



# Алгоритмы и Алгоритмические Языки (C++)

Семинар #2:

1. Нововведения C++ по сравнению с C;
2. Сборка многофайлового проекта в C/C++;
3. Makefile для многофайлового проекта;



# Нововведения в C++ по сравнению с C



# Makefile многофайлового проекта



Сравнение реализаций на C и на C++:

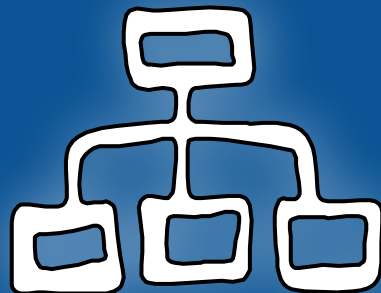
- Структура матрицы на C (см. [10\\_matrix\\_operations](#));
- Этот же проект на базовом C++ (см. [15\\_from\\_c\\_to\\_cpp](#)).

Отличия:

- Новые библиотеки: `<stdlib.h>` -> [<cstdio>/<iostream>](#);
- Дин.память: `malloc()+free()` -> [new](#) + [delete](#);
- Обработка ошибок: `exit()` -> `throw`;
- Контейнер для строк: C-style strings -> [std::string](#);
- Работа с файлами: [std::ifstream](#), [std::ofstream](#);
- Преведение типов: `(double) val` -> [static\\_cast<double>\(val\)](#);



# Сборка многофайлового проекта в C/C++



# Модель сборки проекта в C/C++

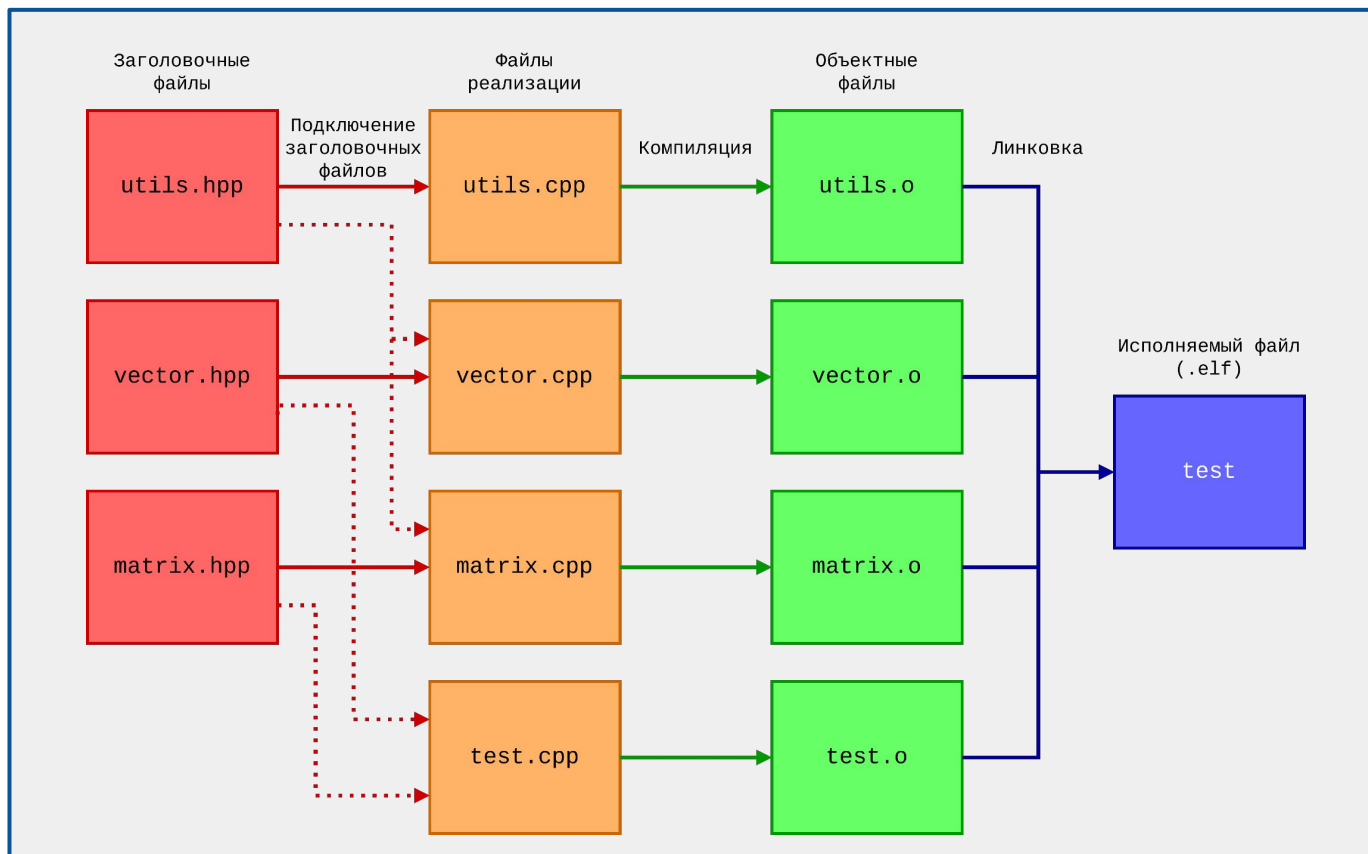


```
> cd examples/15_from_c_to_cpp
```

```
> tree .
```

```
.
├── include
│   ├── matrix.hpp
│   ├── utils.hpp
│   └── vector.hpp
├── Makefile
└── src
    ├── matrix.cpp
    ├── test.cpp
    ├── utils.cpp
    └── vector.cpp
```

# Модель сборки проекта в C/C++



# Модель сборки проекта в C/C++



```
> make
```

```
Building object file build/test.o
```

```
g++ -c src/test.cpp <CFLAGS> -I /home/path/to/include  
-o build/test.o
```

```
Building object file build/utils.o
```

```
Building object file build/vector.o
```

```
Building object file build/matrix.o
```

```
Linking executable build/test
```

```
g++ <LINKFLAGS> build/test.o build/utils.o build/vector.o  
build/matrix.o -o build/test
```



# Шаблон Makefile для многофайлового проекта







# Makefile многофайлового проекта

Полезные ссылки:

- [Шаблон Makefile-a](#) для многофайлового проекта;
- [Makefile](#), адаптированный для проекта [15\\_c\\_from\\_cpp](#).

Обратить внимание:

- “Ленивая” перекомпиляция;
- Ошибки этапа линковки (см. `src/utls.cpp:explanation_str`);
- **[BUG]** Отсутствие пересборки в сценарии:  
`make --> make DEBUG=1`  
(“слишком ленивая” перекомпиляция)
- Что делает каждый из флагов `g++`?

# Вопросы?

