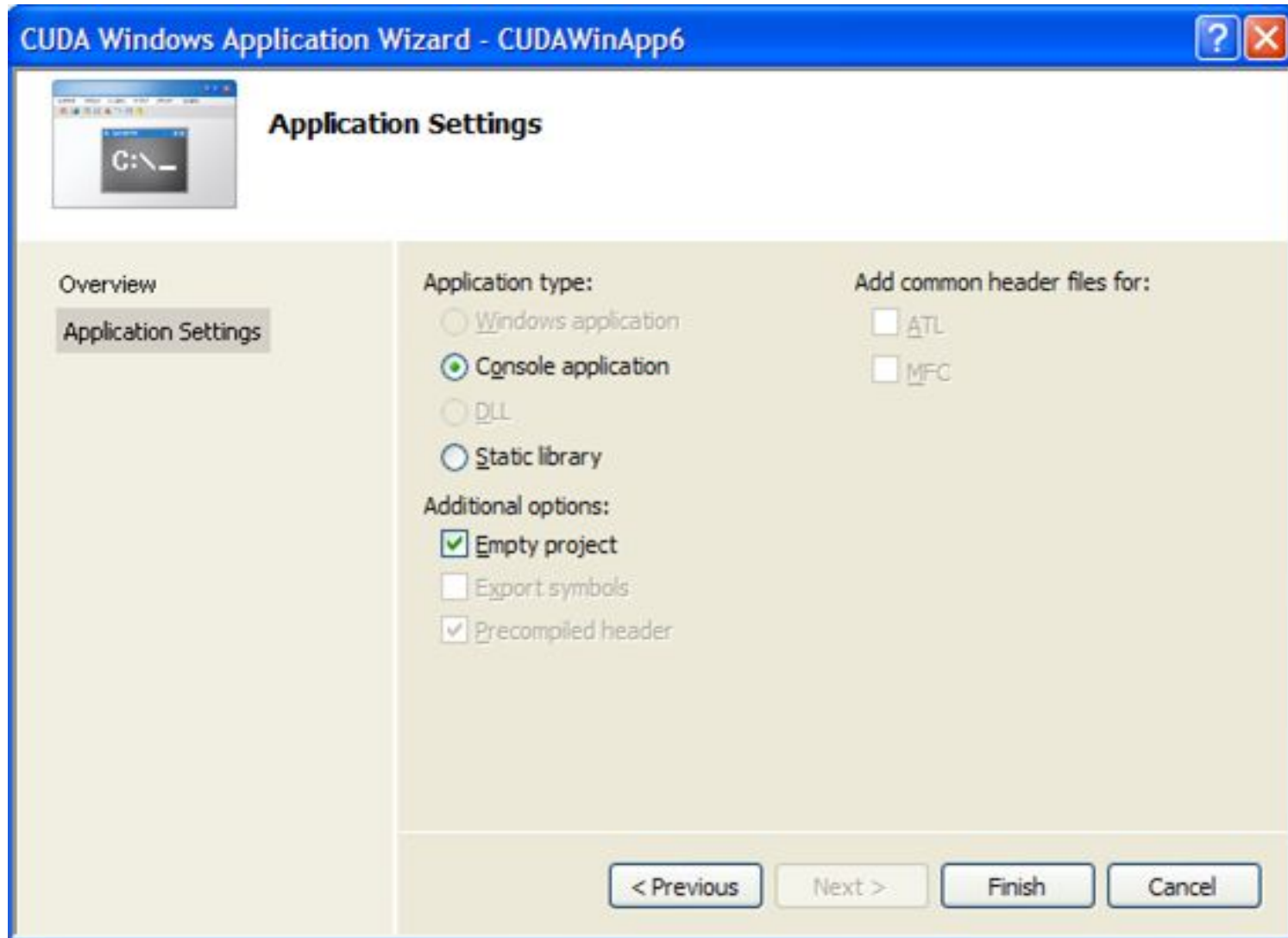
File → New → Project



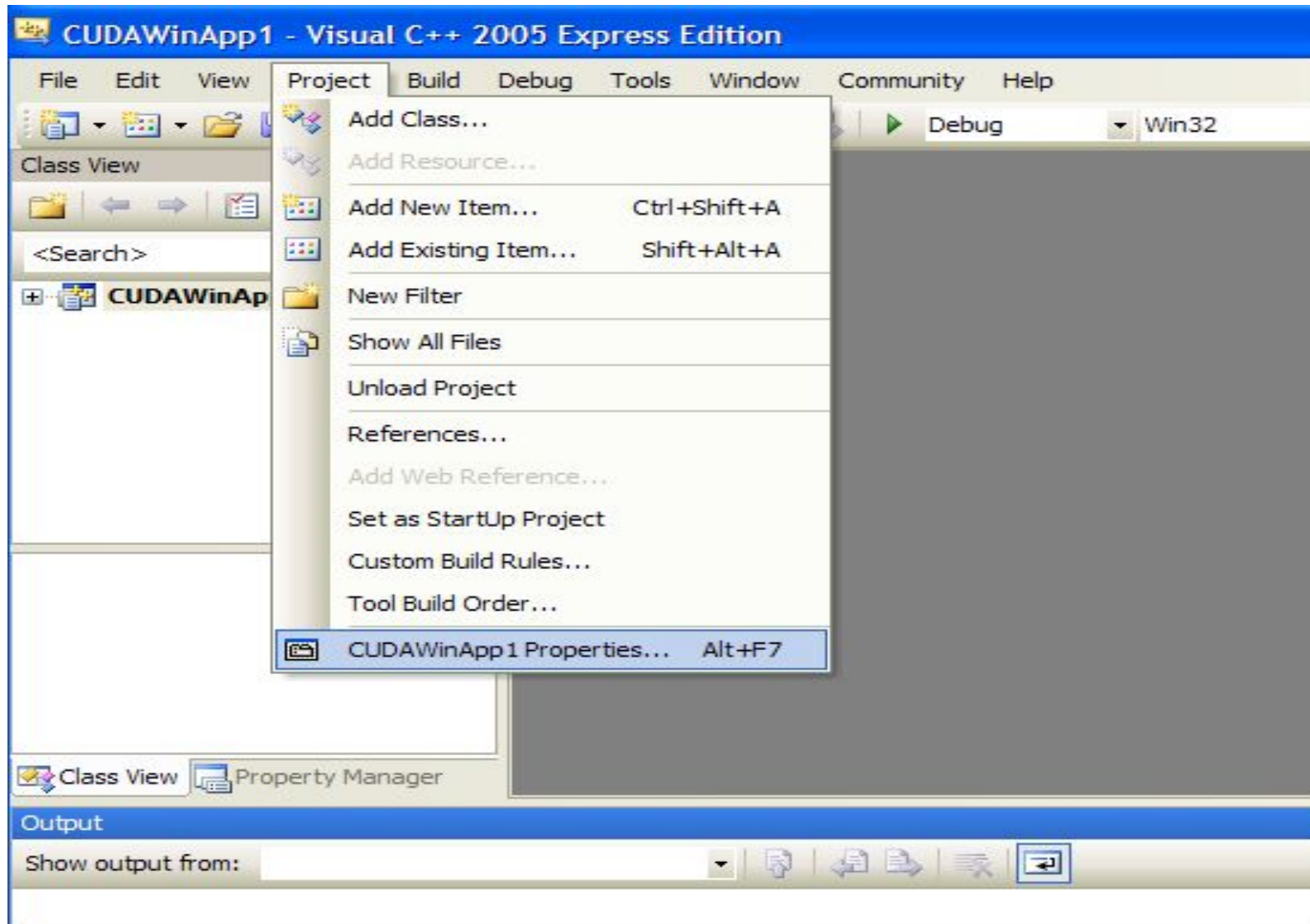
Наш проект - консольное приложение



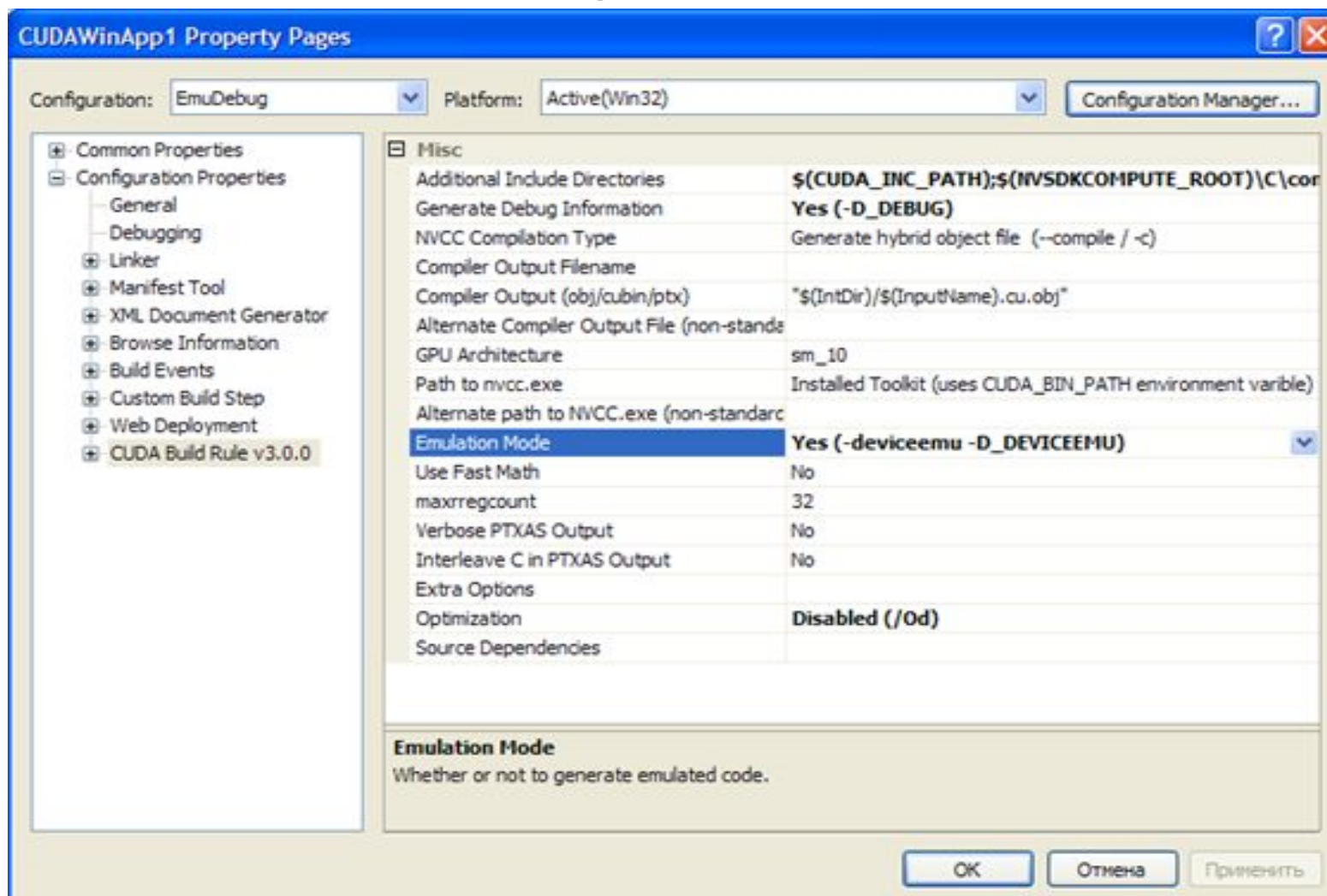
Пустой проект



Работа с CUDA в режиме эмуляции



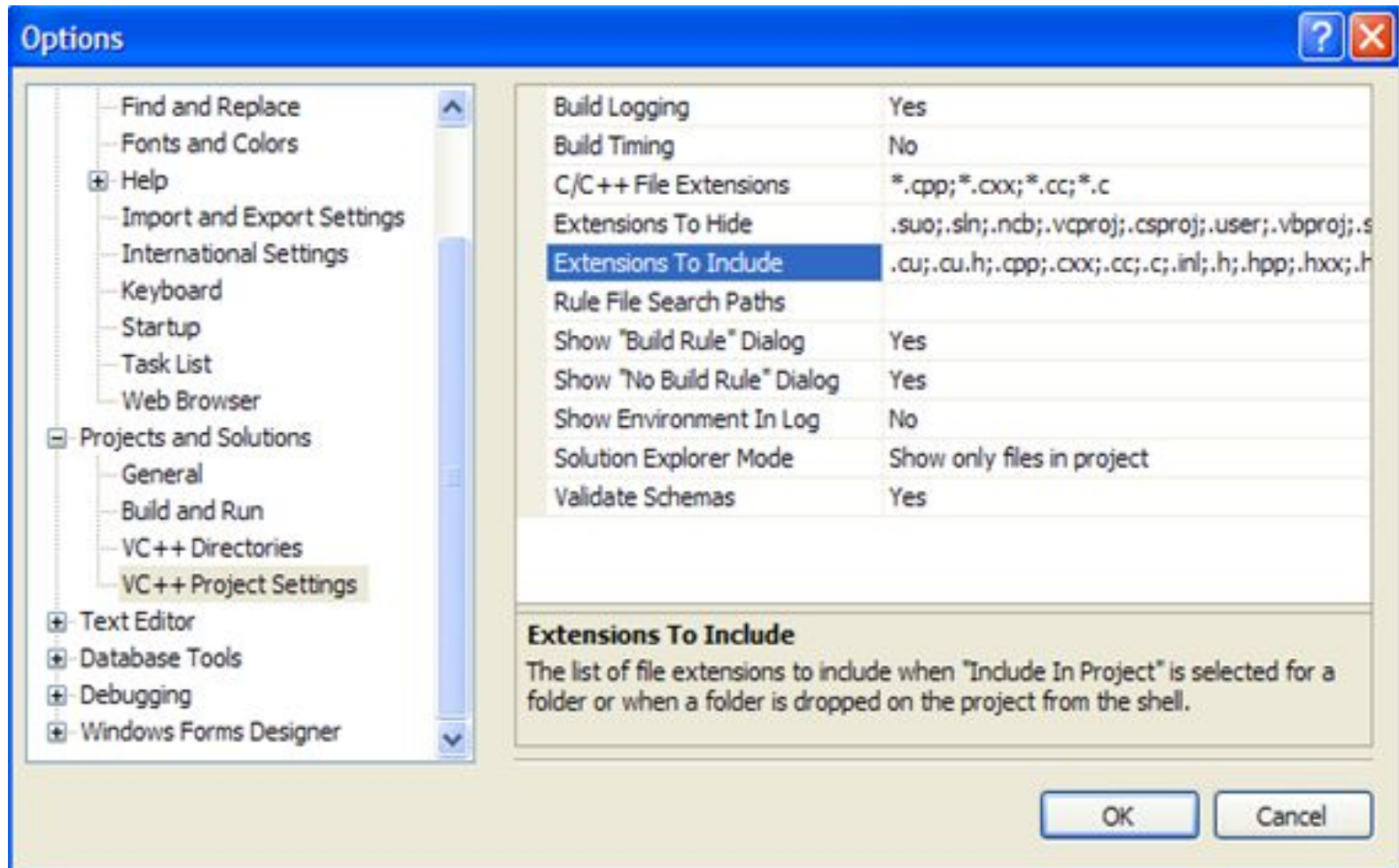
Работа с CUDA в режиме эмуляции



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”

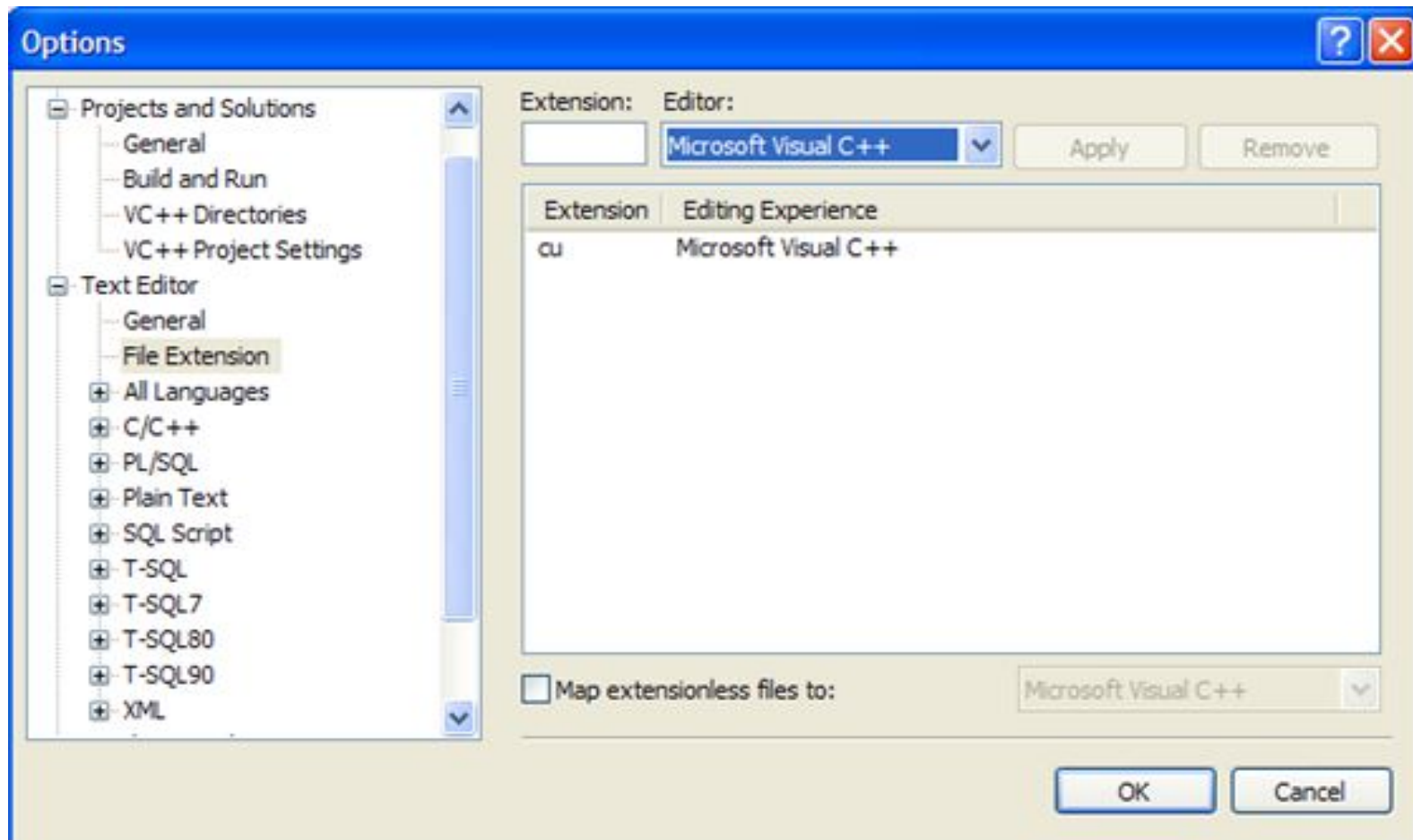
Интеграция NVidia CUDA с Microsoft Visual Studio



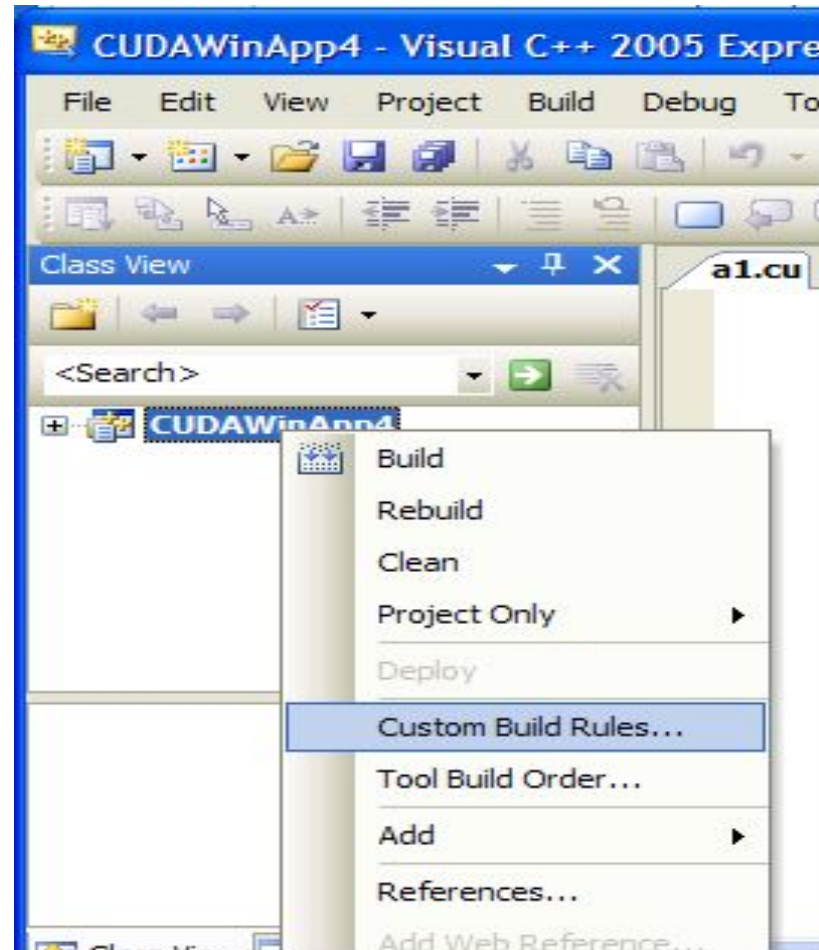
Интеграция NVidia CUDA с Microsoft Visual Studio

- 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;

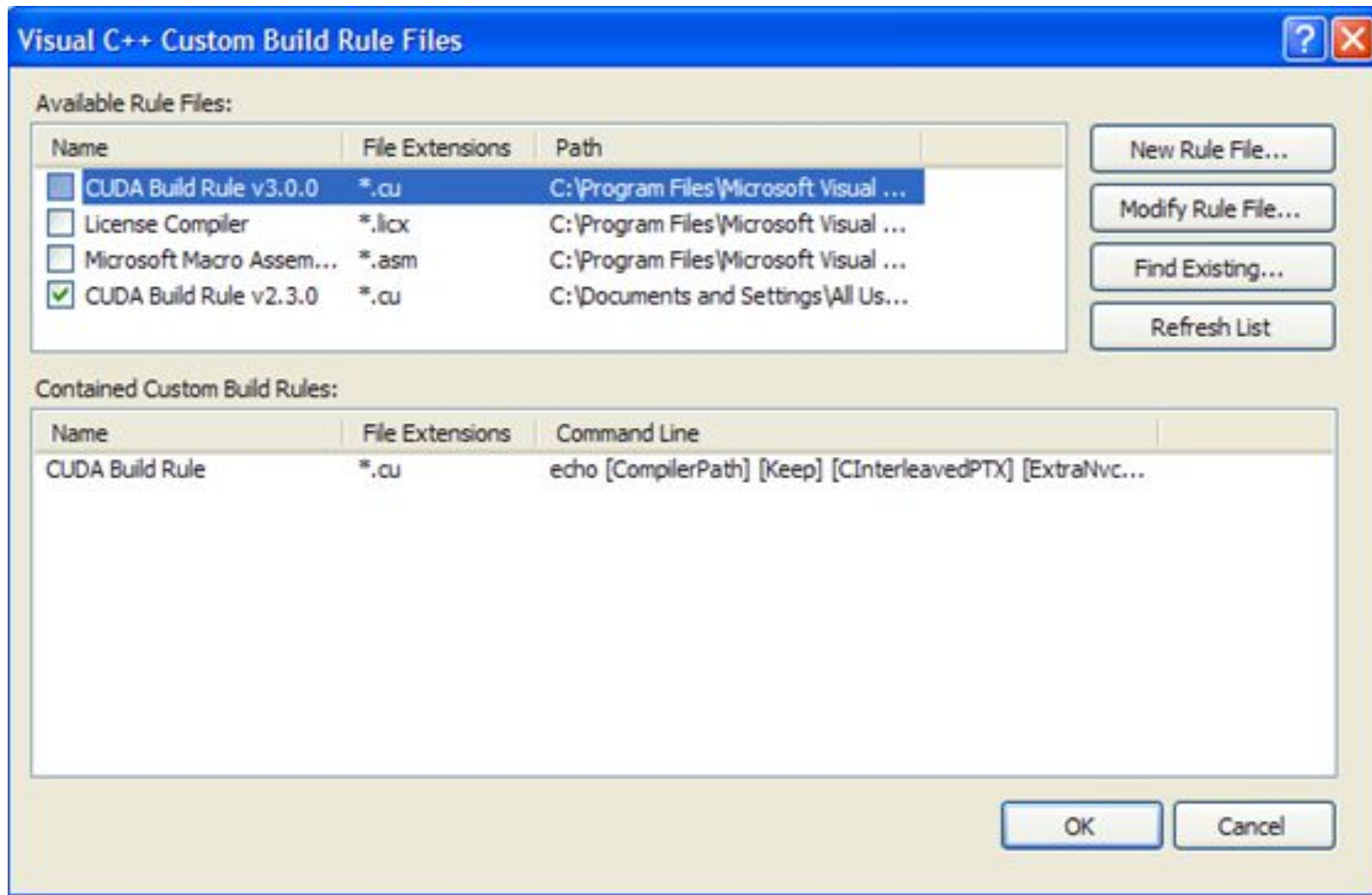
Интеграция NVidia CUDA с Microsoft Visual Studio



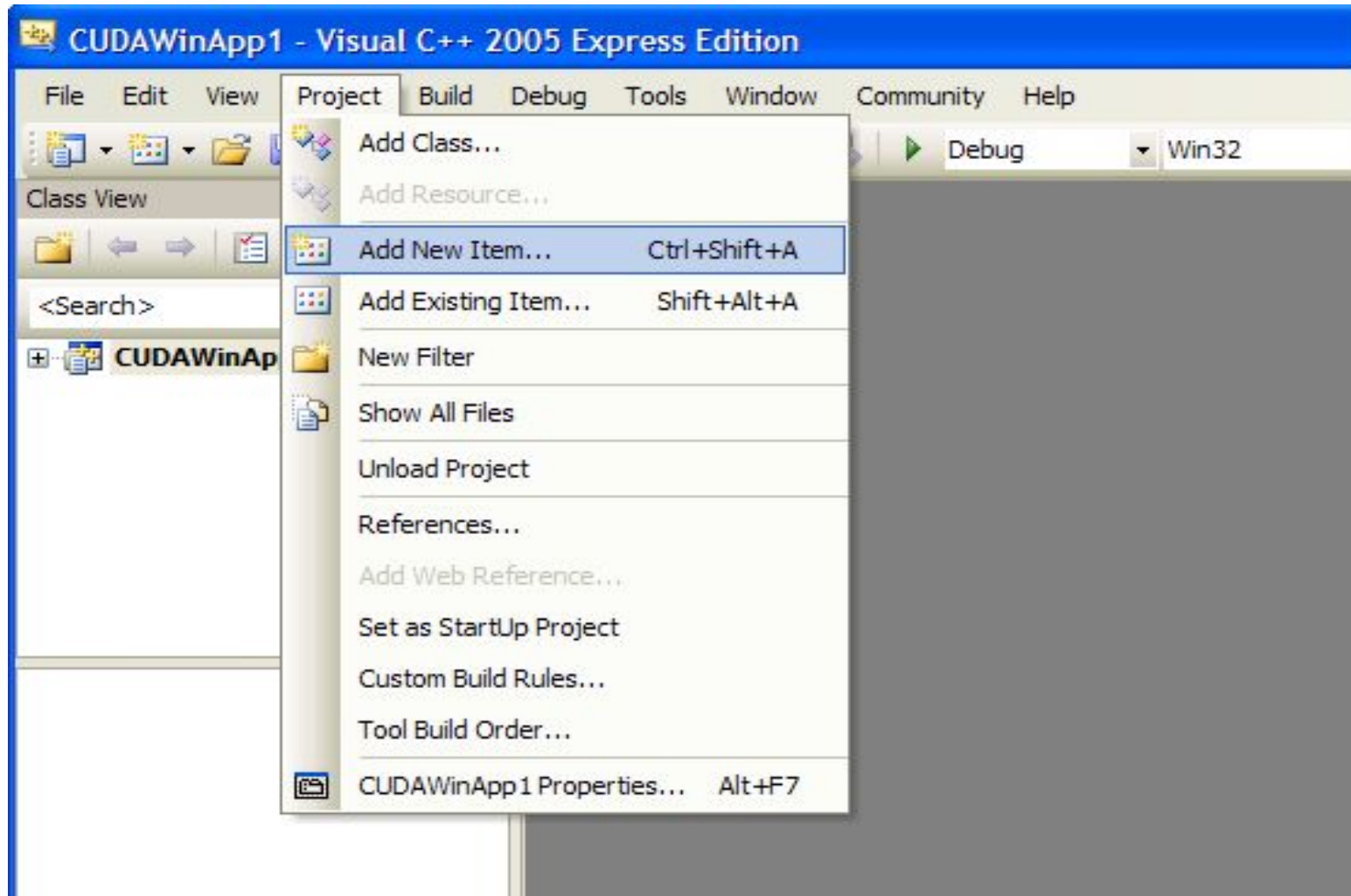
Выбираем правила построения проекта



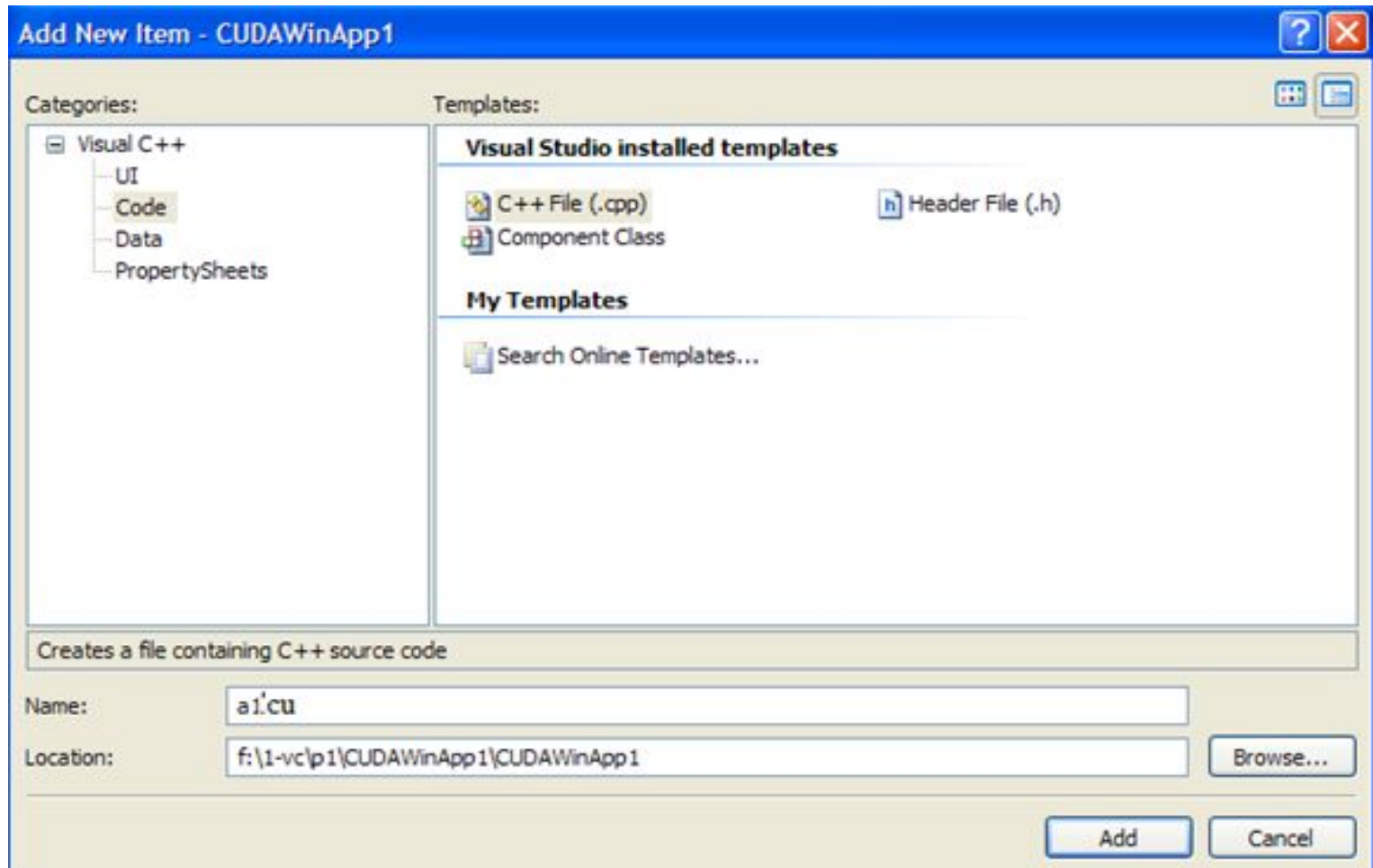
Устанавливаем CUDA Build Rule



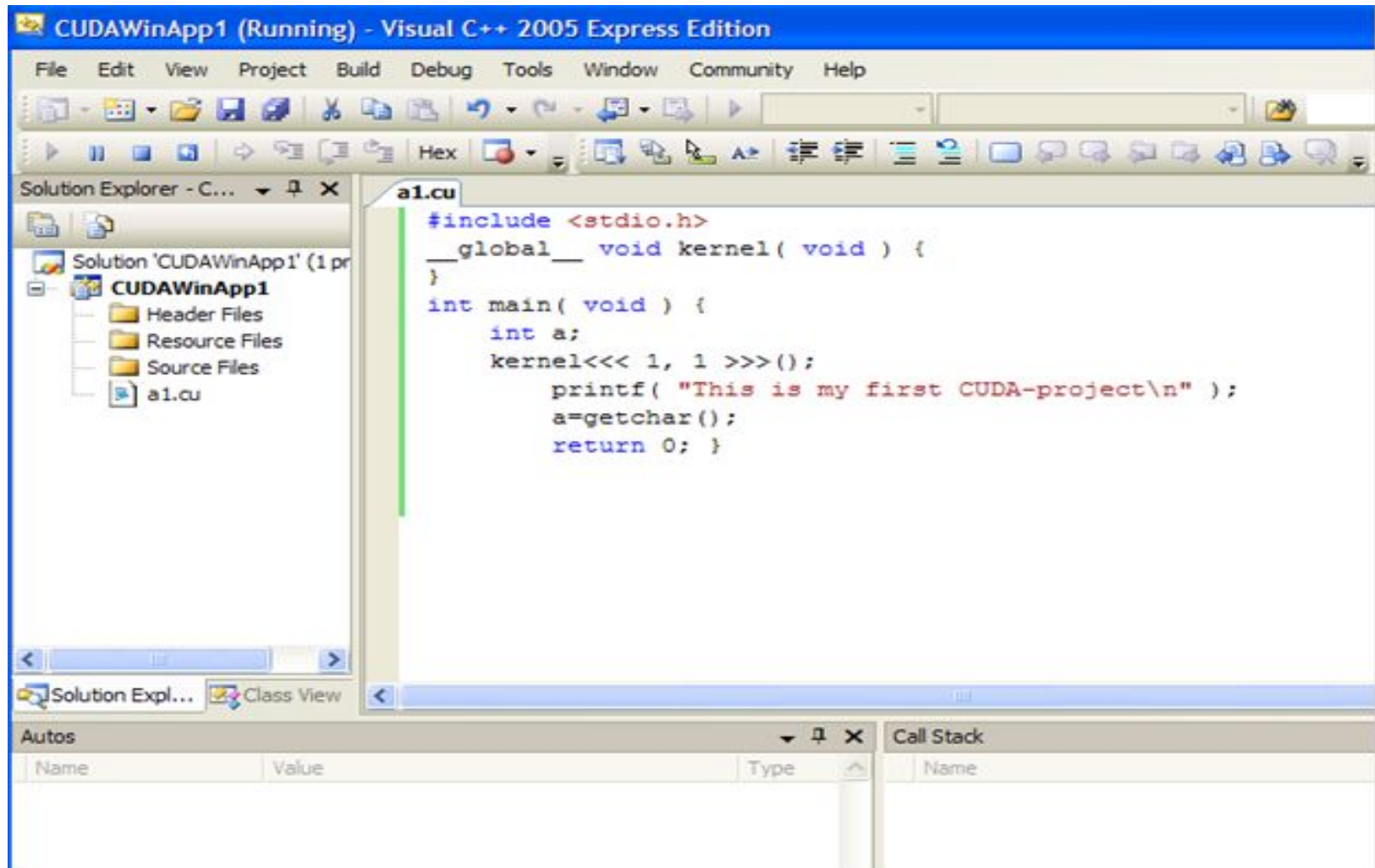
Ввод программы проекта и его запуск



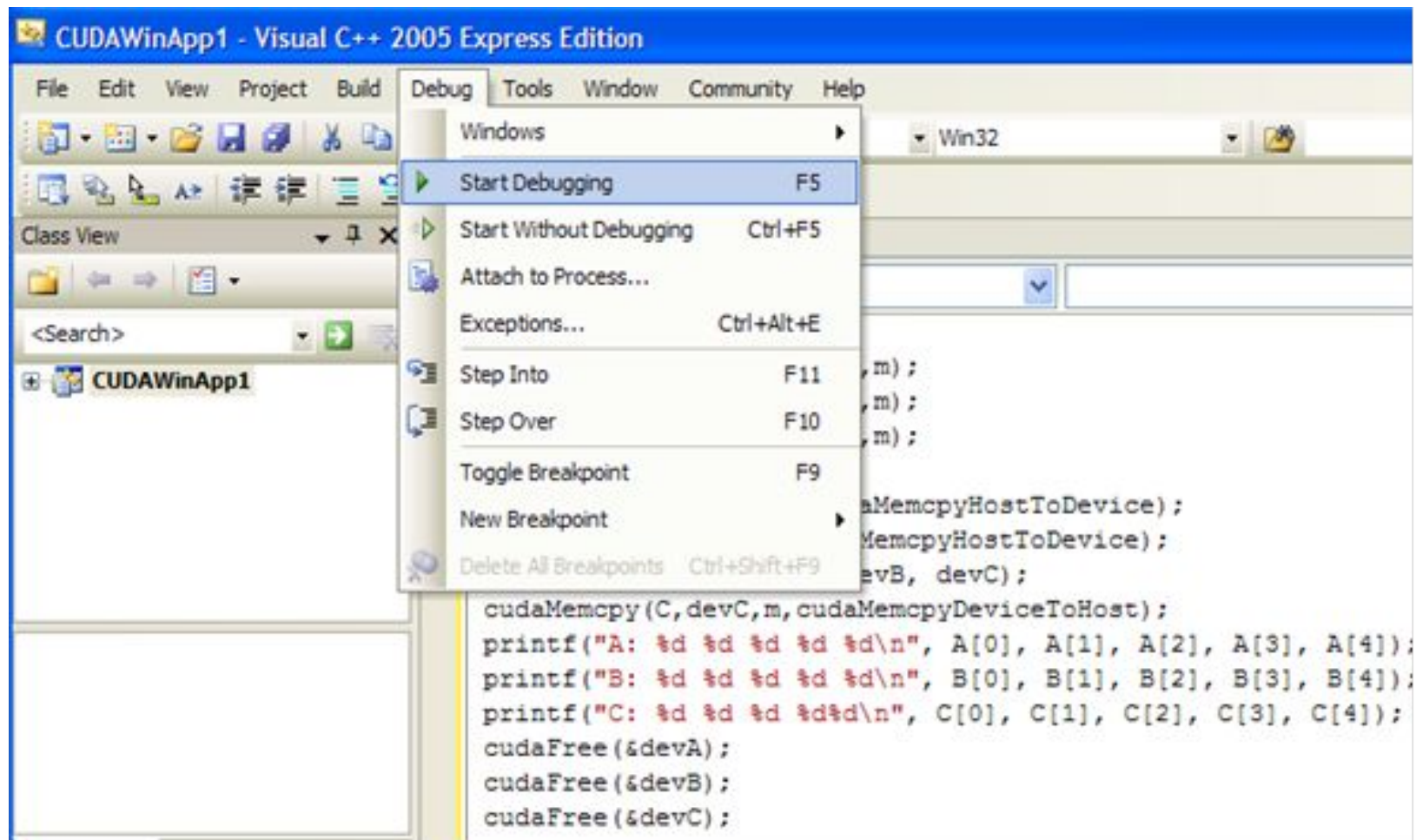
Ввод программы проекта



Ввод программы проекта



Запуск проекта



Результаты работы проекта



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



Вычисление суммы элементов двух массивов (Один блок с N нитеями)

- `#include <stdio.h>`
- `#include <stdlib.h>`
- `//#include <cuda_runtime.h>`
- `//#include <cuda.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