



Объектно-ориентированное программирование

2024



Как со мной связаться?

Старший преподаватель кафедры 806

Дзюба Дмитрий Владимирович

ddzuba@yandex.ru



Что будет в курсе?

Отчетность по курсу рейтинг

5-бальная система	Рейтинговая система
5 - Отлично	90-100
4 – Хорошо	82-89
	75-81
3 - Удовлетворительно	67-74
	60-66
2 - Неудовлетворительно	Менее 60

Баллы даются:

1. Сделанная и сданная Лабораторная работа (7 шт) – 10-15 баллов.
2. Зачет до 30 баллов

Критерий приема лабораторных работ:

- код компилируется и работает на правильных данных
- код корректно обрабатывает ошибки в данных
- студент может рассказать про приемы и алгоритмы применяемые в программе
- студент может ответить на дополнительные вопросы по теме ЛР
- при сдаче ЛР с опозданием максимальный балл опускается до 10 баллов

срок сдачи ЛР -2
недели

ОТ ЛЕКЦИИ ДО ЛЕКЦИИ

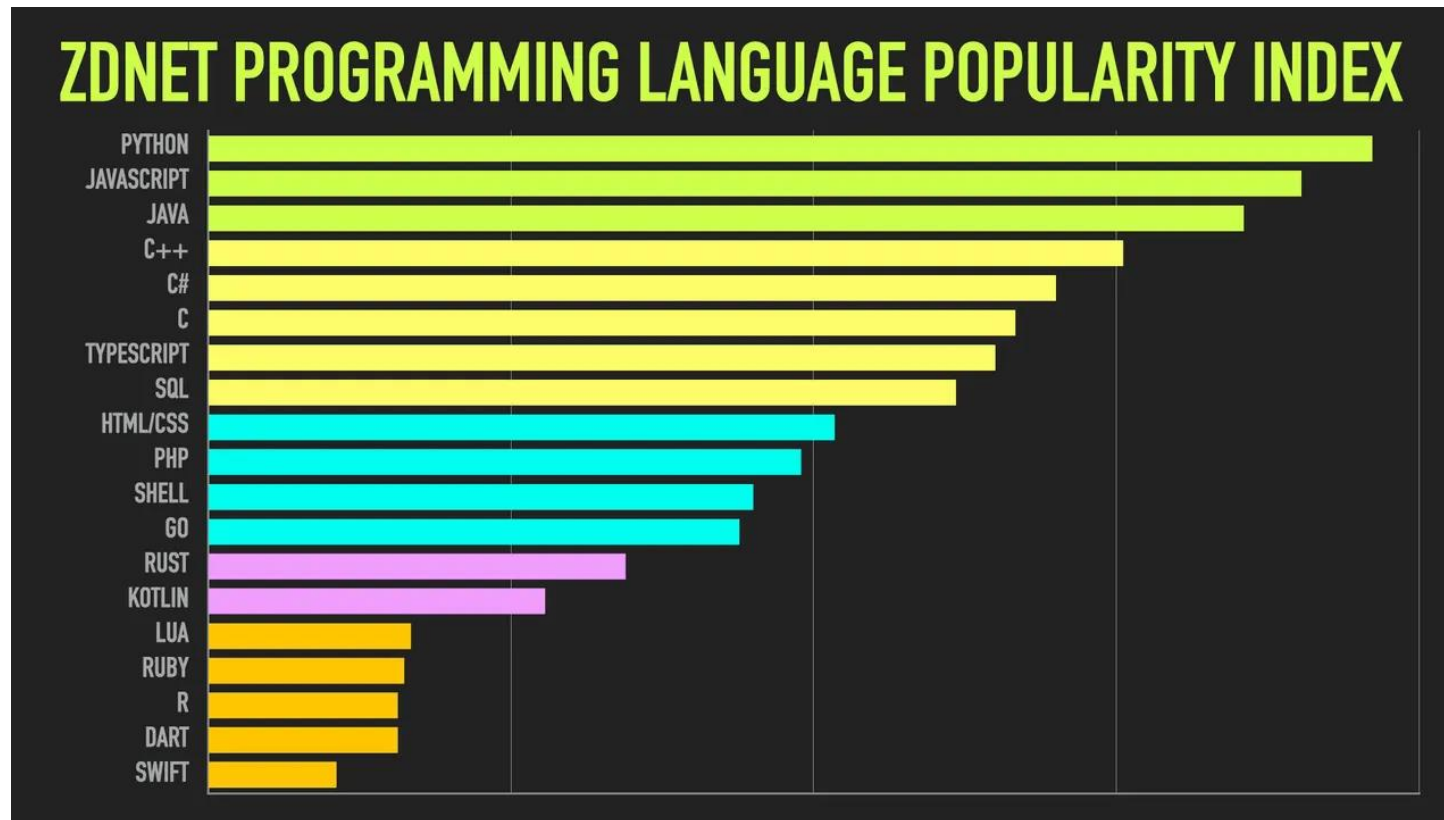
нужно сдать все ЛР

Где что?

- Материалы лекций
- Лабораторные работы
- Вопросы и ответы
- Вспомогательные материалы



Почему C++



Компании, в которых
востребовано знание C++
Билайн, МТС, Касперский, VK,
Yandex ...

<https://www.zdnet.com/article/the-most-popular-programming-languages-in-2024-and-what-that-even-means/>



WEB DEVELOPMENT



GAME DEVELOPMENT



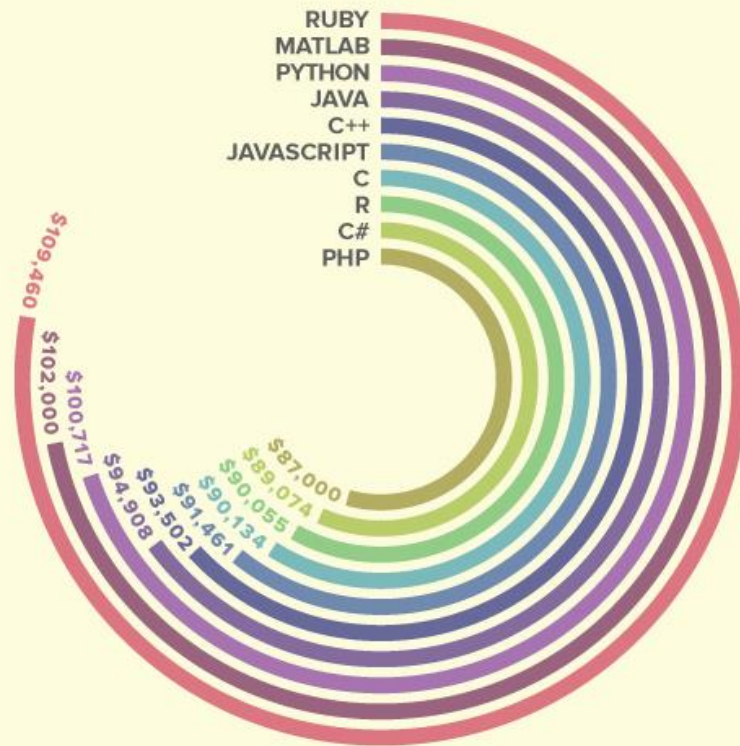
MOBILE APP DEVELOPMENT

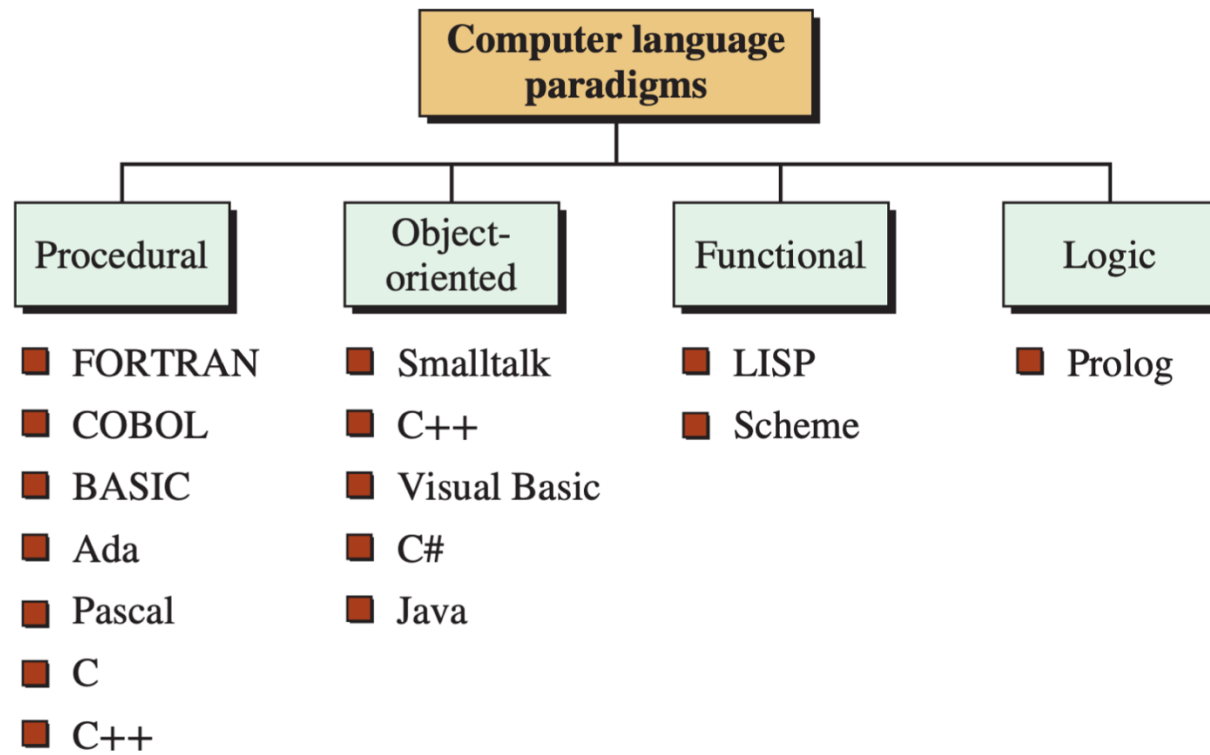


DATA ANALYSIS



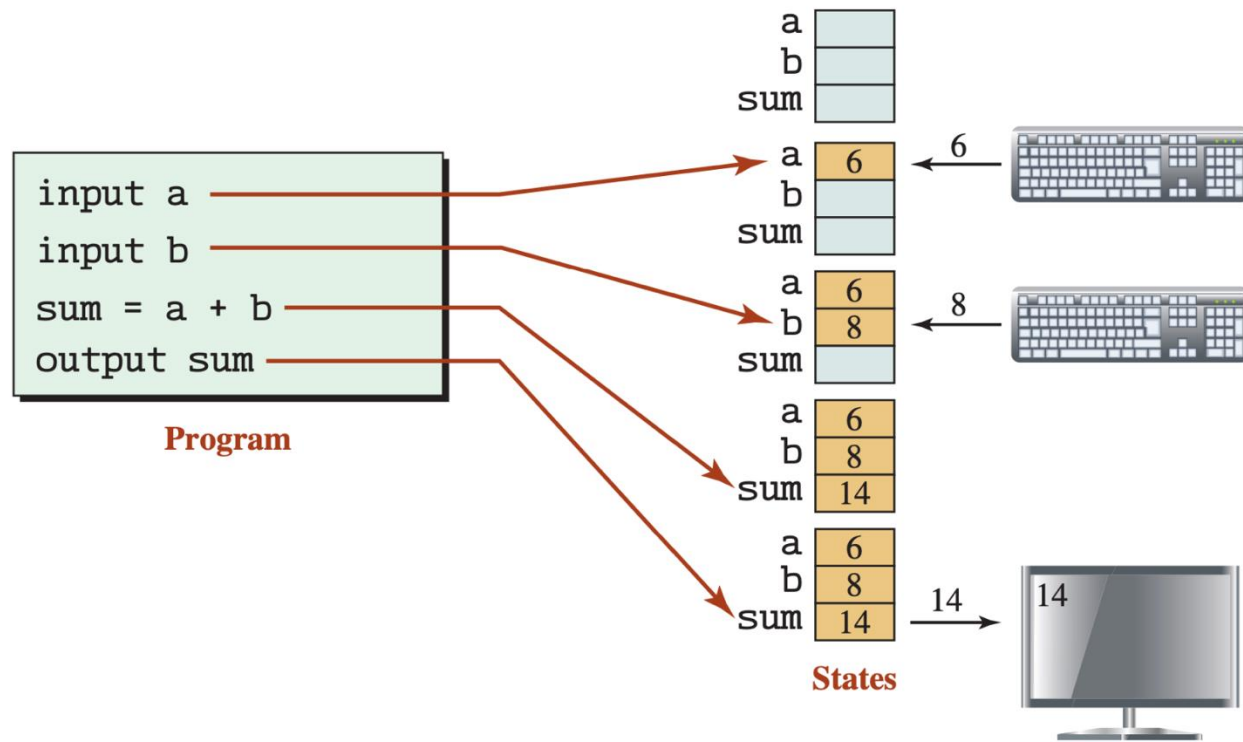
EMBEDDED SYSTEM PROGRAMMING

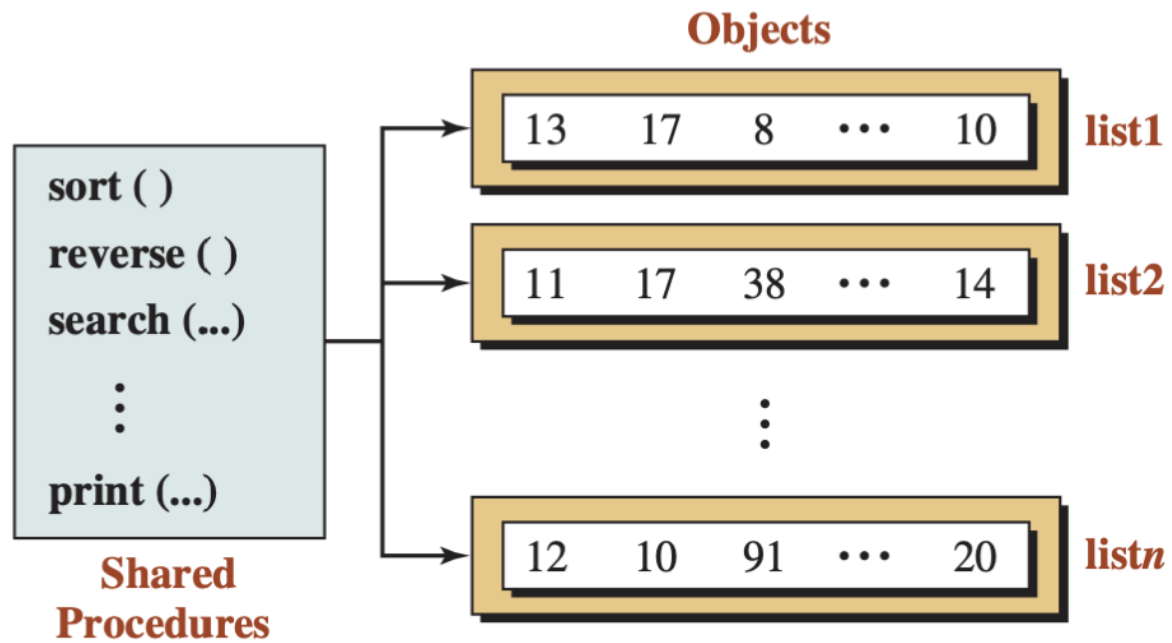




парадигмы

Процедурная парадигма

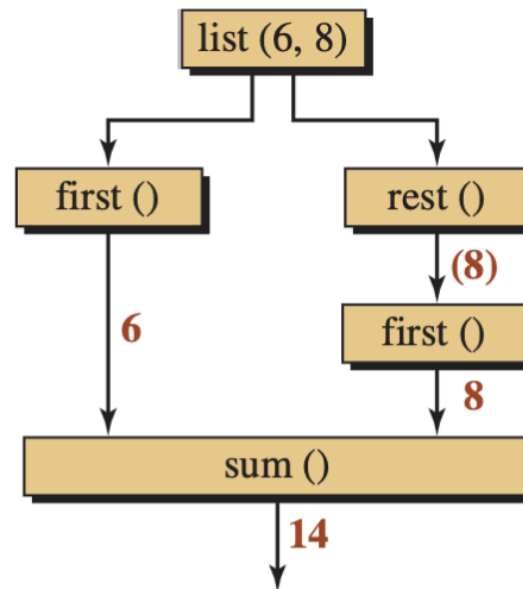




Объектно-
ориентированная
парадигма

```
sum (first (6, 8), first (rest (6, 8)))
```

Code

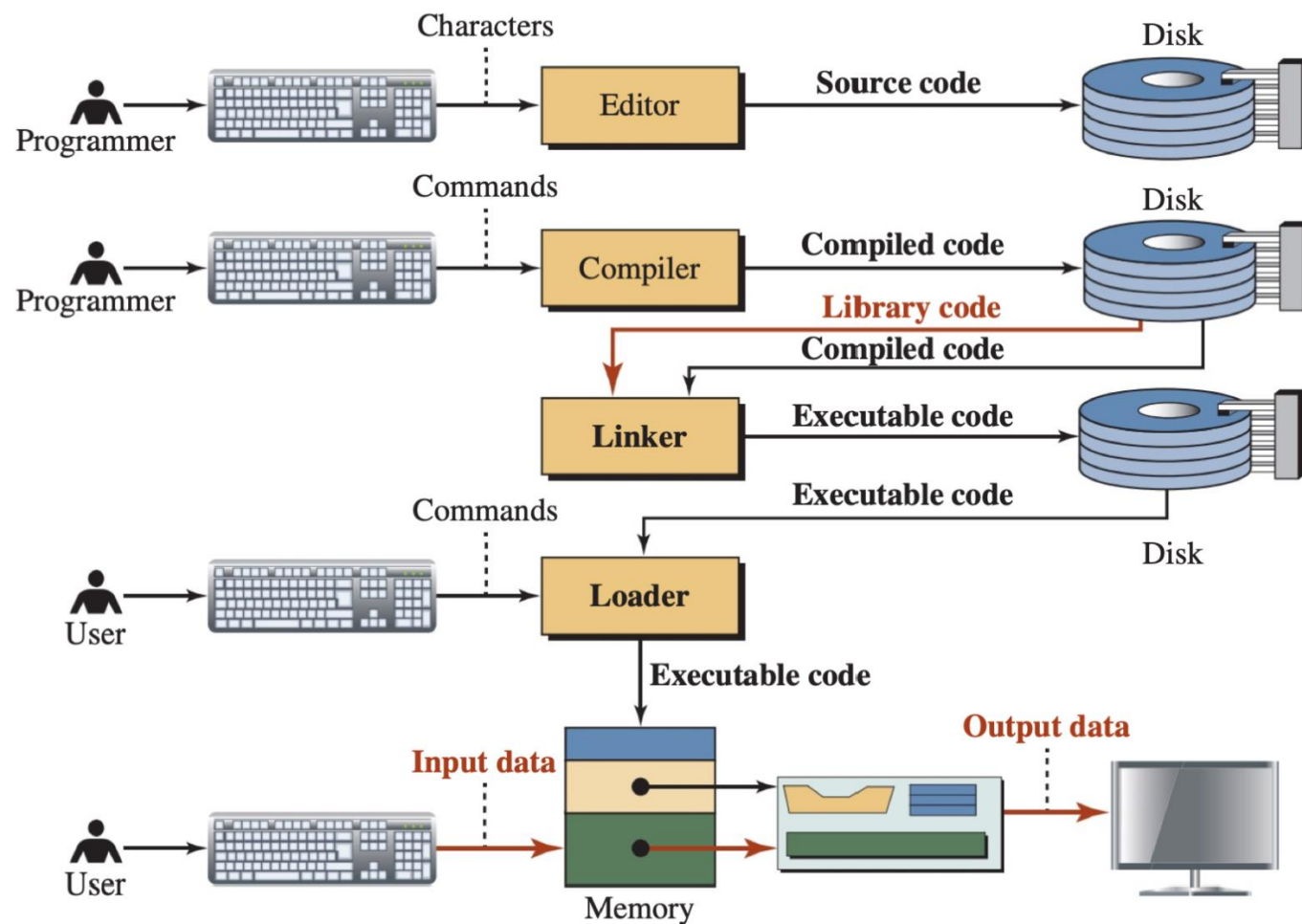


Visualization

Функциональная парадигма



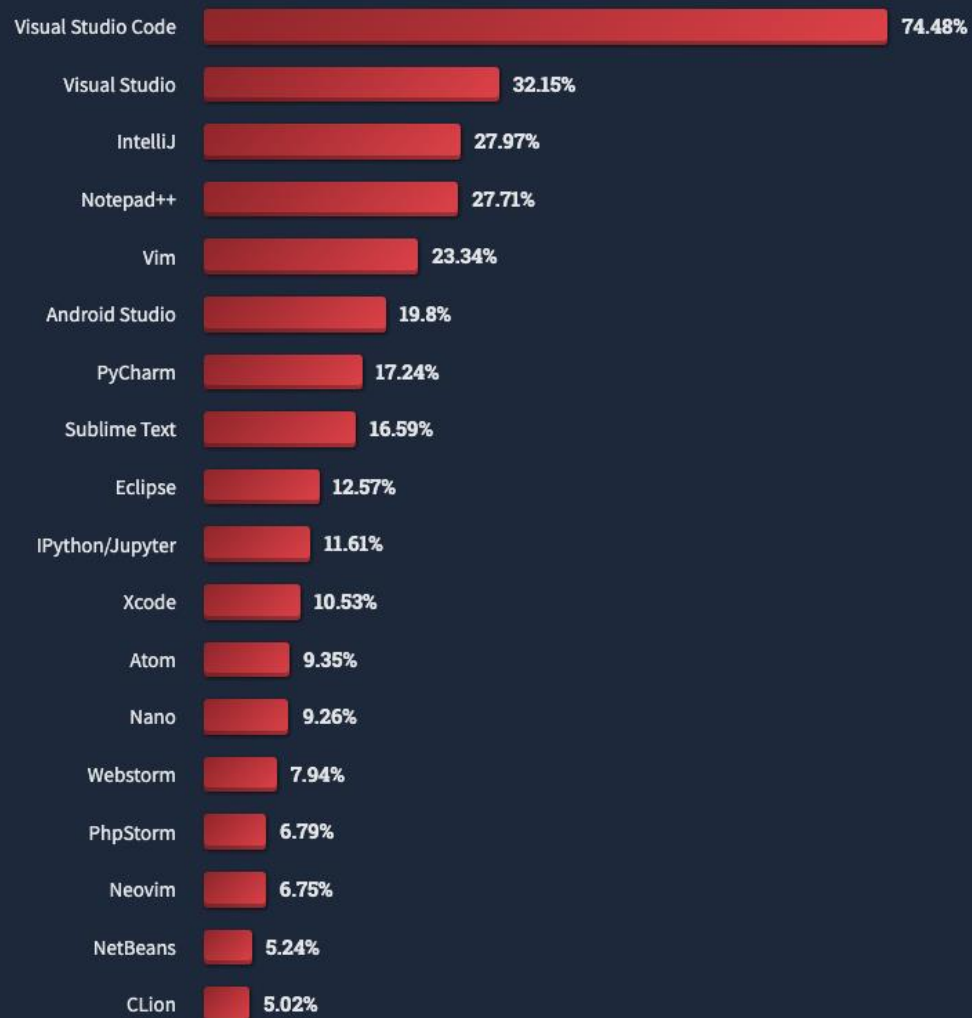
Разработка программ на C++



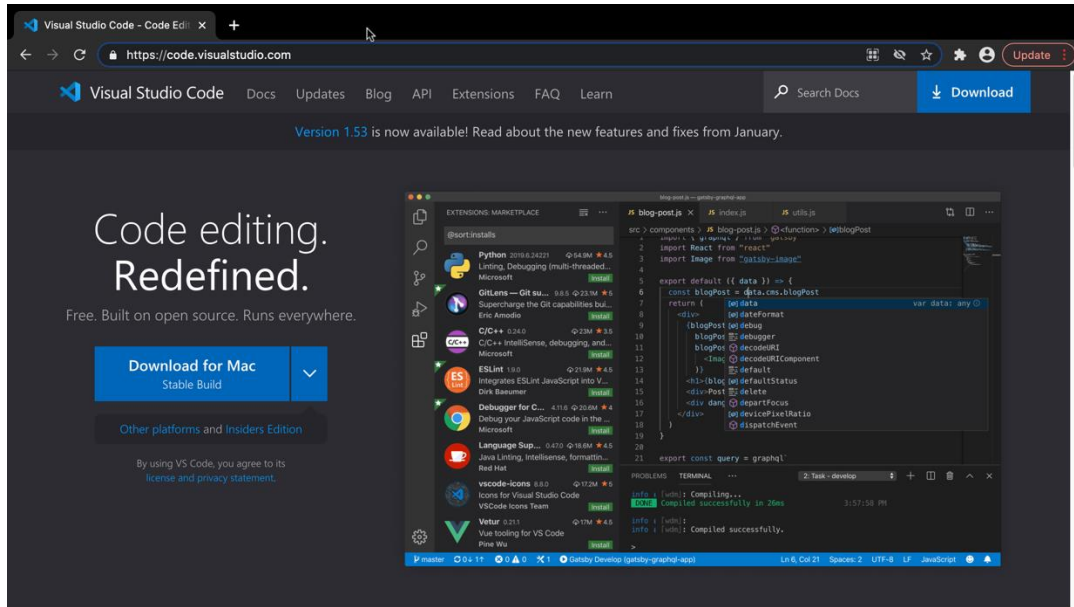
Visual Studio Code

СРЕДА РАЗРАБОТКИ



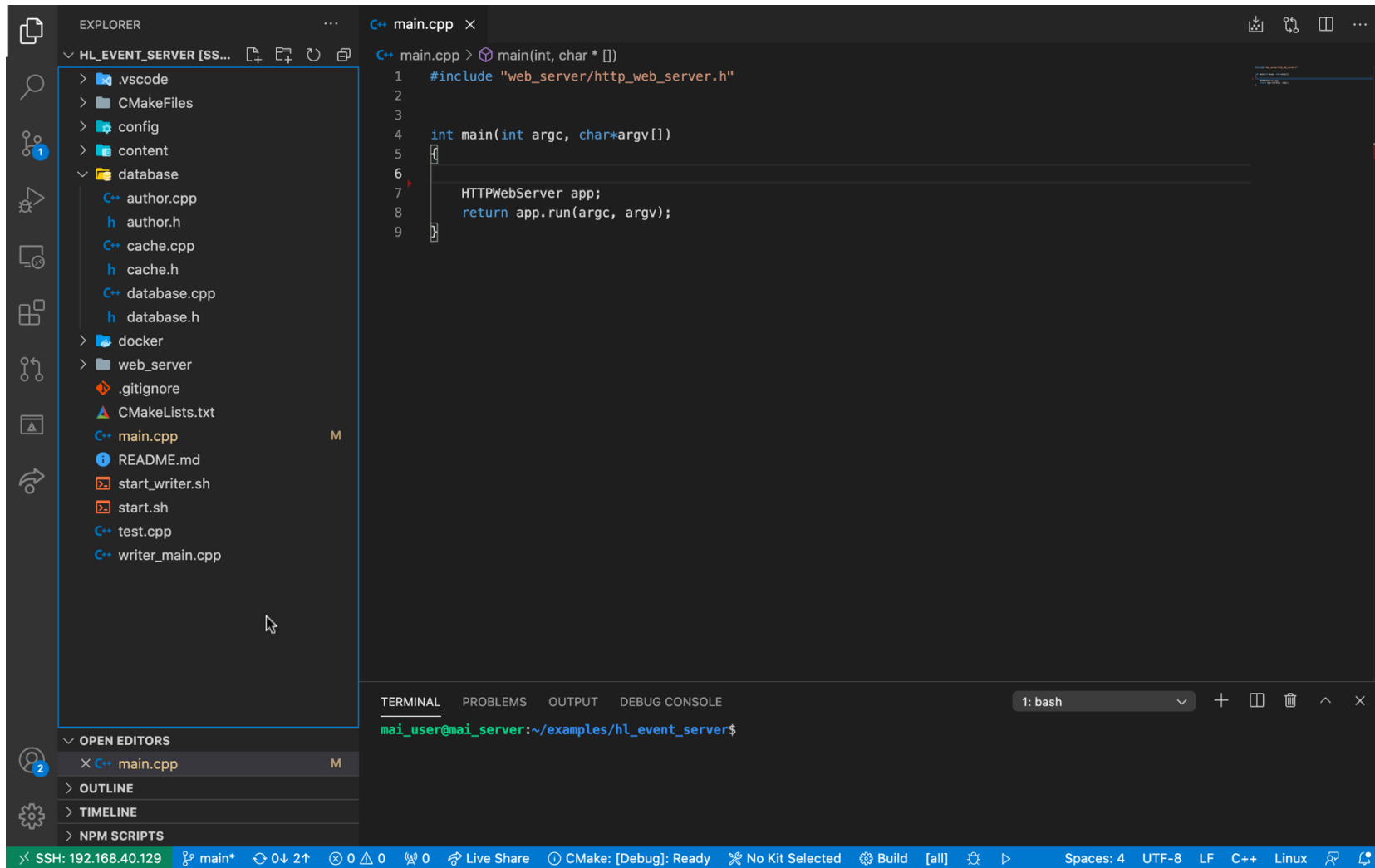


Самый популярный редактор
<https://survey.stackoverflow.co/2023/>

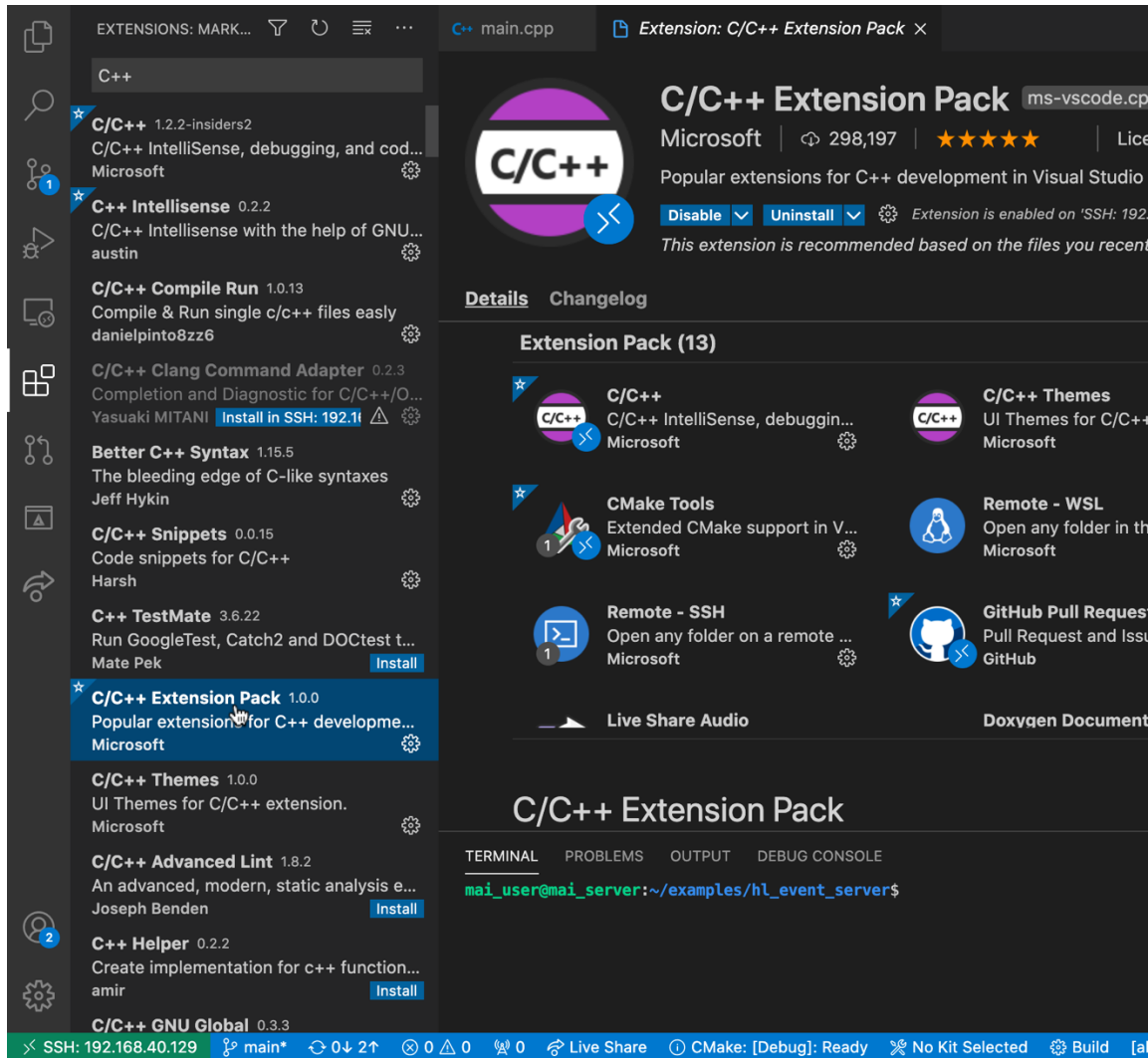


download & install
<https://code.visualstudio.com>

Основные элементы управления



<https://code.visualstudio.com/docs>



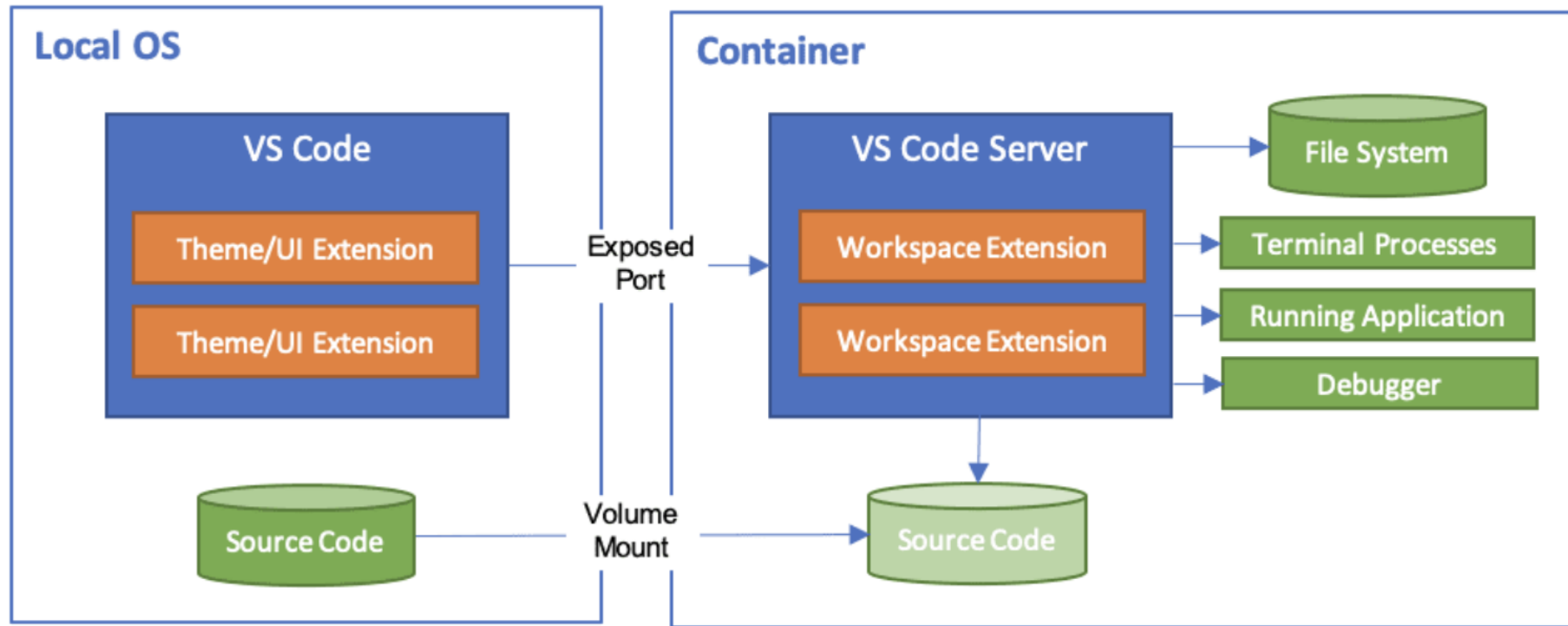
C++ extension pack

GCC 14

ПРИМЕНЯЕМЫЙ
КОМПИЛЯТОР

Docker

[HTTPS://WWW.DOCKER.COM/](https://www.docker.com/)



DevContainer

[HTTPS://CODE.VISUALSTUDIO.COM/DOCS/DEVCONTAINERS/CONTAINERS](https://code.visualstudio.com/docs/devcontainers/containers)

Примеры

[HTTPS://GITHUB.COM/DVDE
MON/MAI_OOP_EXAMPLES
PUBLIC](https://github.com/dvde-mon/mai_oop_examples_public)





cmake

КРОСС-
ПЛАТФОРМЕННАЯ
СБОРКА

ИНСТАЛЛЯЦИЯ

Linux

> sudo apt install cmake

или <https://cmake.org/download/>



простой смаке файл

```
cmake_minimum_required(VERSION 3.10)
```

```
# set the project name
```

```
project(MyProject)
```

```
# add the executable
```

```
add_executable(my_project tutorial.cpp)
```

Библиотеки

```
cmake_minimum_required(VERSION 2.8)
```

```
# Проверка версии CMake.
```

```
# Если версия установленной программы
```

```
# старше указанной, произойдет аварийный выход.
```

```
project(hello_world) # Название проекта
```

```
set(SOURCE_EXE main.cpp)
```

```
# Установка переменной со списком исходников для исполняемого файла
```

```
set(SOURCE_LIB foo.cpp)
```

```
# Тоже самое, но для библиотеки
```

```
add_library(foo STATIC ${SOURCE_LIB})
```

```
# Создание статической библиотеки с именем foo
```

```
add_executable(main ${SOURCE_EXE})
```

```
# Создает исполняемый файл с именем main
```

```
target_link_libraries(main foo) # Линковка программы с библиотекой
```

Более сложный пример

```
cmake_minimum_required(VERSION 3.2)
```

```
project(hl_event_server C CXX)
```

```
SET (EXAMPLE_BINARY "event_server")
```

```
find_package(OpenSSL)
```

```
find_package(Threads)
```

```
find_package(Boost COMPONENTS filesystem system program_options regex REQUIRED)
```

```
include_directories(${Boost_INCLUDE_DIR})
```

```
add_executable(${EXAMPLE_BINARY} main.cpp
```

```
config/config.cpp
```

```
database/database.cpp
```

```
database/cache.cpp
```

```
database/author.cpp)
```

```
target_include_directories(${EXAMPLE_BINARY} PRIVATE "${CMAKE_BINARY_DIR}")
```

```
target_compile_options(${EXAMPLE_BINARY} PRIVATE -Wall -Wextra -pedantic -Werror )
```

```
target_link_libraries(${EXAMPLE_BINARY} PRIVATE
```

```
    ${CMAKE_THREAD_LIBS_INIT}
```

```
    ${Boost_LIBRARIES})
```

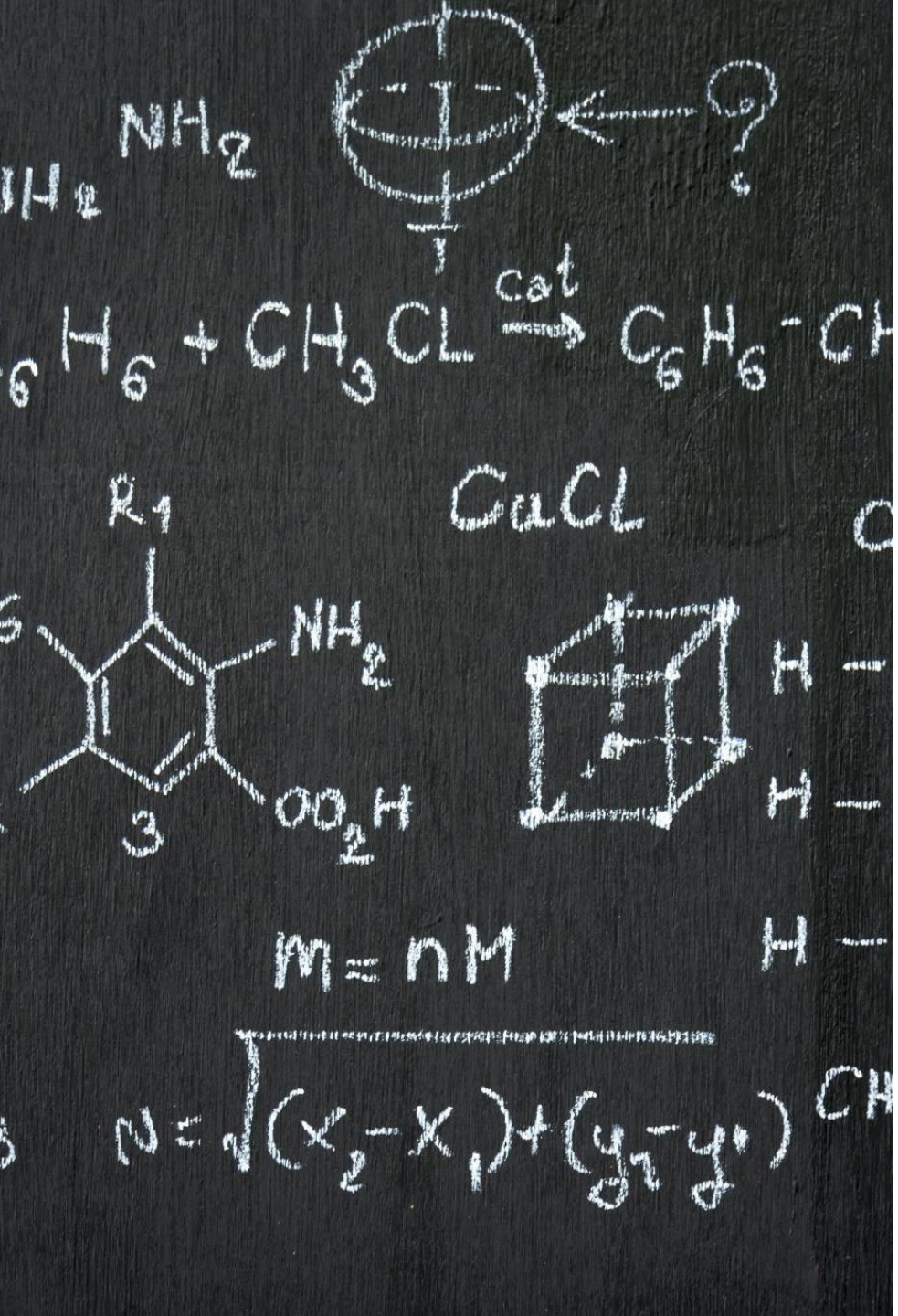
```
set_target_properties(${EXAMPLE_BINARY} PROPERTIES LINKER_LANGUAGE CXX)
```

```
set_target_properties(${EXAMPLE_BINARY} PROPERTIES CXX_STANDARD 17 CXX_STANDARD_REQUIRED ON)
```



Полное описание

[HTTPS://CMAKE.ORG/CM
AKE/HELP/LATEST/GUIDE/
TUTORIAL/INDEX.HTML](https://cmake.org/cmake/help/latest/guide/tutorial/index.html)



План занятия

- ✓ Изучаем базовые конструкции языка C++
- ✓ Изучаем Google Test
- ✓ Типы данных
- ✓ Математические и логические операторы
- ✓ Константы
- ✓ Приведения типов
- ✓ Условия
- ✓ Циклы

простая программа на c++

```
#include <iostream>
int main ()
{
    std::cout << "Это простая программа на C++ ";
    std::cout << "- она показывает ее структуру." << std::endl;
    return 0;
}
```

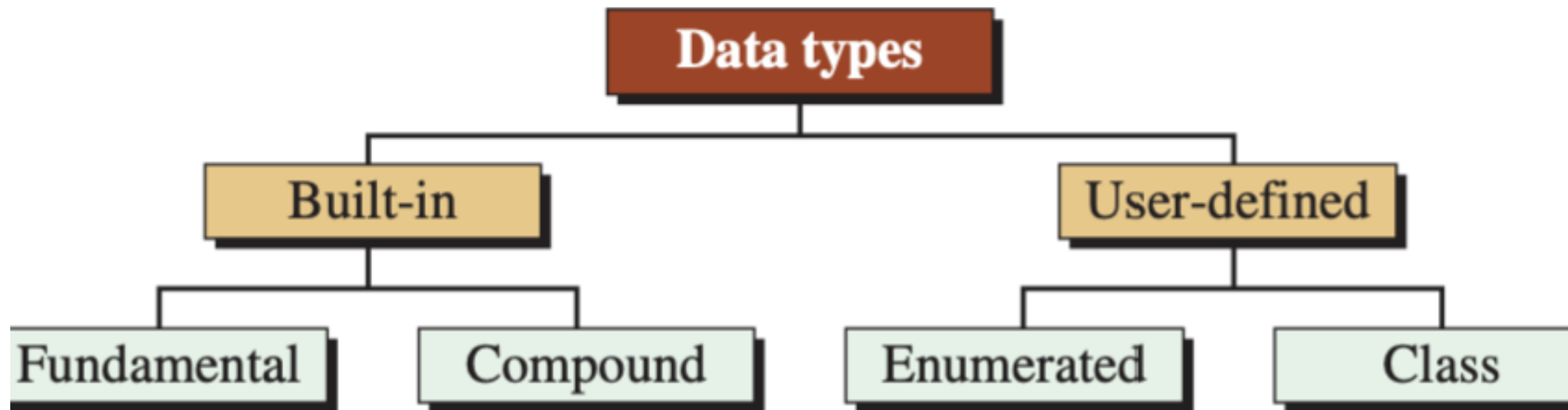


Чуть более сложная программа

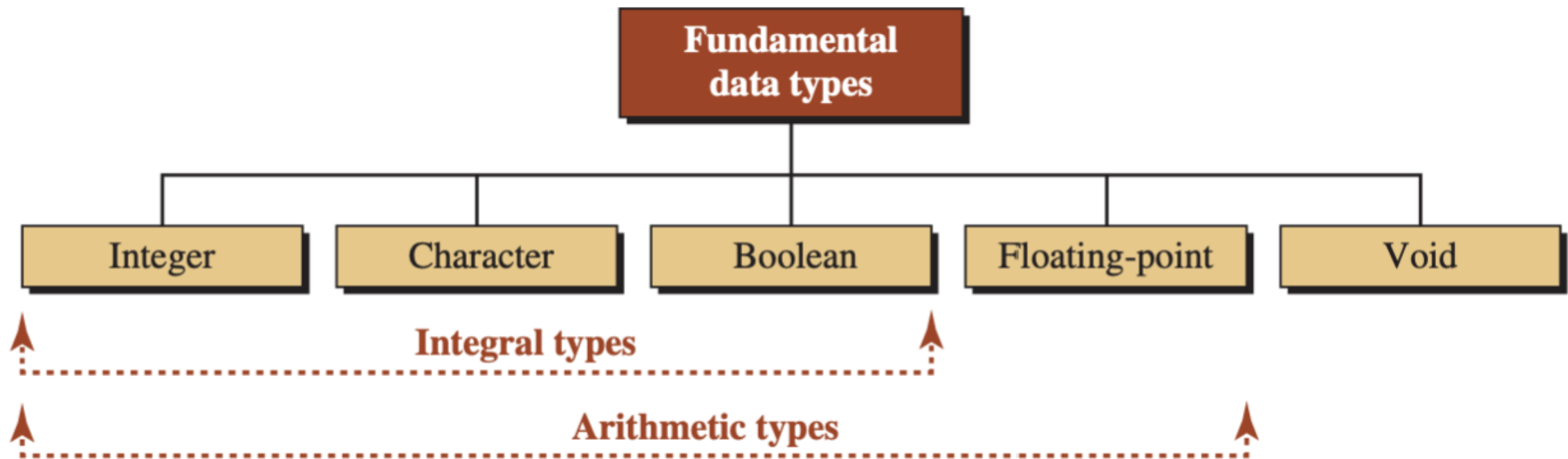
```
#include <iostream>
using namespace std;
int main ()
{
    // Definition
    int num1, num2;
    int sum;
    // Getting inputs
    cout << "Enter the first number: ";
    cin >> num1;
    cout << "Enter the second number: ";
    cin >> num2;
    sum = num1 + num2;
    // Display output
    cout << "The sum is: " << sum << endl;
    return 0;
}
```


alignas	alignof	and	and_eq	asm
auto	bitand	bitor	bool	break
case	catch	char	char16_t	char32_t
class	compl	const	const_cast	constexpr
continue	decltype	default	delete	do
double	dynamic_cast	else	enum	explicit
export	extern	false	float	for
friend	goto	if	inline	int
long	mutable	namespace	new	noexcept
not	not_eq	nullptr	operator	or
or_eq	private	protected	public	register
reinterpret_cast	return	short	signed	sizeof
static	static_assert	static_cast	struct	switch
template	this	thread_local	throw	true
try	typedef	typeid	typename	union
unsigned	using	virtual	void	volatile
wchar_t	while	xor	xor_eq	

КЛЮЧЕВЫЕ СЛОВА C++



Типы данных



Фундаментальные типы данных

Type	Sign	Range	
short int	signed	−32,768	+32,767
	unsigned	0	65,536
int	signed	−2,147,483,648	+2,147,483,647
	unsigned	0	4,294,967,295
long int	signed	−2,147,483,648	+2,147,483,647
	unsigned	0	4,294,967,295

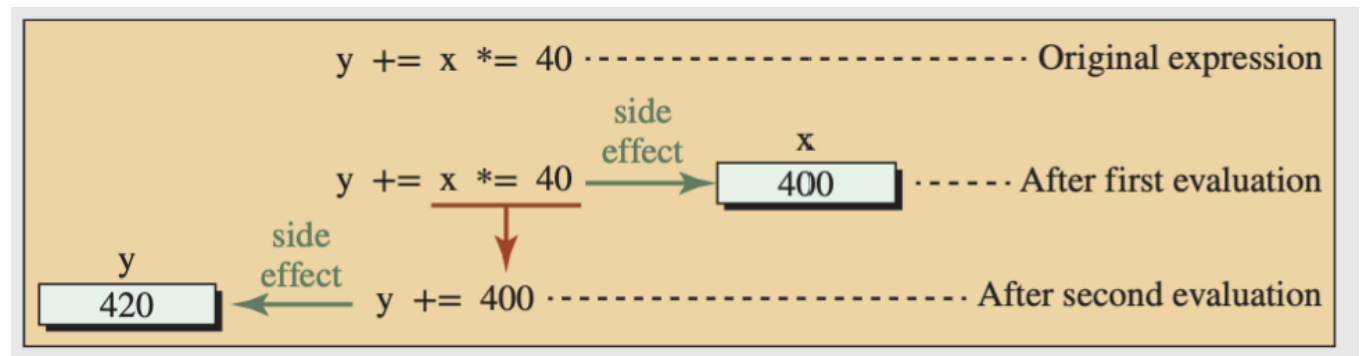
Размеры типов

Literal	Description	Literal	Description
false	// Boolean literal	12897234L	// long integer literal
'A'	// character literal	245.78F	// float literal
"Hello"	// string literal	114.7892	// double literal
234	// integer literal	245.784321L	// long double literal

Литералы

Как вычисляются выражения?

```
#include <iostream>
using namespace std;
int main ( )
{
    int x = 10;
    int y = 20;
    y += x *= 40;
    cout << "Value of y: " << y << endl;
    return 0;
}
```



Инициализация переменных в C++

```
int i1; //undefined value
int i2 = 42; //note: inits with 42
int i3(42); //inits with 42
int i4 = int(42); //inits with 42
int i5{42}; //inits with 42
int i6 = {42}; //inits with 42
int i7{}; //inits with 0
int i8 = {}; //inits with 0
auto i9 = 42; //inits with 42
auto i10{42}; //C++11: std::initializer_list<int>, C++14: int
auto i11 = {42}; //inits std::initializer_list<int> with 42
auto i12 = int{42}; //inits int with 42
int i13(); //declares a function
int i14(7, 9); //compile-time error
int i15 = (7, 9); //OK, inits int with 9 (comma operator)
int i16 = int(7, 9); //compile-time error
```

<https://youtu.be/h-zy1hBqT74>





Переходим в
код

Лабораторная работа №1

1. Установить компилятор C++
2. Установить Cmake
3. Установить Visual Studio Code
4. Установить Google Test
5. Создать аккаунт на GitHub
6. Создать программу на C++
7. Залить программу на GitHub
8. Защитить программу перед преподавателем

C++

Programming:

An Object-Oriented
Approach



BEHROUZ A. FOROUZAN
RICHARD F. GILBERG

Mc
Graw
Hill
Education

Что почитать



Спасибо!

НА СЕГОДНЯ ВСЕ