

УДК 004.42
ББК 32.372
Д40

Клинтон Л. Джеффери

Д40 Создай свой собственный язык программирования. Руководство программиста по разработке компиляторов, интерпретаторов и доменно-ориентированных языков для решения современных вычислительных задач / пер. с англ. С. В. Минца. – М.: ДМК Пресс, 2023. – 408 с.: ил.

ISBN 978-5-93700-140-5

Книга рассказывает о том, как разрабатывать уникальные языки программирования, чтобы сократить время и стоимость создания приложений для новых или специализированных областей применения вычислительной техники. Вы начнете с реализации интерфейса компилятора для вашего языка, включая лексический и синтаксический анализатор, а к концу чтения сможете разрабатывать и воплощать в коде свои собственные языки, позволяющие компилировать и запускать программы.

Издание адресовано разработчикам программного обеспечения, заинтересованным в создании собственного языка. Для изучения материала потребуются опыт программирования на языке высокого уровня, таком как Java или C++.

УДК 004.42
ББК 32.372

First published in the English language under the title 'Build Your Own Programming Language' – (9781800204805)

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN (англ.) 978-1-80020-480-5
ISBN (рус.) 978-5-93700-140-5

Copyright ©Packt Publishing 2021
© Оформление, издание, перевод, ДМК Пресс, 2022

Оглавление

Об авторах	16
О рецензентах	16
Предисловие.....	17
Для кого эта книга	17
Что скрывает обложка	17
Как получить от этой книги максимальную пользу.....	20
Загрузка примеров	20
Видео	20
Цветные иллюстрации	20
Используемые сокращения.....	20
Список опечаток.....	21
Нарушение авторских прав	21
ЧАСТЬ I. ИНТЕРФЕЙСЫ ЯЗЫКА	
ПРОГРАММИРОВАНИЯ.....	23
Глава 1. Зачем создавать еще один	
язык программирования?	25
Итак, вы хотите создать свой собственный язык программирования.....	25
Типы реализации языков программирования.....	26
Организация реализации языка байт-кода.....	27
Языки, используемые в примерах.....	28
Язык и библиотека – в чем разница?	30
Применимость к другим задачам разработки программного	
обеспечения	30
Определение требований к вашему языку	31
Тематическое исследование – требования, которые вдохновили	
на создание языка Unicon	33
Требование Unicon № 1 – сохранять то, что люди любят в Icon.....	34
Требование Unicon № 2 – поддержка крупномасштабных программ,	
работающих с большими данными.....	34
Требование Unicon № 3 – высокоуровневый ввод/вывод	
для современных приложений	35
Требование Unicon № 4 – обеспечить универсально реализуемые	
системные интерфейсы	35
Заключение	36
Вопросы.....	36

Глава 2. Дизайн языка программирования	37
Определение видов слов и пунктуации в вашем языке	38
Определение потока управления	40
Решение о том, какие типы данных поддерживать	41
Атомарные типы.....	41
Составные типы	42
Типы, специфичные для конкретной области	44
Общая структура программы	44
Завершение определения языка Jzero	45
Тематическое исследование – проектирование графических объектов в Unicon	46
Поддержка языка для графики 2D.....	47
Добавление поддержки трехмерной графики.....	49
Заключение	50
Вопросы.....	50
Глава 3. Сканирование исходного кода	52
Технические требования	52
Лексемы, лексические категории и токены.....	53
Регулярные выражения	54
Правила регулярных выражений	54
Примеры регулярных выражений.....	56
Использование UFlex и JFlex.....	57
Раздел заголовка.....	58
Раздел регулярных выражений	58
Написание простого сканера исходного кода	59
Запуск сканера	62
Токены и лексические атрибуты	63
Расширение нашего примера для построения токенов	64
Написание сканера для Jzero	66
Спецификация Jzero flex	66
Код Unicon Jzero	69
Код Java Jzero.....	72
Запуск сканера Jzero.....	75
Регулярных выражений не всегда достаточно	76
Заключение	80
Вопросы.....	80
Глава 4. Парсинг.....	81
Технические требования	81
Анализ синтаксиса	82
Понимание бесконтекстных грамматик.....	83
Написание правил бесконтекстной грамматики	84
Написание правил для программных конструкций	85
Использование йуасс и ВУАСС/J.....	87
Объявление символов в разделе заголовка	88

Составление раздела бесконтекстной грамматики yacc.....	89
Понимание парсеров yacc	90
Устранение конфликтов в парсерах yacc	92
Исправление синтаксических ошибок.....	93
Создание игрушечного примера	93
Написание парсера для Jzero	98
Спецификация Jzero lex	98
Спецификация yacc в Jzero	98
Код Unicon Jzero	103
Код парсера Jzero на языке Java	105
Запуск парсера Jzero	105
Улучшение сообщений об ошибках синтаксиса.....	107
Добавление деталей в сообщения Unicon об ошибках синтаксиса	108
Добавление деталей в сообщения Java об ошибках синтаксиса	108
Использование Merge для создания лучших сообщений об ошибках синтаксиса	109
Заключение	110
Вопросы.....	110
Глава 5. Деревья синтаксиса	111
Технические требования.....	111
Использование GNU make	112
Изучение деревьев	115
Определение типа дерева синтаксиса	115
Деревья разбора в сравнении с деревьями синтаксиса	117
Создание листьев из терминальных символов	119
Обертывание токенов в листья.....	120
Работа со стеком значений YACC	120
Обертка листьев для стека значений парсера	122
Определение нужных вам листьев	123
Построение внутренних узлов из правил производства.....	124
Доступ к узлам дерева в стеке значений	124
Использование фабричного метода узла дерева	126
Формирование деревьев синтаксиса для языка Jzero.....	127
Отладка и тестирование вашего дерева синтаксиса	134
Предотвращение распространенных ошибок в дереве синтаксиса	134
Распечатка вашего дерева в текстовом формате	136
Печать дерева с помощью dot.....	138
Заключение	143
Вопросы.....	143
ЧАСТЬ II. ОБХОДЫ ДЕРЕВА СИНТАКСИСА	145
Глава 6. Таблицы символов	147
Технические требования.....	148
Создание основы для таблиц символов	148
Объявления и области видимости.....	148

Присваивание и разыменование переменных	149
Выбор подходящего обхода дерева для работы	150
Создание и заполнение таблиц символов для каждой области видимости	151
Добавление семантических атрибутов к деревьям синтаксиса	152
Определение классов для таблиц символов и записей в таблицах символов	154
Создание таблиц символов	155
Заполнение таблиц символов	157
Синтез атрибута isConst	159
Проверка наличия необъявленных переменных	160
Идентификация тел методов	160
Выявление использования переменных в теле метода	161
Поиск повторно объявленных переменных	162
Вставка символов в таблицу символов	163
Сообщение о семантических ошибках	163
Обработка пакетов и областей видимости классов в Unicon	164
Искажение имен	165
Вставка self для ссылок на переменные-члены	166
Вставка self в качестве первого параметра в вызовы методов	166
Тестирование и отладка таблиц символов	167
Заключение	169
Вопросы	170
Глава 7. Проверка базовых типов	171
Технические требования	171
Представление типов в компиляторе	171
Определение базового класса для представления типов	172
Подклассификация базового класса для сложных типов	173
Присвоение информации о типе объявленным переменным	175
Синтез типов из зарезервированных слов	177
Наследование типов в списке переменных	178
Определение типа в каждом узле дерева синтаксиса	179
Определение типа в листьях	180
Вычисление и проверка типов во внутренних узлах	182
Проверка типов во время выполнения и вывод типов в Unicon	186
Заключение	188
Вопросы	188
Глава 8. Проверка типов в массивах, вызовах методов и доступах к структурам	189
Технические требования	189
Операции проверки типов массивов	189
Управление объявлениями переменных в массивах	190
Проверка типов при создании массива	191
Проверка типов при обращении к массиву	193
Проверка вызовов методов	194

Вычисление параметров и информации о возвращаемом типе	194
Проверка типов в каждом месте вызова метода	197
Проверка типов в операторах возврата	200
Проверка обращений к структурированным типам	202
Обработка объявлений переменных экземпляра	202
Проверка типов при создании экземпляра	203
Проверка типов при обращении к экземпляру	205
Заключение	208
Вопросы	209
Глава 9. Генерация промежуточного кода	210
Технические требования	210
Подготовка к генерации кода	210
Зачем генерировать промежуточный код?	211
Изучение областей памяти в созданной программе	211
Представление типов данных для промежуточного кода	212
Добавление атрибутов промежуточного кода в дерево	214
Генерация меток и временных переменных	215
Набор инструкций промежуточного кода	218
Инструкции	218
Декларации	219
Аннотирование деревьев синтаксиса метками для потока управления	219
Генерация кода для выражений	222
Генерация кода для потока управления	225
Генерация целевых меток для выражений условий	225
Генерация кода для циклов	228
Генерация промежуточного кода для вызовов методов	229
Проверка сгенерированного промежуточного кода	231
Заключение	232
Глава 10. Раскраска синтаксиса в IDE	233
Загрузка примеров IDE, используемых в этой главе	234
Интеграция компилятора в редактор программиста	236
Анализ исходного кода из среды IDE	236
Отправка выходных данных компилятора в IDE	237
Предотвращение повторного разбора всего файла при каждом изменении	238
Использование лексической информации для раскрашивания токенов	242
Расширение компонента EditableTextList для поддержки цвета	242
Раскрашивание отдельных токенов по мере их создания	242
Подсветка ошибок с использованием результатов разбора	243
Добавление поддержки Java	245
Заключение	247

ЧАСТЬ III. ГЕНЕРАЦИЯ КОДА И СРЕДЫ ВЫПОЛНЕНИЯ 249

Глава 11. Интерпретаторы байт-кода..... 251

Технические требования	251
Понимание, что такое байт-код.....	252
Сравнение байт-кода с промежуточным кодом.....	253
Построение набора инструкций байт-кода для Jzero.....	255
Определение формата файла байт-кода Jzero.....	255
Понимание основ работы стековой машины	258
Реализация интерпретатора байт-кода	259
Загрузка байт-кода в память	259
Инициализация состояния интерпретатора	261
Выборка инструкций и продвижение указателя инструкции	263
Декодирование инструкций	264
Выполнение инструкций	265
Запуск интерпретатора Jzero	268
Написание среды выполнения для Jzero.....	269
Запуск программы Jzero.....	269
Изучение iconx, интерпретатора байт-кода Unicon	270
Понимание целенаправленного байт-кода	271
Сохранение информации о типе во время выполнения	271
Выборка, декодирование и выполнение инструкций.....	272
Создание остальной части среды выполнения	272
Заключение	273
Вопросы.....	273

Глава 12. Генерация байт-кода 274

Технические требования	274
Преобразование промежуточного кода в байт-код Jzero	275
Добавление класса для инструкций байт-кода	276
Соответствие адресов промежуточного кода адресам байт-кода.....	276
Реализация метода генератора байт-кода.....	278
Генерация байт-кода для простых выражений	278
Генерация кода для обработки указателей.....	280
Генерация байт-кода для безусловных и условных переходов	281
Генерация кода для вызовов методов и возвратов	282
Обработка меток и других псевдоинструкций промежуточного кода.....	284
Сравнение ассемблера байт-кода с двоичными форматами	285
Вывод байт-кода в формате ассемблера	285
Вывод байт-кода в двоичном формате	287
Линковка, загрузка и включение среды выполнения	288
Пример Unicon – генерация байт-кода в icont	288
Заключение	290
Вопросы.....	290

Глава 13. Генерация собственного кода.....292

Технические требования.....	292
<i>Принятие решения о генерации собственного кода.....</i>	<i>292</i>
Знакомство с набором инструкций x64	293
Добавление класса для инструкций x64.....	294
Соответствие областей памяти регистровым режимам адресации x64.....	294
Использование регистров	295
Начинаем с нулевой стратегии.....	296
Преобразование промежуточного кода в код x64	299
Соответствие адресов промежуточного кода местоположению в x64	300
Реализация метода генератора кода x64	303
Генерация кода x64 для простых выражений.....	304
Генерация кода для обработки указателей.....	305
Генерация собственного кода для безусловных и условных переходов	306
Генерация кода для вызовов методов и возвратов	307
Обработка меток и псевдоинструкций	309
Генерация выходных данных x64.....	311
Запись кода x64 в формате ассемблера.....	311
Переход от ассемблера к объектному файлу	312
Линковка, загрузка и включение среды выполнения.....	313
Заключение	314
Вопросы.....	315

Глава 14. Реализация операторов

и встроенных функций316

Реализация операторов.....	316
Подразумевают ли операторы аппаратную поддержку, и наоборот	317
Добавление конкатенации строк в генерацию промежуточного кода.....	318
Добавление конкатенации строк в интерпретатор байт-кода.....	319
Добавление конкатенации строк в собственную среду выполнения	322
Написание встроенных функций	323
Добавление встроенных функций в интерпретатор байт-кода	323
Написание встроенных функций для использования в реализации собственного кода.....	324
Интеграция встроенных функций со структурами управления	325
Разработка операторов и функций для Unicorn	326
Написание операторов в Unicorn.....	327
Разработка встроенных функций Unicorn	329
Заключение	330
Вопросы.....	330

Глава 15. Структуры управления доменами 331

Понимание необходимости новой структуры управления	331
Определение структуры управления	332
Устранение избыточных параметров	333
Сканирование строк в Icon и Unicon	333
Среды сканирования и их примитивные операции	334
Устранение избыточных параметров с помощью структуры управления	336
Рендеринг областей в Unicon	337
Отображение 3D-графики из списка отображения	337
Указание областей рендеринга с помощью встроенных функций	338
Изменение графических уровней детализации с помощью вложенного рендеринга областей	339
Создание структуры управления рендерингом областей	340
Добавление зарезервированного слова для рендеринга областей	340
Добавление правила грамматики	341
Проверка wsection на семантические ошибки	342
Генерация кода для структуры управления wsection	343
Заключение	345
Вопросы	345

Глава 16. Сборка мусора 347

Оценка важности сборки мусора	347
Подсчет ссылок на объекты	349
Добавление подсчета ссылок в Jzero	350
Генерация кода для распределения кучи	350
Изменение сгенерированного кода для оператора присваивания	352
Учет недостатков и ограничений, связанных с подсчетом ссылок	353
Пометка реальных данных и очистка остальных	354
Организация областей памяти кучи	355
Обход базиса для пометки живых данных	357
Восстановление живой памяти и размещение ее в непрерывных фрагментах	361
Заключение	363
Вопросы	364

Глава 17. Заключительные размышления 365

Размышления о том, что изучено при написании этой книги	365
Решение о том, куда двигаться дальше	366
Изучение дизайна языков программирования	366
Изучение реализации интерпретаторов и машин байт-кода	367
Приобретение опыта в оптимизации кода	368
Мониторинг и отладка выполнения программ	369
Проектирование и реализация IDE и строителей GUI	369
Изучение ссылок для дальнейшего чтения	370

Изучение дизайна языков программирования.....	370
Изучение реализации интерпретаторов и машин байт-кода.....	371
Приобретение опыта работы с собственным кодом и оптимизации кода.....	371
Мониторинг и отладка выполнения программ.....	372
Проектирование и реализация IDE и строителей GUI	372
Заключение	373
ЧАСТЬ IV. ПРИЛОЖЕНИЕ.....	375
Приложение. Основы Unicon.....	377
Запуск Unicon.....	377
Использование объявлений и типов данных Unicon	379
Объявление различных типов компонентов программы	379
Использование атомарных типов данных.....	381
Организация нескольких значений с помощью структурных типов	382
Оценка выражений.....	384
Формирование базовых выражений с помощью операторов.....	384
Вызов процедур, функций и методов	387
Итерации и выбор того, что и как выполнять	388
Генераторы.....	389
Отладка и вопросы окружения	390
Изучение основ отладчика UDB	390
Переменные окружения.....	391
Препроцессор.....	391
Мини-справочник функций	393
Избранные ключевые слова.....	398
Оценки	400
Глава 1.....	400
Глава 2.....	400
Глава 3.....	401
Глава 4.....	401
Глава 5.....	402
Глава 6.....	402
Глава 7.....	403
Глава 8.....	403
Глава 11.....	404
Глава 12.....	404
Глава 13.....	405
Глава 14.....	405
Глава 16.....	407