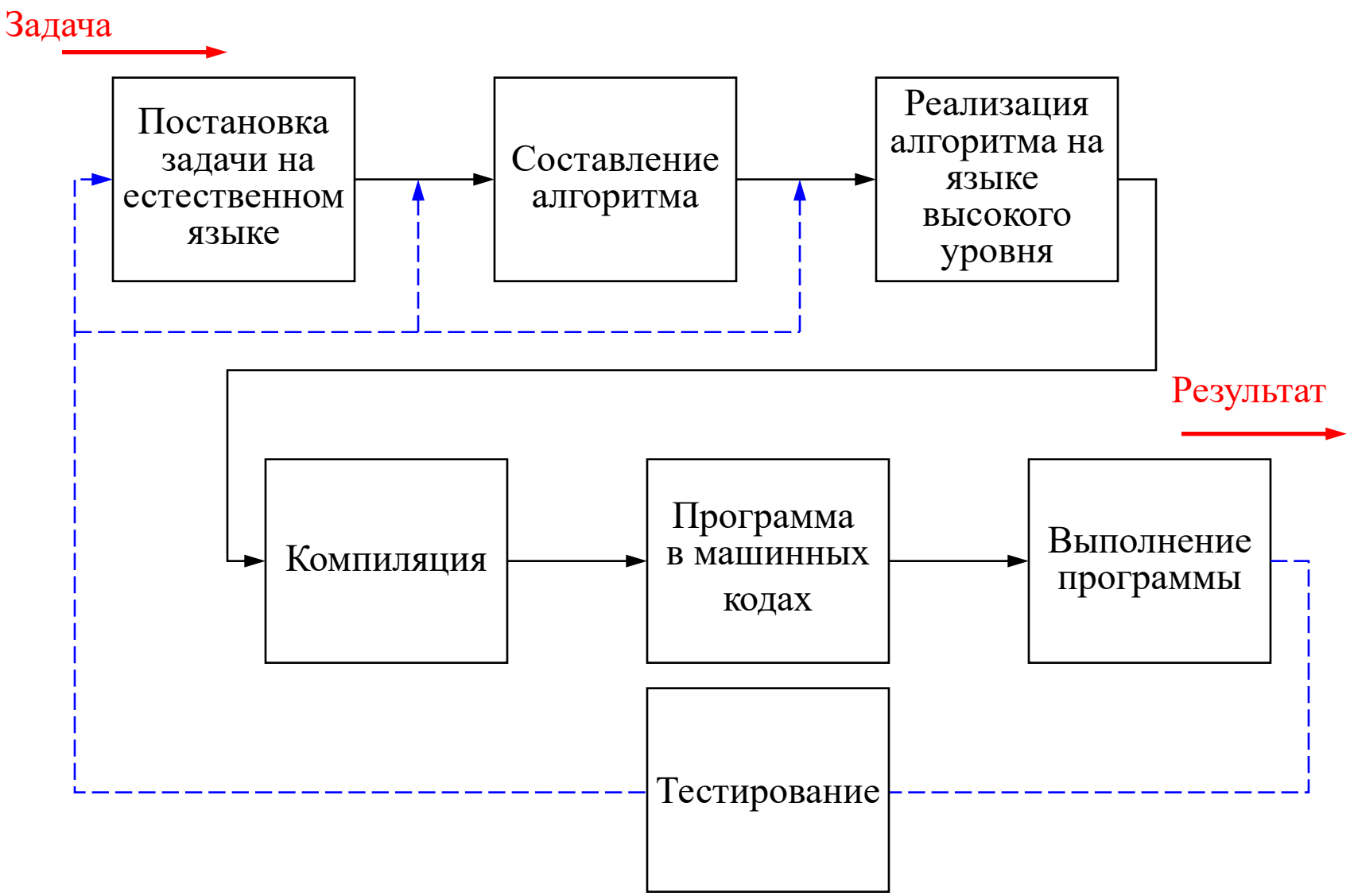
# ЯЗЫКИ ПРОГРАММИРОВАНИЯ

# Этапы реализации задачи программирования

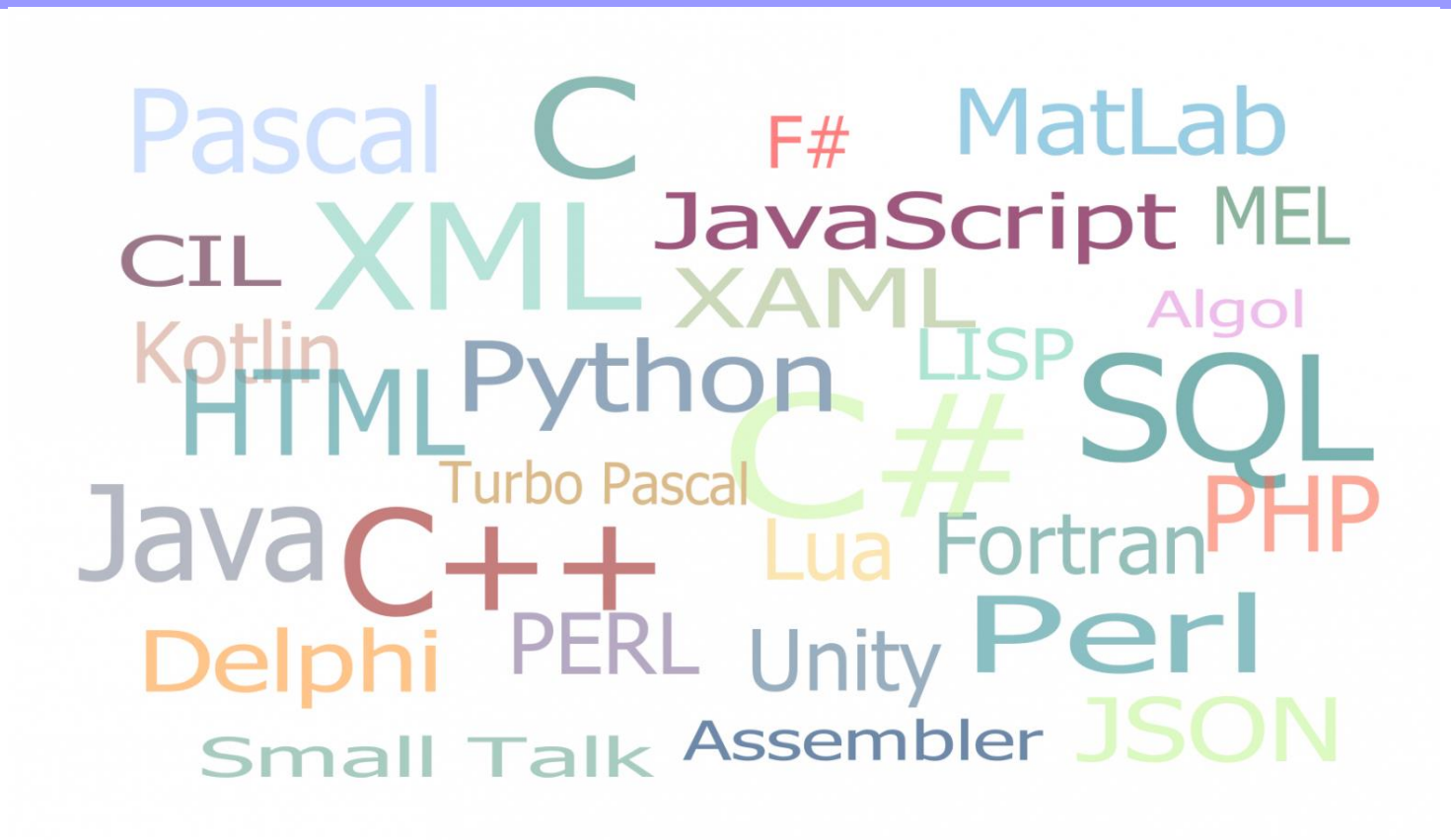


# РАЗВИТИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

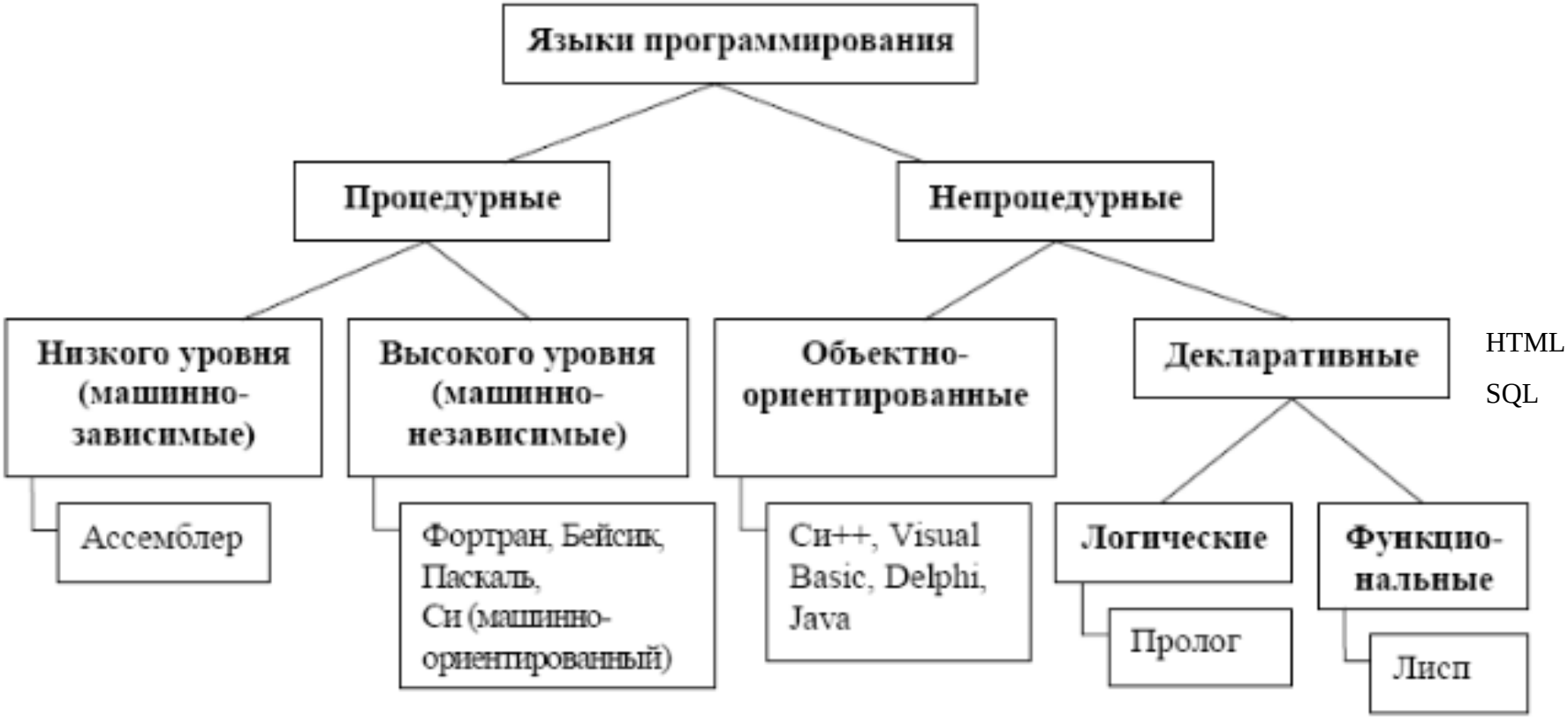
**Язык программирования** — формальный язык, предназначенный для записи компьютерных программ

Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением

~9000



# Классификация языков программирования




HTML

SQL

# Классификация языков программирования

| Компилируемые | Интерпретируемые |
| --- | --- |
| C, C++, Pascal | Visual Basic Script (VBScript), JavaScript, Python, PHP |

Условно компилируемые: C#, .Net, Java для Java-машины

# Классификация языков программирования

**Универсальные**

семейство Pascal/Delphi, C/C++, C#, Java

**Специализированные**

*Математические вычисления:* Fortran, F#
*Математическое моделирование*: MatLab, Wolfram (Mathematica)
*Искусственный интеллект*: LISP,
*На основе передачи сообщений*: Small Talk,
*Многопоточные приложения:* Cw,
*Веб-разработка*: Perl, PHP, JavaScript
*Базы данных*: SQL
*Компьютерные игры*: Lua, Unity, Godot, Twine
*Компьютерная графика*: MEL (Maya), MAX Script (3ds Max)
*Бухгалтерия*: 1С

# Классификация языков программирования

**Алгоритмические**

*Pascal, C++, Java, C#*

**Описания данных**

*XML, XAML, JSON, HTML, DDL SQL*

**Низкоуровневые**

*Assembler*

**Высокоуровневые**

*любой объектно-ориентированный или поддерживающий сложные типы данных язык*

# Показатели энергетической эффективности

| | Energy |
|---|---|
| (c) C | 1.00 |
| (c) Rust | 1.03 |
| (c) C++ | 1.34 |
| (c) Ada | 1.70 |
| (v) Java | 1.98 |
| (c) Pascal | 2.14 |
| (c) Chapel | 2.18 |
| (v) Lisp | 2.27 |
| (c) Ocaml | 2.40 |
| (c) Fortran | 2.52 |
| (c) Swift | 2.79 |
| (c) Haskell | 3.10 |
| (v) C# | 3.14 |
| (c) Go | 3.23 |
| (i) Dart | 3.83 |
| (v) F# | 4.13 |
| (i) JavaScript | 4.45 |
| (v) Racket | 7.91 |
| (i) TypeScript | 21.50 |
| (i) Hack | 24.02 |
| (i) PHP | 29.30 |
| (v) Erlang | 42.23 |
| (i) Lua | 45.98 |
| (i) Jruby | 46.54 |
| (i) Ruby | 69.91 |
| (i) Python | 75.88 |
| (i) Perl | 79.58 |

| | Time |
|---|---|
| (c) C | 1.00 |
| (c) Rust | 1.04 |
| (c) C++ | 1.56 |
| (c) Ada | 1.85 |
| (v) Java | 1.89 |
| (c) Chapel | 2.14 |
| (c) Go | 2.83 |
| (c) Pascal | 3.02 |
| (c) Ocaml | 3.09 |
| (v) C# | 3.14 |
| (v) Lisp | 3.40 |
| (c) Haskell | 3.55 |
| (c) Swift | 4.20 |
| (c) Fortran | 4.20 |
| (v) F# | 6.30 |
| (i) JavaScript | 6.52 |
| (i) Dart | 6.67 |
| (v) Racket | 11.27 |
| (i) Hack | 26.99 |
| (i) PHP | 27.64 |
| (v) Erlang | 36.71 |
| (i) Jruby | 43.44 |
| (i) TypeScript | 46.20 |
| (i) Ruby | 59.34 |
| (i) Perl | 65.79 |
| (i) Python | 71.90 |
| (i) Lua | 82.91 |

| | Mb |
|---|---|
| (c) Pascal | 1.00 |
| (c) Go | 1.05 |
| (c) C | 1.17 |
| (c) Fortran | 1.24 |
| (c) C++ | 1.34 |
| (c) Ada | 1.47 |
| (c) Rust | 1.54 |
| (v) Lisp | 1.92 |
| (c) Haskell | 2.45 |
| (i) PHP | 2.57 |
| (c) Swift | 2.71 |
| (i) Python | 2.80 |
| (c) Ocaml | 2.82 |
| (v) C# | 2.85 |
| (i) Hack | 3.34 |
| (v) Racket | 3.52 |
| (i) Ruby | 3.97 |
| (c) Chapel | 4.00 |
| (v) F# | 4.25 |
| (i) JavaScript | 4.59 |
| (i) TypeScript | 4.69 |
| (v) Java | 6.01 |
| (i) Perl | 6.62 |
| (i) Lua | 6.72 |
| (v) Erlang | 7.20 |
| (i) Dart | 8.64 |
| (i) Jruby | 19.84 |

# СОЗДАНИЕ И РАЗВИТИЕ ЯЗЫКА C++

**C++** - компилируемый, статически типизированный язык программирования общего назначения.

Язык возник в начале 1980-х годов, когда сотрудник фирмы Bell Laboratories Бьёрн Страуструп (1950) придумал ряд усовершенствований к языку C (Си) под собственные нужды.

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

В 1998 году был ратифицирован международный стандарт языка C++: ISO/IEC 14882:1998 «Standard for the C++ Programming Language».

Последняя стабильная на текущий момент версия стандарта - C++20 (анонсирован C++24)

# СОЗДАНИЕ И РАЗВИТИЕ ЯЗЫКА С++

**Принципы Страуструпа, положенные в основу концепции языка:**

• Универсальный язык со статическими типами данных, эффективностью и переносимостью языка С.
• Поддержка процедурного программирование, абстракции данных, объектно-ориентированного программирования и обобщённого программирования.
•Свобода выбора программиста, даже если это даст ему возможность выбирать неправильно.
•Максимальная совместимость с С.
•Отсутствие разночтений между С и С++: любая конструкция, которая допустима в обоих этих языках, должна в каждом из них обозначать одно и то же и приводить к одному и тому же поведению программы.
•Отсутствие особенностей, которые зависят от платформы или не являются универсальными.
•Никакое языковое средство не должно приводить к снижению производительности программ, не использующих его.
•Не требовать слишком усложнённой среды программирования.

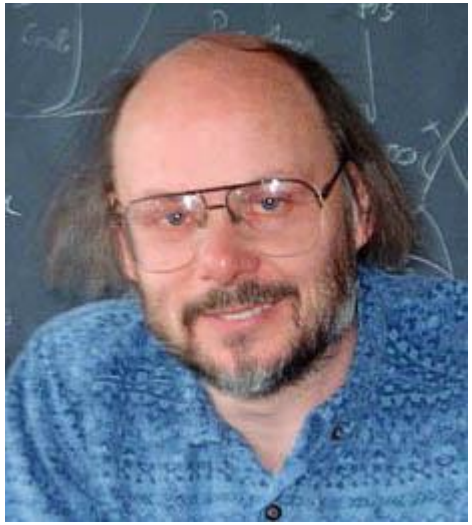
# Сравнение языков семейства C

| Характеристика | C | C++ | C# | Описание |
|---|---|---|---|---|
| Императивный | + | + | + | Императивный язык должен описывать не столько саму задачу, сколько её решение |
| ООП | - | + | + | Использует три парадигмы ООП: наследование, инкапсуляцию и полиморфизм |
| Функциональный | - | -/+ | +/- | Позволяет записывать программу как композицию функций. В чистом функциональном языке нет переменных |
| Рефлексивный | - | - | -/+ | Возможность программы на данном языке оперировать собственным кодом как данными |
| Обобщенное программирование | - | + | + | Обобщенное программирование позволяет записывать алгоритмы, принимающие данные любого типа |
| Логический | - | - | - | Использует логику предикатов для описания баз данных и процедур логического вывода и принятия решений |
| Декларативный | - | - | -/+ | Декларативный язык описывает не столько решение проблемы, сколько саму проблему, а решение уже должен определять компилятор |
| Разделительный | +/- | +/- | -/+ | Содержит специальные конструкции для поддержки распараллеливания программы на несколько компьютеров или процессоров |
| Статическая типизация | + | + | + | Переменные и параметры методов/функций связываются с типами в момент объявления и не могут быть изменены позже |

# Сравнение языков семейства C

| Характеристика | C | C++ | C# | Описание |
|---|---|---|---|---|
| Динамическая типизация | - | -/+ | -/+ | Переменные и параметры методов/функций связываются с типами в момент присваивания значения, а не в момент объявления переменной или параметра. |
| Явная типизация | + | + | + | Типы переменных и параметров указываются явно |
| Неявная типизация | - | -/+ | -/+ | Типы переменных и параметров не указываются явно |
| Поддержка try/catch | - | + | + | Поддержка обработки исключений с помощью try/catch или эквивалентной конструкции |
| Динамические массивы | - | +/- | +/- | Наличие встроенных в язык массивов способных изменять свой размер |
| Контроль границ массивов | - | -/+ | + | Определение допустимости текущего значения индекса |
| Множественное наследование | x | + | - | Возможность наследовать класс сразу от нескольких классов |
| Макросы | -/+ | -/+ | - | Обработка кода программы до времени её компиляции и/или выполнения |
| Перегрузка функций | - | + | + | Возможность перегрузки функций/методов по количеству и типам параметров |
| Значения параметров по умолчанию | - | + | + | Возможность при вызове функции/метода опускать некоторые параметры, чтобы при этом подставлялось значение по умолчанию, указанное при определении функции |

# КЛЯТВА СТРАУСТРУПА



"Я обязуюсь прилежно комментировать свой код, не использовать `goto` и следить за состоянием своих потоков и выделением памяти. Я обязуюсь не оставлять мусора в системе и избегать однобуквенных переменных.

Ресет"