**IT-Колледж “Сириус”**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ДОКЛАД**

по дисциплине “Введение в специальность”

на тему “ «Мертвые» языки программирования”

Выполнила:  
Студентка группы

1.9.7.2  
Тяпкова Виктория Александровна

Принял:

Старший преподаватель  
Тенигин Альберт Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT-Колледж “Сириус”  
2022

**Содержание**

[Введение 3](#_Toc119720575)

[Глава 1. Программирование. Языки программирования 4](#_Toc119720576)

[Глава 2. История возникновения языков программирования 5](#_Toc119720577)

[Глава 3. Первые языки программирования 7](#_Toc119720578)

[3.1 Первый высокоуровневый язык 7](#_Toc119720579)

[3.2 Последующие языки программирования 8](#_Toc119720580)

[Глава 4. «Мёртвые» языки программирования и почему они стали такими 9](#_Toc119720581)

[4.1 Языки, которые подвержены исчезновению в ближайшее десятилетие 9](#_Toc119720582)

[4.2«Мёртвые» языки программирования 10](#_Toc119720583)

[Заключение 13](#_Toc119720584)

[Список используемой литературы 14](#_Toc119720585)

[Приложение 16](#_Toc119720586)

# Введение

Программирование является одной из актуальных и прогрессивных сфер деятельности. Языки программирования выступают его основным инструментом для общения, работы и творчества в Интернет-пространстве.

На данный момент количество языков программирования переваливает за тысячу. Но пользующихся популярность сейчас всего десяток. Почему же такая большая разница в цифрах?

Всё дело в том, что языки программирования, как и многие другие концепции, идеи и разработки со временем теряют свою актуальность. С каждым разом создаются всё более понятные и простые. При том, успех языка не всегда связан с его техническими составляющими. Часто, достаточно, чтобы он оказался в нужное время, в нужном месте. Про какие же языки программирования забыли и почему они пылятся, оставшись лишь в памяти компьютера?

**Актуальность проекта:**

За последние 50 лет программирование плотно вошло в нашу жизнь. Оно тесно переплетается с множеством сфер деятельности: мультипликация, музыка, кино, журналистика, игры, дизайн и многое другое [[1](#первая)].

**Цель проекта:**

Познакомиться с историей развития языков программирования и программирования в целом. А также выяснить, какие языки программирования считаются «мёртвыми» и почему.

**Задачи проекта:**

* Познакомиться с началом появления и развития программирования
* Ознакомиться с первыми языками программирования и их последующим развитием
* Узнать, что означает понятие «мёртвый» язык и какие языки входят в состав этого понятия на сегодняшний день
* Выявить причину, по которой «мёртвые» языки программирования потеряли свою актуальность
* Сделать соответствующие выводы

# Глава 1. Программирование. Языки программирования

Для того, чтобы погрузиться в историю «мёртвых» языков программирования, предлагаю ознакомиться с «основой», то есть с понятиями «программирование» и «языки программирования».

Что же такое программирование? У данного определения есть множество понятий, причём в разных науках. Например, в математике есть такие понятия, как *линейное программирование, динамическое программирование, стохастическое программирование.* Но все они не имеют ничего общего с нужным нам определением программирования [[2](#вторая)].

В «Математической энциклопедии»: Программирование - 1) процесс составления *программы,* плана действий; 2) дисциплина, изучающая методы и приемы составления программ.

В других словарях «программирование»:

— действие по подготовке программы выполнения задания [[4](#четвертая)].

— научная и практическая деятельность по созданию программ [[5](#пятая)] и др.

В своей работе я буду использовать следующее определение: Программирование - процесс составления упорядоченной последовательности действий для ЭВМ [[3](#третья)].

Далее познакомимся с понятием «язык программирования». Для начала выявим, что такое язык и что он представляет.

Язык - ес­те­ст­вен­но воз­ник­шая в че­ло­ве­че­ском об­ще­ст­ве и раз­ви­ваю­щая­ся сис­те­ма об­ле­чён­ных в зву­ко­вую фор­му зна­ко­вых еди­ниц, спо­соб­ная вы­ра­зить всю со­во­куп­ность по­ня­тий и мыс­лей че­ло­ве­ка и пред­на­зна­чен­ная пре­ж­де все­го для це­лей ком­му­ни­ка­ции [[6](#шестая)].

Язык программирования - система слов и правил, используемая для создания программ для компьютера (Научно-технический словарь).

# Глава 2. История возникновения языков программирования

Многие знают, что первым, кто изложил свои идеи по архитектуре и работе ЭВМ, которые позже были обращены в реальность, является Чарльз Бэббидж.

Бэббидж – математик, один из создателей «Аналитического общества» в Кэмбриджском университете, изобретатель первого компьютера [[7](#седьмая)]. Также он интересовался инженерным делом, точнее работой поездов. Внёс свой вклад в развитие поездов, придумав специальные «очистители» дорожных путей. Он участвовал в создании Астрономического общества и в создании единого стандарта астрономических вычислений. Именно благодаря работе над исправлением ошибок в счётных таблицах у Чарльза Бэббиджа появились первые мысли о механизации труда по подсчётам.

В 1822 году Бэббиджем была опубликована статья с описанием машины, которая могла бы заменить ручные подсчёты. Благодаря его сильным знаниям в математике, она должна была уметь вычислять значение разности многочленов до 6 степени, с точностью до 18-го знака. Позже Ч. Бэббидж приступает к проектированию данного изобретения. Эта идея до того заинтересовала многих, что проект спонсировался правительством Великобритании. К несчастью, данный проект затянулся на 9 лет вместо планируемых 3, поэтому спонсирование прекратилось. В итоге, проект так и не был завершён. Но Чарльз не отчаивается и начинает заниматься новой разработкой более общей версии механического компьютера.

Уже после смерти математика, во второй половине 19-го века, современники воплощают задумки Бэббиджа в реальность. Благодаря чему современная копия вычислительной машины есть в Маунтин Вью (Калифорния) и лондонском музее [[8](#восьмая)].

А как на счёт языков программирования?

Первым программистом считается Ада Лавлейс – дочь поэта Байрона. Она познакомилась с Чарльзом Бэббиджем в 1832 году (на том момент ей было всего 17 лет).

Их совместная работа началась в 1842 году, после прочтения цикла своих лекций об Аналитической машине в Туринском университете Бэббиджем. Он попросил Лавлейс перевести на английский конспекты, составленные Луиджи Менабреа (- итальянский генерал; политический деятель и математик) на основе его лекций.

Центральным же моментом в её работе стало составление программы вычисления чисел Бернулли. 10 июля 1843 года она пишет Бэббиджу: "Я хочу вставить в одно из моих примечаний кое-что о числах Бернулли в качестве примера того, как неявная функция может быть вычислена машиной без того, чтобы предварительно быть разрешённой с помощью головы и рук человека. Пришлите мне необходимые данные и формулы". Кроме того, что Бэббидж прислал необходимые данные, он также определил последовательность действий, которые должны лежать в основе программы. Но при этом он допустил ошибку, которая была исправлена Адой. Через несколько недель программа была готова. Так появилась первая компьютерная программа [[9](#девятая)].

# Глава 3. Первые языки программирования

3.1 Первый высокоуровневый язык

Мы узнали, что первую компьютерную программу написала Ада Лавлейс при совместной работе с Чарльзом Бэббиджем. Но когда и кем был разработан «настоящий» (то есть функционирующий) язык программирования?

Таким языком программирования считается Plankalkül (Планкалкюль). Создан он аж через сто лет после триумфа Ады Лавлейс, в период с 1942 по 1946 года немецким инженером Конрадом Цузе, более известным как создатель первого компьютера.

Свои первые электромеханические вычислительные машины Z1-Z3 он создал в 1937-1941 гг. Они работали по принципу фон Неймана, а также были прототипами Аналитической машины Чарльза Бэббиджа. В процессе создания своей следующей работы, ЭВМ Z4, к Цузе приходит идея способа программирования компьютера, в основе которой лежит система исчисления шагов решения задач, с помощью числовых и символьных обозначений. Так появляется Планкалкюль (kalkül - «исчисление» и plan – «план»). Данный язык планировался для исчисления [[10](#десятая)].

**Структура языка**

Переменные делятся на четыре вида.

* «Variablen» (Входные переменные) — это входные переменные подпрограмм, доступные только для чтения; начинаются с буквы «V» и номера.
* «Zwischenwert» (Значения промежуточные), доступны для чтения и записи, предназначены для хранения промежуточных вычисляемых значения; начинаются с «Z» и номера.
* «Resultatwerte» (Результат) — в этих переменных возвращается результат вычисления; начинаются с «R» и номера.
* «Indizes» (Индексы) — переменные цикла, начинаются с «i», дальше может быть номер.

Имеются три вида переменных:

* Для целого неотрицательного указывается размерность в битах. Есть две формы записи «0» — один бит, «n.0» — «n» бит, например, 8.0 — один байт
* Кортеж указывается в скобках, например (3.0, 4.0) — это две переменные в три и четыре бита, кортеж должен иметь более одного элемента.
* Массив записывается через точку, к примеру: 4.5.0 — массив из четырёх элементов по пять бит в каждом, 32.(0, 8.0, 16.0) — 32 кортежа, в каждом из которых три переменных: один бит, восемь и  
  шестнадцать.

Пример программы, написанной на данном языке: см. рисунок 1.

3.2 Последующие языки программирования

**Язык Ассемблера**

Это мнемонический язык (мнемоника - способ запоминания информации, с помощью изменения типа информации), разработанный Никлаусом Виртом в 1948 году. Этот язык очень облегчил жизнь программистам, потому что вместо привычного кода в виде комбинации цифр, появляется код, состоящий из слов и фраз. Из-за чего ошибки в программах стало находить гораздо легче, а объём программы в разы уменьшился [[11](#одиннадцать)].

Пример программы, написанной на языке Ассемблера: см. рисунок 2.

**Shortcode**

Данный язык был предложен Джоном Мокли в 1949 году. А первая реализация была написана Вильгельмом Шмидтом в том же году. Основной идеей реализации Shortcode было то, чтобы компьютер воспринимал алгебраические выражения в привычном виде. Реализация не удалась в полной мере, поэтому конечный вариант вышел в свет лишь в 1952 году благодаря А.Б. Тонику и Дж. Р. Логану [[12](#двенадцатая)].

Пример программы, написанной на Shortcode: см. рисунок 3.

**Autocoder**

Это не является как таковым языком программирования, а относится к тем программам, которые облегчают создание кодов на машинном языке. Первые Autocoders были разработаны Роем Голдфингером и выпущены в 1955 году. Данная программа настолько набрала популярность за достаточно короткое время, что другие компании стали выпускать продукты, похожие на этот. А Университет штата Пенсильвания разработал компилятор "Dual Autocoder Fortran Translator", который помог написать строки инструкций автокодера. Данный автокодер мог свободно чередоваться со строками программы, что оптимизировало скорость внутренних циклов, а также позволяло кодировать на уровне символьных машин [[13](#тринадцатая)].

Пример программы, с использованием Autocoder: см. рисунок 4

# Глава 4. «Мёртвые» языки программирования и почему они стали такими

Для начала предлагаю ознакомиться с понятием «Мёртвый язык программирования»:

**Мёртвый язык программирования** – такой язык программирования, который перестал пользоваться популярностью, но остался в истории программирования.

Но просто так язык не может перестать пользоваться популярностью, соответственно, существуют какие-либо причины.

Приведу список самых распространённых причин для исчезновения языков программирования:

1. Существует другой язык с большим количеством совместимостей с различными компиляторами.
2. Сложная структура языка. Другие языки могут быть намного проще.
3. Отсутствие «материальной подушки». Чтобы популяризировать новый язык программирования требуется немало средств. Ведь для существования языка какое-то время нужно, чтобы он:

* широко поддерживался разработчиками компиляторов и производителями ПК
* был полезным для значительного класса приложений
* использовался разработчиками ПО.

1. Доступны для узкого круга пользователей и др.

4.1 Языки, которые подвержены исчезновению в ближайшее десятилетие

Существуют различные мнения о том, какие же языки скоро перестанут быть так широко использованы, как сейчас. Это связано с тем, что у каждого пользователя имеется свой «любимый» и «нелюбимый» язык программирования.

Я составила свой топ-3 языков, которые могут исчезнуть в ближайшие годы (опиралась я на рейтинг Stack Overflow на 2021 год):

1. F#. Во-первых, он имеет малочисленную аудиторию пользователей. Во-вторых, обладает слабой поддержкой мобильных платформ.
2. Elixir. Основной причиной спада и до того невысокой популярности является то, что существует Ruby-язык, появившийся раньше и уже набравший большую популярность и немалую группу пользователей.
3. PHP. Основная причина – слабая безопасность. К тому же данный язык появился на рынке в 1995 году, на ряду с ним вышли другие языки программирования, которые превосходили его: Python, Java, JavaScript и др.

4.2«Мёртвые» языки программирования

«Мёртвых» языков программирования не так много. Поподробнее расскажу о языках, которые принесли в развитие программирования весомый вклад. Познакомимся с их историей, вкладом, который они принесли и историей исчезновения.

**Cobol** (см. рисунок 5)

История: был разработан в 1959 году, а выпущен лишь через 10 лет, в 1969 году комитетом CODASYL. Он стал первым компьютерным языком, благодаря которому компьютеры можно стало активно использовать в повседневной жизни.

Вклад: Данная заслуга во многом принадлежит обычному человеку, работающему в банке. Некий Томас рос в семье, где отец работал плотником. Но данными навыками мальчик не обладал, поэтому мать предложила обучиться популярному на тот момент языку – Cobol, что и сделал юноша. Далее он начинает работать в банке в Торонто, с каждым днём всё больше и больше программируя на перфокартах коды для транзакций. Таким образом, проработав более 20 лет, ему удалось написать около десятков миллионов строк кода.

Многие банки до сих пор пользуются кодом Томаса, который позволяет правильно подсчитать сумму транзакций [[14](#четырнадцатая)].

Причина исчезновения: Во-первых, данный язык в большинстве своём был создан для бизнес сферы, соответственно, круг пользователей небольшой. Во-вторых, Cobol-очень сложный язык, из-за чего компиляторы Cobol нуждались в еще более мощной вычислительной мощности, чем могли дать на тот момент компьютеры.

**Basic** (см. рисунок 6)

История: история создания данного языка берет своё начало еще с XIX. Английскому миссионеру пришла в голову идея создать язык на основе английского, вычленив оттуда самые простые и нужные слова. Такой язык создавался для туземцев и иммигрантов, потому что он и правда очень легко запоминался. Таким образом в XIX веке появляется новый язык – Basic English. В 1964 году Томас Курц и Джон Камени решают, что такой же простой язык должен существовать и для программирования. Поэтому на свет выходит язык программирования с названием BASIC (Beginner's All—purpose Symbolic Instruction Code»). Достаточно быстро он завоевал место в сердцах многих программистов. Со временем данный язык программирования постоянно обновляется, у него выходят новые модели [[15](#пятнадцатая)].

Вклад: BASIC стал первым языком программирования доступным для всех, кто хотел попробовать себя в сфере программирования. Кроме того, он оказал техническое влияние, ведь это первый язык, где можно было производить вычисления в режиме реального времени.

Причина исчезновения: Нужно помнить о том, что BASIC – это простота. Поэтому «настоящие» программисты не воспринимали его, как язык для программирования. В большей мере им пользовались студенты и школьники. Но несмотря на это, он всё же был популярным. BASIC стоял долгое время на одном уровне с С#, потому что они были очень похожи. Но постепенно С# начал продвигаться вперед, в то время как BASIC стоял на месте. Таким образом, этот язык программирования начал угасать [[16](#шестнадцатая)].

**APL** (см. рисунок 7)

История: APL (или А Programming Language) был создан в 1957 году профессором Гарварда Кеннетом Айверсоном. Изначально, он разрабатывался для описания алгоритмов прикладной математики. Позже, при переходе Кеннета в компанию IBM язык был рассмотрен как язык программирования и к тому времени APL уже успел обрести заинтересованность у многих. Впоследствии он модернизировался и создавались новые версии (APL2; Dyalog APL; APL/700 и пр.)

Вклад: Основная заслуга языка заключается в том, что у него есть инструменты для работы с массивами, что в разы облегчило жизнь его пользователям. Плюс, APL стал фундаментом для языков программирования, которые выходили после него.

Причина исчезновения: Так как язык предназначался для пользования в сфере прикладной математики, то он содержал в себе множество математических символов, которых нет на стандартной клавиатуре. Поэтому IBM выпустила свою клавиатуру. Она была предназначена только для программирования на APL. Это и стало явной причиной «гибели» языка [[17](#семнадцатая)].

**CLU** (см. рисунок 8)

История: Концепт создания данного языка очень прост: Барбара Лисков совместно с её учениками захотела принести в мир новую идею, связанную с абстрактными типами данных. Реализовать это получилось с помощью языка программирования CLU.

Вклад: Данный язык послужил толчком для реализации множества последующих языков программирования. А именно: C++; Python; Ruby; Perl; Sather; Lua.

Причина исчезновения: Она проста: ведь этот язык программирования создавался лишь для преподнесения новой идеи, не более [[18](#восемнадцатая)].

**Smalltalk** (см. рисунок 9)

История: Smalltalk был разработан и выпущен компанией Xerox PARC. Изначально он использовался как прототип тех языков программирования, которые участвуют в нашем обиходе сейчас. Smalltalk стал первым языком с объекто-ориентированным программированием, который обрёл популярность [[19](#девятнадцатая)].

Вклад: Именно Smalltalk ввел тенденция ООП, что является огромной заслугой.

Причина исчезновения: Кроме того, что у языка существовали свои недочёты, его попросту сместил вышедший на свет Java [[20](#двадцатая)].

# Заключение

Программирование берет своё начало с XIX века. Тогда оно было «закрытой книгой», которую впервые начали познавать Чарльз Бэббидж и Ада Лавлейс.

Сейчас программирование является одной из ведущих сфер деятельности. Без него не могло бы существовать множество других сфер.

В рамках данного проекта программирование было рассмотрено мною с точки зрения истории и развития его главных составляющих.

Понятие «мёртвые» языки программирования» достаточно абстрактная. Об этом ведутся постоянные дискуссии, потому что каждый пользователь имеет свой сформированный круг языков, который он применяет в своей работе. При том, хочу сказать то, что существуют такие компании и предприятия, которые продолжают работать на них. Соответственно, можно сделать вывод о том, что нет полностью «мёртвых» языков, есть лишь забытые многими.

# Список используемой литературы

1. GeekBrains Роль программирования в современном мире / GeekBrains [Электронный ресурс] // ЯндексДзен: [сайт]. — URL: https://dzen.ru/media/geekbrain/rol-programmirovaniia-v-sovremennom-mire-5ebd1ebc007045124bbc33ef (дата обращения: 06.10.2022).
2. А.А. Тюгашев. Основы программирования. Часть I. – СПб: Университет ИТМО, 2016. – 160 с.// с. 9-10

1. [Большая советская энциклопедия](https://ru.wikipedia.org/wiki/%D0%91%D0%BE%D0%BB%D1%8C%D1%88%D0%B0%D1%8F_%D1%81%D0%BE%D0%B2%D0%B5%D1%82%D1%81%D0%BA%D0%B0%D1%8F_%D1%8D%D0%BD%D1%86%D0%B8%D0%BA%D0%BB%D0%BE%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F" \l "%D0%A2%D1%80%D0%B5%D1%82%D1%8C%D0%B5_%D0%B8%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5" \o "Большая советская энциклопедия) : [в 30 т.] / гл. ред. [А. М. Прохоров](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%85%D0%BE%D1%80%D0%BE%D0%B2,_%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80_%D0%9C%D0%B8%D1%85%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D0%B8%D1%87). — 3-е изд. — М. : Советская энциклопедия, 1969—1978.
2. ГОСТ Р 60.0.0.4-2019/ИСО 8373:2012. Роботы и робототехнические устройства. Термины и определения
3. ГОСТ 19781-90. Обеспечение систем обработки информации программное. Термины и определения
4. Н. Д. Арутюнова Большая российская энциклопедия - электронная книга / Н. Д. Арутюнова [Электронный ресурс] // Большая российская энциклопедия : [сайт]. — URL: https://bigenc.ru/linguistics/text/4924604 (дата обращения: 21.10.2022).
5. История создания программирования / [Электронный ресурс] // ISTMIRA : [сайт]. — URL: https://www.istmira.com/drugoe-razlichnye-temy/15323-istorija-sozdanija-programmirovanija.html (дата обращения: 10.11.2022).
6. Александр Неверов Чарльз Бэббидж - портрет, биография, личная жизнь, причина смерти, вычислительная машина / Александр Неверов [Электронный ресурс] // 24СМИ : [сайт]. — URL: https://24smi.org/celebrity/26360-charlz-bebbidzh.html (дата обращения: 10.11.2022).
7. Рекомендуемая форма библиографической ссылки: Луховицкая Э.С. Ада и Бэббидж //Препринты ИПМ им. М.В.Келдыша. 2016 № 143 15 с. doi:10.20948/prepr-2016-143 — URL: http://library.keldysh.ru/preprint.asp?id=2016-143
8. И. А. Казакова Plankalkül-первый высокоуровневый язык программирования. // Прикладная информатика.. - 2012. - №5(41). - С. 128-129.
9. История появления Ассемблера // БазаЗнаний URL: http://www.physics.altspu.ru/community/wiki/DmitriyPylkov (дата обращения: 11.11.2022).
10. Short Code // Национальная библиотека им. Н. Э. Баумана Bauman National Library URL: https://ru.bmstu.wiki/Short\_Code (дата обращения: 11.11.2022).
11. Autocoder - Autocoder // ВикибриФ URL: https://ru.wikibrief.org/wiki/Autocoder (дата обращения: 12.11.2022)
12. PatientZero COBOL - древний код, который управляет вашими деньгами / PatientZero [Электронный ресурс] // Хабр : [сайт]. — URL: https://habr.com/ru/post/532554/ (дата обращения: 16.11.2022).
13. semen\_grinshtein История языков программирования: от BASIC к Visual Basic / semen\_grinshtein [Электронный ресурс] // Хабр : [сайт]. — URL: https://habr.com/ru/post/315470/ (дата обращения: 18.11.2022).
14. PatientZero Расцвет и упадок Visual Basic / PatientZero [Электронный ресурс] // Хабр : [сайт]. — URL: https://habr.com/ru/post/456692/ (дата обращения: 18.11.2022).
15. Elena\_sm Язык программирования APL / Elena\_sm [Электронный ресурс] // Itnan.ru : [сайт]. — URL: https://itnan.ru/post.php?c=1&p=275177 (дата обращения: 18.11.2022).
16. ВикибриФ CLU (язык программирования) / ВикибриФ [Электронный ресурс] // ВикибриФ : [сайт]. — URL: https://ru.wikibrief.org/wiki/CLU\_(programming\_language) (дата обращения: 18.11.2022).
17. Smalltalk / [Электронный ресурс] // Национальная библиотека им. Н. Э. Баумана Bauman National Library : [сайт]. — URL: https://ru.bmstu.wiki/Smalltalk#.D0.A1.D0.B8.D0.BD.D1.82.D0.B0.D0.BA.D1.81.D0.B8.D1.81\_.D1.8F.D0.B7.D1.8B.D0.BA.D0.B0\_Smalltalk (дата обращения: 18.11.2022).
18. Hexlet: о программировании и IT 10 самых влиятельных языков программирования / Hexlet: о программировании и IT [Электронный ресурс] // Яндекс.Дзен : [сайт]. — URL: https://dzen.ru/media/hexlet/10-samyh-vliiatelnyh-mertvyh-iazykov-programmirovaniia-609282cca38d215d4ea79da0 (дата обращения: 18.11.2022).

# Приложение

Рисунок 1(https://ai2-s2-public.s3.amazonaws.com/figures/2017-08-08/56c54595bb26e00c4870266466583aaf8ed986fb/6-Figure3-1.png).

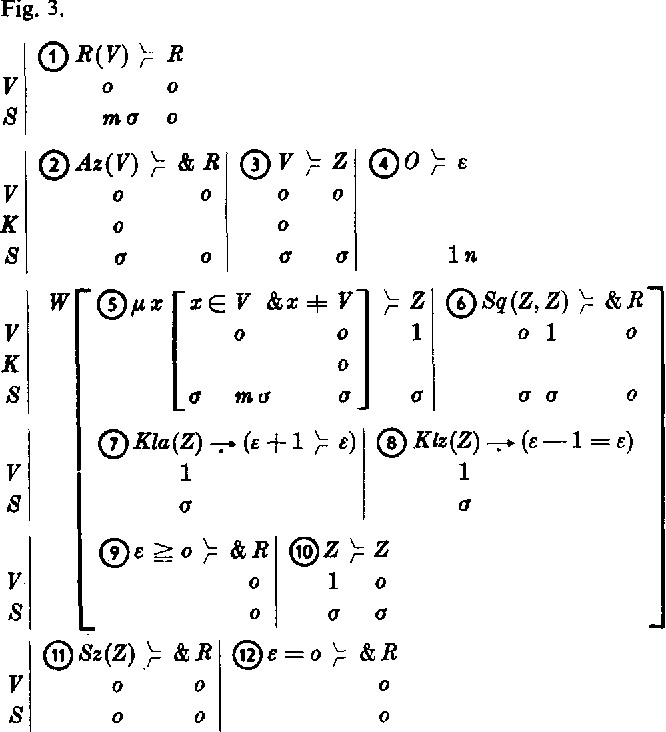


Рисунок 2(https://mypresentation.ru/documents\_6/01dd77a09e03833c81e29c9a68491cd4/img3.jpg).



Рисунок 3(<https://blog.sucuri.net/wp-content/uploads/2017/03/wpxss1.jpg>).

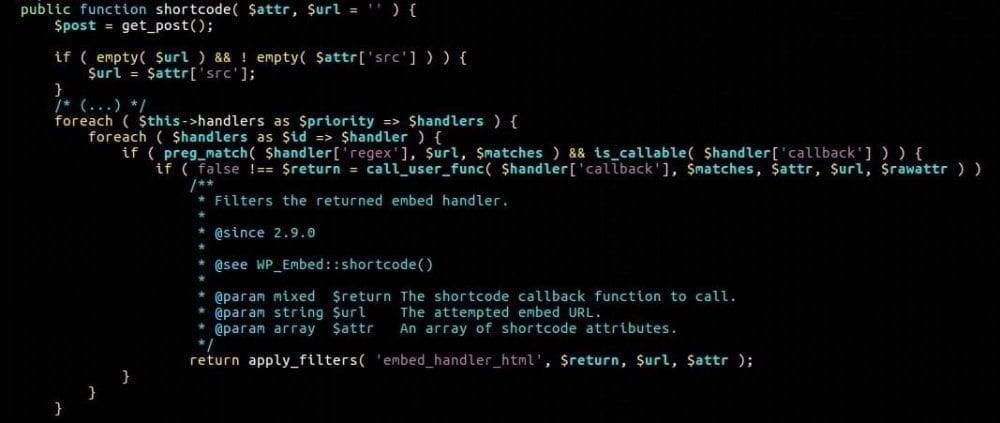


Рисунок 4(<https://en.wikipedia.org/wiki/Autocode>).

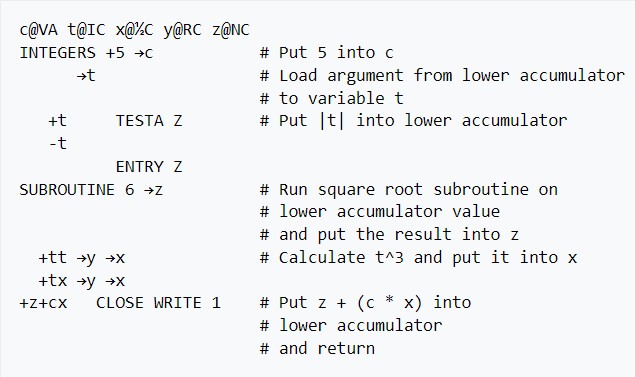


Рисунок 5(hhttps://www.compgramotnost.ru/wp-content/uploads/2018/09/COBOL.jpg).

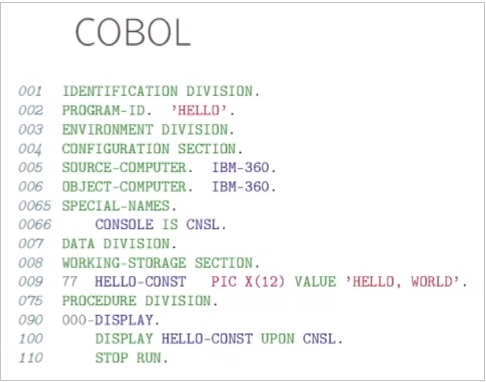


Рисунок 6(<https://toto-school.ru/800/600/https/ds04.infourok.ru/uploads/ex/1041/000678e0-f8078bb7/img33.jpg>).

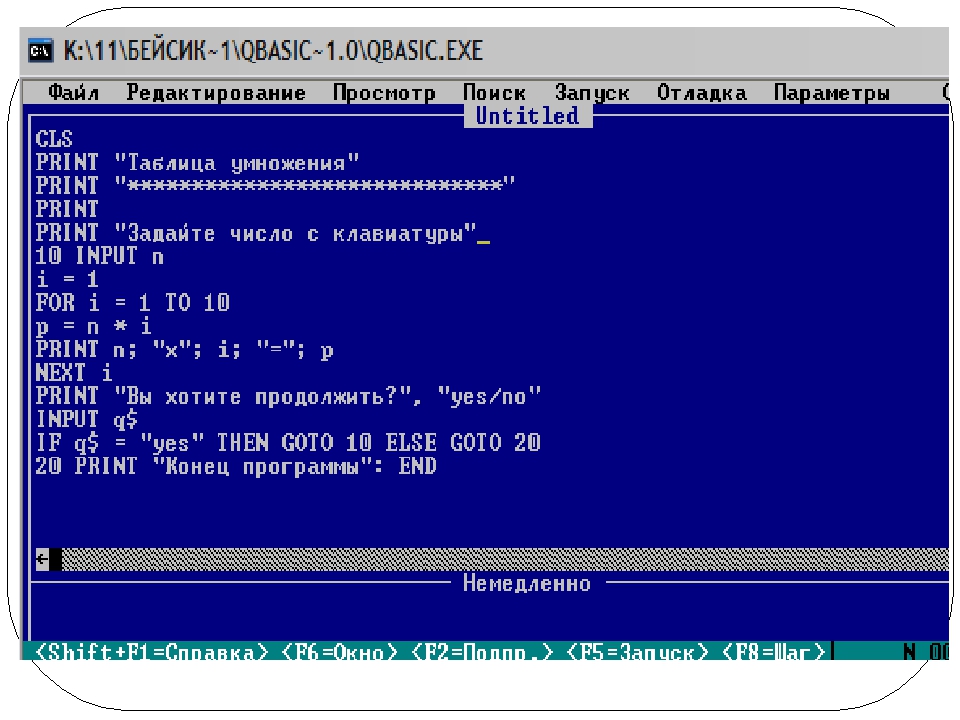


Рисунок 7(https://img-lb.fireden.net/sci/image/1638/30/1638306105315.jpg).

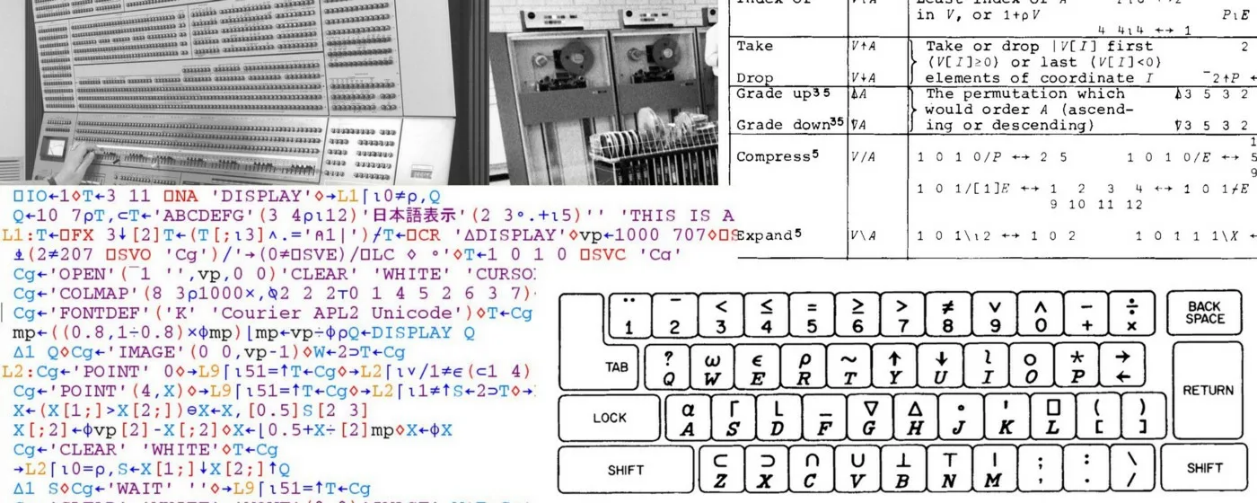


Рисунок 8(https://ru.wikipedia.org/wiki/Клу).

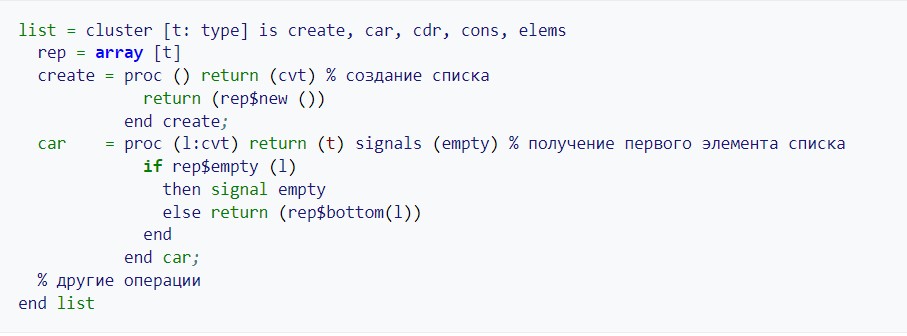


Рисунок 9(https://cf.ppt-online.org/files/slide/s/sNevZQUm534GucKL7Dw9Cx6Y1TznB2hoPyOtiq/slide-4.jpg).

