Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Иркутский национальный исследовательский технический

университет»

Институт информационных технологий и анализа данных

О Т Ч Ё Т

|  |  |
| --- | --- |
| о прохождении | учебной практики |
|  | (вид практики: учебная/производственная) |
| технологической (проектно-технологической) практики | |
| (тип практики: технологическая/научно-исследовательская работа/преддипломная и др.) | |
| в | ИРНИТУ |
|  | (наименование профильной организации) |

Обучающегося Васильева Никиты Сергеевича

(ФИО, группа, подпись)

Руководитель практики от института ИТиАД Кононенко Р.В., доцент института ИТиАД

(ФИО, должность, подпись)

Руководитель образовательной программы

hh.ru: <https://hh.ru/resume/93e51b7dff0efd6fda0039ed1f506571426941>

Кононенко Р.В., доцент института ИТиАД

(ФИО, должность, подпись)

Оценка по практике

(ФИО, подпись, дата)

Содержание отчета на стр. Приложение к отчету на стр.

Superjob: <https://www.superjob.ru/resume/razrabotchik-c-55732969.html>

Иркутск 2025

# Индивидуальное задание на прохождение

**учебной практики: технологической (проектно-технологической) практики**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| для | Васильева Никиты Сергеевича | | |  |  |
| (ФИО обучающегося полностью) | | | | | |
| обучающегося | | 1 | курса | группы | Исиб-24-1 |

по направлению подготовки Информационные системы и технологии профиль Интеллектуальные системы обработки информации и управления

Место прохождения практики: ИРНИТУ

Сроки прохождения практики с «16» 06 2025 г. по «29» 06 2025 г.

Цели и задачи прохождения практики: Основной целью практики является изучение теории и получение практических навыков в области разработки компиляторов через создание полноценного прототипа для учебного языка программирования

Содержание практики, вопросы, подлежащие изучению:

1. Формирование требований к АС

1.1. Обследование объекта и обоснование необходимости создания АС

Объектом автоматизации процесс трансляции исходного кода, написанного на высокоуровневом языке программирования, в исполняемую форму. В текущий момент данный процесс для разрабатываемого языка Ccb отсутствует. Необходимость создания АС (компилятора) обусловлена учебными целями: получение практических навыков в области теории и практики построения компиляторов, применение знаний по структурам данных и алгоритмам для решения комплексной задачи, создания завершенного программного продукта (прототипа) в рамках учебной практики.

1.2 Формирование требований пользователя к автоматизированной системе (АС)

Пользователь (разработчик-студент, инженер-новичек) формулирует следующие требования к создаваемой системе:

Система должна принимать на вход текстовый файл с исходным кодом на языке Ccb.

Система должна выполнять лексический, синтаксический и семантический анализ исходного кода.

В случае обнаружения ошибок на этапе компиляции, система должна выводить информативное сообщение об ошибке.

Система должна генерировать промежуточное представление (байт-код) на основе корректного исходного кода.

Работа с системой должна осуществлятся через интерфейс командной строки (CLI), без графического интерфейса.

1.3 Оформление отчёта о выполненной работе и заявки на разработку АС (тактико-техническое задание)

Название проекта: Компилятор для учебного языка программирования CCb.  
Разработчик: Васильев Никита Сергеевич  
Цель разработки: Изучение фундаментальных принципов работы компиляторов, включая этапы лексического анализа, синтаксического анализа (парсинга), генерации кода и его исполнения.  
Обоснование: Наиболее наглядный способ освоения теории компиляции — реализация собственного компилятора. Данный проект позволит получить практический опят, который невозможно приобрести только теоретическим изучением.  
Краткое описание: Разработка прототипа компилятора для простого императивного языка Ccb. Компилятор будет преобразовывать исходный код в байт-код с последующим его исполнением на стековой виртуальной машине.

2. Разработка концепции АС

2.1. Изучение объекта

Изучены основные этапы работы компилятора: лексический анализ, синтаксический анализ, генерация промежуточного кода, исполнение. Проанализированы подходы к реализации каждого этапа.

2.2. Проведение необходимых научно-исследовательских работ

Проведен анализ существующих подходов к разработке компиляторов для учебных целей. Рассмотрены следующие архитектуры: прямая интерпретация через обход Абстрактного Синтаксического Дерева (АСД), компиляция в байт-код с последующим исполнением на виртуальной машине (ВМ), компиляция в нативный код целевой платформы.

2.3. Разработка вариантов концепции АС.

Вариант 1 (Выбранный): Реализация полного цикла: Лексер → Парсер (АСД) → Генератор байт-кода → Виртуальная машина. Этот вариант полностью охватывает все ключевые этапы и обеспечивает четкое разделение модулей.

Вариант 2 (Отклоненный): «Tree-walking» интерпретатор, исполняющий кодй напрямую с АСД. Отклонен, так как не включает этапы генерации байт-кода и разработки ВМ, что является одной из целей проекта.

2.4. Оформление отчёта о выполненной работе

Текущее состояние: Анализ требований завершён. Выбрана концепция компилятора с генерацией байт-кода и его исполнением на стековой виртуальной машине. Для реализации будет использоваться язык C и его стандартная библиотека, без привлечения сторонних зависимостей и генераторов парсеров.

3. Техническое задание

Целью создания программного средства является разработка прототипа компилятора для учебного языка Ccb. Компилятор должен демонстрировать ключевые стадии преобразования исходного кода в исполняемую форму. Основным функциональным назначением является обеспечение трансляции и исполнения программ, написанных на Ccb. Язык поддерживает следующие конструкции: типы данных (только целые числа), объявление через ключевое слово let и присваивание, арифметические операции, условные конструкции, циклы, ввод/вывод.

4. Эскизный проект

4.1. Разработка предварительных проектных решений по системе и её частям

Лексический анализатор (Лексер): Будет реализован на основе конечного автомата, который преобразует последовательность символов в поток токенов. Синтаксический анализатор (Парсер): Будет реализован метод рекурсивного спуска для построения Абстрактного Синтаксического Дерева (АСД) из потока токенов. Генератор кода: Будет реализован в виде функции, рекурсивно обходящей АСД и генерирующей на его основе линейную последовательность инструкций (байт-код). Виртуальная машина (ВМ): Будет реализована как стековая машина с циклом fetch-decode-execute для поочередного выполнения инструкций байт-кода.

4.2 Разработка документации на АС и её части

Будет подготовлен следующий комплект документации:

README.md — руководство пользователя: описание проекта, спецификации языка Ccb, инструкции по сборке и запуску, примеры программ.

Исходный код с комментариями, описывающими логику работы ключевых модулей.

5.1 Разработка проектных решений по системе и её частям

В рамках проектирования системы выполнено следующее:

После прочтения файла функцией ReadFile() (модуль main) в работу вступает модуль lexer. Он создает лексер вызовом NewLexer(const char\* source), затем многократно вызывает NextToken(), формируя динамический массив структур Token до тех пор, пока не вернётся маркер TOKEN\_EOF.

Полученный поток токенов поступает в модуль parser. Парсер инициализируется через NewParser(Lexer\* l); непосредственный разбор выполняет функция ParseProgram(): она строит полное абстрактное синтаксическое дерево (root-узел Program) с использованием вспомогательных процедур ParseStatement(), ParseExpression() и др.

Дерево передаётся в модуль codegen. Центральная функция Compile(Program\* ast) обходит AST и формирует объект Chunk, содержащий вектор байт-кода (code) и таблицу констант (constant). На этом этапе конкретные операции превращаются в опкоды — например OP\_ADD, OP\_EQUAL, OP\_JUMP\_IF\_FALSE и т. д.

Получившийся Chunk исполняет vm. Точка входа Interpret(const char\* source) инициализирует стек, загружает Chunk и, двигая указатель ip, вызывает внутреннюю функцию Run(). Та реализует байт-кодовую машину: операции со стеком (Push, Pop), доступ к глобальным переменным, арифметику, сравнения, ввод-вывод (OP\_IN, OP\_OUT) и управление потоком (OP\_LOOP, OP\_JUMP, OP\_JUMP\_IF\_FALSE).

Модуль main связывает все воедино: читает файл (ReadFile), печатает заголовок строки «Compilling and running ...», передаёт исходник в Interpret, по коду возврата (INTERPRET\_OK, INTERPRET\_COMPILE\_ERROR, INTERPRET\_RUNTIME\_ERROR) устанавливает конечный exit()-статус (0, 65 или 70)

Планируемые результаты практики: Планируемый результатом практики является полностью рабочий прототип компилятора для учебного языка Ccb. Он будет представлять собой консольное приложение, способное принимать на вход файл с исходным кодом, компилировать его в байт-код и исполнять на втроенной виртуальной машине. В результате будут получены практические навыки в области теории компиляции, проектирования архитектуры программных систем и применения алгоритмов на практике.

Руководитель практики от института ИТиАД

/ Кононенко Р.В. /

(подпись

# Согласовано:

Руководитель ООП

/Кононенко Р.В./

(подпись

« » 2025 г.

С настоящим индивидуальным заданием и с программой практики ознакомлен(а), задание принято к исполнению

«16» 06 2025 г.

(подпись)

**ДНЕВНИК**

прохождения практики

|  |  |  |  |
| --- | --- | --- | --- |
| обучающегося | | | Васильева Никиты Сергеевича, ИСИб-24-1 |
|  |  |  | (фамилия, имя, отчество, группа) |
| курс | 1 | | |
| направление | | | Информатика и вычислительная техника |
| профиль | | Интеллектуальные системы обработки | |
| информации и управления | | | |
| в |  |  | ИРНИТУ |
|  |  |  | (наименование профильной организации) |

Иркутск 2025

Руководителем практики от структурного подразделения назначен:

Кононенко Роман Владимирович, доцент института ИТиАД

(ФИО, должность)

# Рабочий график (план) прохождения практической подготовки

(заполняется обучающимся)

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Период практики | Содержание выполненных работ | Подпись руководителя практики от  структурного подразделения |
| 1 | 16.06.2025 | Формализация требований к языку Ccb (синтаксис, ключевые слова, операторы). Изучение теоретических основ: этапы компиляции, лексический и синтаксический анализ. |  |
| 2 | 17.06.2025 | Согласовывание плана проекта с руководителем практики. Создание репозитория проекта (Git). Разработка архитектуры проекта: определение модулей (lexer, parser, codegen, vm) и их взаимодействия. Создание резюме на hh.ru и superjob. |  |
| 3 | 18.06.2025 | Настройка сборочной системы (Makefile/Cmake). Реализация общих структур данных: токен (Token), узел АСД (AstNode), структура для байт-кода (BytecodeChunk). |  |
| 4 | 19.06-20.06.2025 | Реализация функции для сканирования исходного кода. Распознавание всех типов токенов: ключевые слова (let, if, while), операторы, числа, идентификаторы. |  |
| 5 | 21.06.2025 | Реализация парсера на основе рекурсивного спуска. Построение Абстрактного синтаксического дерева (АСД) для арифметических выражений с учетом приоритета операторов. |  |
| 6 | 23.06.2025 | Добавление в парсер обработки управляющих конструкций: присваивание переменных, условные операторы и циклы. |  |
| 7 | 24.06.2025 | Проектирование набора инструкций (опкодов для виртуальной машины. Реализация обхода АСД для генерации байт-кода для простых выражений и работы с константами. [экскурсия] |  |
| 8 | 25.06.2025 | Реализация генерации кода для работы с переменными. Генерация кода для условных переходов и циклов. |  |
| 9 | 26.05.-27.06.2025 | [экскурсия] Реализация основного цикла ВМ. Реализация стека операндов и логики выполнения всех разработанных опкодов (арифметика, переходы, ввод/вывод). |  |
| 10 | 28.06.2025 | Объединение всех модулей в единую программу. Написание тестовых программ на Ccb для проверки всего функционала. Отладка и исправление ошибок. Написание документации с описанием языка и инструкциями по запуску. Финальная чистка кода, добавление комментариев. Подготовка отчета о проделанной работе. |  |

|  |  |
| --- | --- |
| Дата фактического прибытия |  |
| обучающегося в структурное подразделение | 16.06.2025 |
| Дата фактического убытия |  |
| обучающегося из структурного подразделения | 29.06.2025 |

|  |  |
| --- | --- |
| Руководитель образовательной программы | Кононенко Р.В. |
|  | (ФИО, подпись) |
| Директор института | Говорков А.С. |
|  | (ФИО, подпись) |

# Содержание

**Введение 12**

[**Ссылки на материалы проекта**](#_bookmark0) **13**

[**Список использованных материалов 1**](#_bookmark2)**4**

**Введение**

Целью учебной практики является изучение фундаментальных принципов построения трансляторов и получения практических навыков в разработке компилятора для собственного учебного языка программирования. В процессе практики будет приобретён опыт в проектировании архитектуры компилятора, реализации его ключевых этапов: лексического анализа, синтаксического анализа с построением абстрактного синтаксического дерева (АСД), генерации промежуточного байт-кода, а также в разработке стековой виртуальной машины для его исполнения. Основное внимание будет уделено последовательной реализации ковейера компиляции начиная с преобразования исходного кода в токены (лексер) и построения синтаксического дерева (парсер), и заканчивая генерацией исполняемого байт-кода и созданием виртуальной машины, способной его выполнить.