

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ

Кафедра «**АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ**»

«История развития языков программирования»
Вариант 15

Выполнил:
студент гр. АСОИЗ 201
Лижбанов Владислав Сергеевич

Проверила:

Могилев, 2022

Содержание

Введение.....	3
Основные этапы исторического развития языков программирования.....	5
Разнообразие языков программирования.....	9
Трансляция.....	11
Заключение.....	13
Список использованных источников.....	14

Введение

Определимся с терминами. Программа - это набор инструкций для конкретного исполнителя (механизма, устройства и т.д.), а язык программирования - знаковая система, предназначенная для записи программ.

Первые программисты появились задолго до компьютеров, с которыми они обычно ассоциируются. Автомат-гуманоид Аль-Джазари (1206 год), ткацкий станок Жаккара (1804) (рис.№1)

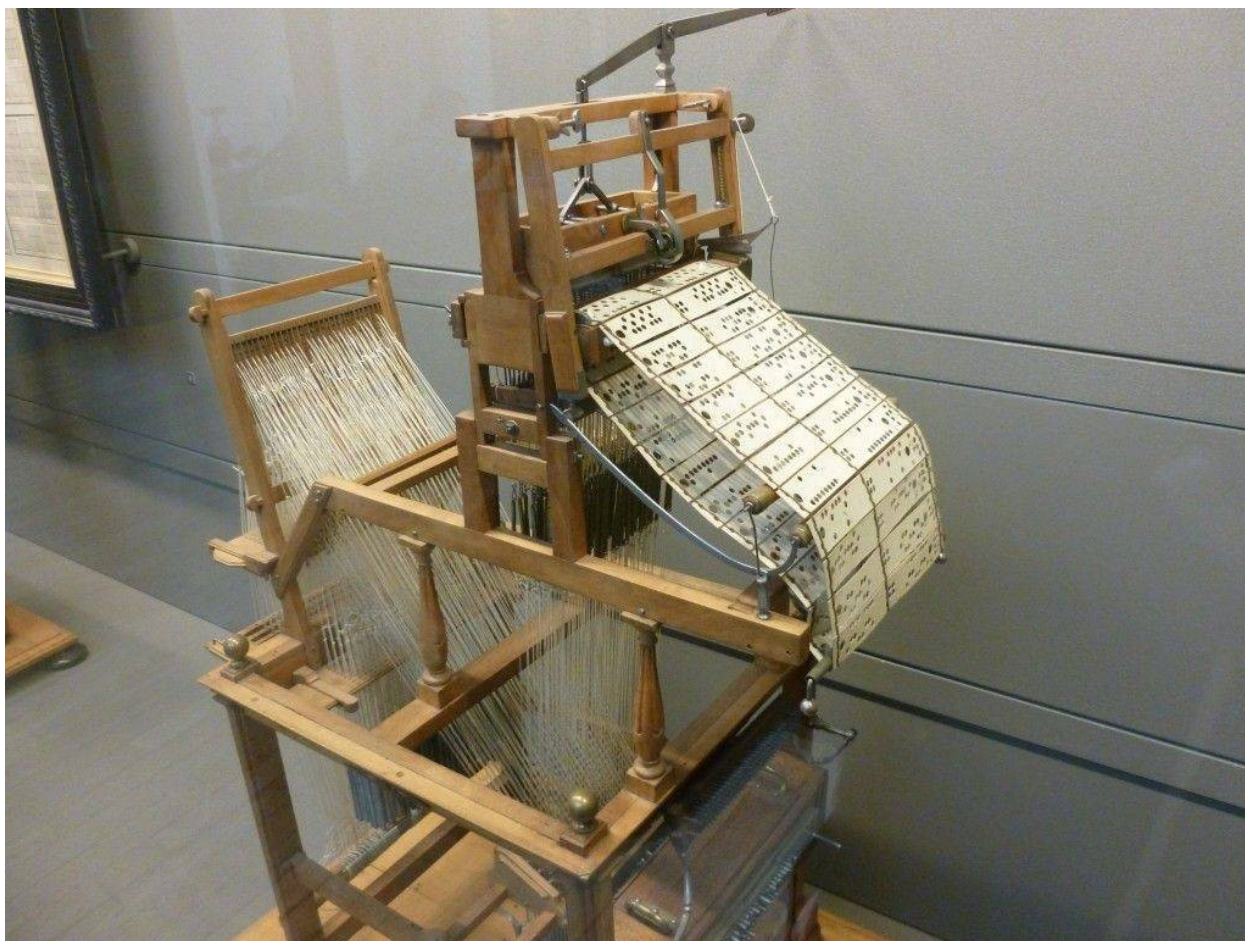


Рисунок №1 Реконструкция ткацкого станка Жаккара

"Аналитическая машина" Бэббиджа (1843) были программируемыми устройствами. Автомату алгоритм действий задавался с помощью кулачков и зажимов, станку и "разностной машине" - с помощью перфокарт. Ада Августа Лавлейс, дочь самого Байрона, даже написала для устройства Бэббиджа (которое он так и не собрал) программу, решавшую уравнение Бернулли.

Программу можно представить как набор последовательных команд, то есть алгоритм, для объекта, то есть исполнителя, который должен их выполнить для достижения определенной цели.

Так можно условно запрограммировать человека, составив для него к примеру инструкцию "как приготовить оладьи", а он начнет четко ей следовать. При этом инструкция, она же программа, для человека будет написана на так называемом естественном языке, например, русском или английском.

Все же программируют не людей, а вычислительные машины, используя при этом специальные языки. Необходимость в особых языках связана с тем, что машины не в состоянии "понимать" наши, то есть человеческие естественные для нас языки.

Инструкции для машин пишут на языках программирования, которые характеризуются формальностью, то есть синтаксической однозначностью (например, в них нельзя менять местами определенные слова) и ограниченностью (имеют строго определенный набор слов и символов).

Основные этапы исторического развития языков программирования

Первые программы писались на машинном языке, так как для ЭВМ того времени еще не существовало развитого программного обеспечения, а машинный язык – это единственный способ взаимодействия с аппаратным обеспечением компьютера, так называемым "железом".

Каждую команду машинного языка непосредственно выполняет то или иное электронное устройство. Данные и команды записывали в цифровом виде, например, в шестнадцатеричной или двоичной системах счисления.

Двоичная система	Шестнадцатеричная система
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Таблица №1: Примеры чисел в шестнадцатеричной или двоичной системах счисления

Человеку воспринимать программу на таком языке сложно. Кроме того, даже небольшая программа состояла из множества строк кода. Ситуация осложнялась еще и тем, что каждая вычислительная машина понимает лишь свой машинный язык. ЭВМ были похожи на огромные шкафы, которые обычно стояли в огромном машинном зале и делали нужные вычисления быстрее, чем на механических или электрических калькуляторах. (Рис №2)

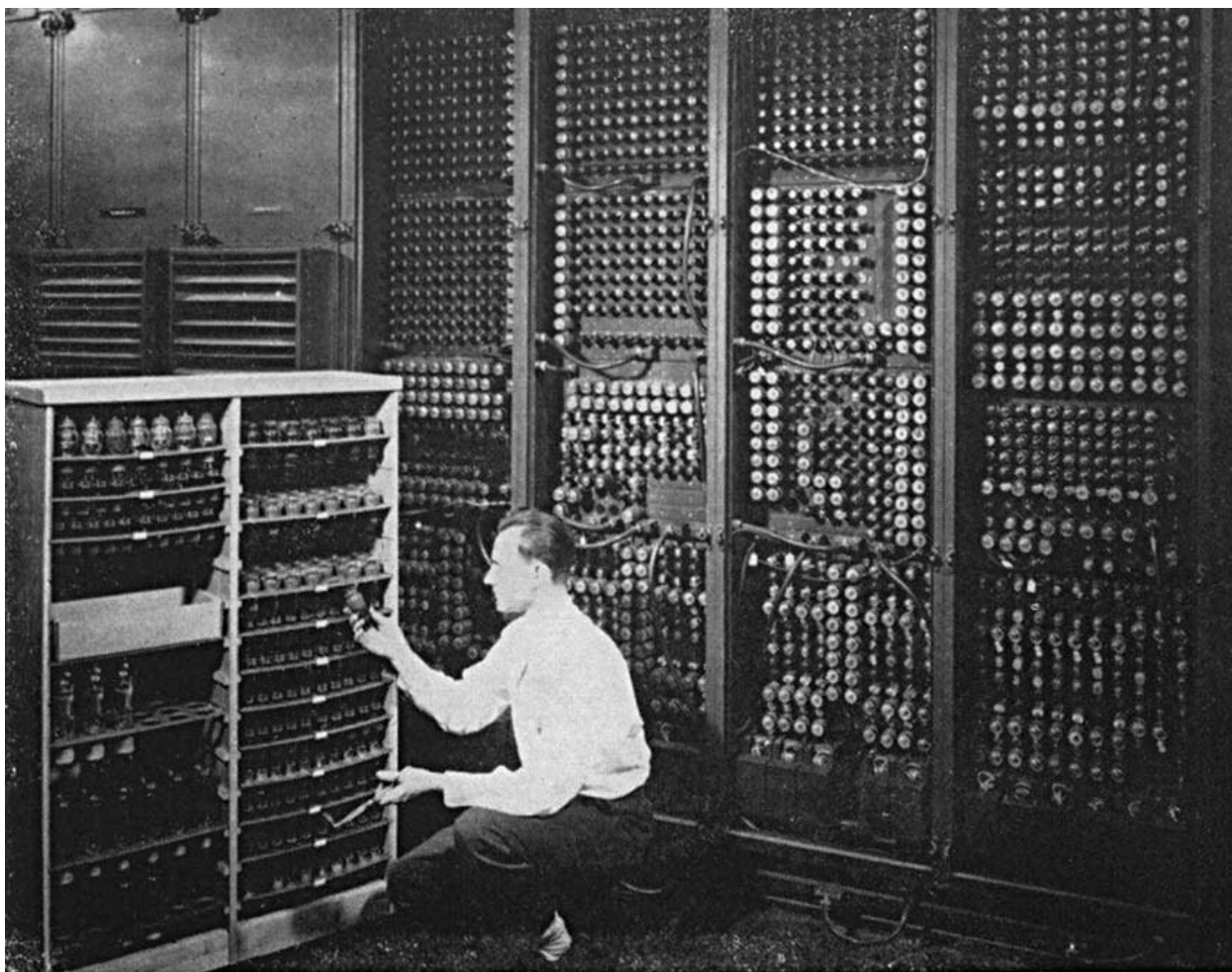


Рисунок №2 Одна из первых вычислительных машин

Людам, в отличие от машин, более понятны слова, чем наборы цифр. Стремление человека оперировать словами, а не цифрами привело к появлению ассемблеров. Это языки, в которых вместо численного обозначения команд и областей памяти используются словесно-буквенные.

При этом появляется проблема: машина не в состоянии понимать слова. Необходим какой-нибудь переводчик на ее родной машинный язык. Поэтому, начиная со времен ассемблеров, под каждый язык программирования создаются трансляторы – специальные программы, преобразующие программный код с языка программирования в машинный код. Ассемблеры на сегодняшний день продолжают использоваться. В системном программировании с их помощью создаются низкоуровневые интерфейсы операционных систем, компоненты драйверов.

После ассемблеров наступил расцвет языков так называемого высокого уровня, такие как:

1. C++
2. C#
3. Delphi
4. Fortran
5. Java
6. JavaScript

Для них потребовалось разрабатывать более сложные трансляторы, так как языки высокого уровня куда больше удобны для человека, чем для вычислительной машины.

В отличие от ассемблеров, которые остаются привязанными к своим типам машин, языки высоко уровня обладают переносимостью. Это значит, что, написав один раз программу, программист без последующего редактирования может выполнить ее на любом компьютере, если на нем установлен соответствующий транслятор. Программа-транслятор для данной ЭВМ при трансляции исходного кода сама адаптирует его под эту ЭВМ.

Следующим значимым шагом было появление объектно-ориентированных языков, что в первую очередь связано с усложнением разрабатываемых программ.

Покол - ения	Языки программирования	Характеристика
1	Машинные	Ориентированы на использование в конкретной ЭВМ, сложны в осознании, требуют хорошего знания архитектуры ЭВМ
2	Ассемблеры, Макроассемблеры	Более удобны для использования, но по-прежнему машинно-зависимы
3	Языки высокого уровня	Мобильные, человеко- ориентированные, проще в освоении
4	Непроцедурные, объектно-Ориентированные, языки запросов, параллельные	Ориентированы на непрофессионального пользователя и на ЭВМ с параллельной архитектурой
5	Языки искусственного интеллекта, экспертных систем и баз знаний, естественные языки	Ориентированы на повышение интеллектуального уровня ЭВМ и интерфейса с языками

Таблица №2: Эволюция языков программирования

С помощью таких языков программист как бы управляет виртуальными объектами. Мыслить в рамках объектов-сущностей, описывать их взаимодействие, обобщать объекты в классы и устанавливать между ними наследственные связи, – все это делает программу по-своему похожей на реальный мир, на то, как его воспринимает человек.

На сегодняшний день в большинстве случаев реализация крупных проектов осуществляется с помощью объектно-ориентированных возможностей языков.

Основные принципы структурирования в случае ООП связаны с различными аспектами базового понимания предметной задачи, которое требуется для оптимального управления соответствующей моделью:

1. **абстракция** для выделения в моделируемом предмете важного для решения конкретной задачи по предмету
2. **инкапсуляция** для быстрой и безопасной организации собственно иерархической управляемости
3. **наследование** для быстрой и безопасной организации родственных понятий
4. **полиморфизм** для определения точки, в которой единое управление лучше распараллелить или наоборот — собрать воедино.

То есть фактически речь идёт о прогрессирующей организации информации согласно первичным семантическим критериям: «важное/неважное», «ключевое/подробности», «родительское/дочернее», «единое/множественное». Прогрессирование, в частности, на последнем этапе даёт возможность перехода на следующий уровень детализации, что замыкает общий процесс.

Хотя существуют и другие современные парадигмы программирования, поддерживаемые другими или теми же языками.

Разнообразие языков программирования

В настоящее время существует множество различающихся и похожих между собой языков программирования. Причина такого явления становится понятна, если представить то количество и разнообразие задач, которые на сегодняшний день решаются с помощью вычислительной техники. Для решения разных задач требуются разные инструменты, то есть разные языки и подходы к программированию. (Рис.№3)

Какой язык используете сейчас для работы

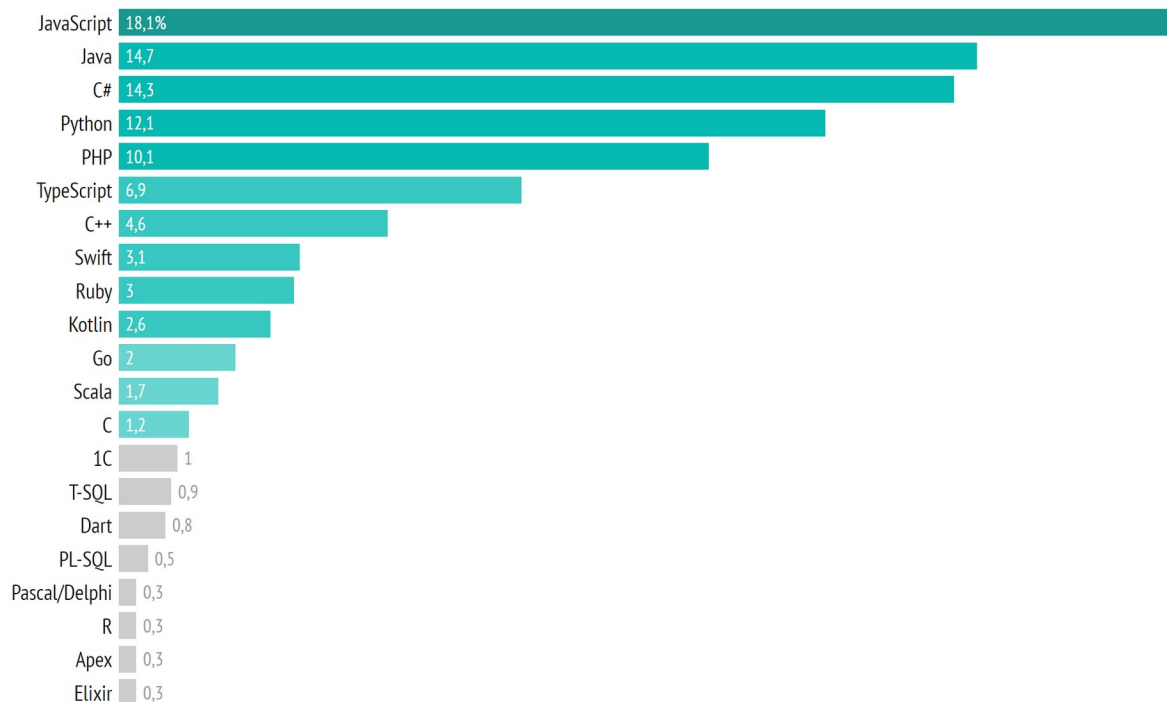


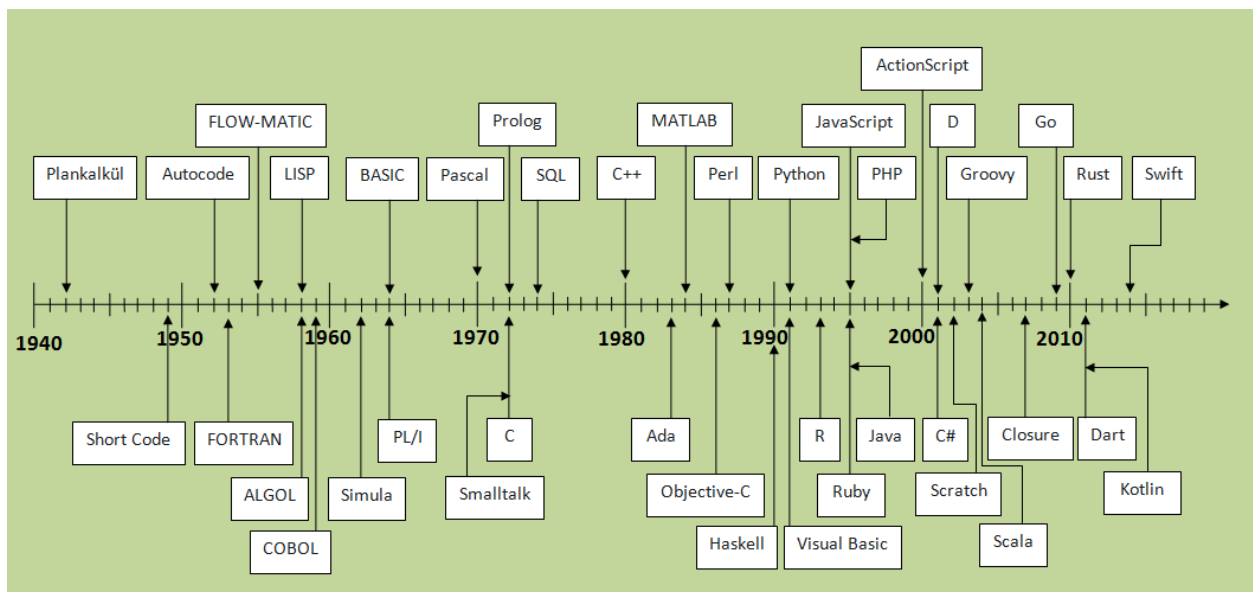
Рисунок №3 Популярность использования разных языков программирования в процентном соотношении

Разработка новых языков программирования, обладающий теми или иными преимуществами, велась как в прошлом, так и ведется сейчас. Эволюционируют, подстраиваясь под запросы нового времени, и старые языки программирования.

Все многообразие языков можно классифицировать по разным критериям. Например, по типу решаемых задач:

1. языки системного или прикладного назначения,
2. языки для web-разработки
3. организации баз данных
4. разработки мобильных приложений

Среди наиболее популярных на сегодняшний день можно отметить Java, C, C++, C#, JavaScript, PHP, и в том числе Python. Так же есть менее популярные языки, которые приведены на (рис.№4)



Рисунок№4 Временная ветка создания языков программирования

Трансляция

Ранее было сказано, что для перевода кода с языка программирования высокого уровня на машинный язык требуется специальная программа – транслятор.

(Рис. №5)

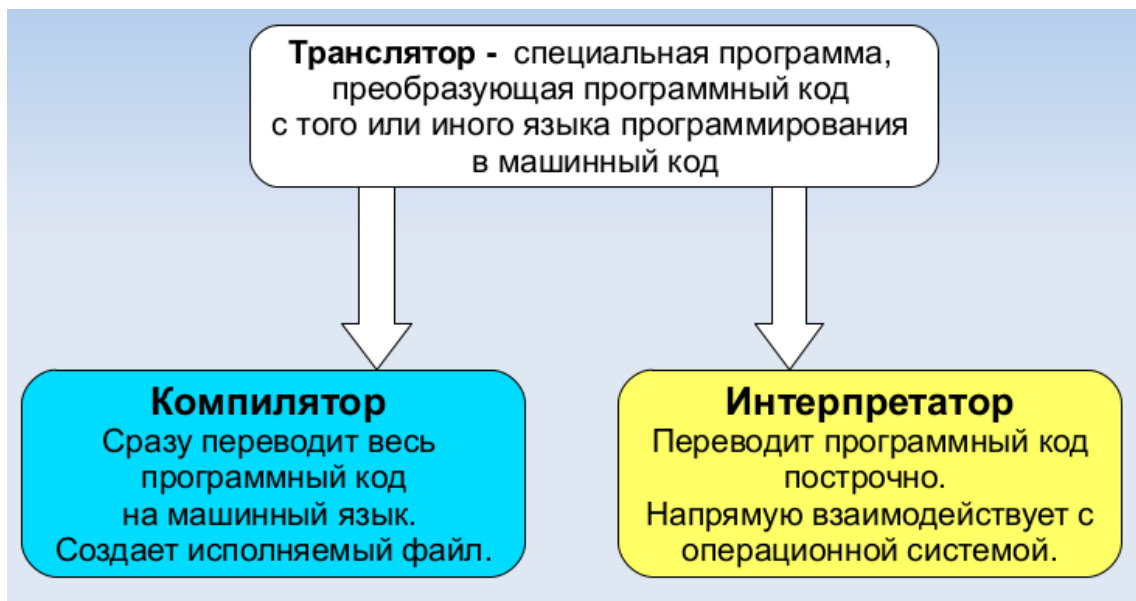


Рисунок №5 Схема действия Транслятора

Заложенный в транслятор алгоритм такого перевода сложен. При этом существует два основных способа трансляции — компиляция программы или ее интерпретация.

При компиляции весь исходный программный код (тот, который пишет программист) сразу переводится в машинный. Создается так называемый отдельный исполняемый файл, который никак не связан с исходным кодом. Выполнение исполняемого файла обеспечивается операционной системой (ОС). После того как получен исполняемый файл, для его чтения транслятор уже не нужен.

При интерпретации выполнение кода происходит последовательно (условно можно сказать, строка за строкой). Грубо говоря, операционная система взаимодействует с интерпретатором, а не с файлом, содержащим программный код. Интерпретатор же, прочитав очередную часть исходного кода, переводит его в машинный (или не совсем машинный, но "понятный" для ОС) и "отдает" его ОС.

ОС исполняет этот код и ждет следующей "подачки" от интерпретатора. Питон именно такой язык. Он интерпретируемый язык программирования.

Выполнение откомпилированной программы происходит быстрее, так как она представляет собой готовый машинный код. Однако на современных компьютерах снижение скорости выполнения при интерпретации обычно не заметно. Кроме того, интерпретируемые языки обладают рядом преимуществ, среди которых отсутствие подготовительных действий для исполнения программы, что может быть важным для тех, кто только начинает изучать программирование.

Заключение

Мир программирования огромен.

История языков программирования увлекательна. Кто бы мог подумать, что алгоритм середины 19 века проложит путь для технологического общества, в котором мы живем сегодня. От ранних машинных кодов до сложного, удобночитаемого кода, на котором сегодня работают наши любимые технологии, языки программирования прошли долгий путь.

У языков программирования разные пороги вхождения. Начать работать в Python относительно просто. При выборе языка для изучения исходите из своих целей, знаний и уровня подготовки. Ищите и используйте возможности для образования и самообразования. Это не только курсы, книги и статьи, но и онлайн-тренажеры.

Несомненно, компьютерное программирование продолжит развиваться, как это происходило в течение последних 150 лет, и интересно смотреть, что принесет будущее.

Список использованных источников

1. Germn. Краткая и на 146% точная история языков программирования
- <https://habr.com/ru/post/481822/>
2. Владимир Языков. Языки программирования: для чего нужны, какие популярны, как выбрать и с чего начать изучение- <https://netology.ru/blog/03-2022-programming-languages>
3. История языков программирования
- https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%82%D0%BE%D1%80%D0%B8%D1%8F_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F
4. Какой язык программирования учить прямо сейчас: 9 самых востребованных
- <https://trends.rbc.ru/trends/education/601c1a6b9a79472c4806230ahttps://trends.rbc.ru/trends/education/601c1a6b9a79472c4806230a>
5. Harry McCracken. Fifty Years of BASIC, the Programming Language That Made Computers Personal - <https://time.com/69316/basic/>