МИНИСТЕРСТВО ОБРАЗОВАНИЯ СТАВРОПОЛЬСКОГО КРАЯ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«СТАВРОПОЛЬСКИЙ РЕГИОНАЛЬНЫЙ МНОГОПРОФИЛЬНЫЙ КОЛЛЕДЖ»

**ИНДИВИДУАЛЬНЫЙ ПРОЕКТ**

по дисциплине: ОУД.09 Информатика

**Тема:**

**«История развития языков программирования»**

**Выполнил:**
обучающийся группы П-11
Желнов Милан Альбертович

**Руководитель:**
Косторнова Людмила Николаевна

Ставрополь, 2019 г.

Содержание

[Введение 3](#_Toc125349)

[Глава 1. История возникновения языков программирования. 5](#_Toc125350)

[1.1. Начало развития 5](#_Toc125351)

[1.2. Язык программирования Ассемблер 5](#_Toc125352)

[1.3. Язык программирования FORTRAN 6](#_Toc125353)

[1.4. Язык программирования Algol 58 7](#_Toc125354)

[1.5. Язык программирования C 7](#_Toc125355)

[1.6. Язык программирования C# 8](#_Toc125356)

[1.7. Язык программирования C++ 9](#_Toc125357)

[1.8. Язык программирования Java 10](#_Toc125358)

[Глава 2. Современные языки программирования 13](#_Toc125359)

[2.1. Язык программирования Swift. 13](#_Toc125360)

[2.2. Япрограммирования Scala 13](#_Toc125361)

[2.3. Язык программирования Go. 15](#_Toc125362)

[Список литературы и интернет-источников 18](#_Toc125363)

# Введение

Вы когда-нибудь использовали приложение на мобильном телефоне или программу на компьютере? А задумывались ли вы при этом, на каком языке программирования написаны сайты, интернет страницы, скрипты?

С каждым днем наш мир становится более информационным и компьютер заменит в скором времени человека в разной деятельности. Чтобы это обеспечить, создается большое количество программного обеспечения с помощью языков программирования. На сегодняшний день существует большое количество языков программирования, которые помогают программистам решать задачу любого уровня: от простых веб-сайтов до сложных и нагруженных приложений.

Каждый язык программирования выделяется на фоне остальных своими особенностями (область применения, возможности, уровень доступа к машине), но стоит знать языки, которые оставили большой след в истории языков программирования. Многие из них применяются и до сих пор, обгоняя современные языки программирования.

**Цель:** проследить историю возникновения языков программирования от самых начальных этапов до наших дней.

**Объект:** история развития языков программирования.

**Предмет:** языки программирования.

**Гипотеза:** почему одни языки программирования забыты, а другие легли в основу новых и используются до сих пор?

**Задачи:**

1. Провести изучение и теоретический анализ литературы по данной проблеме.
2. Узнать историю возникновения языков программирования.
3. Выделить и рассмотреть два уровня языков программирования.
4. Рассмотреть языки программирования в хронологическом порядке.
5. Подготовить презентацию сопровождения проекта.

**Актуальность.**

Современные дети очень быстро развиваются. В два года они уже неплохо разбираются в родительских гаджетах, а к первому классу готовы сконструировать собственного двигающегося робота. Детей все больше увлекает интернет и программирование. В погоне за прогрессом, иногда полезно остановиться, вспомнить историю, задуматься о будущем, чтобы с новыми силами вновь окунуться в работу, осваивать, а может быть и создавать новые технологии, новые языки программирования.

Актуальность данной темы, с моей точки зрения, обусловлена тем, что прогресс компьютерных технологий определил процесс появления новых разнообразных знаковых систем для записи алгоритмов – языков программирования.И дать ответ на основной вопрос: «Почему одни языки программирования остались на стадии зарождения и про них все забыли, а другие легли в основу новых, которые живут и позволяют человечеству развивать информационные технологии»?

# Глава 1. История возникновения языков программирования.

## Начало развития

Изначально принцип работы электронно-вычислительной машины устроен так, что компьютер может принимать машинный код(0 и 1 - последовательность перепада напряжения)

Такой подход имел большие затруднения. Программисту трудно было контролировать программу, был большой риск допустить критическую ошибку (к примеру, перепутав 0 и 1)

Еще одна проблема заключалась в том, что за разработку языков программирования брались инженеры, а не программисты, поэтому операции представлялись в виде последовательности номеров операций.

Программирование появилось задолго до 50-х годов XX века. Первые идеи высказал ещё Чарльз Бэббидж (1792-1871), которого по праву считают отцом компьютера. Он ничего не знал о транзисторах, микросхемах и мониторах, но достаточно точно описал основные принципы, на которых будут строиться все вычислительные машины.

Развила идею графиня Ада Лавлейс (1815-1852). Её место в истории до сих вызывает немало споров, но одно абсолютно точно – именно Ада фактически стала первым известным программистом. Благодаря её трудам стало понятно, что путь к эффективному использованию машин – алгоритмы, описанные в коде.

Решение проблемы с языком программирования было крайне просто: представить цифры в буквы, что и повлекло созданием языка программирования низкого уровня – **Ассемблера**. Хотя такойспособ итребовал лишних ресурсов для перевода из символов в цифры, но, учитывая проблему, жертвы были минимальны.

## Язык программирования Ассемблер

Язык Ассемблера — машинно-ориентированный язык программирования низкого уровня.

Имеется много вариаций языков, заточенные под определенные платформы. Они, имея подобный вид, являются несовместимыми.

Многие компиляторы с языков программирования высокого уровня выдают результат в виде программы на языке ассемблера, которую в дальнейшем обрабатывает ассемблер. Результатом ассемблирования может быть объектный модуль, который потом с помощью редактора связей переходит в исполняемый файл.

Основными преимуществами языка Ассемблера являются:

1. Быстродействие;
2. Безопасность;
3. Эффективность за счет исполняемой платформы;
4. Понимание программного кода.

Однако у языка есть и свои недостатки:

1. Трудное чтение кода из-за однотипных операций и действий;
2. Высокий порог вхождения;
3. Многие платформы не нуждаются в Ассемблере;
4. Трудоемкий процесс написания программного кода.

Язык Ассемблера в наше время находит себе применение в многих отраслях, где необходима низкая затрата и высокое быстродействие, однако в различных сферах его нецелесообразно использовать.

## Язык программирования FORTRAN

Фортран является первым языком высокого уровня. Создан был в периоде с 1954 по 1957 года группой программистов под руководством Джона Бэкуса в корпорации IBM.

Основной задачей при разработке языка программирования Фортранбылосоздать альтернативу языка Ассемблер. Первое руководство появилось уже в 1956 году с компилятором (который был оптимизирующий). Без него люди отказывались программировать так как терялась скорость работы программы.

Одно из преимуществ Фортрана является большое количество библиотек и подпрограмм написанных на нем.

Фортран в СССР появился позже, чем на Западе, поскольку поначалу в СССР более перспективным языком считался Алгол. Внедрению языка программирования Фортран поспособствовало общение советских физиков со своими коллегами из CERN, где в 1960-х годах почти все расчёты велись с использованием программ на Фортране.

Первый советский компилятор с Фортрана был создан в 1967 году для машины Минск-2, однако он не получил большой известности.

Широкое внедрение Фортрана началось после создания в 1968 году компилятора ФОРТРАН-ДУБНА для машины БЭСМ-6.

Современный Фортран адаптирован под новые тенденции программирования, такие как: модульное и обобщенное программирование, объектно-ориентированное программирование (ООП)

В конце 1980-х — начале 1990-х физик Андрей Зарецкий создал серию детских книжек, одним из главных героев которых был профессор Фортран. В книге доступным языком объясняется устройство компьютера, а также основы компьютерной грамотности и программирования.

## Язык программирования Algol 58

Язык программирования Алгол был разработан в 1958 году на недельной конференции в ETH как универсальный язык программирования для широкого круга применений.

В ходе работы возникали большие трудности непринципиального характера. Так, например, один из членов комитета вспоминал «десятичную бурю» — крайне резкую дискуссию между американскими и европейскими участниками по поводу того, какой именно символ должен быть использован в качестве разделителя целой и дробной частей числа. Чтобы избежать конфликтов по мелким вопросам, было решено, что описание Алгола будет трёхуровневым, включающим уровень описаний, публикаций и реализации.

Отличительным свойством языка стала то, что программа представлялась не как свободная последовательность команд, а как блочная структура, состоящая из описанных и отдаленных друг от друга частей.

Были выделены структурные управляющие конструкции: ветвление, циклы, последовательные участки, ограниченные ключевыми словами begin и end

В Алгол была доступна рекурсия, что позволяло описывать программу более понятно за счет сходства с математическим описанием алгоритмической задачи.

## 1.5.Язык программирования C

Си (от англ.C) - компилируемый язык программирования общего назначения, разработанный как продолжение языка Би (бинарных отношений).

Конструкции языка близко сопоставляются типичным машинным конструкциям, благодаря чему он нашел применение в проектах, для которых был свойствен язык ассемблер

Язык программирования Си оказал существенное влияние на индустрию программного обеспечения. Его синтаксис заложил основу для C++, C#, Java и Objective-C.

Язык был разработан в лабораториях BellLabsв период с 1969 по 1973 годы. Язык так был назван по той причине, что буква C третья по счету в английском алфавите, а основы берет с языка B(второй буквы по счету в английском алфавите)

К 1973 году язык Си стал достаточно силён, и большая часть ядра UNIX, первоначально написанная на ассемблере, была переписана на Си. Это было одно из самых первых ядер операционных систем, написанное на языке, отличном от ассемблера

Язык программирования Си отличается минимализмом. Авторы изначально хотели, чтоб программа компилировалась легко с помощью однопроходного компилятора.

Код на Си можно писать на низком уровне абстракции, что делает его похожим на язык ассемблер, поэтому его называют часто языком низкого или среднего уровня, но по строгой классификации он является языком высокого уровня

После появления язык был хорошо принят, потому что, он позволял быстро создавать компиляторы для новых платформ, а также позволял программистам довольно точно представлять, как выполняются их программы.

Благодаря близости к языкам низкого уровня программы на Си работали эффективнее написанных на многих других языках высокого уровня

Язык широко применяется при разработке операционных системе и высокопроизводительного программного кода. Первые реализации языком программирования Java, Python и PHP были написаны на C.

В целом, язык дает большие возможности для написания качественного кода, но при неправильном использовании он позволяет допустить критические ошибки.

## 1.6. Язык программирования C#

C# - объектно-ориентированный язык программирования, разработанный в 1998-2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтауота как язык разработки приложений для платформы Microsoft .NETFramework.

В поддержку новшеств Microsoft выпустила инструментарий для разработки приложений – платформу .NET. Она также объединяла «под одной крышей» несколько языков программирования, что было в новинку для того времени. Еще одним новшеством платформы .NET была технология активных серверных страниц ASP.NET (ActiveServerPage). С её помощью можно было относительно быстро разработать веб-приложения, взаимодействующие с базами данных.

Язык программирования C# был создан специально для ASP.NET. На C# полностью была написана и сама ASP.NET.

Название **«Си шарп»** происходит от буквенной музыкальной нотации, где латинской букве C соответствует нота До, а знак диез (англ. sharp) означает повышение соответствующего ноте звука на полутон. Название так же можно представить как несколько «+».

Название языка программирования принято называть «Си шарп».

C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR.

CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом языке C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.

Язык C# широко используется в различных сферах разработки: разработка программного обеспечения, веб-разработка(ASP.NET), гейм-дев (Unity 3D), разработка мобильных приложений.

## 1.7. Язык программирования C++

C++ - компилируемый, статически типизированный язык программирования общего назначения. Он поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование.

C++ широко используется для разработки ПО. Область его применения включает создание операционных систем, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр.

Язык возник в начале 1980-х годов, когда сотрудник фирмы BellLabs**Бьёрн Страуструп** придумал ряд усовершенствований к языку C под собственные нужды. Когда в конце 1970-х годов Страуструп начал работать в BellLabs над задачами теории очередей (в приложении к моделированию телефонных вызовов), он обнаружил, что попытки применения существующих в то время языков моделирования оказываются неэффективными, а применение высокоэффективных машинных языков слишком сложно из-за их ограниченной выразительности.

Вспомнив опыт своей диссертации, Страуструп решил дополнить язык **C** возможностями, имеющимися в языке Симула. Страуструп добавил к нему возможность работы с классами и объектами, что добавило больше возможностей для моделирования.

До начала официальной стандартизации язык развивался в основном силами Страуструпа в ответ на запросы программистского сообщества.

Имя языка, получившееся в итоге, происходит от оператора унарного постфиксного инкремента C ++ (увеличение значения переменной на единицу). Имя C+ не было использовано потому, что является синтаксической ошибкой в C и, кроме того, это имя было занято другим языком. Язык также не был назван D, поскольку «является расширением C и не пытается устранять проблемы путём удаления элементов C».

Стандартная библиотека C++ включает в себя набор средств, которые должны быть доступны для любой реализации языка, чтобы обеспечить программистам удобное пользование языковыми средствами. Стандартная библиотека C++ включает в себя часть стандартной библиотеки C. Стандарт C++ содержит нормативную ссылку на стандарт C от 1990 года и не определяет самостоятельно те функции стандартной библиотеки, которые заимствуются из стандартной библиотеки C.

Доступ к возможностям стандартной библиотеки C++ обеспечивается с помощью включения в программу (посредством директивы #include) соответствующих стандартных заголовочных файлов. Всего в стандарте C++11 определено 79 таких файлов

Неоднократно предпринимались попытки предложить альтернативы C++, как для прикладного, так и для низкоуровневого программирования.Старейшим конкурентом C++ в задачах низкого уровня является Objective-C, совмещающий Си с объектной моделью Smalltalk.Прямым потомком на данный момент является язык D.

## 1.8. Язык программирования Java

Java -сильно типизированныйобъектно-ориентированный язык программирования, разработанный компаниейSunMicrosystems(в последующем приобретённой компаниейOracle). Приложения Java обычнотранслируютсяв специальныйбайт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощьювиртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года.

Изначально язык назывался Oak («Дуб»), разрабатывался Джеймсом Гослингом для программирования бытовых электронных устройств. Из-за того, что язык с таким названием уже существовал, вскоре Oak был переименован в Java. Назван в честь марки кофе Java, которая, в свою очередь, получила наименование одноимённого острова (Ява), поэтому на официальной эмблеме языка изображена чашка с горячим кофе. Существует и другая версия происхождения названия языка, связанная с аллюзией на кофе-машину как пример бытового устройства, для программирования которого изначально язык создавался.

Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

Часто к недостаткам концепции виртуальной машины относят снижение производительности.Ряд усовершенствований несколько увеличил скорость выполнения программ на Java:

* применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с возможностью сохранения версий класса в машинном коде,
* широкое использование платформенно-ориентированного кода (native-код) в стандартных библиотеках,
* аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например, технология Jazelle, поддерживаемая некоторыми процессорами архитектуры ARM).
* Следующие успешные проекты реализованы с привлечением Java (J2EE) технологий: RuneScape,Amazon, eBay, LinkedIn, Yahoo! Компании в основном фокусируются на Java (J2EE) технологиях: SAP, IBM, Oracle. В частности, СУБД OracleDatabase включает JVM как свою составную часть, обеспечивающую возможность непосредственного программирования СУБД на языке Java, включая, например, хранимые процедуры.
* Программы, написанные на Java, имеют репутацию более медленных и занимающих больше оперативной памяти, чем написанные на языке C. Тем не менее, скорость выполнения программ, написанных на языке Java, была существенно улучшена с выпуском в 1997—1998 годах так называемого JIT-компилятора в версии 1.1 в дополнение к другим особенностям языка для поддержки лучшего анализа кода (такие, как внутренние классы, класс StringBuffer, упрощенные логические вычисления и т. д.). Кроме того, была произведена оптимизация виртуальной машины Java — с 2000 года для этого используется виртуальная машина HotSpot. По состоянию на февраль 2012 года, код Java 7 приблизительно в 1,8 раза медленнее кода, написанного на языке Си.

Некоторые платформы предлагают аппаратную поддержку выполнения для Java. К примеру, микроконтроллеры, выполняющие код Java на аппаратном обеспечении вместо программной JVM, а также основанные на ARM процессоры, которые поддерживают выполнение байт-кода Java через опцию Jazelle.

# Глава 2. Современные языки программирования

## 2.1. Язык программирования Swift.

**Swift**— открытый мультипарадигмальный компилируемый язык программирования общего назначения. Создан компанией Apple в первую очередь для разработчиков iOS и macOS. Swift работает с фреймворками Cocoa и CocoaTouch и совместим с основной кодовой базой Apple, написанной на Objective-C.Swift задумывался как более лёгкий для чтения и устойчивый к ошибкам программиста язык, нежели предшествовавший ему Objective-C

Старший вице-президент по разработке программного обеспечения Apple Крейг Федериги во время анонса этого продукта заявил, что язык программирования Swift был заложен ещё в платформе NeXT (ОС NeXTSTEP выпускалась в **1989—1995 годах**), которая стала основой для современной macOS, а затем и iOS.

**2 июня 2014 года** на конференции WWDC язык Swift был официально представлен вместе с бесплатным руководством по использованию языка объёмом в 500 страниц, доступным на сервисе «iBookStore».

8 июня 2015 года компания Apple объявила о выпуске новой версии Swift 2.0.

3 декабря 2015 года была выпущена бета версия Swift 3.0 с поддержкой операционных систем OS X, iOS и Linux и лицензированная под открытой лицензией Apache 2.0.

10 апреля 2016 года Google объявила о намерениях сделать Swift так называемым «первым языком» для Android.

19 сентября 2017 года была выпущена версия Swift 4.0.

Часть функций языка выполняется быстрее по сравнению с другими подобными языками. Например, сортировка комплексных объектов выполняется в 3,9 раз быстрее, чем в Python, и почти в 1,5 раза быстрее, чем в Objective-C.

Код, написанный на Swift, может работать вместе с кодом, написанным на языках программирования C и Objective-C в рамках одного и того же проекта.

Apple разделила код Swiftна несколько открытых репозиториев:

* компилятор и стандартная библиотека;
* библиотеки ядра;
* менеджер пакетов;
* клонированные репозитории.

##

## 2.2. Язык программирования Scala

Scala—мультипарадигмальный язык программирования, спроектированный кратким и типобезопасным для простого и быстрого создания компонентного программного обеспечения, сочетающий возможности функционального и объектно-ориентированного программирования.

Первые версии языка созданы в 2003 году коллективом лаборатории методов программирования Федеральной политехнической школы Лозанны под руководством Мартина Одерски, язык реализован для платформ Java и JavaScript.

Язык был создан в 2001—2004 годах в Лаборатории методов программирования EPFL. Он стал результатом исследований, направленных на разработку улучшенной языковой поддержки компонентного программного обеспечения. За основу при разработке языка были взяты две идеи:

* Язык программирования компонентного ПО должен быть масштабируемым. Поэтому внимание было сконцентрировано на механизмах абстракции, композиции и декомпозиции
* Масштабируемая поддержка компонентов может быть предоставлена языком программирования, унифицирующим и обобщающим объектно-ориентированное и функциональное программирование.

Язык был выпущен для общего пользования на платформе JVM в январе 2004 года и на платформе .NET в июне 2004 года, в 2016 году создан LLVM-компилятор (ScalaNative

Прежде всего, язык впитал значительное число концепций и синтаксических соглашений Java и C#.Абстрактные типы в Scala очень похожи на абстрактные типы сигнатур в SML и OCaml, обобщённые в контексте полноценных компонентов.

Scala-программы во многом похожи на Java-программы, и могут свободно взаимодействовать с Java-кодом. Язык включает единообразную объектную модель — в том смысле, что любое значение является объектом, а любая операция — вызовом метода. При этом является также функциональным языком в том смысле, что функции — это полноправные значения.

В языке используется чистая объектно-ориентированная модель, похожая на применяемую в Smalltalk: каждое значение — это объект, и каждая операция — это отправка сообщения

Каждая функция — это значение. Язык предоставляет легковесный синтаксис для определения анонимных и каррированных функций.

Scala может взаимодействовать с кодом, написанным на Java. Все классы из пакета java.lang уже подключены по умолчанию, в то же время другие должны быть подключены явно.

Основные веб-фреймворки, написанные на Scala — Play, Lift. Социальная сеть LinkedIn использует микрофреймворкScalatra для поддержки своего Signal API. Важная составляющая инфраструктуры разработки на Scala — средство автоматической сборки Sbt — также написана на Scala.

##

## 2.3. Язык программирования Go.

Go (часто также Golang)— компилируемый многопоточный язык программирования, разработанный внутри компании Google. Первоначальная разработка Go началась в сентябре 2007года, а его непосредственным проектированием занимались Роберт Гризмер, Роб Пайк и Кен Томпсон, занимавшиеся до этого проектом разработки операционной системы Inferno.

Язык Go разрабатывался как язык программирования для создания высокоэффективных программ, работающих на современных распределённых системах и многоядерных процессорах. Он может рассматриваться как попытка создать замену языкам Си и C++. По словам Роба Пайка, «Go был разработан для решения реальных проблем, возникающих при разработке программного обеспечения в Google». В качестве основных таких проблем он называет:

* медленную сборку программ;
* неконтролируемые зависимости;
* использование разными программистами разных подмножеств языка;
* затруднения с пониманием программ, вызванные неудобочитаемостью кода, плохим документированием и так далее;
* дублирование разработок;
* высокую стоимость обновлений;
* несинхронные обновления при дублировании кода;
* сложность разработки инструментария;
* проблемы межъязыкового взаимодействия.

Основные возможности языка Go:

* Go — язык со строгой статической типизацией. Доступен автоматический вывод типов, для пользовательских типов — «утиная типизация».
* Полноценная поддержка указателей, но без возможности применять к ним арифметические операции, в отличие от C/C++/D.
* Строковый тип со встроенной поддержкой юникода.
* Использование динамических массивов, хэш-таблиц, срезов (слайсов), вариант цикла для обхода коллекции.
* Средства функционального программирования: неименованные функции, замыкания, передача функций в параметрах и возврат функциональных значений.
* Автоматическое управление памятью со сборщиком мусора.
* Средства объектно-ориентированного программирования, но без поддержки наследования реализации (наследуются только интерфейсы). По большому счёту, Go является процедурным языком с поддержкой интерфейсов.
* Средства параллельного программирования: встроенные в язык потоки (goroutines), взаимодействие потоков через каналы и другие средства организации многопоточных программ.
* Достаточно лаконичный и простой синтаксис, основанный на Си, но существенно доработанный, с большим количеством синтаксического сахара.

В основном критика языка фокусируется на отсутствии в нём тех или иных популярных средств, предоставляемых другими языками. В первую очередь это средства обобщённого программирования (generics) и структурная обработка исключений. Также часто критикуется отсутствие «полноценного ООП» (фактически — наследования реализации), перегрузки функций, переопределения операторов.

**Вывод**.

В представленном мною проекте была рассмотрена история возникновения языков программирования, исследованы как языки низкого, так и высокого уровней.

Можно предположить, что в ближайшем будущем, человеческие познания в этой сфере, помогут произвести на свет языки, умеющие принимать, обрабатывать и передавать информации в виде мысли, слова, звука или жеста.

Уже сейчас появляются языки программирования, которые можно назвать языками сверхвысокого уровня. Они предназначены не для написания прикладных программ, а для обработки больших массивов данных и статистики. В их числе R, MATLAB и Julia.

Можно предполагать, что в будущем появятся человекоориентированные языки, на которых написать программу будет не сложнее, чем пост или инстаграмм. С другой стороны, квантовые компьютеры потребуют свой особый, квантовый ассемблер.

На сегодняшний день, любое предприятие, имеет в своем штате группу программистов, обладающими знаниями программирования различными языками, которые редактируют, изменяют, и модифицируют программы используемыми сотрудниками предприятия. Это говорит о том, что на рынке труда пользуются спросом люди, обладающими знаниями и опытом работы с различными языками программирования.

В наше время знание языков программирования не просто показатель образованности человека, а и необходимость. Если вы хотите идти в ногу со временем, для вас просто обязательно знание, хотя бы одного языка программирования, поэтому моя специальность 09.02.03 «Программирование в компьютерных системах» будет всегда востребована.

На мой взгляд, все вопросы рассмотрены, цель достигнута.

# Список литературы и интернет-источников

1. Дмитрий М., Алексей Н. «История компьютерной эры», 2016 год;
2. Ревич Ю.В. «Практическое программирование микроконтроллеров на AtmelAVR на языке ассемблера», 2014 год;
3. Доусон М. «Изучаем C++ через программирование игр», 2017 год;
4. Липпман С., Лажойе Ж., Му Б. «Язык программирования C++. Базовый курс», 2014 год;
5. Валерий К. «Язык программирования Go. Руководство 2016», 2016 год;
6. Михаил Ф. «Библия C#» (3 издание), 2016 год;
7. Михаил Ф. «DirectX и C++. Искусство программирования» (
8. Уильям. К., Брайн. К., Эрик. Ст-М. «Go в действии», 2015 год;
9. Герберт Ш. «Java 8. Руководство для начинающих» (6-е издание), 2015 год;
10. Мартин. О., Лекс С., Билл В. «Scala. Профессиональное программирование», 2018 год.