

2.2. Получить профессиональные навыки по эксплуатации и сопровождению прикладного программного обеспечения

Использование системы LMS Moodle

Создание и развертывание собственного курса на платформе организации

Гришутенко Павел



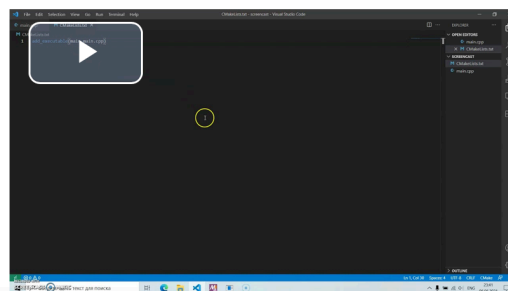
Работа с CMake

Курс в формате микрообучения посвящен работе с утилитой автоматизированной сборки Cmake.

В состав курса входят несколько обучающих материалов теории. Тестовые задания для проверки усвоенного материала. А также практические работы и некоторые сторонние материалы, которые могут быть полезны при работе с утилитой.



Содержание



Уроки

Занятие 1



Занятие 1



К занятию 1



Тест по теме занятия 1

Занятие 2



Занятие 2



Тест по теме занятия 2

Занятие 3



Занятие 3



К занятию 3



Тест по теме занятия 3

Справочные материалы



Уроки по CMake из официальной документации






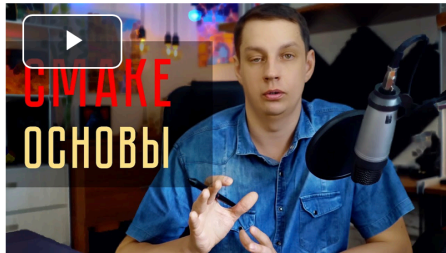
Справочные материалы






Дополнительный практикум

Справочные материалы

-  Уроки по CMake из официальной документации
-  Справочные материалы
-  Дополнительный практикум



footer

-  Глоссарий по теме
-  Автор курса
-  Обсуждение

Царулкова Анастасия Витальевна

Пример составленных заданий

Занятие 1

Cmake – утилита для автоматизации сборки программ на языке C/C++.

Проблема: при сборке проекта возникают проблемы с различными зависимостями, добавлением библиотек, добавлением ресурсов, адаптация сборки под различные ОС.

Cmake решает эти проблемы простым методом. Создается единый файл с конфигурацией сборки, после чего можно вызывать сборку одной консольной командой. Добавление, например, одной библиотеки осуществляется с помощью новой строки в файл конфигурации.

Основной файл, содержащий конфигурацию сборки, носит название CMakeLists.txt

Задание

1. Напишите простую программу на языке C/C++ в файл main.cpp.

```
#include <stdio.h>

int main()
{
    printf("Hello Cmake");
    return 0;
}
```

2. Напишите CMakeLists.txt файл.

```
cmake_minimum_required (VERSION 3.17)

cmake_project(helloCmake)

add_executable(helloCmake main.cpp)
```

3. Создайте каталог build перейдите в него и выполните следующие команды.

Важно: опция после ключа -G зависит от использования компилятора, в своем случае я использую компилятор MinGW, на Unix системе опция может выглядеть так "Unix Makefiles". Более точную информацию посмотрите в документации Cmake по вашей системе и компилятору.

```
C:\Users\grish\Documents\cpp\cpp-projects\cmakeEdu\build>cmake .. -G"MinGW Makefiles"
```

После этой команды мы сконфигурировали makefile, который скажет системе как собирать проект.

Сборка осуществляется командой

Занятие 3

Давайте попробуем воспользоваться знаниями, полученными после прохождения занятий 1 и 2, чтобы собрать небольшую программу на одном из самых популярных GUI фреймворков QT.

Стоит отметить, что существует множество нюансов при сборке программ, написанных с использованием QT, которые мы рассматривать не станем, но некоторые из таких нюансов мы посмотрим: QT имеет собственную систему сборки QMake, QT использует ряд специфичных CMake опций в своей работе.

Задание

1. Скачайте и установите QT.
2. Давайте запишем простую программу в файле main.cpp.

```
#include <QtWidgets>

int main(int argc, char** argv)
{
    QApplication app(argc, argv);
    QWidget wgt;

    QWidget* pwgt1 = new QWidget(&wgt);
    QPalette pal1;
    pal1.setColor(pwgt1->backgroundRole(), Qt::blue);
    pwgt1->setPalette(pal1);
    pwgt1->resize(100, 100);
    pwgt1->move(25, 25);
    pwgt1->setAutoFillBackground(true);

    QWidget* pwgt2 = new QWidget(&wgt);
    QPalette pal2;
    pal2.setBrush(pwgt2->backgroundRole(), QBrush(QPixmap(":/stone.jpg")));
    pwgt2->setPalette(pal2);
    pwgt2->resize(100, 100);
    pwgt2->move(75, 75);
    pwgt2->setAutoFillBackground(true);

    wgt.resize(200, 200);
    wgt.show();
    return app.exec();
}
```

3. В проекте мы используем файл ресурсов, он нужен чтобы мы могли добавить внешние данные в нашу программу, например, в нашем случае мы используем изображение stone.jpg.

Пример выполнения работы

6. Во второй категории создайте два вопроса с открытым (кратким) ответом.

Банк вопросов

Выберите категорию: Категория 2 для Тест 1 (2)

Категория 2 для общих вопросов в контексте «Тест 1».

Фильтры тегов не применялись

Фильтр по тегам...

☐ Отображать текст вопроса в списке вопросов

Параметры поиска

☒ Также отображать вопросы, находящиеся в подкатегориях

☐ Также отображать старые вопросы

Создать новый вопрос...

Тип Вопрос	Действия	Создан	Последнее изменение
Название вопроса / ID-номер		Имя / Фамилия / Дата	Имя / Фамилия / Дата
<input type="checkbox"/> Вопрос на знание платформы	Редактировать	Pavel Grishutenko 21 Сентябрь 2021, 16:11	Pavel Grishutenko 21 Сентябрь 2021, 16:11
<input type="checkbox"/> Вопрос на знание платформы	Редактировать	Pavel Grishutenko 21 Сентябрь 2021, 16:10	Pavel Grishutenko 21 Сентябрь 2021, 16:10

С выбранными:

7. Создайте в курсе тест на 5 баллов, набрав его из вопросов первой и второй категорий.

Редактирование теста: Тест 1

Вопросы: 5 | Этот тест открыт

Максимальная оценка 5,00 Сохранить

Распределить Выбери несколько элементов

Итоговый балл: 5,00

☐ Перемешать

Страница 1

1 Вопрос на знание платформы Какие инструменты использ...

2 Вопрос на знание платформы Паттерн проектирования ис...

3 Вопрос на знание платформы Разработка IOS приложений...

Страница 2

4 Вопрос на знание платформы Хранение данных осуществ...

Страница 3

5 Вопрос на знание платформы Основные языки разработк...