

УДК 004.43

## РАЗРАБОТКА ЯЗЫКА ПРОГРАММИРОВАНИЯ, СОВМЕЩАЮЩЕГО ПРЕИМУЩЕСТВА КОМПИЛИРУЕМЫХ И ИНТЕРПРЕТИРУЕМЫХ ЯЗЫКОВ, И СМАРТ-КОМПИЛЯТОРА К НЕМУ

**Козлов В.С.** (ФГБОУ ВО «МГТУ им. Г.И. Носова»/Проектная школа, г. Магнитогорск)

**Научный руководитель – доцент кафедры ИиИБ, Мазнин Д.Н.**

(ФГБОУ ВО «МГТУ им. Г.И. Носова», г. Магнитогорск)

**Введение.** В мире существует множество языков программирования, большинство из которых можно поделить на две группы: компилируемые и интерпретируемые. Компилируемые языки обладают высокой производительностью, позволяют взаимодействовать с операционной системой и внешними устройствами. Интерпретируемые обладают платформенной независимостью, представляют информативные ошибки, упрощают процесс разработки, но их производительность существенно ниже.

На сегодняшний день многим начинающим разработчикам сложно перейти с такого простого языка, как Python, на C++, поэтому необходим промежуточный язык программирования, который помогал бы им в этом.

Цель работы - создать язык программирования, который был бы удобнее C++ и производительнее любого интерпретируемого языка, и компилятор к нему.

Для выполнения поставленной цели были определены следующие задачи:

1. Изучить преимущества и недостатки различных языков программирования
2. Разработать концепцию и синтаксис языка программирования
3. Разработать компилятор к языку программирования

**Основная часть.** Придумывать «еще одну замену C++» - нецелесообразно: программист должен писать «что он хочет сделать, а не как». Необходим высокоуровневый язык программирования, который сумел бы заменить интерпретируемые языки в некоторых сценариях использования и отвечал бы всем современным требованиям: имел простой синтаксис, обеспечивал высокую безопасность при работе с памятью и несколькими потоками.

Самым главным в языке программирования является его синтаксис, потому что это способ выражения идей в программном коде. Простой синтаксис позволяет программисту быть эффективнее других разработчиков, сопоставимых по уровню, что уменьшает расходы и ускоряет выход продукта на рынок. Это преимущество стало основной причиной популярности интерпретируемых языков.

Не менее актуальным для языка программирования является его «безопасность». Без стабильности и безопасности производительность не имеет смысла, потому что самое главное в любой программе - корректность исполнения. Наиболее важным является безопасность при работе с памятью, так как большинство современных дефектов вызваны уязвимостями в ней.

Многопоточные программы становятся всё более актуальными, так как с каждым годом количество ядер в серверных и потребительских процессорах увеличивается. Увеличение количества потоков у программы может существенно повысить производительность, поэтому главным критерием оптимизации в современных приложениях является возможность эффективной работы с несколькими потоками. Этим преимуществом пользоваться трудно, потому что разработчику необходимо следить за тем, чтобы не вызвать «гонок за данными» и взаимных блокировок. Более того, асинхронный код не всегда повышает производительность программы.

Синтаксис языка будет C-подобным, потому что это определенный стандарт в отрасли. Часть конструкций будет взята из C++, но упрощена при помощи компилятора. Потенциально небезопасные конструкции будут полностью исключены из языка.

Главным концептуальным решением является отказ от заголовочных файлов и файлов ресурсов, так как они ограничивают средство разработки, создавая следующие проблемы:

- Низкий уровень глобальной оптимизации
- Невозможность компилятора отследить весь код во время сборки для полного его анализа
- Необходимость пересобирать проект, при изменении одного заголовочного файла

Еще одним решением является почти полный отказ от атрибутов. Компилятор, обладая возможностью полностью отслеживать код во время сборки, сможет расставлять атрибуты автоматически. Это решение позволит упростить язык и повысить независимость оптимизации от программиста.

**Выводы.** Результатом работы будет является язык программирования и компилятор к нему. Язык программирования будет сочетать в себе преимущества компилируемых и интерпретируемых языков, благодаря чему сможет найти применения в программировании различных мобильных устройств, устройств из мира интернет вещей, высокопроизводительных программ, а также в обучении молодых специалистов.

#### **Список использованных источников:**

1. Cimpanu C. 70 percent of all security bugs are memory safety issues: [Электронный ресурс]. URL: <https://www.zdnet.com/article/microsoft-70-percent-of-all-security-bugs-are-memory-safety-issues>. (Дата обращения 03.02.2023).
2. Herb S. Can C++ be 10x simpler & safer?: [Электронный ресурс]. URL: <https://herbsutter.com/2022/09/19/my-cppcon-2022-talk-is-online-can-c-be-10x-simpler-safer>. (Дата обращения 16.12.2022).
3. Kehrer P. Memory Unsafety in Apple's Operating Systems: [Электронный ресурс]. URL: <https://langui.sh/2020/07/24/apple-memory-safety>. (Дата обращения 06.02.2023).