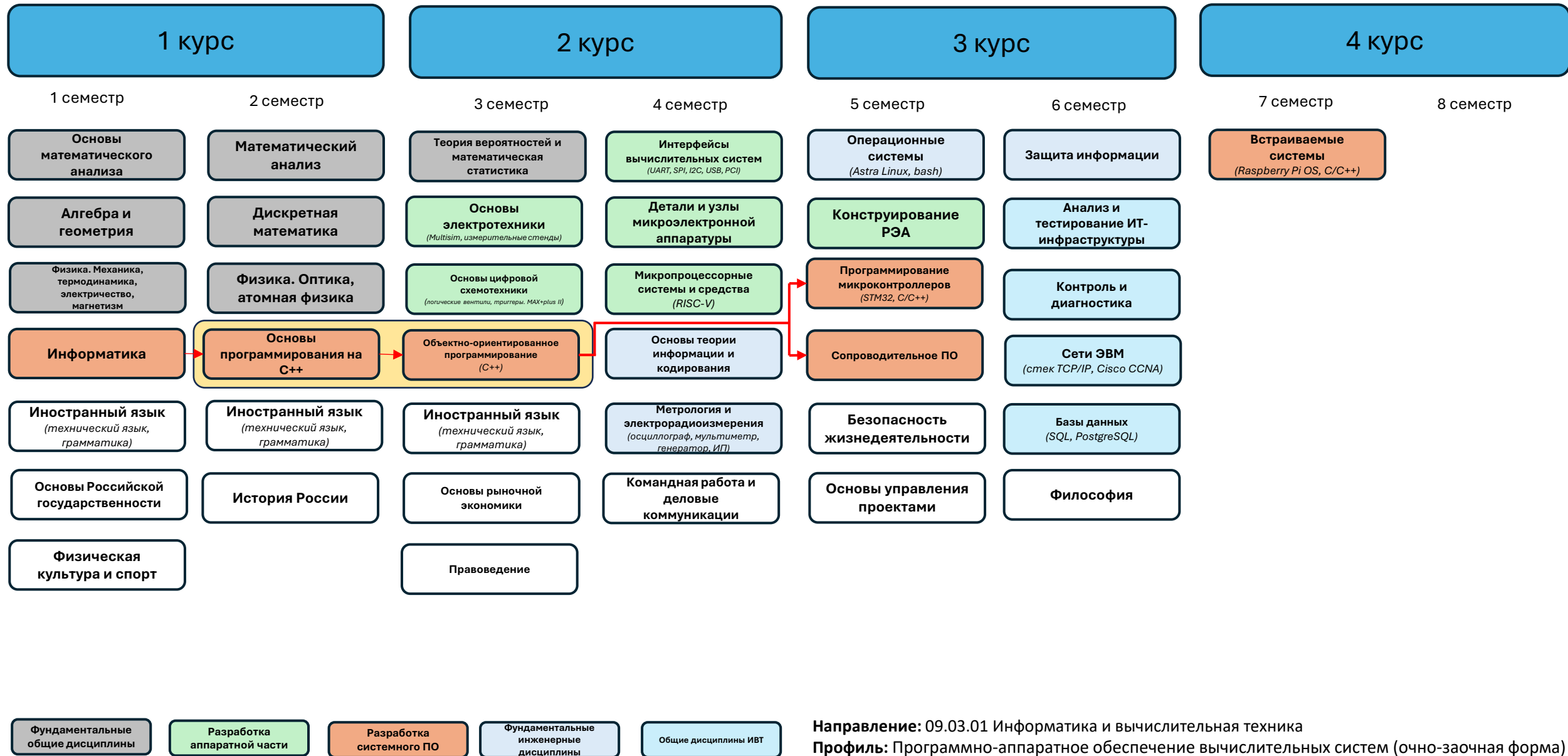




Введение в курс



Учебный план бакалавриата Института МПСУ





Основные определения

Основы программирования на C++

Программирование — процесс создания компьютерных программ с помощью языков программирования.

Язык программирования — формальный язык, предназначенный для записи компьютерных программ

Алгоритм — это последовательность действий, записанная на формальном языке и приводящая к заданной цели за конечное число шагов.



Программирование

Входные данные



Результат





Пример





Программирование



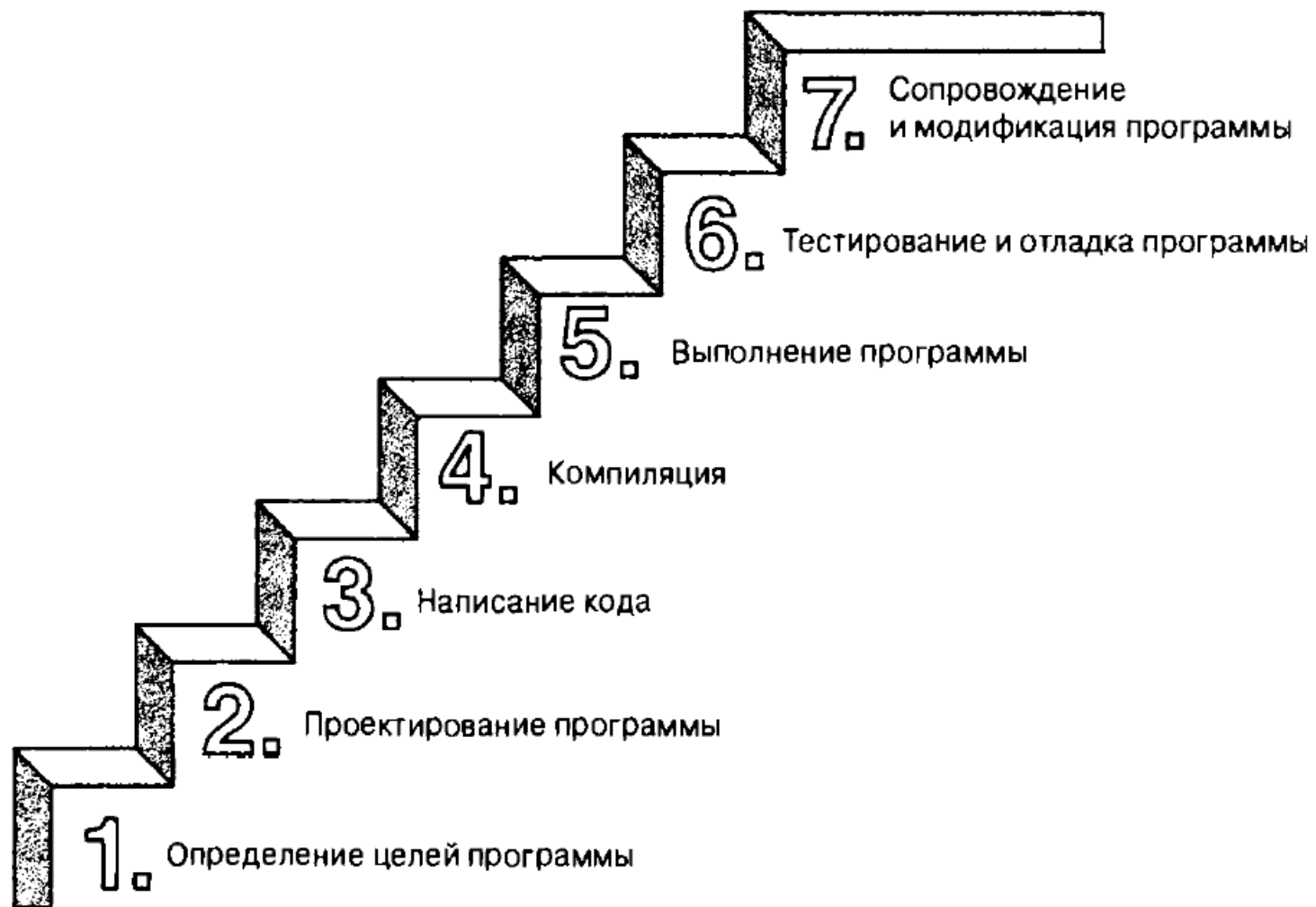
Входные данные



Результат



Этапы программирования















Почему мы учим именно язык «С»

Jan 2021	Jan 2020	Change	Programming Language	Ratings	Change
1	2	⬆	C	17.38%	+1.61%
2	1	⬇	Java	11.96%	-4.93%
3	3		Python	11.72%	+2.01%
4	4		C++	7.56%	+1.99%
5	5		C#	3.95%	-1.40%
6	6		Visual Basic	3.84%	-1.44%
7	7		JavaScript	2.20%	-0.25%
8	8		PHP	1.99%	-0.41%
9	18	⬆	R	1.90%	+1.10%
10	23	⬆	Groovy	1.84%	+1.23%












Почему мы учим именно язык «С»

Jan 2022	Jan 2021	Change	Programming Language		Ratings	Change
1	3	↑		Python	13.58%	+1.86%
2	1	↓		C	12.44%	-4.94%
3	2	↓		Java	10.66%	-1.30%
4	4			C++	8.29%	+0.73%
5	5			C#	5.68%	+1.73%
6	6			Visual Basic	4.74%	+0.90%
7	7			JavaScript	2.09%	-0.11%
8	11	↑		Assembly language	1.85%	+0.21%
9	12	↑		SQL	1.80%	+0.19%
10	13	↑		Swift	1.41%	-0.02%












Почему мы учим именно язык «С»

Feb 2023	Feb 2022	Change	Programming Language		Ratings	Change
1	1			Python	15.49%	+0.16%
2	2			C	15.39%	+1.31%
3	4	▲		C++	13.94%	+5.93%
4	3	▼		Java	13.21%	+1.07%
5	5			C#	6.38%	+1.01%
6	6			Visual Basic	4.14%	-1.09%
7	7			JavaScript	2.52%	+0.70%
8	10	▲		SQL	2.12%	+0.58%
9	9			Assembly language	1.38%	-0.21%












Почему мы учим именно язык «С»

Feb 2024	Feb 2023	Change	Programming Language		Ratings	Change
1	1			Python	15.16%	-0.32%
2	2			C	10.97%	-4.41%
3	3			C++	10.53%	-3.40%
4	4			Java	8.88%	-4.33%
5	5			C#	7.53%	+1.15%
6	7	^		JavaScript	3.17%	+0.64%
7	8	^		SQL	1.82%	-0.30%
8	11	^		Go	1.73%	+0.61%
9	6	v		Visual Basic	1.52%	-2.62%




Почему мы учим именно язык «С»

Jan 2025	Jan 2024	Change	Programming Language		Ratings	Change
1	1			Python	23.28%	+9.32%
2	3	^		C++	10.29%	+0.33%
3	4	^		Java	10.15%	+2.28%
4	2	v		C	8.86%	-2.59%
5	5			C#	4.45%	-2.71%
6	6			JavaScript	4.20%	+1.43%
7	11	^^		Go	2.61%	+1.24%
8	9	^		SQL	2.41%	+0.95%
9	8	v		Visual Basic	2.37%	+0.77%



Почему мы учим именно язык «С»

Jan 2026	Jan 2025	Change	Programming Language		Ratings	Change
1	1			Python	22.61%	-0.68%
2	4	↑		C	10.99%	+2.13%
3	3			Java	8.71%	-1.44%
4	2	↓		C++	8.67%	-1.62%
5	5			C#	7.39%	+2.94%
6	6			JavaScript	3.03%	-1.17%
7	9	↑		Visual Basic	2.41%	+0.04%
8	8			SQL	2.27%	-0.14%
9	11	↑		Delphi/Object Pascal	1.98%	+0.19%
10	18	↑↑		R	1.82%	+0.81%



Статистика по github (2023)

Language	Primary Language for % of Repositories		Occur in % of Repositories		Total Number of Bytes	
Python		23.21%		31.94%		1.10 GB
TypeScript		13.19%		18.83%		0.19 GB
JavaScript		7.50%		29.60%		0.35 GB
Rust		4.19%		5.52%		63.21 MB
Go		4.02%		5.17%		0.18 GB
C#		4.02%		5.05%		0.31 GB
C++		3.84%		8.61%		1.14 GB
Jupyter Notebook		3.24%		5.14%		1.07 GB
Java		2.88%		4.30%		0.48 GB
Kotlin		2.71%		4.38%		60.23 MB
C		2.63%		8.37%		3.80 GB



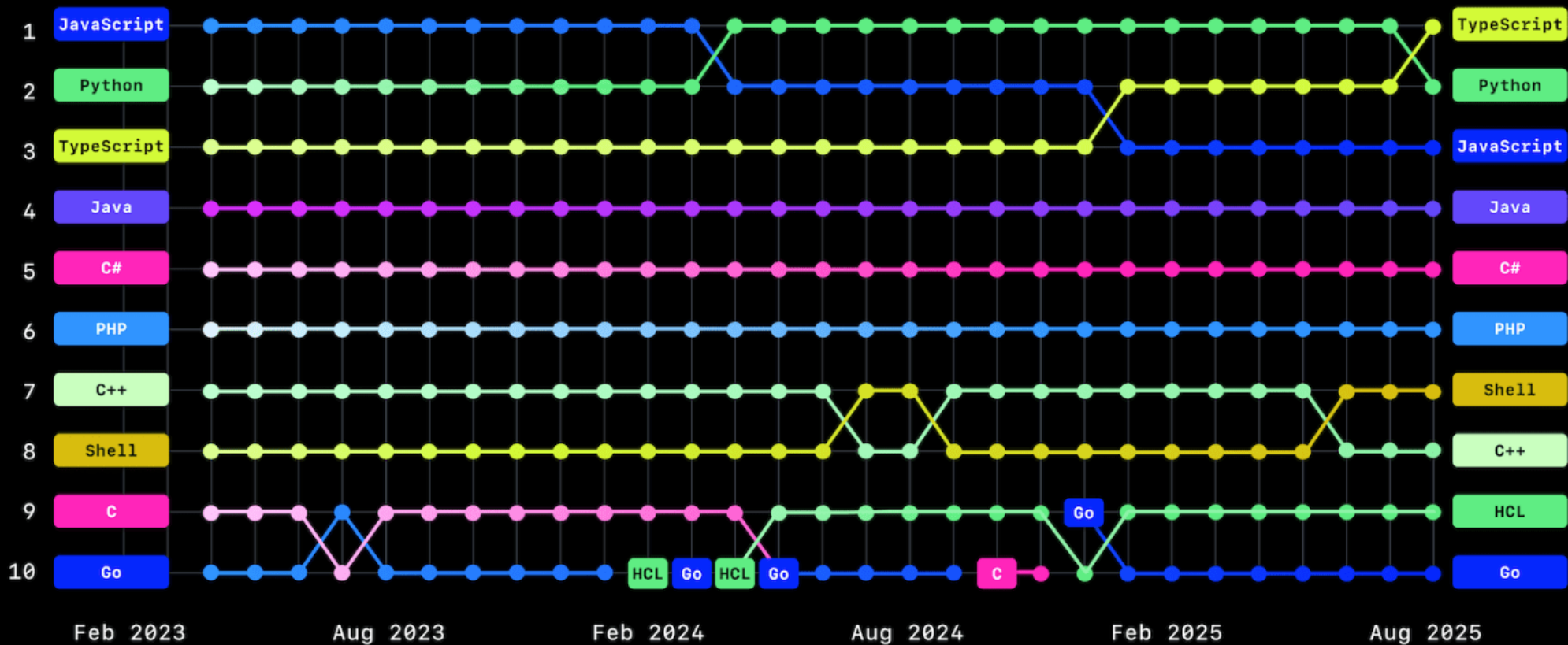
Статистика по github (2024)

# Ranking	Programming Language	Percentage (YoY Change)	YoY Trend
1	Python	18.172% (+0.129%)	
2	JavaScript	15.278% (-1.341%)	
3	Go	12.275% (+0.151%)	
4	C++	9.750% (+0.785%)	
5	Java	7.959% (-0.074%)	
6	TypeScript	7.045% (-0.298%)	
7	C	5.947% (+0.499%)	
8	C#	4.222% (+0.163%)	
9	PHP	3.690% (+0.050%)	



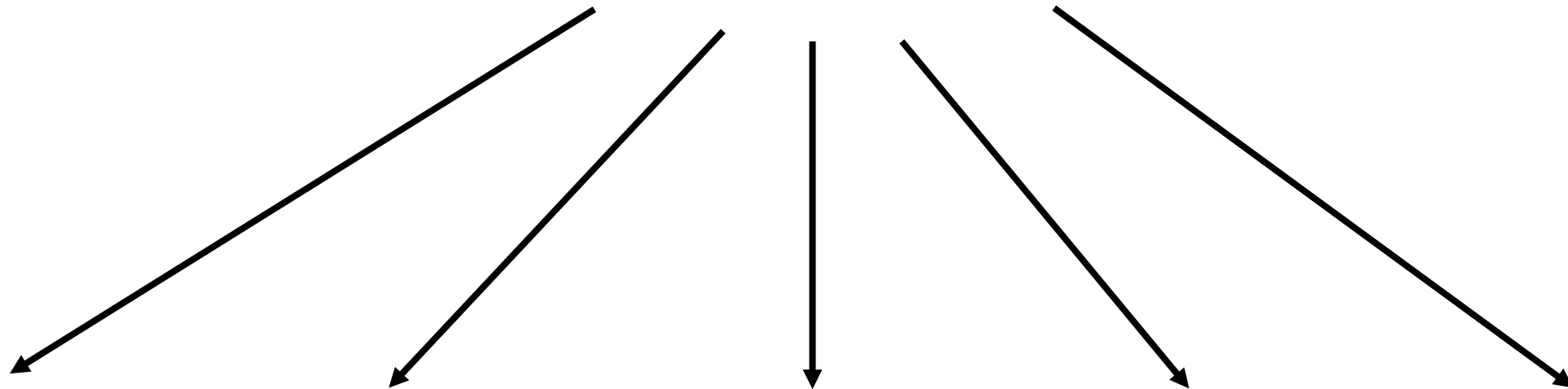
Статистика по github (2025)

Top 10 programming languages on GitHub 2023-2025





Области применения языков программирования



Программирование
микроконтроллеров



- C
- Assembler

Веб разработка



- PHP
- C#
- JS/TS
- Go

Приложения для
смартфонов



- Java
- Kotlin

Научные
исследования



- Python
- Matlab
- C++

Разработка
приложений



- C++
- Java
- Python

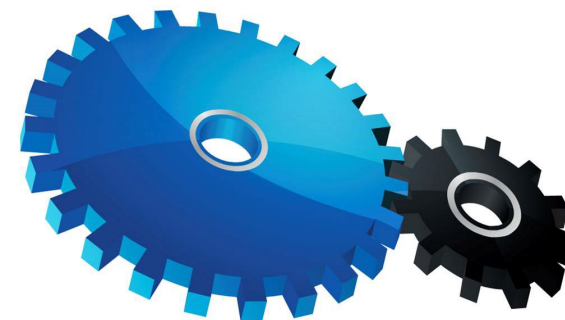
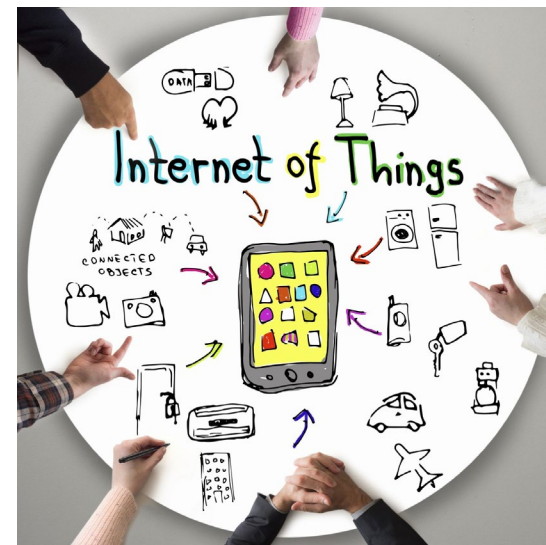


Почему мы учим именно язык «С»



Язык С применяется:

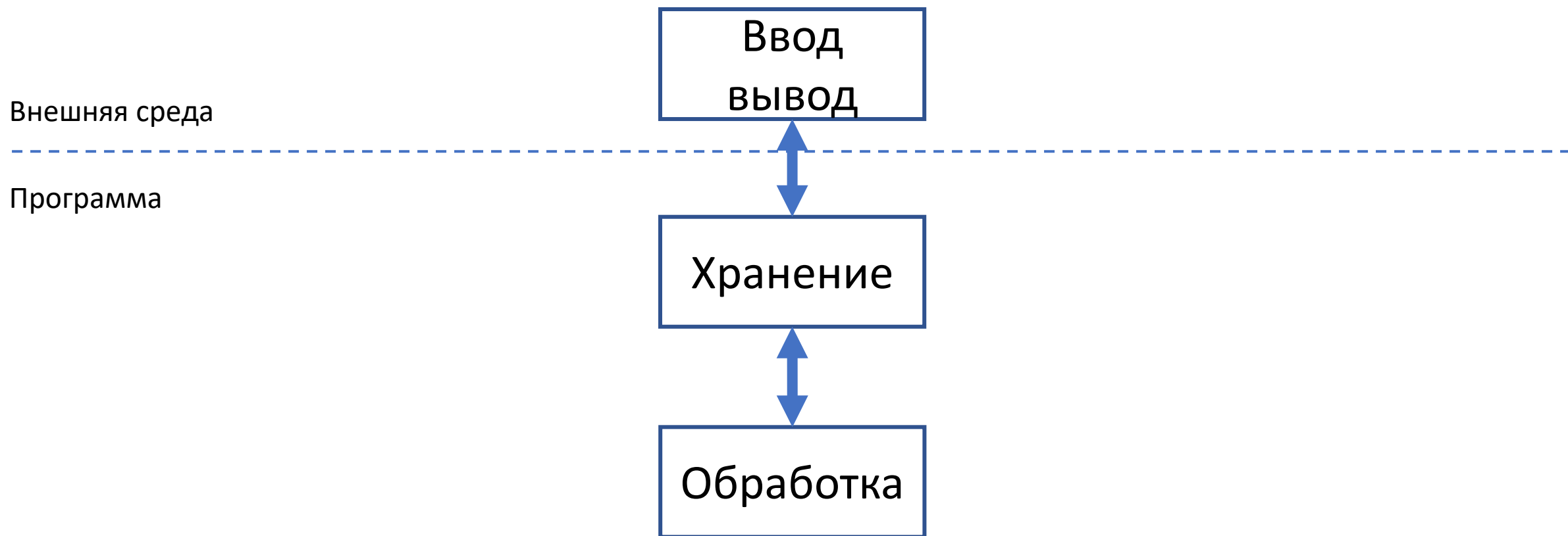
- Микроконтроллеры
- Интернет вещей
- Операционные системы
- Драйверы



Drivers

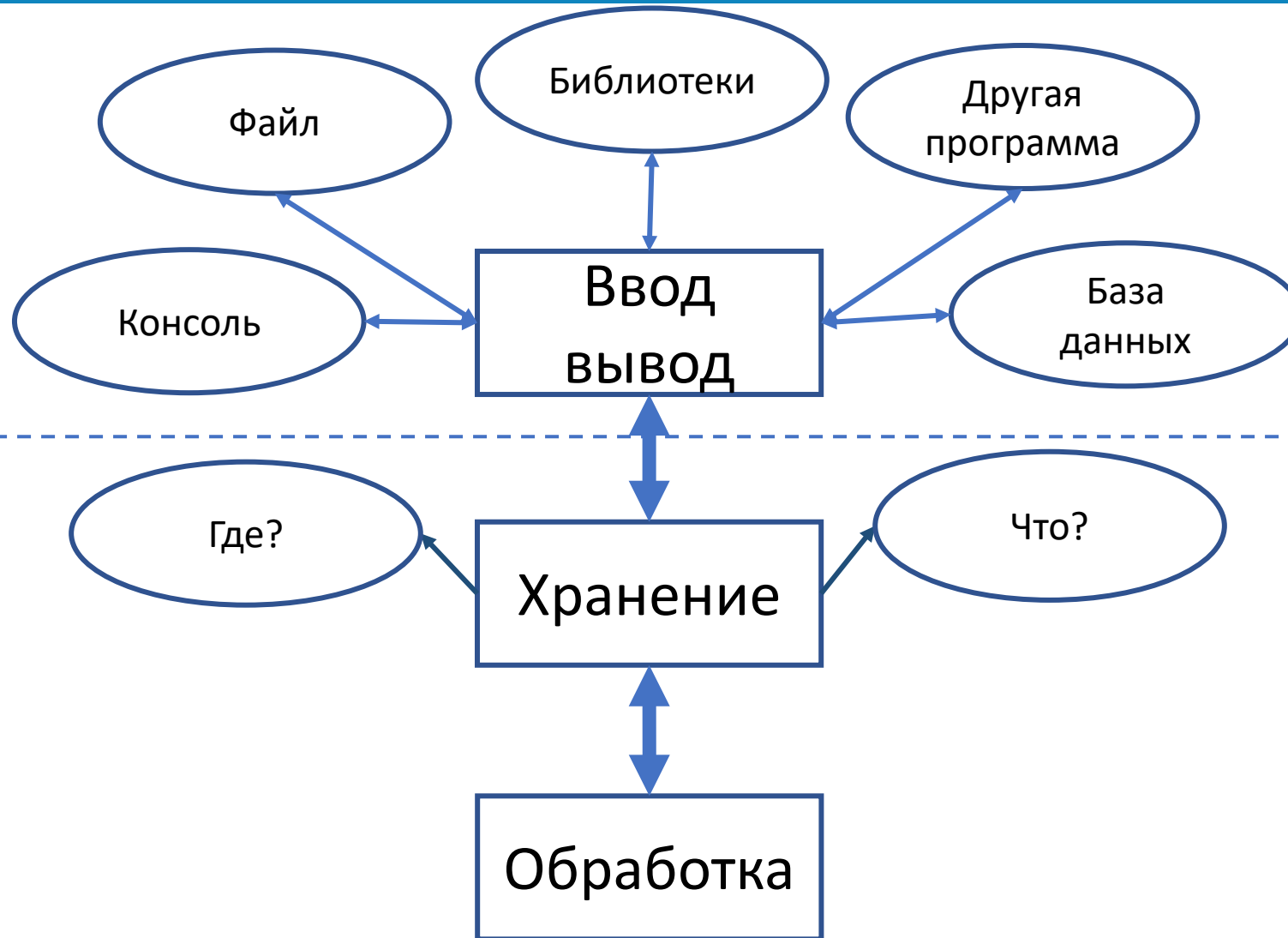


Дерево языка



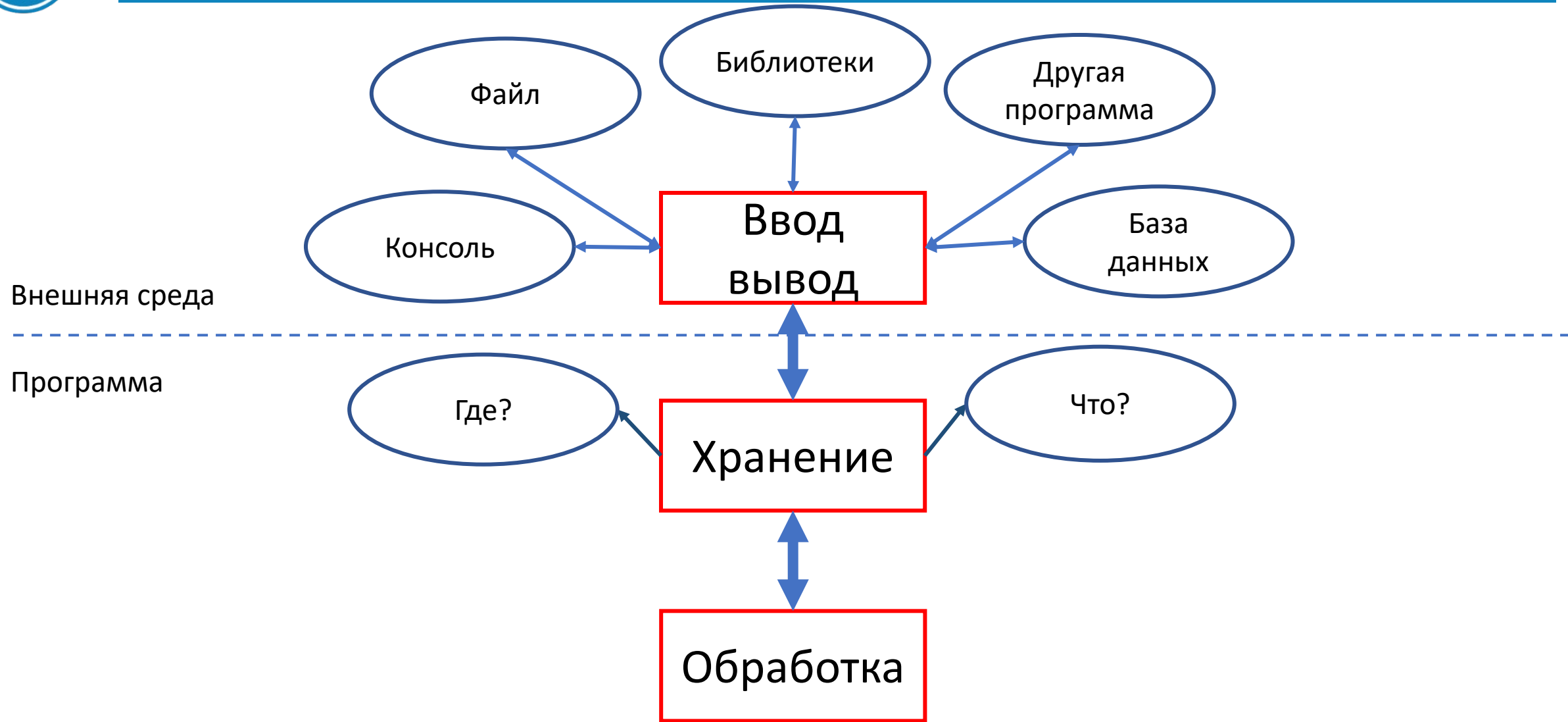


Дерево языка





Дерево языка





Немного про алгоритмы



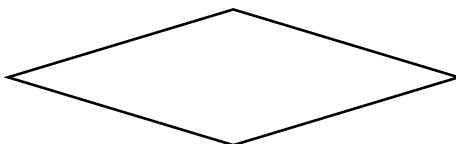
Проектирование программы



Начало/конец



Операция



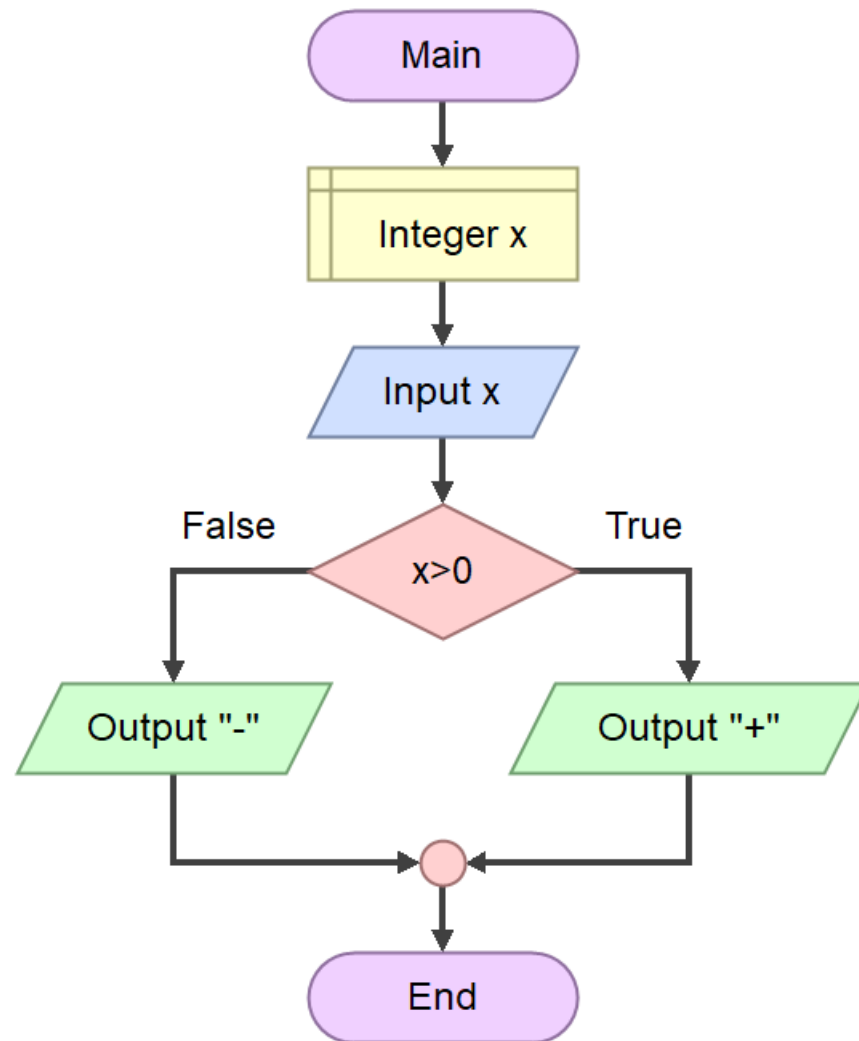
Условие



Ввод/вывод



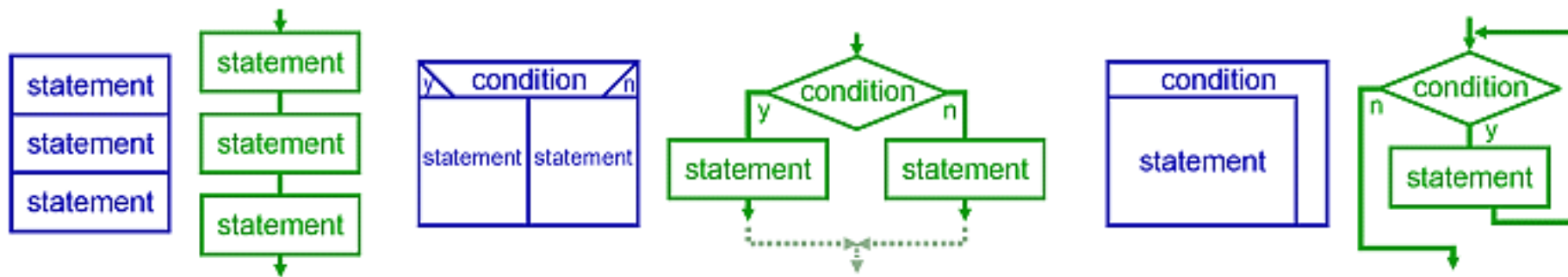
Цикл





Structured program theorem

Теорема Бёма-Якопини - положение структурного программирования, согласно которому любой исполняемый алгоритм может быть преобразован к структурированному виду, то есть такому виду, когда ход его выполнения определяется только при помощи трёх структур управления: **последовательной** (англ. sequence), **ветвлений** (англ. selection) и повторов или **циклов** (англ. repetition, cycle).





Оценка алгоритмов

Задача: Нужно найти максимальный элемент в списке

```
for (i = 0; i < n; i++) {  
    bool is_max = true;  
    for (j = 0; j < n; j++) {  
        if (a[j] > a[i]) {  
            is_max = false;  
            break;  
        }  
    }  
    if (is_max) return a[i];  
}
```

?

```
max = a[0];  
for (i = 1; i < n; i++) {  
    if (a[i] > max)  
        max = a[i];  
}
```



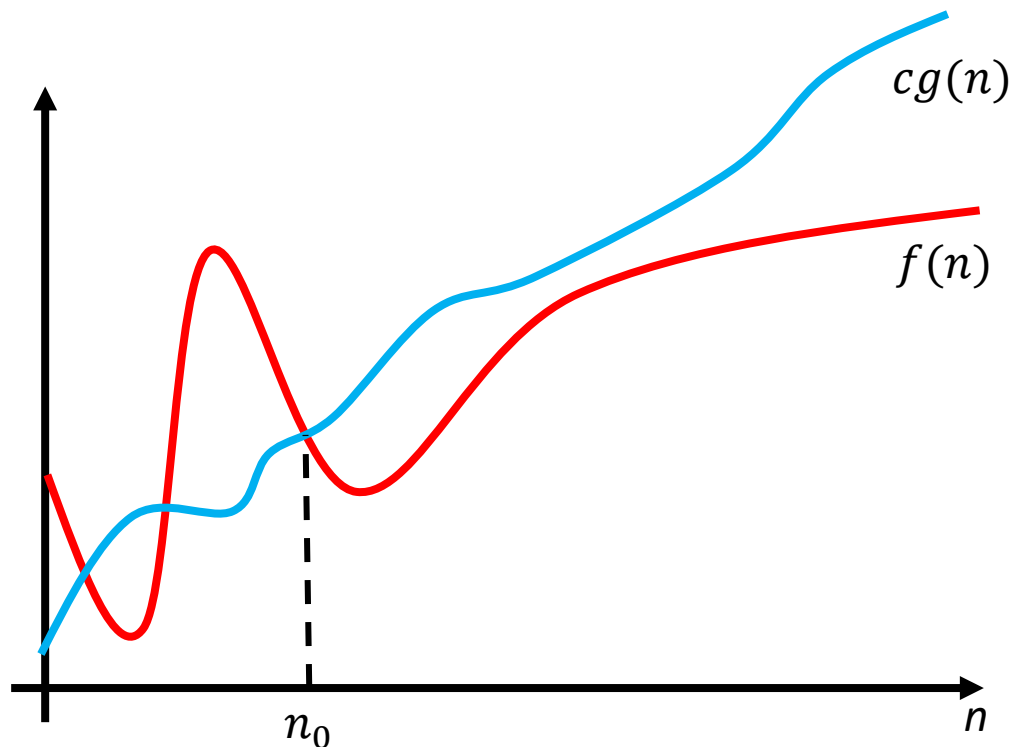
Сложность алгоритмов

Сложность алгоритма - теоретическая оценка количества ресурсов (времени и памяти), необходимых алгоритму для обработки входных данных определенного размера (n).

Сложность по времени: Сколько шагов/операций потребуется алгоритму.

Сложность по памяти: Сколько дополнительной памяти (помимо самих входных данных) потребуется алгоритму.

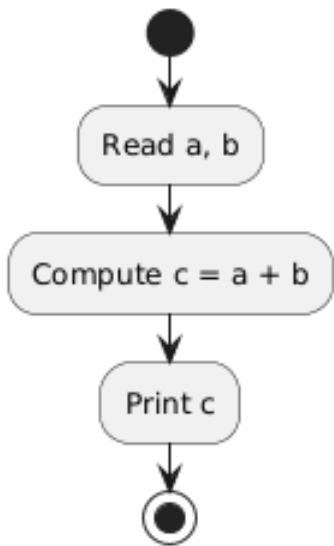
$f(n) = O(g(n))$, если $\exists c, n_0 > 0$ тогда $\forall n \geq n_0$ выполняется $0 \leq f(n) \leq cg(n)$



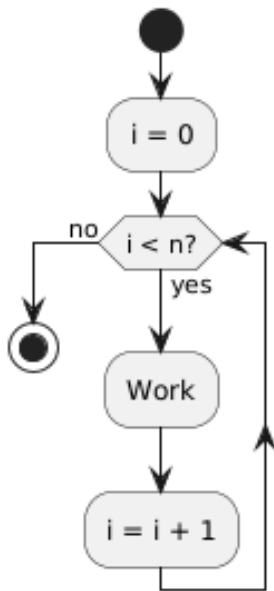
$f(n)$ — функция, которую мы анализируем (например, время работы алгоритма)
 $g(n)$ — “эталонная” функция, с которой мы сравниваем рост $f(n)$



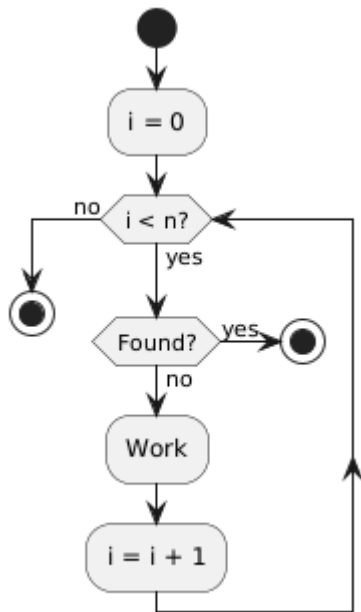
Примеры сложности алгоритмов



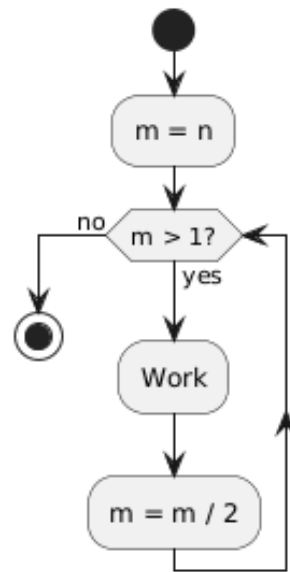
$O(1)$



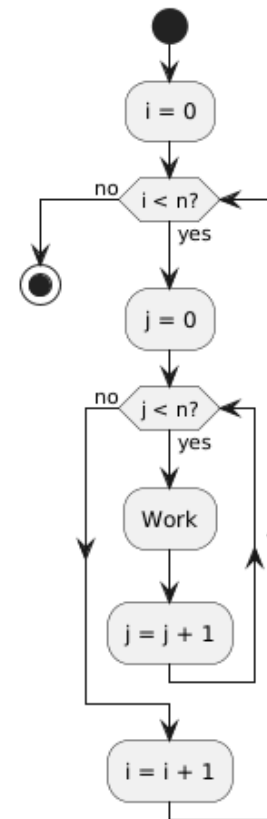
$O(n)$



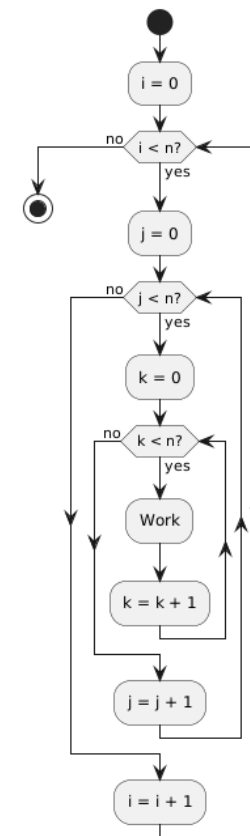
$O(n)$



$O(\log(n))$



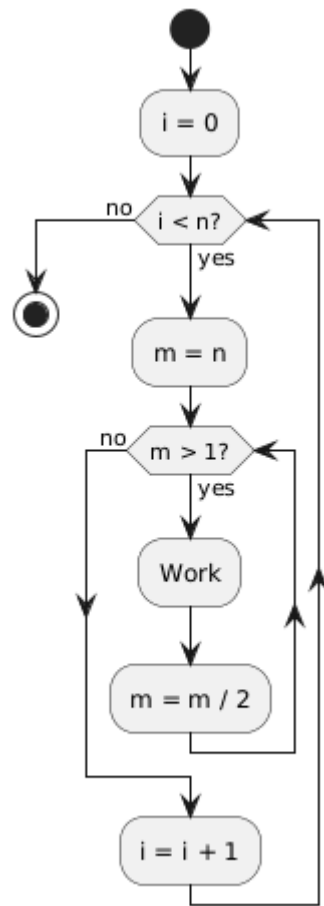
$O(n^2)$



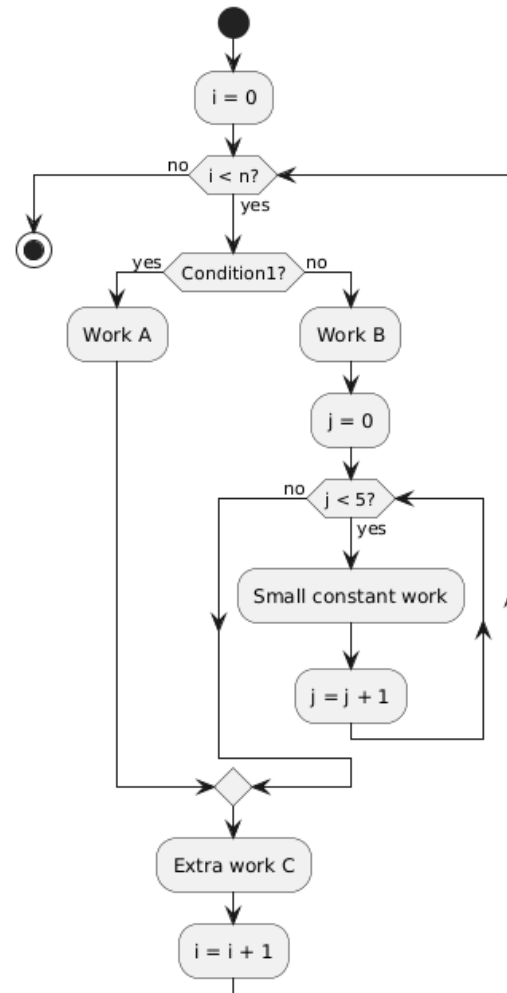
$O(n^3)$



Примеры сложности алгоритмов



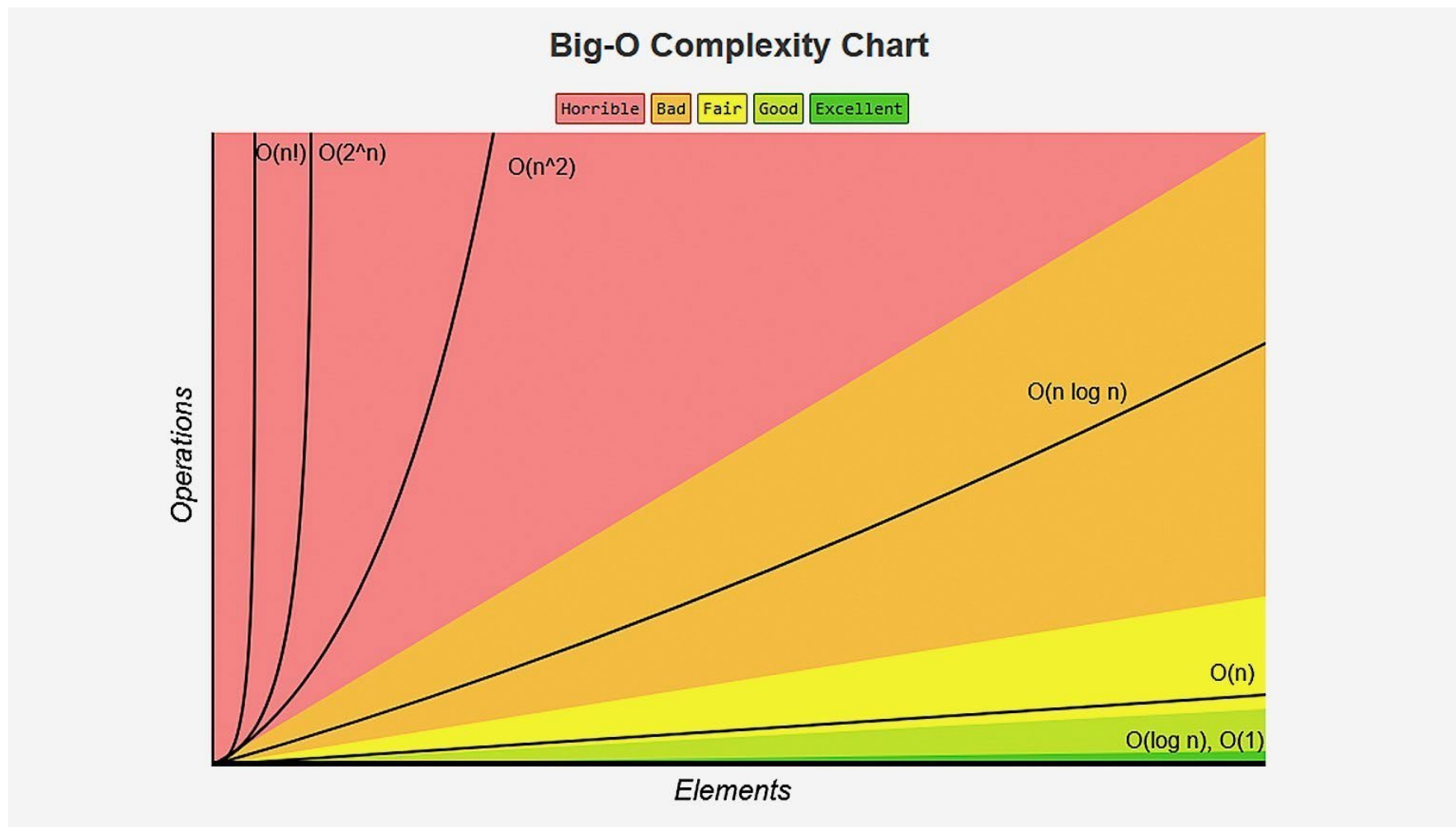
$O(n \log(n))$



$O(n)$



Вычислительная сложность



Алгоритм – это последовательность действий, записанная на формальном языке и приводящая к заданной цели за конечное число шагов.