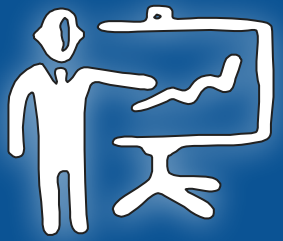
Алгоритмы и Алгоритмические Языки

Семинар #17:

1. План на модуль и система оценивания.
2. Многофайловые программы и раздельная компиляция.
3. Нововведения в C++ по сравнению с C.

17.01.2025

План на модуль
и система оценивания



Система оценивания

ИТОГ = 0,1×ACT + 0,5×(0,45×HW1 + 0,55×HW2) + 0,4×EXAM.

Активность (ACT) выставляется на основе N небольших контестов (будут выдаваться на семинарах).

Домашние задания (HW1, HW2):

- ДЗ №1 – выдача требований 27.01 - 02.02.
- ДЗ №2 – выдача требований 10.02 - 16.02.
- Каждый проект содержит 10 требований,
  оценка – сумма баллов по отдельным требованиям.
- Дедлайны – как в 1-2 модулях.

Экзамен – теоретические задачи по материалу лекций в ejudge.

Разрыв образование/промышленность

Контесты по программированию		**Работа в промышленности**
Локальную абстрактную задачу	**Ориентация на**	Целостный продукт
500 строк кода	**Объём задачи**	500000 строк кода
Узкий и постоянный	**Контекст работы**	Широкий и меняющийся
Импульсная фрагментарная	**Характер работы**	Непрерывная итерационная
Только со своим	**Работа с кодом**	Со своим и чужим
Индивидуальная	**Разработка**	Командная
Мало	**Стандарты и литература**	Много
Нет	**Документация и сопровождение**	Обязательны

Промышленное программирование

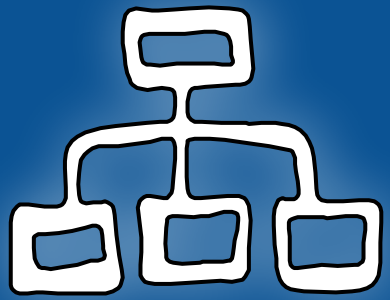
Техники промышленного программирования

- Модульность и интерфейсы
- Переиспользование ПО
- Версионирование ПО
- Внутреннее самотестирование ПО
- Удобство использования ПО
- Разработка и использование библиотек

Что будем изучать?

- Классы в C++, раздельная компиляция
- Иерархии классов и шаблоны в C++
- Система контроля версий Git
- Разработка тестов для библиотеки
- «Синтаксический сахар» в C++
- Система сборки Make

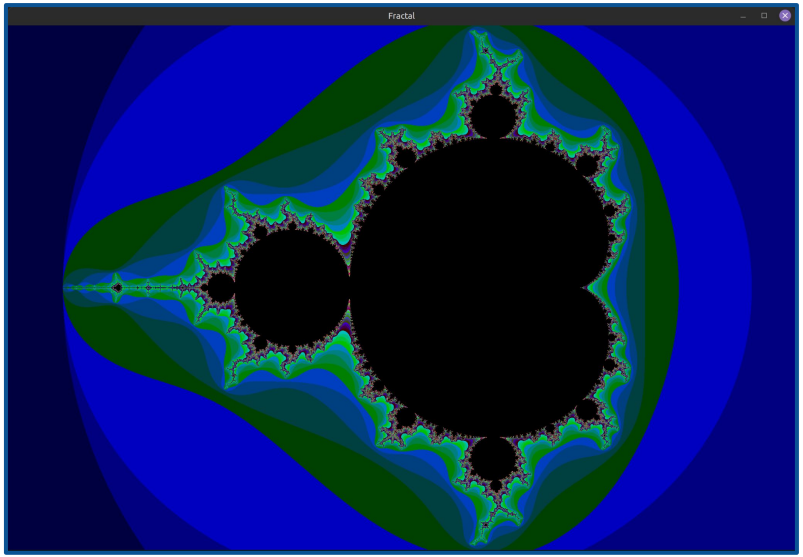
Многофайловые программы и раздельная компиляция



Минимальное графическое приложение

Составные части:

- Графическая библиотека.
- Вычисления цветов фрактала.
- Вспомогательные функции.
- Управление:
  перемещение (WASD) + zoom (OP).



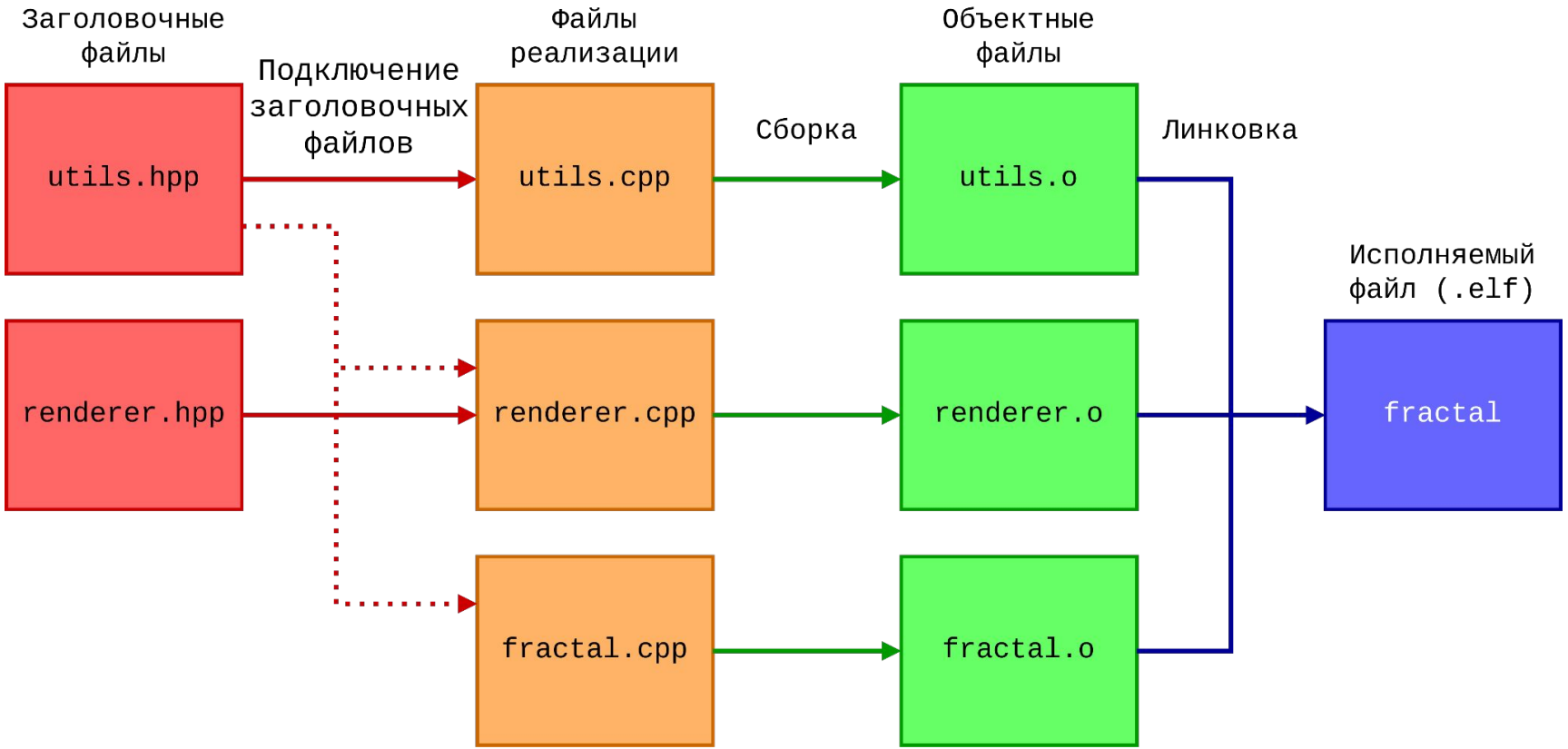
Как разбить программу на части?
Как потом состыковать эти части?

Модель сборки проекта в C/C++

```
❯ cd examples/17_basic_cpp
❯ tree .
.
├── build
│   ├── fractal
│   ├── fractal.o
│   ├── renderer.o
│   └── utils.o
├── include
│   ├── renderer.hpp
│   └── utils.hpp
...
```

```
❯ cd examples/17_basic_cpp
❯ tree .
...
├── Makefile
└── src
    ├── fractal.cpp
    ├── renderer.cpp
    └── utils.cpp
```

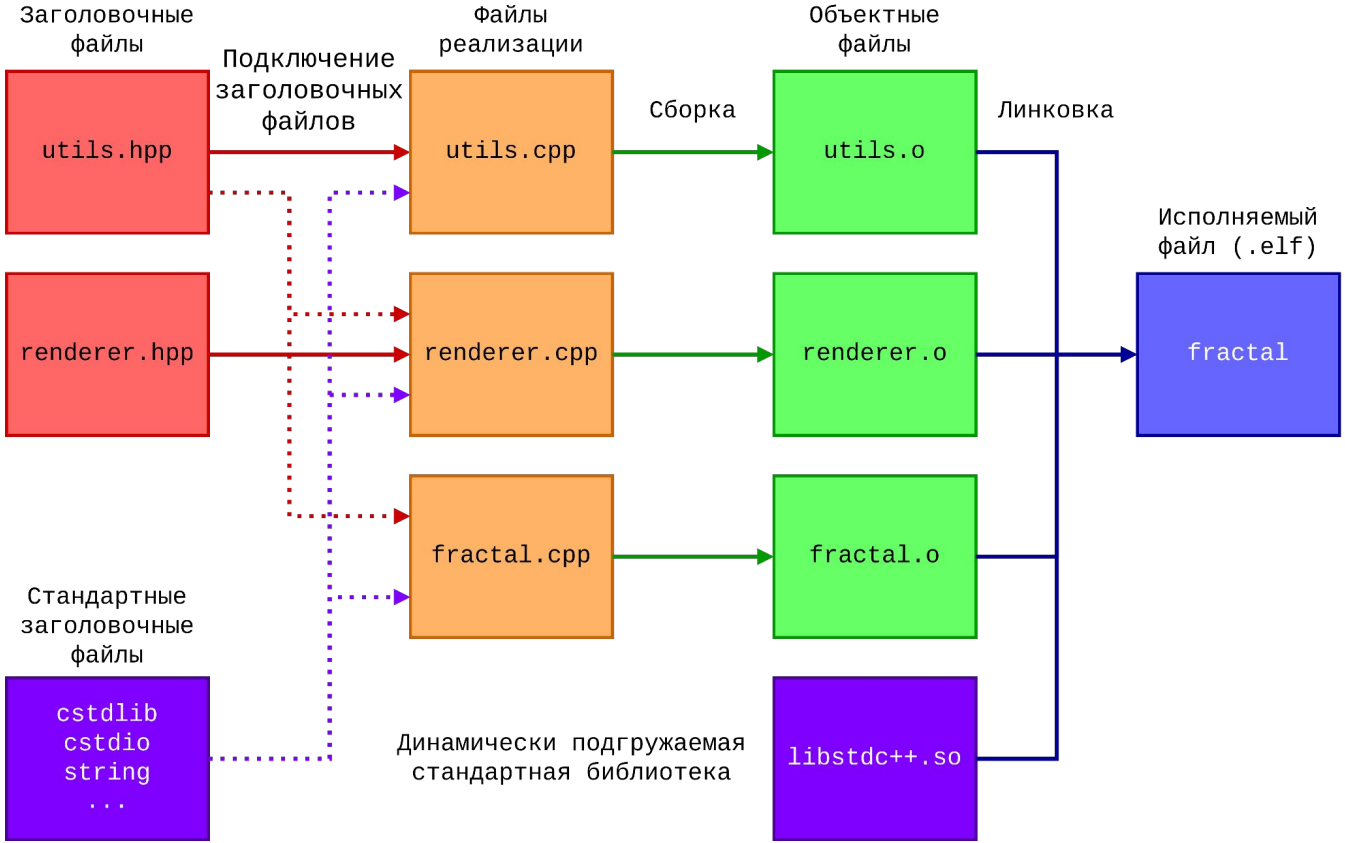
Модель сборки проекта в C/C++



Подключение стандартной и др. библиотек

```
❯ readelf -d build/fractal
Dynamic section at offset 0x6c58 contains 32 entries:
  Tag        Type                         Name/Value
 0x0000000000000001 (NEEDED) Shared library: [libc.so.6]
 0x0000000000000001 (NEEDED) Shared library: [libX11.so.6]
 0x0000000000000001 (NEEDED) Shared library: [libXext.so.6]
 0x0000000000000001 (NEEDED) Shared library: [libstdc++.so.6]
 0x0000000000000001 (NEEDED) Shared library: [libgcc_s.so.1]
 0x0000000000000001 (NEEDED) Shared library: [libpthread.so.0]
 ...
```

Подключение стандартной и др. библиотек



Заголовочный файл и файл реализации

Заголовочный файл: минимальный набор того, что нужно для использования компонента.

Файл реализации: всё остальное.

Защита от повторного подключения

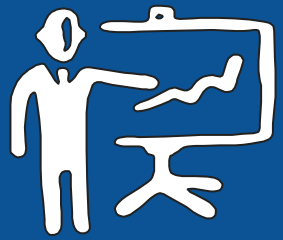
```
#ifndef HEADER_GUARD_UTILS_H_INCLUDED
#define HEADER_GUARD_UTILS_H_INCLUDED

// Объявления макроопределений, функций, типов.

#endif // HEADER_GUARD_UTILS_H_INCLUDED
```

Нововведения в C++ по сравнению с C

Вопросы?



Красивые иконки взяты с сайта handdrawngoods.com