



Алексей Постоногов

**Сборник языков
программирования А-я**

«Автор»

2026

Постоногов А. Н.

Сборник языков программирования А-я / А. Н. Постоногов —
«Автор», 2026

Это каталог по языкам программирования. В нем собраны все языки, упорядоченные в алфавитном порядке, что позволит быстро находить нужное для каждого. Приведены ключевые характеристики: год создания, основные парадигмы, типичные сферы применения, сильные и слабые стороны. Каталог охватывает широкий спектр технологий, отражающих эволюцию программирования. Издание будет полезно начинающим разработчикам, которые хотят сориентироваться в многообразии языков, опытным инженерам, подбирающим оптимальный инструмент для задачи, а также преподавателям и исследователям, изучающим историю и тенденции развития программных технологий. Каталог поможет не только выбрать подходящий язык, но и глубже понять принципы работы современных ИТсистем.

Алексей Постоногов

Сборник языков программирования А-я

Вступление.

Прежде чем начать изучать или уже разрабатывать, нужно понять для себя свою цель. Это направление для начала карьеры или задумка для проекта.

Если карьера, то выбираем просто по своему вкусу - по популярности, или сложности, или другим предпочтениям.

А если проект, то лучше выбрать снаряжение сразу максимально подходящее по функционалу и возможностям, чтобы где нибудь в середине спустя пару месяцев не пришлось все переделывать.

В этих страницах лежит краткий обзор на все что вам может пригодиться под любые цели и интересы.

Также важно понимать, большинство языков и инструментов программирования очень похожи, как обычные человеческие речи — английский и английский(США) или Русский и украинский. Поэтому программы почти понимают друг друга, но из-за разниц ошибки неизбежны. Тем более что в мире программирования единственный символ, может стать фатальным для всего проекта.

1

А — он существует в головах тех, кто на нём говорит.

Как народная солянка, рецепт которой совершенствуется по сей день. У этого языка программирования нет исходной даты появления и нет автора: разные преподаватели и инженеры независимо друг от друга придумывали свой мини язык и называли его просто «А». В учебных курсах он служил тренажёром: студенты учились формулировать логику «если X, то Y», не отвлекаясь на сложный синтаксис. В компаниях его брали как простой DSL для бизнес правил («если заказ дороже 10 000, то отправить на проверку»), а в научных прототипах — как черновик логики для агентов и роботов, который потом переносили в рабочий код.

Этот язык не вычисляет — он задаёт правила. Как должностная инструкция для сотрудников или законодательство для граждан: не делает работу сам, а говорит, как её выполнять. В физике он не моделирует падение или энергию, но может описать правила, по которым модель должна существовать: «считать падение завершённым при скорости ниже порога», «выдать предупреждение, если энергия отклонилась больше чем на 5%».

2

А#

Как младший брат C# с очень узкой специализацией. Это не самостоятельный промышленный язык, а экспериментальный или учебный диалект, который часто создавали в академической среде, чтобы «покрутить» отдельные идеи синтаксиса и типизации — и сразу на базе экосистемы .NET.

Его собирали не для того, чтобы писать на нём сервисы или игры, а чтобы отработать концепции: как по-другому описать интерфейсы, как упростить работу с событиями или как сделать строже проверку типов. Отсюда и название — оно сразу отсылает к C#, показывая: «мы играем на той же площадке, но пробуем свои правила».

Где применяют: в учебных проектах, исследовательских прототипах и иногда как внутренний DSL в .NET командах, когда нужно быстро проверить идею без создания полноценного фреймворка. Чего не умеет: не годится для продакшн-систем — у него нет стабильной компиляции, широкой поддержки и готовых библиотек. Это скорее «песочница для идей», чем рабочий инструмент.

3

А++

Как учебный конструктор «собери свой язык»: название нарочно сделано нарочито громким, будто «ещё круче, чем C++», но на деле это не промышленный инструмент, а демонстрационный язык. Его придумали специально, чтобы на паре страниц показать, как вообще устроены разные парадигмы: вот тут у нас кусочек функционального стиля, тут — объектного, а тут — логического.

Появился в академической среде как иллюстрация: преподаватели брали A++ и на его примере показывали студентам: «Смотрите, вот так выглядит рекурсия, а вот так — полиморфизм, и всё это в одном крошечном синтаксисе». Отсюда и стиль — нарочно гиперболизированный, чтобы сразу цеплять взгляд и подчёркивать: «это про идеи, а не про продакшн».

Где применяют: только в лекциях, разборах и статьях про основы языков программирования — как наглядный пример того, как можно упаковать несколько подходов в одну минисистему. Чего не умеет: не годится для реальных задач — у него нет компилятора для продакшна, нет экосистемы и нет цели быть полезным за пределами учебной доски. Это скорее «шпаргалка в форме языка», чем инструмент для работы.

4

ABAP

Как корпоративный язык-долгожитель: родился не ради красоты кода, а чтобы внутри гигантских ERP-систем (вроде SAP) можно было быстро «докрутить» нужную логику под конкретный завод или бухгалтерию. Его придумали как рабочий инструмент для бизнес-задач: «пересчитать зарплату по новому тарифу», «проверить остаток на складе», «сформировать отчёт для налоговой».

Появился в 1980-х вместе с развитием SAP: тогда бизнесу нужны были свои программы, которые не столько пишут алгоритмы, сколько адаптируют систему под процессы компании. Отсюда и специфика: ABAP заточен под работу с таблицами, отчётами и транзакциями — всё максимально «по делу», без лишних абстракций.

Где применяют: в компаниях, где стоит SAP — это бухгалтерия, закупки, логистика, производство. На нём дорабатывают стандартные процессы, пишут отчёты и интеграции. Это не про стартапы и не про мобильные приложения, а про стабильную работу больших корпоративных систем. Чего не умеет: не годится для игр, веб-сервисов или научных расчётов — его сила в связке с SAP и обработке бизнес-данных, а не в универсальных задачах.

5

ABAP++

Как «тюнингованная версия» из гаража: вроде и значок «плюс-плюс» наклеили, и обвес модный прицепили, а по факту это всё тот же рабочий инструмент, только в воображении. Никакого официального релиза, никакого компилятора — это концепт-шутка, учебный пример и ироничная зарисовка в одном флаконе.

Появился в лекциях и статьях как способ наглядно показать: «Смотрите, вот старый ABAP — понятный, строгий, заточенный под отчёты и таблицы. А вот ABAP++ — это если бы мы захотели добавить ему модные абстракции, паттерны и современный синтаксис. Как бы это выглядело и почему половина идей тут вообще не прижилась бы». Преподаватели брали этот «язык» как тренажёр: на его примере студенты разбирали, что из новых подходов реально нужно в корпоративной разработке, а что только усложняет жизнь бухгалтеру, которому нужен простой отчёт.

Где применяют: нигде. Ни в одном реальном проекте, ни в одной компании. Только в учебных разборах, на семинарах и в статьях про эволюцию языков — чтобы на контрасте показать, как меняется подход к бизнес-логике. Чего не умеет: вообще ничего не умеет, потому что не существует как рабочий инструмент. Это не технология, а способ объяснить идею: «вот так мы раньше делали, вот так хотим, а вот это — просто забавный мысленный эксперимент».

6

ABC

Как школьный конструктор из крупных деталей: его специально делали так, чтобы новичок с первых строк понял, что вообще происходит в программировании. Название не про «алфавит», а про простоту: А В С — значит «понятно с первого взгляда».

Появился в 1980 х как альтернатива сложному Basic: авторы хотели показать, что код может быть коротким, логичным и без лишней мишуры — чтобы на паре строк можно было реально что-то посчитать или обработать список. Отсюда и стиль: минимум синтаксического шума, максимум смысла — будто читаешь почти обычный текст.

Где применяют: в основном в обучении и исторических разборах — как пример того, как можно сделать язык предельно ясным. Иногда его вспоминают, когда проектируют новые учебные языки или DSL, где важна читаемость. Чего не умеет: не годится для больших систем, современных веб сервисов или тяжёлых вычислений — он не про мощность, а про наглядность. Это скорее «учебный эталон простоты», чем рабочий инструмент.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.