

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой ИУ6

А.В. Пролетарский
« » 2026 г.

КОМПИЛЯТОР СТЕКОВОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ

Техническое задание

Листов 6

Студент

ИУ6-83Б
(Группа)

(Подпись, дата)

В.К. Залыгин
(И.О. Фамилия)

Руководитель

(Подпись, дата)

Б.И. Бычков
(И.О. Фамилия)

2026 г.

1 ВВЕДЕНИЕ

Настоящее техническое задание распространяется на разработку программы «Компилятор стекового языка программирования» [КСЯП], используемой для трансляции в байткод Wasm и верификации программ и предназначенной для создания исполняемых файлов по текстам программ, написанных программистом на конкатенативном стековом языке программирования Cat.

Актуальность разработки обусловлена новизной выбора цели трансляции. С точки зрения языка, прямая трансляция в WebAssembly позволит избежать дополнительных накладных расходов, связанных с тяжеловесной машиной .NET CLR, увеличить скорость работы и снизить потребление памяти Cat-программ. С точки зрения виртуальной машины, текстовый формат программ которой основан на использовании S-выражений, WebAssembly получит инструмент описания программ при помощи конкатенативного языка, что является более распространенным для стековых языков.

2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Компилятор для стекового языка программирования разрабатывается в соответствии с тематикой кафедры «Компьютерные системы и сети».

3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Основное назначение компилятора заключается в создании исполняемых файлов путем трансляции и верификации написанных программистом на исходном языке Cat текстов в байткод, соответствующий спецификации виртуальной машины WebAssembly.

4 ИСХОДНЫЕ ДАННЫЕ, ЦЕЛИ И ЗАДАЧИ

4.1 Исходными данными для разработки являются следующие материалы:

4.1.1 Simple Type Inference for HigherOrder StackOriented Languages [Электронный ресурс]. URL: <https://dcreager.net/remarkable/Diggins2008a.pdf> (дата обращения: 13.10.2025).

4.1.2 Cat language repository [Электронный ресурс]. URL: <https://github.com/cdiggins/catlanguage> (дата обращения: 13.10.2025).

4.1.3 WebAssembly Specification 2.0 [Электронный ресурс]. URL: <https://webassembly.github.io/spec/versions/core/WebAssemblyU2.0.pdf> (дата обращения: 11.11.2025).

4.1.4 Cat language repository [Электронный ресурс]. URL: <https://github.com/cdiggins/catlanguage> (дата обращения: 13.10.2025).

4.2 Цель работы

Целью работы является прототип компилятора стекового языка программирования для трансляции и верификации программ и создания исполняемых файлов.

4.3 Решаемые задачи

4.3.1 Анализ синтаксиса и семантики различных стековых языков программирования, выделение и сравнение их особенностей.

- 4.3.2 Анализ средств статического анализа стековых языков программирования.
- 4.3.3 Анализ устройства байткода и модели исполнения WebAssembly.
- 4.3.4 Выбор технологии и средств разработки для программного обеспечения (далее ПО).
- 4.3.5 Проектирование программных компонентов: парсера, статического анализатора, кодогенератора.
- 4.3.6 Разработка структуры программного обеспечения и его компонентов.
- 4.3.7 Реализация компонентов ПО с использованием выбранных средств разработки.
- 4.3.8 Сборка программного обеспечения и его комплексное тестирование.
- 4.3.9 Разработка технологии построения SSA-представления текстов программ на стековых языках.

5 ТРЕБОВАНИЯ К ПРОГРАММНОМУ ИЗДЕЛИЮ

5.1 Требования к функциональным характеристикам

5.1.1 Выполняемые функции:

- статический анализ программы на предмет наличия ошибок несовпадения типов и исчерпания стека;
- генерация байткода WebAssembly.

5.1.2 Исходные данные:

- текст программы на исходном языке, который является подмножеством языка программирования Cat;
- управляющие флаги, регулирующие работу компилятора: выбор режима работы (только верификация или верификация с последующей кодогенерацией), уровня критичности выводимых в процессе работы сообщений, названия выходного файла.

5.1.2 Результаты:

- в случае успешной операции – файл с байткодом WebAssembly или сообщение, подтверждающее корректность программы;
- в случае неуспешной операции – сообщение с описанием ошибки.

5.2 Требования к надежности

5.2.1 Предусмотреть контроль консистентности управляющих флагов, передаваемых при вызове компилятора.

5.2.2 Предусмотреть контроль синтаксической корректности текста на исходном языке.

5.3 Условия эксплуатации

Условия эксплуатации в соответствии с СанПиН 2.2.2/2.4.1340-03.

5.4 Требования к составу и параметрам технических средств

Минимальная конфигурация технических средств, на которых может быть запущен компилятор:

- количество ядер процессора – 1 шт;
- объем ОЗУ – 1 Гб.

5.5 Требования к информационной и программной совместимости

5.5.1 Программа должна иметь интерфейс командной строки.

5.5.2 Программа должна работать под управлением операционных систем семейства Linux.

5.6 Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

5.7 Требования к транспортированию и хранению

Требования к транспортировке и хранению не предъявляются.

5.8 Специальные требования

Сгенерировать установочную версию ПО.

6 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

6.1 Разрабатываемые программные модули должны быть самодокументированы, т.е. тексты программ должны содержать все необходимые комментарии.

6.2 В состав сопровождающей документации должны входить:

6.2.1 Расчетно-пояснительная записка на 55-60 листах формата А4 (без приложений).

6.2.2 Техническое задание (Приложение А).

6.2.3 Руководство программиста (Приложение Б).

6.3.4 Исходный текст программного модуля вывода типов (Приложение В).

6.4 Графическая часть должна быть выполнена на 6 листах формата А1 (копии формата А3/А4 включить в качестве приложений к расчетно-пояснительной записке):

6.3.1 Схема процесса трансляции программы в байткод WebAssembly и ее исполнения.

6.3.2 Результаты анализа стековых языков.

6.3.3 Схема алгоритма вывода типов.

6.3.4 Функциональная диаграмма общего процесса работы.

6.3.5 Диаграмма компоновки.

6.3.6 Схема алгоритма построения SSA-представления текстов программ на стековых языках.

7 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Требования не предъявляются.

8 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Этапы разработки курсовой работы указаны в таблице 1.

Таблица 1 — Этапы разработки

№	Название этапа	Срок даты, %	Отчетность
1	2	3	4
1.	Разработка техническо-го задания	9.02.2026 – 28.02.2026, 5%	Утвержденное техниче-ское задание и задание на выпуск-ную квали-фикацион-ную работу
2.	Анализ требований и уточнение специфика-ций (эскизный проект)	28.02.2026 – 14.03.2026, 25%	Спецификации ПО.
3.	Проектирование струк-туры программной си-стемы, проектирование компонентов (техниче-ский проект)	14.03.2026 – 01.04.2026, 50%	Диаграмма компоновки, функциональная диа-грамма общего процесса работы, схема алгоритма вывода типов
4.	Реализация компонен-тов и автономное тести-рование компонентов. Сборка и комплексное тестирование.	01.04.2026 – 14.05.2026, 75%	Тексты программных компонентов. Тесты, ре-зультаты тестирования.
5.	Разработка документа-ции.	14.05.2026 – 25.05.2026, 90%	Расчетно-пояснительная записка.
6.	Прохождение нормокон-троля, проверка на антиплагиат, получение рецензии, подготовка доклада и предзащита.	25.05.2026 – 6.06.2026, 95%	Иллюстративный мате-риал, доклад, рецензия, справки о нормо-кон-троле и проценте плаги-ата.
7.	Защита выпускной ква-лификацион-ной рабо-ты.	1.06.2026 – 04.07.2026, 100%	—

9 ПОРЯДОК КОНТРОЛЯ И ПРИЕМА

9.1 Порядок контроля

Контроль выполнения осуществляется руководителем еженедельно.

9.2 Порядок защиты

Защита осуществляется перед государственной экзаменационной комиссией (ГЭК).

9.3 Срок защиты

Срок защиты определяется в соответствии с планом заседаний ГЭК.

10 ПРИМЕЧАНИЕ

В процессе выполнения работы возможно уточнение отдельных требований технического задания по взаимному согласованию руководителя и исполнителя.