

# Содержание

<b>От издательства</b> .....	15
<b>Вступительное слово</b> .....	16
<b>Об авторе</b> .....	17
<b>О рецензентах</b> .....	18
<b>Предисловие</b> .....	19
<b>Глава 1. Первые шаги</b> .....	23
Технические требования.....	24
Основные факты .....	24
Что такое CMake? .....	25
Как она работает? .....	26
Этап конфигурирования .....	27
Этап генерирования.....	27
Этап сборки .....	28
Установка CMake на различных платформах .....	30
Docker .....	31
Windows .....	32
Linux.....	33
macOS.....	34
Сборка из исходного кода.....	34
Работа с командной строкой .....	35
Командная строка CMake.....	35
Генерирование сборочной системы проекта .....	35
Сборка проекта .....	41
Установка проекта.....	44
Выполнение скрипта.....	45
Выполнение командной программы .....	45
Выполнение готовой конфигурации технологического процесса.....	46
Получение справки .....	46
Командная строка CTest.....	47
Командная строка CPack.....	47
Графический интерфейс CMake .....	48

Командная строка CCMake .....	49
Навигация по файлам и каталогам проекта .....	49
Дерево исходного кода.....	50
Дерево сборки .....	50
Списковые файлы .....	51
Файл проекта.....	51
Файл кеша.....	52
Файл определения пакета .....	53
Сгенерированные файлы .....	53
Файлы в формате JSON и YAML .....	53
Файлы готовых конфигураций.....	54
Файловый API.....	54
Журнал конфигурирования .....	54
Игнорирование файлов в Git .....	55
Скрипты и модули .....	55
Скрипты .....	56
Служебные модули .....	56
Модули поиска .....	57
Резюме .....	57
Для дополнительного чтения.....	58
<b>Глава 2. Язык CMake .....</b>	<b>59</b>
Технические требования.....	60
Основы синтаксиса языка CMake .....	60
Комментарии.....	61
Вызов команд .....	62
Аргументы команд .....	63
Аргументы в квадратных скобках .....	64
Аргументы в кавычках .....	65
Аргументы без кавычек.....	66
Работа с переменными.....	67
Ссылки на переменные.....	68
Переменные окружения .....	69
Кешированные переменные .....	71
Как правильно использовать области видимости переменных в CMake.....	72
Списки .....	74
Управляющие конструкции в CMake .....	76
Условные блоки .....	76
Синтаксис условных команд.....	76
Циклы .....	80
Цикл while() .....	80
Цикл foreach().....	80
Определения команд .....	82
Макросы .....	83

Функции.....	84
Процедурная парадигма в CMake .....	85
К вопросу о соглашениях об именовании .....	87
Наиболее употребительные команды.....	87
Команда message() .....	87
Команда include() .....	89
Команда include_guard().....	90
Команда file() .....	90
Команда execute_process() .....	91
Резюме .....	91
Для дополнительного чтения.....	92
<b>Глава 3. Использование CMake в популярных IDE .....</b>	<b>93</b>
Знакомство с IDE .....	94
Выбор IDE.....	95
Выбирайте функционально насыщенную IDE .....	95
Выбирайте IDE, широко поддерживаемую в вашей организации.....	96
Не выбирайте IDE, исходя из целевой ОС и платформы.....	96
Выбирайте IDE, поддерживающую удаленную разработку .....	96
Установка комплекта инструментов.....	97
Использование примеров из этой книги совместно с IDE .....	98
Начало работы с CLion IDE .....	99
Почему она может вам понравиться.....	100
Первые шаги .....	101
Крутая штука: накачанный отладчик .....	103
Начало работы с Visual Studio Code .....	103
Почему она может вам понравиться.....	104
Первые шаги .....	105
Крутая штука: контейнеры разработчика .....	105
Начало работы с Visual Studio IDE.....	106
Почему она может вам понравиться.....	107
Первые шаги .....	107
Крутая штука: отладка в режиме горячей перезагрузки .....	108
Резюме .....	109
Для дополнительного чтения.....	110
<b>Глава 4. Подготовка первого проекта на основе CMake .....</b>	<b>111</b>
Технические требования.....	112
Основные директивы и команды .....	112
Задание минимальной версии CMake .....	113
Определение языков и метаданных .....	113
Разбиение проекта на части.....	115
Управление областью видимости с помощью каталогов .....	117

Когда использовать вложенные проекты.....	119
Внешние проекты должны оставаться внешними .....	119
Размышления о структуре проекта.....	120
Определение окружения.....	125
Определение операционной системы.....	125
Кросс-компиляция: что такое хост-система и целевая система?.....	126
Сокращенные переменные .....	126
Информация о хост-системе.....	127
Архитектура платформы 32- или 64-разрядная?.....	128
Каков порядок байтов в системе? .....	128
Конфигурирование комплекта инструментов .....	129
Задание стандарта C++.....	129
Принудительная поддержка стандарта.....	130
Расширения производителя.....	130
Межпроцедурная оптимизация .....	131
Проверка возможностей, поддерживаемых компилятором.....	132
Компиляция тестового файла .....	132
Запрет внутренней сборки .....	134
Резюме .....	134
Для дополнительного чтения.....	135
<b>Глава 5. Цели .....</b>	<b>136</b>
Технические требования.....	137
Понятие цели .....	137
Определение исполняемых целей .....	138
Определение библиотечных целей .....	139
Специальные цели.....	139
Граф зависимостей .....	140
Визуализация зависимостей.....	142
Задание свойств целей.....	143
Что такое транзитивные требования использования? .....	144
Конфликты распространяемых свойств .....	147
Знакомство с псевдоцелями .....	148
Импортированные цели .....	149
Псевдонимы целей .....	149
Интерфейсные библиотеки.....	149
Объектные библиотеки.....	150
Цели сборки .....	151
Написание специальных команд .....	152
Использование специальной команды в качестве генератора .....	153
Использование специальной команды в качестве точки вклинивания.....	154
Резюме .....	155
Для дополнительного чтения.....	156

---

<b>Глава 6. Генераторные выражения.....</b>	157
Технические требования.....	158
Что такое генераторные выражения? .....	158
Основные синтаксические правила составления выражения общего вида ...	159
Вложенность .....	160
Условное расширение.....	161
Вычисление булева значения .....	162
Логические операторы .....	162
Сравнения.....	162
Запросы.....	163
Запросы и преобразования .....	163
Работа со строками, списками и путями.....	163
Параметризация конфигурирования и платформы сборки .....	165
Настройка комплекта инструментов.....	166
Запрос целеориентированной информации .....	168
Экранирование.....	170
Примеры.....	170
Конфигурации сборки.....	170
Системно-зависимые однострочные выражения .....	171
Интерфейсные библиотеки с зависящими от компилятора флагами.....	171
Вложенные генераторные выражения .....	172
Различие между условным выражением и вычислением оператора BOOL .....	173
Резюме .....	174
Для дополнительного чтения.....	175

---

<b>Глава 7. Компиляция исходного кода на C++ с помощью CMake .....</b>	176
Технические требования.....	177
Основы компиляции .....	177
Как устроена компиляция .....	178
Начальная конфигурация.....	180
Требование конкретных возможностей от компилятора .....	181
Управление исходными файлами для целей.....	182
Конфигурирование препроцессора.....	183
Задание путей к включаемым файлам.....	183
Определения препроцессора .....	184
Избегайте доступа к закрытым полям класса в автономных тестах.....	185
Использование фиксаций git для отслеживания откомпилированной версии .....	186
Конфигурирование заголовочных файлов .....	187
Конфигурирование оптимизатора .....	188
Общий уровень .....	190
Встраивание функций.....	191

Разворачивание циклов.....	192
Векторизация циклов.....	193
Управление процессом компиляции .....	194
Уменьшение времени компиляции .....	195
Предкомпиляция заголовочных файлов .....	195
Унифицированные сборки.....	197
Поиск ошибок.....	199
Конфигурирование ошибок и предупреждений.....	199
Отладка сборки .....	200
Предоставление информации отладчику.....	202
Резюме .....	204
Для дополнительного чтения.....	205
<b>Глава 8. Компоновка исполняемых файлов и библиотек .....</b>	<b>206</b>
Технические требования.....	207
Основы компоновки .....	207
Сборка библиотек разных типов .....	211
Статические библиотеки .....	212
Разделяемые библиотеки .....	212
Разделяемые модули .....	214
Позиционно-независимый код (PIC).....	215
Решение проблем с помощью ODR .....	216
Разбираемся с динамически скомпонованными	
символами-дубликатами.....	218
Используйте пространства имен – не полагайтесь на компоновщик.....	220
Порядок компоновки и неразрешенные символы .....	220
Символы, на которые нет ссылок.....	223
Вынесение логики main() для тестирования.....	224
Резюме .....	227
Для дополнительного чтения.....	228
<b>Глава 9. Управление зависимостями в CMake .....</b>	<b>229</b>
Технические требования.....	230
Использование уже установленных зависимостей .....	230
Нахождение пакетов с помощью команды CMake <code>find_package()</code> .....	231
Написание модулей своими силами .....	236
Нахождение унаследованных пакетов с помощью <code>FindPkgConfig</code> .....	242
Использование зависимостей, отсутствующих в системе.....	244
FetchContent .....	245
Простой пример программы чтения YAML .....	246
Скачивание зависимостей .....	248
Обновление и исправление .....	251
Использование установленной зависимости, если это возможно .....	251
ExternalProject .....	253

---

Резюме .....	255
Для дополнительного чтения.....	255
<b>Глава 10. Использование модулей C++20.....</b>	<b>256</b>
Технические требования.....	257
Что такое модули C++20? .....	257
Написание проектов с поддержкой модулей C++20 .....	260
Активация экспериментальной поддержки в CMake 3.26 и 3.27 .....	260
Включение поддержки для CMake 3.28 и старше.....	262
Задание требований к компилятору.....	262
Объявление модуля C++ .....	263
Конфигурирование комплекта инструментов .....	264
Резюме .....	265
Для дополнительного чтения.....	266
<b>Глава 11. Каркасы тестирования .....</b>	<b>267</b>
Технические требования.....	268
Почему нужно задуматься об автоматизированных тестах? .....	268
Применение CTest для стандартизации тестирования в CMake.....	269
Режим сборки и тестирования.....	270
Режим тестирования .....	271
Опрос тестов.....	272
Фильтрация тестов .....	272
Тасование тестов .....	273
Обработка ошибок.....	274
Повтор тестов.....	274
Управление выводом .....	275
Разное .....	276
Создание элементарного автономного теста для CTest.....	277
Структурирование проектов для тестирования .....	281
Каркасы автономного тестирования .....	285
Catch2.....	286
GoogleTest .....	288
Работа с GTest.....	288
GMock .....	291
Генерирование отчетов о покрытии тестами .....	296
Применение LCOV для генерирования отчетов о покрытии .....	298
Как избежать SEGFAULT .....	301
Резюме .....	302
Для дополнительного чтения.....	303
<b>Глава 12. Инструменты анализа программ .....</b>	<b>304</b>
Технические требования.....	305

Принудительное форматирование .....	305
Использование статических проверщиков.....	309
clang-tidy .....	312
Cpplint.....	312
Cppcheck .....	313
include-what-you-use .....	313
Link What You Use.....	313
Динамический анализ с помощью Valgrind.....	314
Memcheck .....	315
Memcheck-Cover .....	318
Резюме .....	320
Для дополнительного чтения.....	321
<b>Глава 13. Генерирование документации .....</b>	<b>322</b>
Технические требования.....	323
Добавление Doxygen в проект .....	323
Генерирование документации, выглядящей современно .....	328
Обогащение вывода специальным HTML-кодом.....	330
Резюме .....	333
Для дополнительного чтения.....	334
<b>Глава 14. Установка и упаковка .....</b>	<b>335</b>
Технические требования.....	336
Экспорт без установки.....	336
Установка проектов в систему .....	339
Установка логических целей .....	340
Использование конечного каталога по умолчанию для разных	
платформ .....	342
Публичные заголовки .....	343
Низкоуровневая установка.....	345
Установка командами install(FILES) и install(PROGRAMS).....	345
Работа с целыми каталогами .....	348
Вызов скриптов в процессе установки .....	351
Установка зависимостей времени выполнения.....	353
Создание повторно используемых пакетов.....	354
Проблемы перемещаемости целей .....	354
Установка файлов экспорта целей .....	356
Написание простых конфигурационных файлов .....	357
Создание продвинутых конфигурационных файлов .....	360
Генерирование файлов версий пакетов .....	363
Определение компонентов .....	365
Как использовать компоненты в find_package().....	365
Как используются компоненты в команде install().....	365

---

Управление символическими ссылками для версионированных разделяемых библиотек.....	367
Упаковка с помощью CPack .....	368
Резюме .....	370
Для дополнительного чтения.....	371
<b>Глава 15. Создание профессионального проекта .....</b>	<b>372</b>
Технические требования.....	373
Планирование работы .....	373
Структура проекта .....	377
Разделяемые и статические библиотеки .....	378
Файлы проекта .....	379
Сборка и управление зависимостями.....	380
Сборка библиотеки Calc .....	382
Сборка исполняемого файла консольного калькулятора Calc .....	384
Тестирование и анализ программы .....	388
Подготовка модуля Coverage .....	390
Подготовка модуля Memcheck .....	391
Тестовые сценарии .....	392
Добавление инструментов статического анализа .....	394
Установка и упаковка .....	396
Установка библиотеки.....	397
Установка исполняемого файла .....	398
Упаковка с помощью CPack.....	399
Документирование .....	399
Генерирование технической документации .....	400
Написание нетехнических документов для профессионального проекта .....	402
Резюме .....	404
Для дополнительного чтения.....	405
<b>Глава 16. Создание готовых конфигураций CMake .....</b>	<b>406</b>
Технические требования.....	407
Использование готовых конфигураций, определенных в проекте.....	407
Написание файла готовой конфигурации.....	408
Определение готовых конфигураций, зависящих от этапа .....	409
Общие характеристики всех готовых конфигураций .....	410
Уникальные имена полей.....	410
Факультативные поля .....	410
Ассоциация с готовыми конфигурациями этапа конфигурирования .....	411
Определение готовых конфигураций этапа конфигурирования.....	411
Определение готовых конфигураций этапа сборки.....	413
Определение готовых конфигураций этапа тестирования.....	414

---

Определение готовых конфигураций этапа упаковки .....	417
Добавление готовой конфигурации установки .....	418
Определение готовых конфигураций технологического процесса .....	419
Добавление условий и макросов .....	421
Резюме .....	423
Для дополнительного чтения .....	423
<b>Приложение .....</b>	<b>424</b>
Команда string() .....	424
Поиск и замена .....	425
Модификация .....	425
Сравнение .....	426
Хеширование .....	426
Генерирование .....	427
JSON .....	427
Команда list() .....	428
Чтение .....	428
Поиск .....	428
Модификация .....	429
Упорядочение .....	429
Команда file() .....	430
Чтение .....	430
Запись .....	430
Файловая система .....	431
Преобразование путей .....	431
Передача .....	432
Блокировка .....	432
Архивация .....	432
Команда math() .....	433
<b>Предметный указатель .....</b>	<b>434</b>

# От издательства

## *Отзывы и пожелания*

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте [www.dmkpress.com](http://www.dmkpress.com), зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com); при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу [http://dmkpress.com/authors/publish\\_book/](http://dmkpress.com/authors/publish_book/) или напишите в издательство по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

## *Список опечаток*

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com). Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

## *Нарушение авторских прав*

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

# Вступительное слово

В постоянно развивающемся и видоизменяющемся ландшафте C++ уверенное владение программой CMake является неоценимым преимуществом для любого разработчика, стремящегося писать эффективный, удобный для сопровождения и масштабируемый код. Книга «CMake для C++» Рафала Свидзиньски станет маяком, который проведет как начинающих, так и опытных программистов сквозь бушующие волны CMake.

Эта книга – не просто руководство, это путешествие. Начинается она с основ, так чтобы даже человек, только приступающий к изучению программы CMake, мог разобраться в ее понятиях. Постепенно читатель знакомится с более продвинутыми методами и в конце концов осваивает весь потенциал CMake.

Характерной особенностью этой книги является прагматический подход. В текст вплетены реальные примеры и рекомендованные практики, благодаря чему читатель не только усваивает концепции, но и учится эффективно применять их в своих проектах.

Прочитав всю книгу, читатель не только будет глубоко понимать CMake, но и обретет новую уверенность в своих способностях справляться с многочисленными сложностями разработки на C++. Он будет вооружен знаниями и навыками, которые позволят писать более чистый и эффективный код, встав тем самым на путь, ведущий к рождению профессионала.

«CMake для C++» – не просто книга, это инструмент, который поднимает навыки разработки на C++ на новую высоту. Равно начинающему и эксперту эта книга поможет раскрыть полный потенциал CMake и сделать код более надежным, пригодным для сопровождения и масштабируемым.

Александр Кушнир,  
главный программист, компания Biosense Webster

# Об авторе

Рафал Свидзиньски работает инженером в Google, имеет за плечами 12-летний опыт разработки полного цикла. Возглавлял проекты для таких промышленных гигантов, как Cisco Meraki, Amazon и Ericsson, всей душой предан инновациям. Лондонец по духу, он всегда на переднем крае технического прогресса и занимается кучей личных проектов. Его недавнее увлечение применением ИИ в здравоохранении отражает стремление к прогрессу, несущему пользу людям. Рафал высоко ценит безупречное качество и мастерство и делится знаниями на канале YouTube и в своих книгах.

*Зое – без тебя эта книга не была бы написана.*

# О рецензентах

**Эрик Нуар** получил степень магистра технических наук в Высшей государственной школе электротехники, электроники, информатики, гидравлики и телекоммуникаций и степень PhD по информатике в университете Версаль-Сен-Кантен-ан-Ивельин, Франция. За плечами Эрика 25 лет написания и компилирования исходного кода на разных языках. Является пользователем CMake с 2006 года, принимал активное участие в ее развитии. В настоящее время работает в компании Antidot, специализирующейся на разработке программного обеспечения для семантического поиска, ИИ и обеспечения доступности контента. Эрик входит в состав исследовательской группы, которая отвечает за включение новых технологий, в т. ч. ИИ и обработку естественного языка, во флагманский продукт компании, Fluid Topics.

В активе **Джованни Романо** 28 лет работы в ИТ – от разработки программного обеспечения до проектирования приложений и компонентов. В настоящее время работает старшим программистом в компании Leica Geosystem AG, где занимается проектированием SDK, микросервисов и серверных компонентов с низкой задержкой. Будучи амбассадором Nokia/Blackberry Qt, он верит в программное обеспечение с открытым исходным кодом и участвует в развитии этого каркаса. В сферу его интересов входят облачные приложения, Kubernetes, Docker и GitOps. Обожает писать на языке C и играть в теннис.

# Предисловие

Создание высококачественного программного обеспечения – не простое дело. Разработчики, изучающие этот вопрос по материалам в сети, часто не могут понять, какая рекомендация самая актуальная, а каким методам пришли на смену более новые и совершенные. К тому же объяснения обычно хаотические, им недостает базы, контекста и структуры.

В книге «CMake для C++» предлагается простой и исчерпывающий подход к построению решений на C++. Вы не только научитесь использовать CMake в своих проектах, но и узнаете, как сделать их пригодными для сопровождения, элегантными и чистыми. Рассматривается автоматизация решения сложных задач, в т. ч. сборки, тестирования и упаковки.

Вы узнаете об организации каталогов с исходным кодом, о целях сборки и о создании пакетов. Вы научитесь компилировать и компоновать исполняемые файлы и библиотеки, поймете все детали этих процессов и будете знать, как оптимизировать каждый шаг. Кроме того, будет показано, как включать в проект внешние зависимости, например сторонние библиотеки, каркасы тестирования, инструменты анализа кода и генераторы документации. Наконец, вы научитесь экспортить и устанавливать свое решение для внутреннего и внешнего использования.

Прочитав эту книгу, вы сможете уверенно использовать CMake на профессиональном уровне.

## Для кого предназначена эта книга

Изучив C++, вы довольно быстро поймете, что одного лишь владения языком недостаточно для организации поставки программного продукта в соответствии с высокими стандартами качества. Книга, которую вы держите в руках, восполняет этот пробел: она адресована всем, кто стремится стать лучшим программистом или даже профессиональным инженером по сборке!

Читайте, если хотите изучить современную CMake с азов или подтянуть отставшие от времени навыки. Книга поможет вам создавать проекты на C++ высочайшего качества и отказаться от других сред сборки.

## Краткое содержание книги

В главе 1 «Первые шаги» рассматривается установка CMake, работа с ее командным интерфейсом и основные строительные блоки проекта на основе CMake.

Глава 2 «Язык CMake» посвящена основным понятиям языка CMake, включая вызовы команд, аргументы, переменные, управляющие структуры и комментарии.

В главе 3 «Использование CMake в популярных IDE» подчеркивается важность **интегрированных сред разработки (IDE)**, даются рекомендации по выбору IDE и приводятся инструкции для Clion, Visual Studio Code и Visual Studio IDE.

В главе 4 «Подготовка первого проекта на основе CMake» вы узнаете, как сконфигурировать базовый проект CMake в файле верхнего уровня, как организовать дерево файлов и как подготовить комплект инструментов разработки.

В главе 5 «Цели» рассматривается понятие логических целей сборки, их свойства и типы, а также определение своих команд для проектов CMake.

В главе 6 «Генераторные выражения» объясняется назначение и синтаксис генераторных выражений, в т. ч. их использование в условных расширениях, запросах и преобразованиях.

Глава 7 «Компиляция исходного кода на C++ с помощью CMake» посвящена процессу компиляции, конфигурированию препроцессора и оптимизатора, а также способам уменьшения времени сборки и упрощения отладки.

В главе 8 «Компоновка исполняемых файлов и библиотек» вы узнаете о механизме компоновки, о различных типах библиотек, о правиле одного определения, о порядке компоновки и о том, как подготовить проект к тестированию.

В главе 9 «Управление зависимостями в CMake» рассматривается работа со сторонними библиотеками, добавление поддержки CMake в те из них, где она не была изначально предусмотрена, а также скачивание внешних зависимостей из интернета.

В главе 10 «Использование модулей C++20» описываются модули из стандарта C++20, их поддержка в CMake и задание соответствующей конфигурации комплекта инструментов.

Глава 11 «Каркасы тестирования» поможет вам понять важность автоматизированного тестирования, освоить встроенную в CMake поддержку тестирования и приступить к автономному тестированию с применением популярных каркасов.

В главе 12 «Инструменты анализа программ» показано, как автоматически форматировать исходный код и обнаруживать программные ошибки на этапах сборки и выполнения.

Глава 13 «Генерирование документации» посвящена использованию Doxygen для автоматизации создания документации по исходному коду и ее стилизации.

В главе 14 «Установка и упаковка» вы научитесь подготавливать проект к выпуску с установкой и без нее, создавать пакеты, допускающие повторное использование, и определять отдельные компоненты, включаемые в пакет.

В главе 15 «Создание профессионального проекта» вы примените все полученные ранее знания к разработке полного проекта профессионального уровня.

Конец ознакомительного фрагмента.  
Приобрести книгу можно  
в интернет-магазине  
«Электронный универс»  
[e-Univers.ru](http://e-Univers.ru)