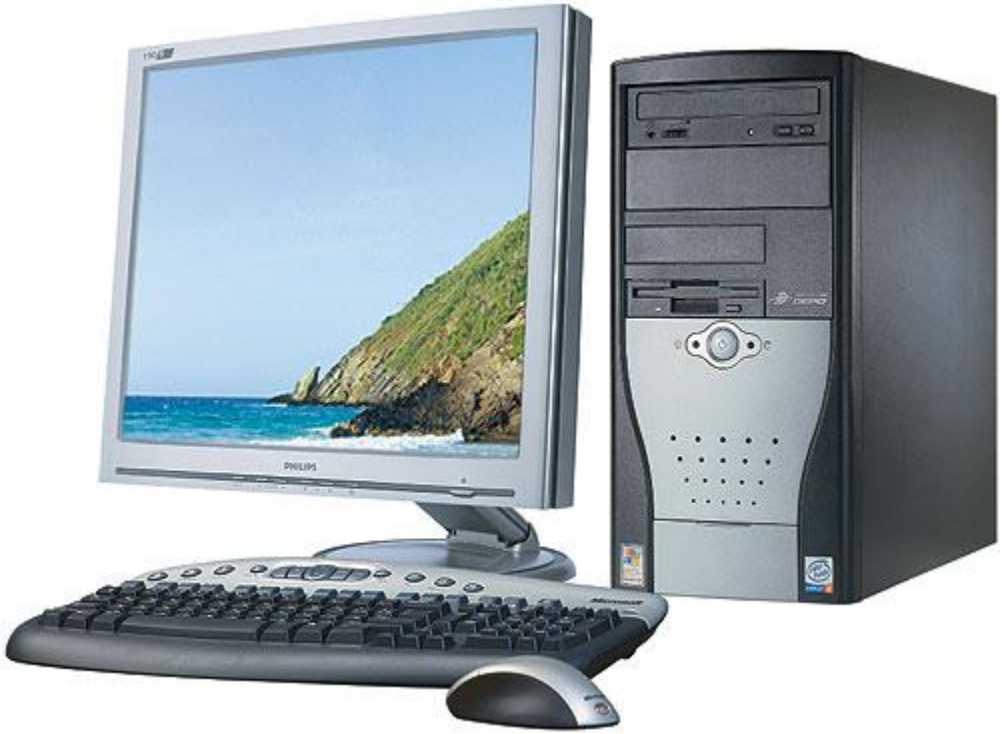
Программирование

Лекция 1

- Что такое программирование?
- Почему C/C++?
- Дорожная карта.
- Базовые понятия.
- Ввод/вывод.



исходная информация (данные)

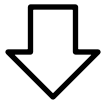
⇩

компьютер

⇩

выходная информация (результат)

Физическое устройство: **полупроводниковые элементы**

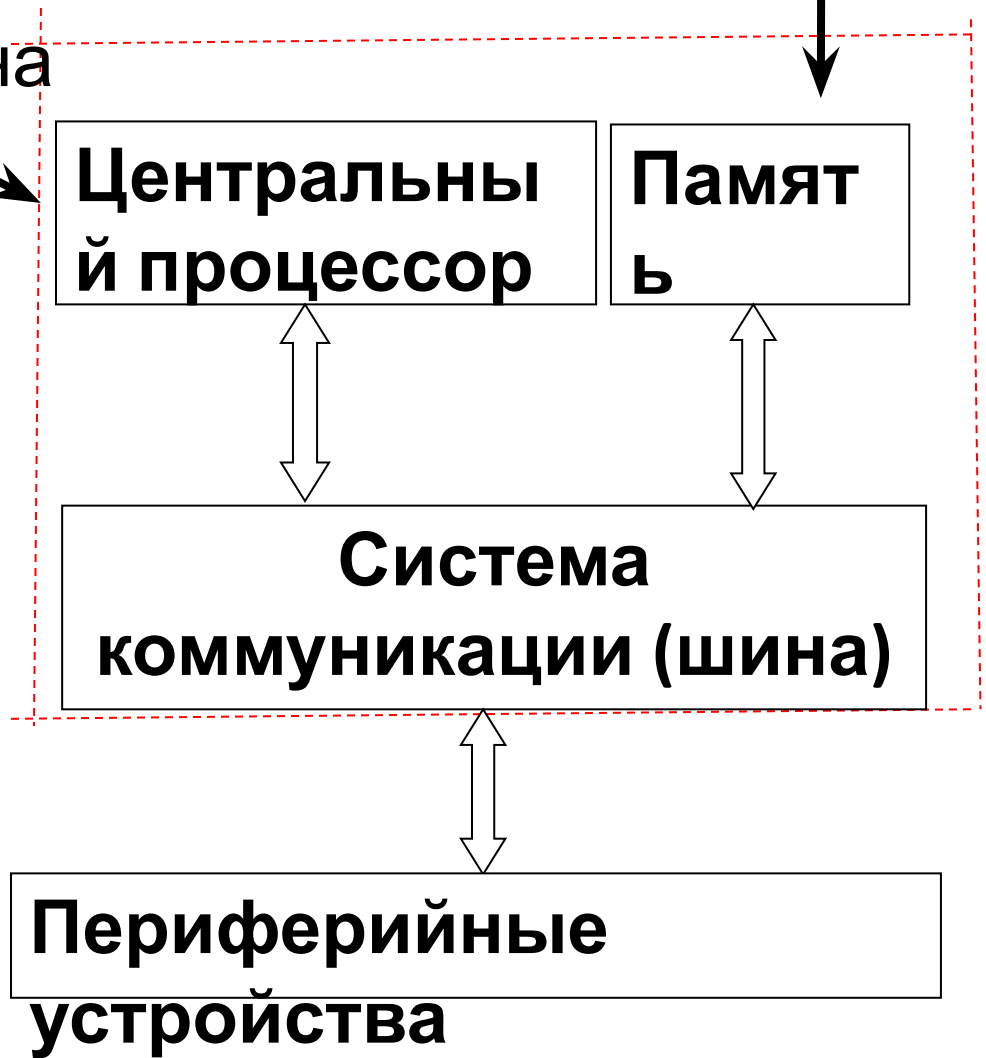


Все данные хранятся и обрабатываются в **двоичном виде**

1111010101010010111100000011110101011011101011001010101010010101010100100000000000000000010101010101111100000110101010101011101010

Набор команд:
арифметических,
логических,
присваивания,
управления, обмена
данными

Хранит данные
в бинарном
виде



Алгоритм – конечная последовательность точно определённых действий (операций), приводящих к однозначному решению поставленной задачи.

Программа – алгоритм решения задачи, записанный
на языке программирования.

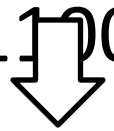
Языки программирования:

- низкого уровня
- высокого уровня

Языки программирования низкого уровня – языки программирования близкие к программированию непосредственно в машинных кодах процессора.

Писать программу в двоичном коде невозможно крайне тяжело…

1111010101010010111100000011110101011011
1010110010101010110000000010101111101010



гораздо более удобно заменить машинные коды мнемониками:

1010101 → **push**

Языки программирования высокого уровня – языки программирования, средства которых обеспечивают запись программы в более простом (чем программы, написанные на ЯП низкого уровня) виде.

C/C++, C#, Java, Python, PHP, R, Basic, Perl, Matlab, Ruby, Delphi…

Основные элементы ЯП

Алфавит – набор символов или групп символов, рассматриваемых как единое целое, с помощью которых составляется текст программы.

Оператор – основная конструкция ЯП, определяющая конкретное действие.

Синтаксис – правила построения конструкций ЯП с помощью алфавита и операторов языка.

Семантика – смысл и правила использования этих конструкций.

Почему C/C++?

Почему C/C++?

языки низкого уровня

↓

язык C

↓

языки высокого уровня

Быстродействие, гибкость, управление памятью ⇩

Разработка системного программного обеспечения ⇩

Операционные системы (windows, unix), базы данных (Oracle, MySQL, MS SQL Server) и др.

Aug 2019	Aug 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.028%	-0.85%
2	2		C	15.154%	+0.19%
3	4	^	Python	10.020%	+3.03%
4	3	v	C++	6.057%	-1.41%
5	6	^	C#	3.842%	+0.30%
6	5	v	Visual Basic .NET	3.695%	-1.07%
7	8	^	JavaScript	2.258%	-0.15%
8	7	v	PHP	2.075%	-0.85%
9	14	^^	Objective-C	1.690%	+0.33%
10	9	v	SQL	1.625%	-0.69%
11	15	^^	Ruby	1.316%	+0.13%
12	13	^	MATLAB	1.274%	-0.09%
13	44	^^	Groovy	1.225%	+1.04%
14	12	v	Delphi/Object Pascal	1.194%	-0.18%
15	10	vv	Assembly language	1.114%	-0.30%
16	19	^	Visual Basic	1.025%	+0.10%
17	17		Go	0.973%	-0.02%
18	11	vv	Swift	0.890%	-0.49%
19	16	v	Perl	0.860%	-0.31%
20	18	v	R	0.822%	-0.14%

* Источник:

Дорожная карта

Мотивация

Программирование – это

- Искусство
- Постоянное обучение, развитие
- Хорошая зарплата
- Зарубежные командировки
- Интересный коллектив

Цели

Изучить основы C/C++

1-й семестр – язык Си

- переменные, операции, структура программы
- ветвления, циклы
- функции
- массивы, строки
- указатели
- структуры данных

2-й семестр – язык C++

- ООП, классы, объекты
- перегрузка операций
- наследование
- полиморфизм

Акцент на практику.

1-й семестр – язык Си

- Консольные приложения.
- Решение учебных задач.
- Реализация проекта.

2-й семестр – язык C++

- Графика.
- Решение учебных задач.
- Реализация проекта.

Базовые понятия

Несколько определений:

Функция – часть программы, которая выполняет законченную последовательность действий и имеет связи с другими функциями.

Оператор – структурная единица языка, в C/C++ всегда оканчивается ;

Переменная – поименованная область памяти.

КОММЕНТАРИИ

Комментарии бывают двух типов:

- Однострочные

```
// Это первый однострочный комментарий
// Это второй однострочный комментарий
```

- Многострочный комментарий

```
/*
    Это многострочный комментарий
    В нем размещается многострочный текст
*/
```

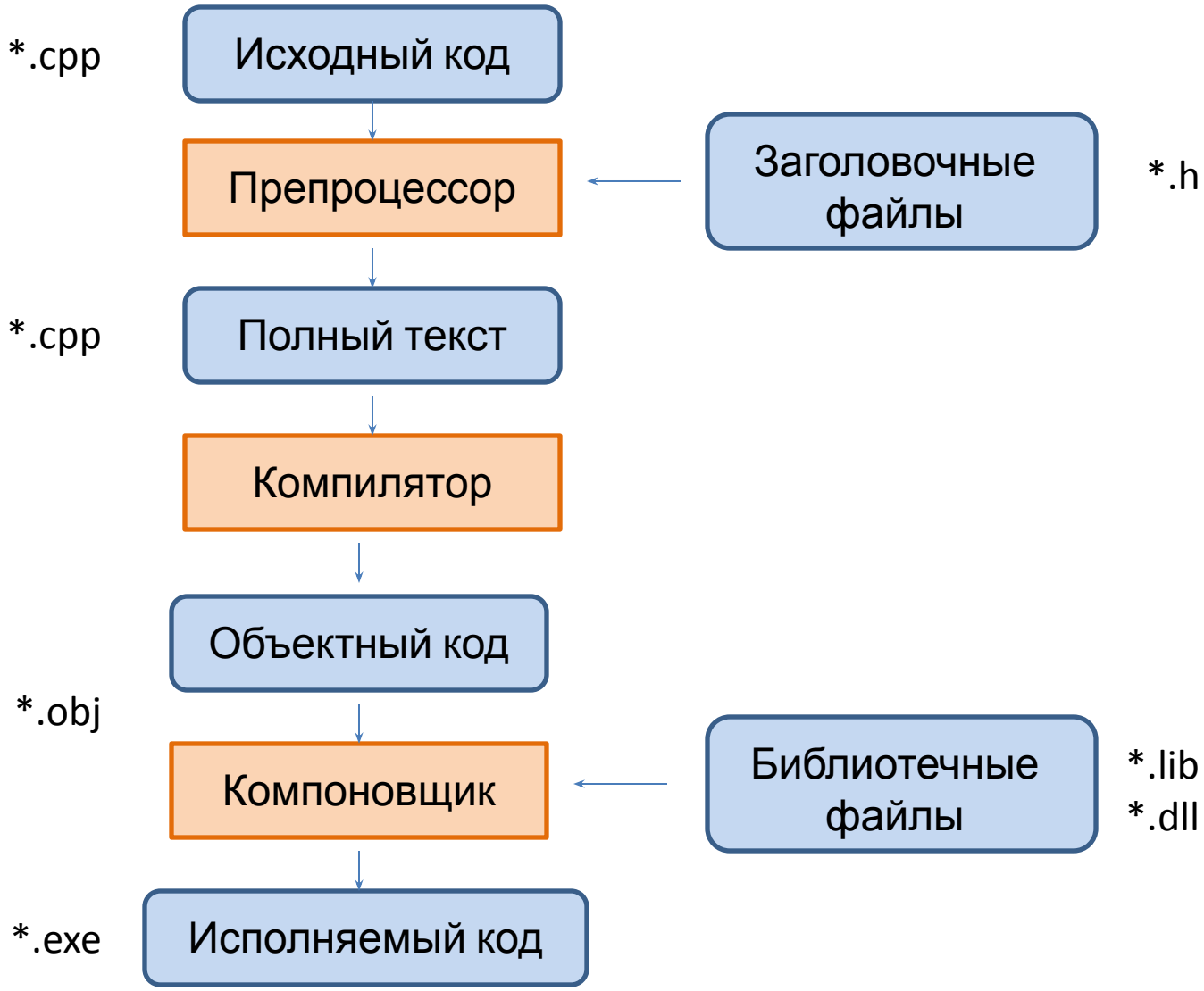
Препроцессор убирает комментарии из текста программы перед компиляцией

Структура программы

```
#include <iostream>   //директива препроцессора
using namespace std; /*использование
    стандартного
                                пространства имен*/
int main()                    //заголовок функции main
{
    cout<<“Hello, world!”;    //вывод на экран
    return 0;
}
```

функция **main**
главная функция
программы

ТРАНСЛЯЦИЯ ИСХОДНОГО КОДА



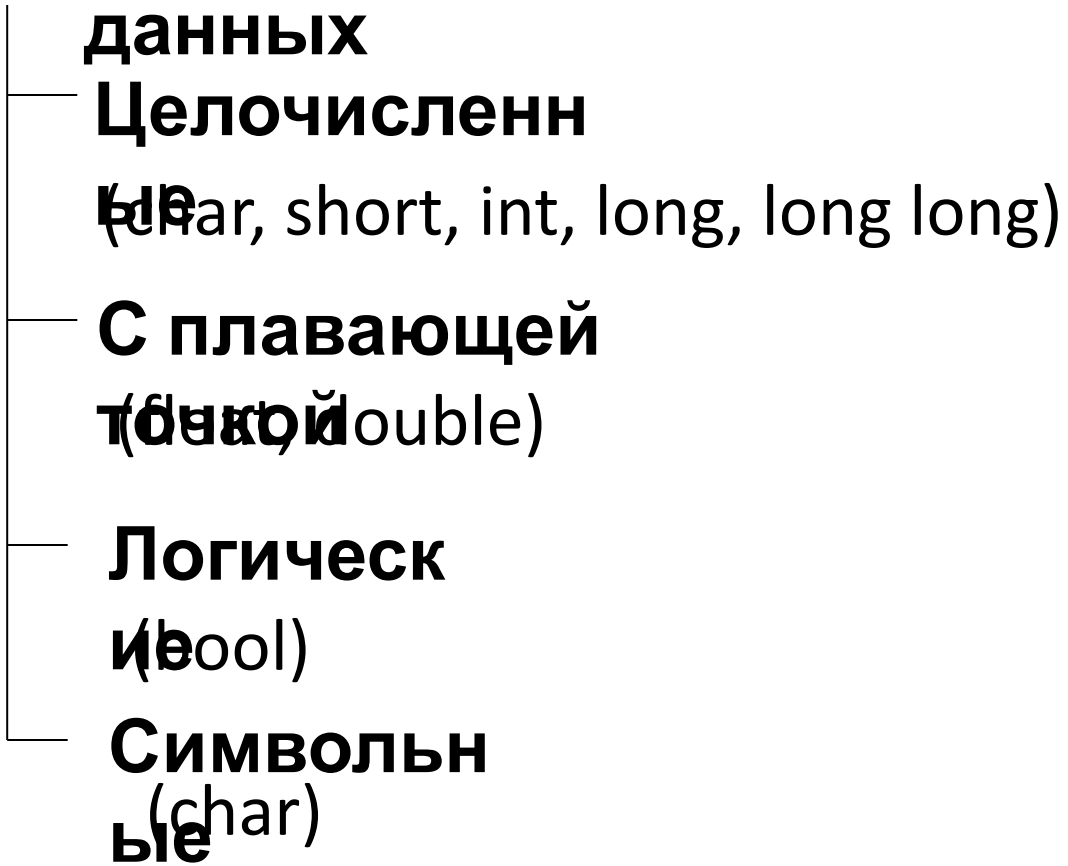
ПЕРЕМЕННЫЕ

Объявить переменную – значит дать ей имя и указать тип данных.

Инициализировать переменную – дать переменной значение.

Тип данных влияет на:

- размер памяти, выделяемый под переменную
- данные, которые могут храниться в переменной



РАЗМЕРЫ СТАНДАРТНЫХ ТИПОВ

Заголовок	Байт
bool	1
char	1
short int (short)	2
long int (long)	4
long long int (long long)	8
unsigned char	1
unsigned short	2
unsigned long	4
unsigned long long	8
float	4
double	8
long double	8
int	системно-зависимый (чаще 4 байт)

Правила именования переменных:

- имя переменной может содержать буквы английского алфавита, цифры и знак подчеркивания;
- язык С является регистрозависимым;
- желательно, чтобы имена были говорящими и короткими

```
int max_index = 100;
bool result;
float pi=3.14;
```

ОПЕРАЦИИ

Деление операций по функциональному назначению

Арифметические	(+, -, *, /, %);
Присваивание	(=);
Сравнение	(==,!=,>=,<=);
Арифметические с присваиванием	(+=,-=,*=,/=);
Инкремент, декремент	(++,--);
Логические	(!,&&,||);
Логические побитовые	(&,^,>>,<<,~);
Адресные	(*,&);
Доступ к полям структур и объединений	(.,->);
Прочие	(?:,*sizeof*()

Операции различаются:

- **Арностью** (количество операндов, принимаемых операцией)
- **Приоритетом** (определяет порядок выполнения операций)
- **Ассоциативностью** (слева направо, справа налево)

ПРИОРИТЕТ ОПЕРАЦИЙ

Операторы											Выполняются
()	[]	->	.	::							слева направо
!	~	++	--	+	-	*	&	(type)	sizeof		справа налево
.*	->*										
*	/	%									слева направо
+	-										слева направо
<<	>>										слева направо
<	<=	>	>=								слева направо
==	!=										слева направо
&											слева направо
^											слева направо
|											слева направо
&&											слева направо
||											слева направо
?:											справа налево
=	+=	-=	*=	/=	%=	&=	^=	|=	<<=	>>=	справа налево
,											слева направо

Ввод/вывод

Библиотека iostream

Операторы **cout** и **cin**

1. Подключение библиотеки:

#include <iostream>

2. Использование стандартного пространства имен:

using namespace std;

В противном случае, необходимо каждый раз указывать стандартное пространство имен

std::cout<<“Hello, world!”;

Вывод на экран: оператор **cout** и операция **<<**

- Вывод фразы :

```
cout<<“Hello, world!”;
```

- Вывод значения переменной:

```
float pi=3.14;
cout<<pi;
```

- Каскадирование операции <<

```
cout<<“Значение числа пи, равно: ”<<pi;
```

- Переход на новую строку endl

```
cout<<“Значение числа пи, равно: ”<<endl<<pi;
```

Для ввода с клавиатуры используются: оператор **cin** и операция **>>**

```
int data;
cout<<“Введите число”;
cin>>data;
```

Операцию >> тоже можно каскадировать.

Библиотека stdio.h

Функции **printf**, **scanf**

Необходимо подключить библиотеку

#include “stdio.h”

- Вывод фразы:

printf(“Hello, world!”);

- Вывод значения переменной

  форматная строка, начинается со **знака %**, обязательный параметр - **тип данных**

float pi=3.14;

printf(“Значение числа пи равно %f”, pi);

i,d – целое число
f – вещественное число
o – восьмиричное число
x – шестнадцатиричное число
c – символ

- Форматный вывод значения переменной:

%[флаг][ширина][.точность]тип

↓

- (минус) – выравнивание по левому краю
+ (плюс) – вывод знака числа
(пробел) – выводить пробел перед данными
0 (ноль) – дополнять поле вывода нулями

```
float pi=3.141592;
printf("Значение числа пи равно %+10.2f", pi);
```

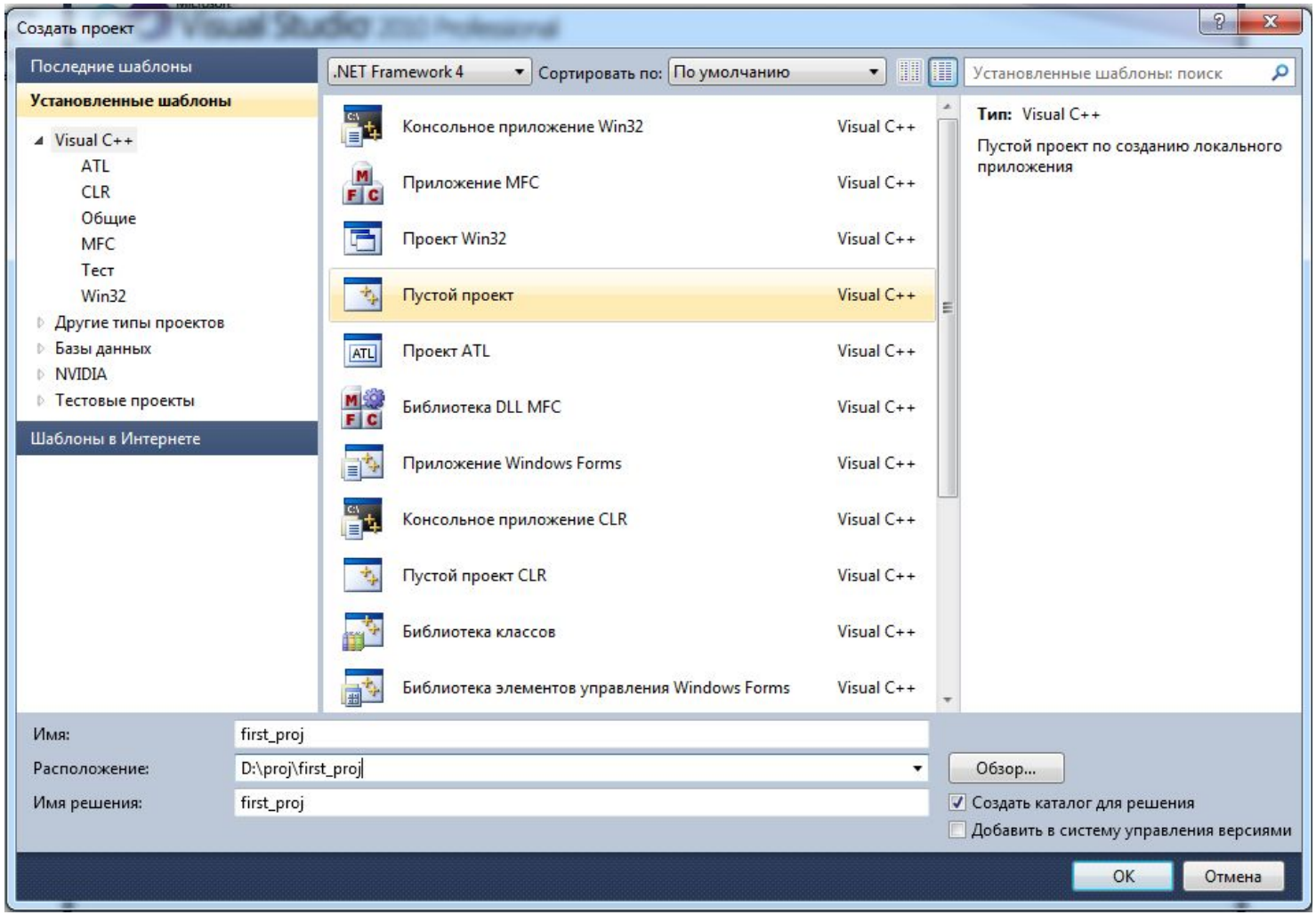
Функция **scanf**

используются: **%тип** и операция взятия адреса **&**

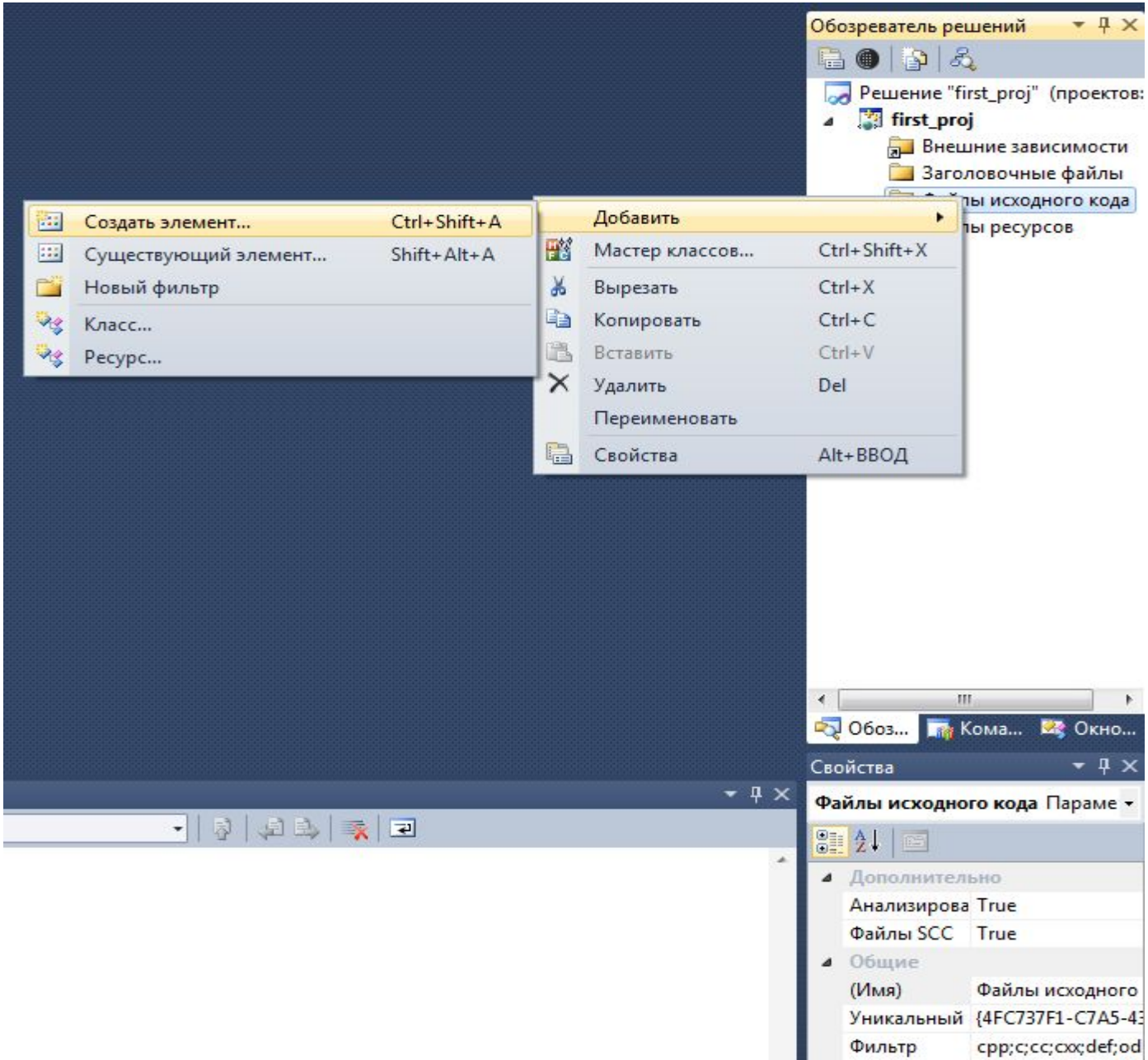
```
int data;
scanf(“%d”,&data);
```

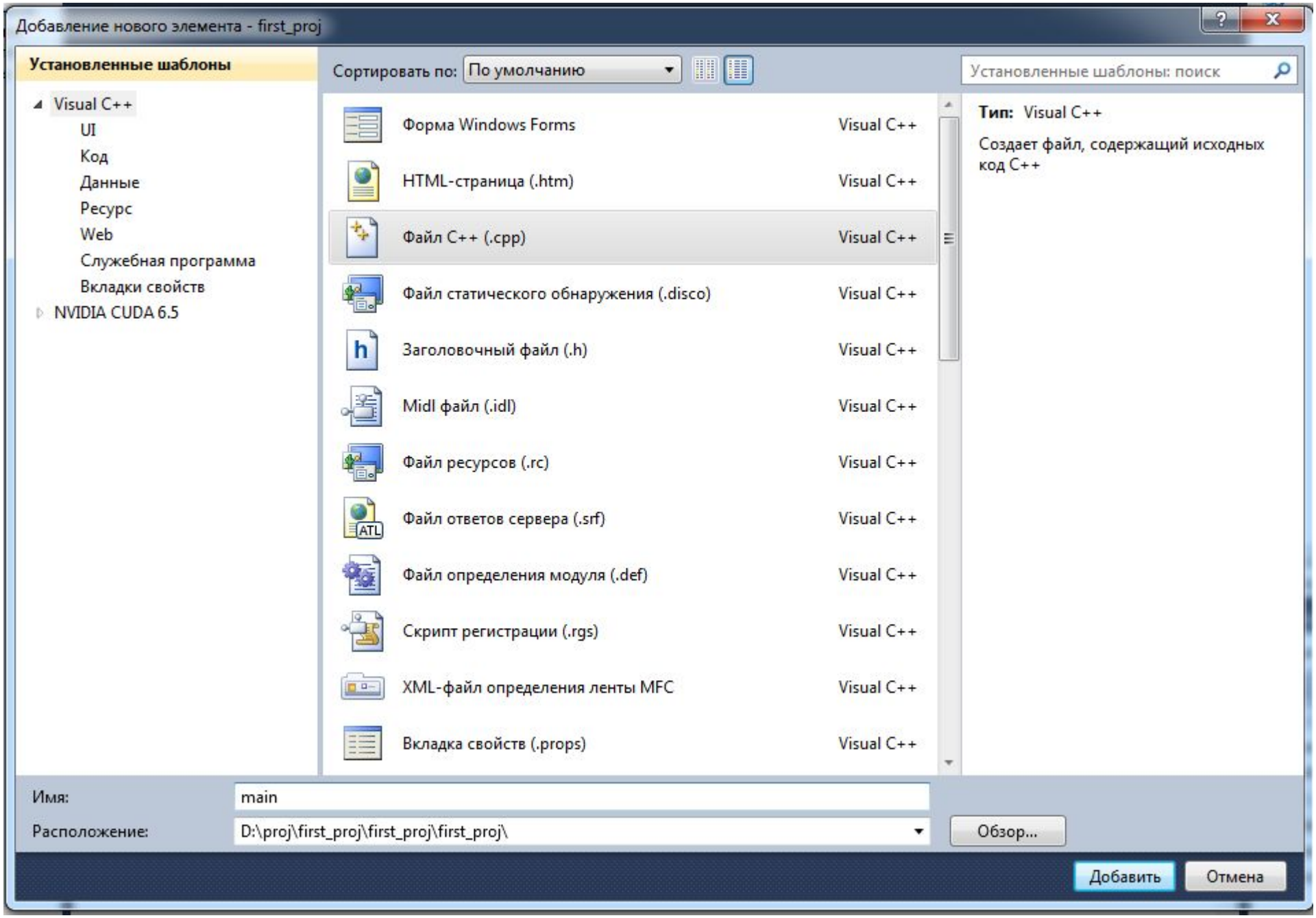
Практика

- Устанавливаем **Visual Studio**
- Создаем консольное приложение (файл->создать->проект->пустой проект)



❑ Добавляем файл с исходным кодом




Добавление нового элемента - first_proj
Установленные шаблоны
Visual C++
UI
Код
Данные
Ресурс
Web
Служебная программа
Вкладки свойств
NVIDIA CUDA 6.5
Сортировать по: По умолчанию
Установленные шаблоны: поиск
Форма Windows Forms Visual C++
HTML-страница (.htm) Visual C++
Файл C++ (.cpp) Visual C++
Файл статического обнаружения (.disco) Visual C++
Заголовочный файл (.h) Visual C++
Midl файл (.idl) Visual C++
Файл ресурсов (.rc) Visual C++
Файл ответов сервера (.srf) Visual C++
Файл определения модуля (.def) Visual C++
Скрипт регистрации (.rgs) Visual C++
XML-файл определения ленты MFC Visual C++
Вкладка свойств (.props) Visual C++
Тип: Visual C++
Создает файл, содержащий исходных код C++
Имя: main
Расположение: D:\proj\first_proj\first_proj\first_proj\
Обзор...
Добавить
Отмена