



# Факультатив по программированию на языке C

**Преподаватель:** Балабаев Сергей Андреевич  
[sergei.balabaev@mail.ru](mailto:sergei.balabaev@mail.ru)



# План занятий

- Занятие – дополнительное
- Оценок не будет
- Можно (и даже нужно) гуглить
- Не стесняемся задавать вопросы
- Если стесняемся – то можно сюда:  
[https://vk.com/sergei\\_balabaev](https://vk.com/sergei_balabaev)



# План занятий

№	Тема	Описание
1	Введение в курс	Языки программирования. Основы работы с Linux.
2	Основы языка C	Написание и компиляция простейших программ с использованием gcc. Правила написания кода. Разбиение программы на отдельные файлы. Make файлы
3	Ввод данных. Библиотеки	Работа со входом/выходом. Статические и динамические библиотеки. Компиляция.
4	Язык ассемблера	Основы анализа программ на языке ассемблер.
5	Хранение данных. Память	Хранение процесса в памяти компьютера. Виртуальная память, сегментация. Секции программы.
6	Хранение данных.	Стек, куча. Типы данных. Преобразования типов. Gdb и отладка Хранение различных типов данных. Указатели, ссылки. Передача аргументов в функцию по ссылке/указателю.
7	Обработка данных	Безопасные функции. Битовые операции – сдвиги, логические операции. Битовые поля.
8	Программирование под встраиваемые ОС	Работа с микрокомпьютером Raspberry Pi



## Рекомендованная литература

1. *Ашарина И.В.* Язык программирования C++. Конспект лекций по курсу “Информатика”. - М.: МИЭТ, 2000. - 112 с.: ил
2. *Брайан Керниган, Деннис Ритчи* - Язык программирования Си Москва: Диалектика, 2020.
3. *Прата С.* Язык программирования C. Лекции и упражнения, 6-е изд. : Пер. с англ. — М. :ООО “И.Д. Вильямс”, 2015. — 928 с.

---

4. *Столяров А.В.* Низкоуровневое программирование. Том 2 Глава 4
5. *Дэвид М. Харрис и Сара Л. Харрис* Цифровая схемотехника и архитектура компьютера. Приложение «C»

---

6. *Igor Zhirkov* Low-Level Programming: C, Assembly, and Program Execution on Intel 64 Architecture
7. *Richard Reese* Understanding and Using C Pointers-O Reilly Media 2013
8. *Suzanne J. Matthews* Dive into Systems <https://diveintosystems.org/>
9. Видеолекции МФТИ – Тимофей Хирьянов (Youtube)



# Вместо введения

## Задача

Необходимо перемножить две матрицы размерностью  $N \times N$

## Решение для $N=2$

$$C_{ij} = \sum_{s=1}^n A_{is} B_{sj} \quad A = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \quad B = \begin{pmatrix} c_1 & d_1 \\ c_2 & d_2 \end{pmatrix}$$

$$\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \begin{pmatrix} c_1 & d_1 \\ c_2 & d_2 \end{pmatrix} = \begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix}$$



# Вместо введения

## Задача

Необходимо перемножить две матрицы размерностью  $N \times N$

## Решение

$$C_{ij} = A_{i1}B_{1j} + A_{i2}B_{2j}$$
$$C_{11} = A_{11}B_{11} + A_{12}B_{21}$$
$$\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \begin{pmatrix} c_1 & d_1 \\ c_2 & d_2 \end{pmatrix} = \begin{pmatrix} a_1c_1 + b_1c_2 & ? \\ ? & ? \end{pmatrix}$$

$$\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \begin{pmatrix} c_1 & d_1 \\ c_2 & d_2 \end{pmatrix} = \begin{pmatrix} a_1c_1 + b_1c_2 & a_1d_1 + b_1d_2 \\ a_2c_1 + b_2c_2 & a_2d_1 + b_2d_2 \end{pmatrix}$$



# Вместо введения

## Задача

Необходимо перемножить две матрицы размерностью  $N \times N$

## Решение

$$C_{ij} = \sum_{s=1}^n A_{is} B_{sj} \quad \begin{pmatrix} A_{11} & * & A_{1n} \\ * & * & * \\ A_{n1} & * & A_{nn} \end{pmatrix} \begin{pmatrix} B_{11} & * & B_{1n} \\ * & * & * \\ B_{n1} & * & B_{nn} \end{pmatrix} = \begin{pmatrix} ? & * & ? \\ * & * & * \\ ? & * & ? \end{pmatrix}$$

$$C_{ij} = A_{i1} B_{1j} + A_{i2} B_{2j} + \dots + A_{in} B_{nj}$$

$$\begin{pmatrix} A_{11} & * & A_{1n} \\ * & * & * \\ A_{n1} & * & A_{nn} \end{pmatrix} \begin{pmatrix} B_{11} & * & B_{1n} \\ * & * & * \\ B_{n1} & * & B_{nn} \end{pmatrix} = \begin{pmatrix} ? & * & ? \\ * & * & * \\ ? & * & ? \end{pmatrix}$$



# Вместо введения

```
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        for (int s = 0; s < n; s++)
            C[i][j] = C[i][j] + A[i][s] * B[s][j];
```

$i=0, j=0, s=0$        $C[0][0] = C[0][0] + A[0][0] * B[0][0];$

$i=0, j=0, s=1$        $C[0][0] = C[0][0] + A[0][1] * B[1][0];$

\*\*\*

$i=0, j=0, s=n-1$        $C[0][0] = C[0][0] + A[0][n-1] * B[n-1][0];$

$$\begin{pmatrix} \boxed{A_{00} * A_{0(n-1)}} \\ * & * \\ A_{(n-1)0} * A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} \boxed{B_{00}} * B_{0(n-1)} \\ * & * \\ B_{(n-1)0} * B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} \boxed{A_{00}B_{00} + \dots + A_{(n-1)0}B_{0(n-1)}} * & * \\ * & * \\ ? & ? \end{pmatrix}$$





# Вместо введения

```
for (int i = 0; i < n; i++)
    for (int s = 0; s < n; k++)
        for (int j = 0; j < n; j++)
            C[i][j] = C[i][j] + A[i][s] * B[s][j];
```

i=0, s=0, j=0

C[0][0] = C[0][0] + A[0][0] \* B[0][0];

i=0, s=0, j=1

C[0][1] = C[0][1] + A[0][0] \* B[0][1];

\*\*\*

i=0, s=0, j=n-1

C[0][n] = C[0][n-1] + A[0][0] \* B[0][n-1];

$$\begin{pmatrix} \boxed{A_{00}} & * & A_{0(n-1)} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} \boxed{B_{00}} & * & \boxed{B_{0(n-1)}} \\ * & * & * \\ B_{(n-1)0} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} \boxed{A_{00}B_{00}} & * & \boxed{A_{00}B_{0(n-1)}} \\ * & * & * \\ ? & * & ? \end{pmatrix}$$



# Вместо введения

```
for (int i = 0; i < n; i++)  
    for (int s = 0; s < n; k++)  
        for (int j = 0; j < n; j++)  
            C[i][j] = C[i][j] + A[i][s] * B[s][j];
```

$i=0, s=1, j=0$        $C[0][0] = C[0][0] + A[0][1] * B[1][0];$

$i=0, s=1, j=1$        $C[0][1] = C[0][1] + A[0][1] * B[1][1];$

\*\*\*

$i=0, s=1, j=n-1$        $C[0][n] = C[0][n-1] + A[0][1] * B[1][n-1];$

$$\begin{pmatrix} A_{00} & * & A_{0(n-1)} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} B_{00} & * & B_{0(n-1)} \\ * & * & * \\ B_{(n-1)0} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} A_{00}B_{00} + A_{01}B_{10} & * & A_{00}B_{0(n-1)} + A_{01}B_{1(n-1)} \\ * & * & * \\ ? & * & ? \end{pmatrix}$$



# Вместо введения

```
for (int j = 0; j < n; j++)  
    for (int s = 0; s < n; s++)  
        for (int i = 0; i < n; i++)  
            C[i][j] = C[i][j] + A[i][s] * B[s][j];
```

j=0, s=0, i=0

C[0][0] = C[0][0] + A[0][0] \* B[0][0];

j=0, s=0, i=1

C[1][0] = C[1][0] + A[1][0] \* B[0][0];

\*\*\*

j=0, s=0, i=n-1

C[n-1][0] = C[n-1][0] + A[n-1][0] \* B[0][0];

$$\begin{pmatrix} A_{00} & * & A_{0(n-1)} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} B_{00} & * & B_{0(n-1)} \\ * & * & * \\ B_{(n-1)0} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} A_{00}B_{00} & * & ? \\ * & * & * \\ A_{(n-1)0}B_{00} & * & ? \end{pmatrix}$$



# Вместо введения

```
for (int j = 0; j < n; j++)
    for (int s = 0; s < n; k++)
        for (int i = 0; i < n; i++)
            C[i][j] = C[i][j] + A[i][s] * B[s][j];
```

j=0, s=1, i=0                      C[0][0] = C[0][0] + A[0][1] \* B[1][0];

j=0, s=1, i=1                      C[1][0] = C[1][0] + A[1][1] \* B[1][0];

\*\*\*

j=0, s=1, i=n-1                      C[n-1][0] = C[n-1][0] + A[n-1][1] \* B[1][0];

$$\begin{pmatrix} A_{00} & * & A_{0(n-1)} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} B_{00} & * & B_{0(n-1)} \\ * & * & * \\ B_{(n-1)0} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} A_{00}B_{00} + A_{01}B_{10} & * & ? \\ * & * & * \\ A_{(n-1)0}B_{00} + A_{(n-1)1}B_{10} & * & ? \end{pmatrix}$$



# Вместо введения

А есть ли разница?

```

for (int j = 0; j < n; j++)
    for (int s = 0; s < n; k++)
        for (int i = 0; i < n; i++)
            C[i][j] = C[i][j] + A[i][s] * B[s][j];

```

j=0, s=1, i=0                      C[0][0] = C[0][0] + A[0][1] \* B[1][0];

j=0, s=1, i=1                      C[1][0] = C[1][0] + A[1][1] \* B[1][0];

\*\*\*

j=0, s=1, i=n-1                      C[n-1][0] = C[n-1][0] + A[n-1][1] \* B[1][0];

$$\begin{pmatrix} A_{00} & * & A_{0(n-1)} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} B_{00} & * & B_{0(n-1)} \\ * & * & * \\ B_{(n-1)0} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} A_{00}B_{00} + A_{01}B_{10} & * & ? \\ * & * & * \\ A_{(n-1)0}B_{00} + A_{(n-1)1}B_{10} & * & ? \end{pmatrix}$$



# Вместо введения

## Хранение массива в памяти

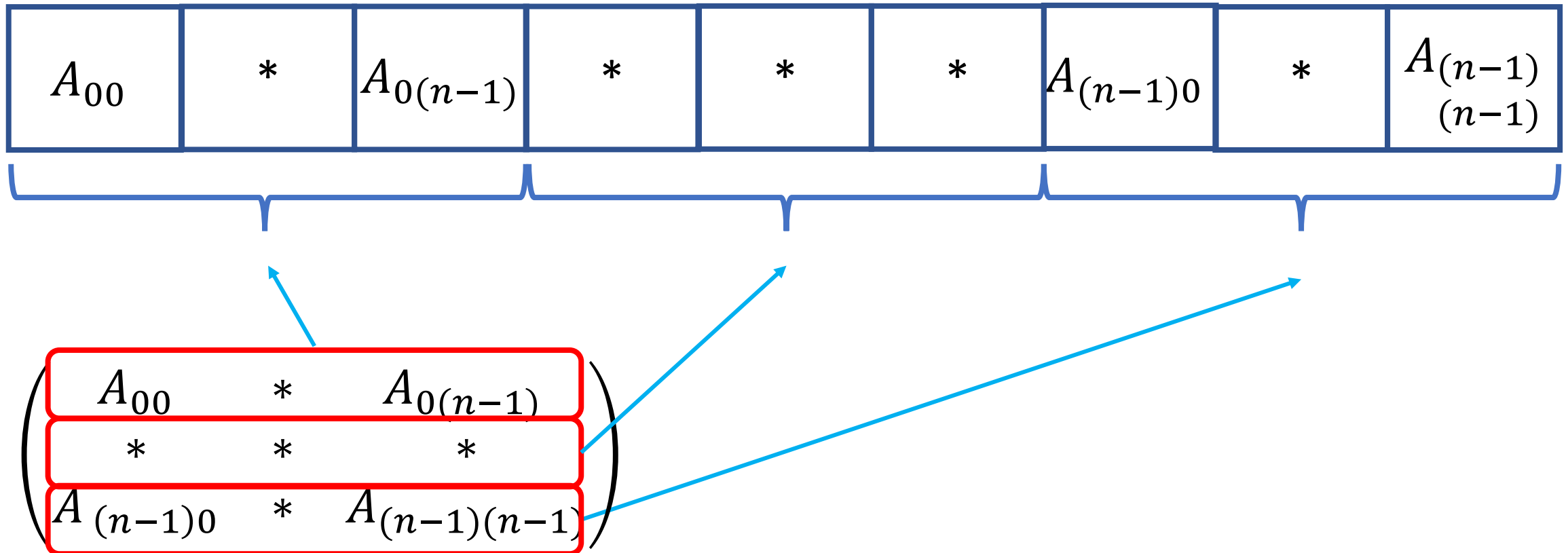
$A_{00}$	*	$A_{0(n-1)}$	*	*	*	$A_{(n-1)0}$	*	$A_{(n-1)(n-1)}$
----------	---	--------------	---	---	---	--------------	---	------------------

$$\begin{pmatrix} A_{00} & * & A_{0(n-1)} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix}$$



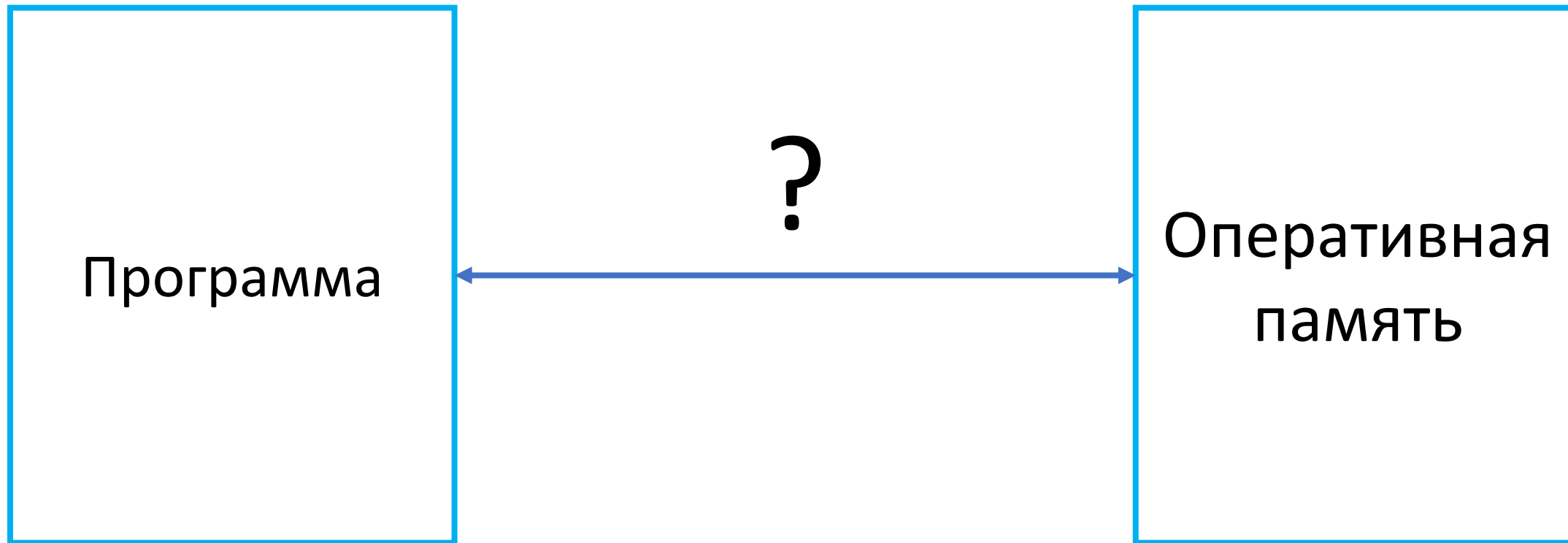
# Вместо введения

## Хранение массива в памяти





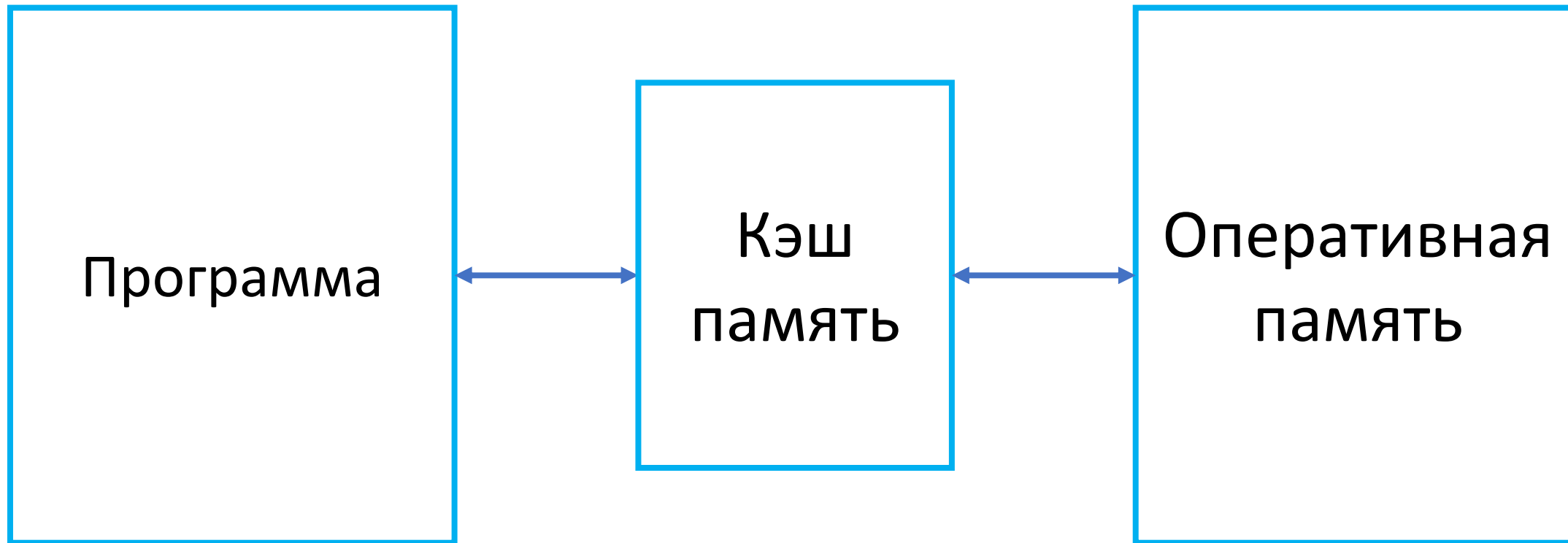
# Вместо введения





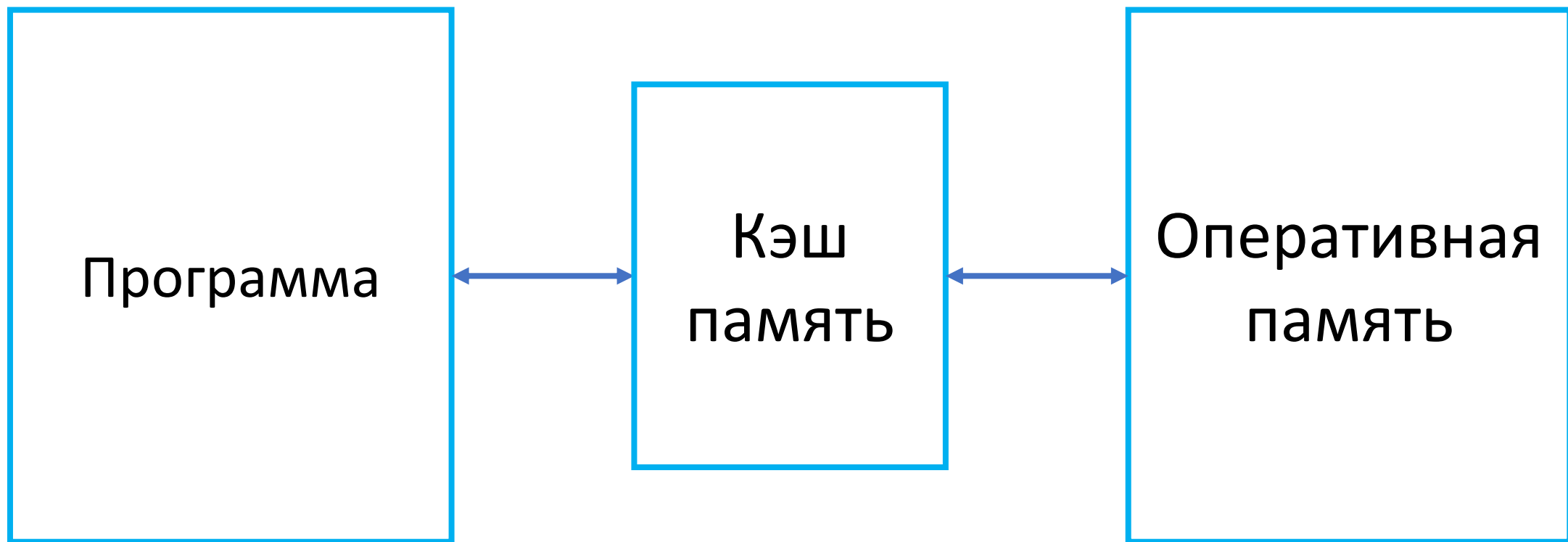


# Вместо введения





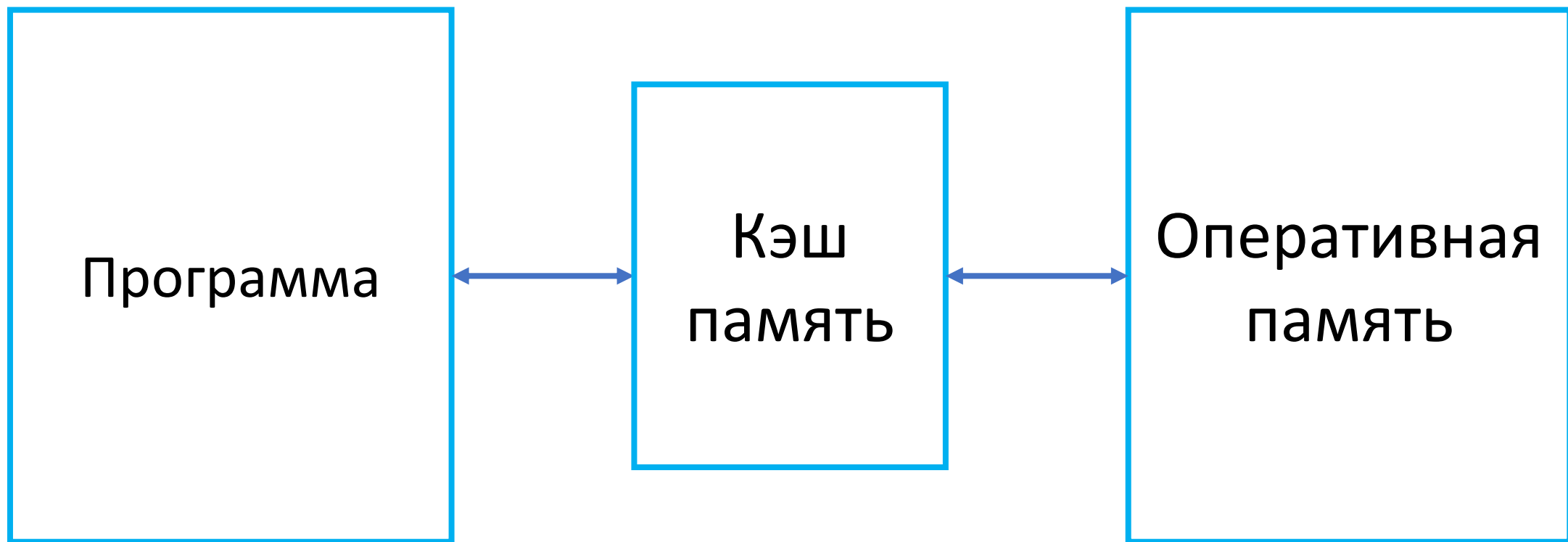
# Вместо введения



1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---



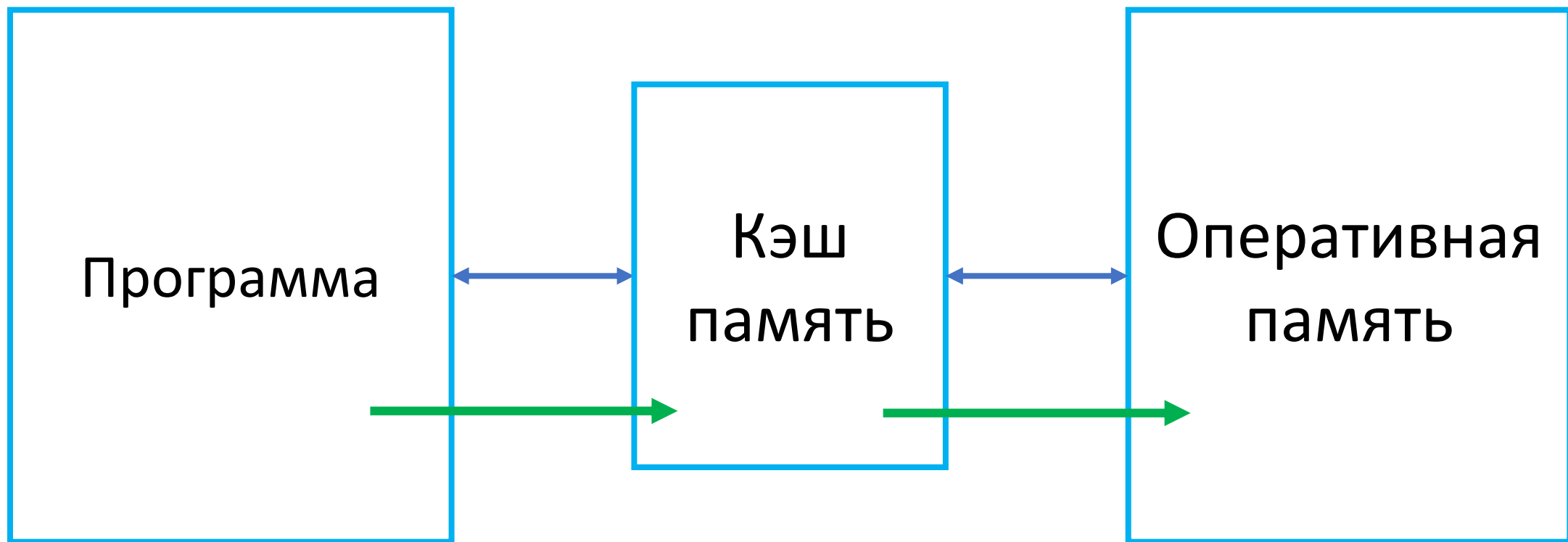
# Вместо введения



1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---



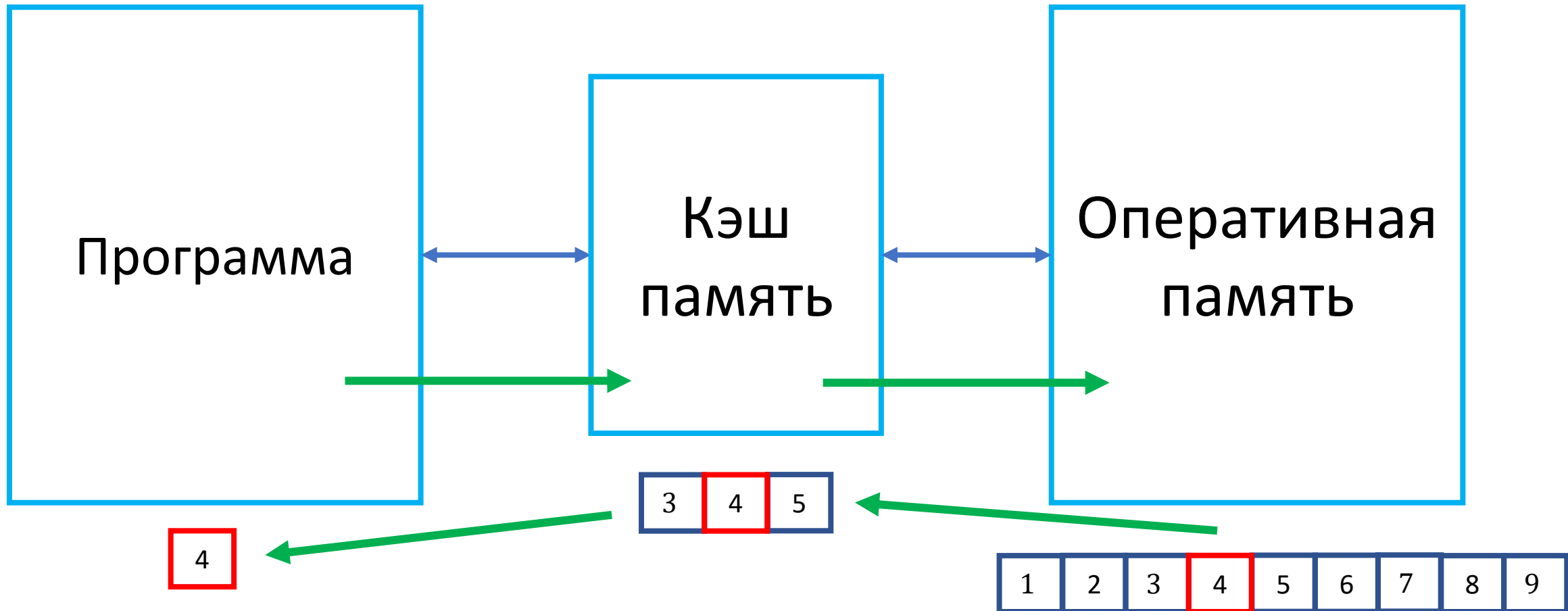
# Вместо введения



1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

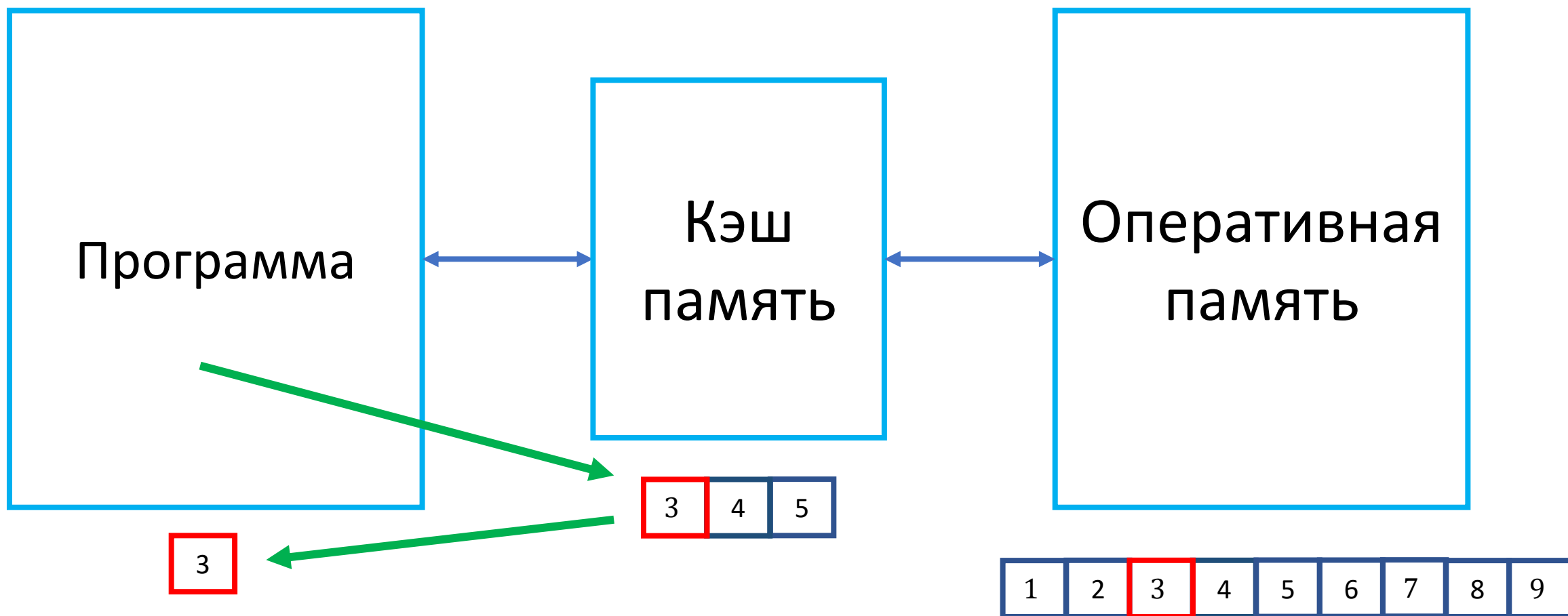


# Вместо введения





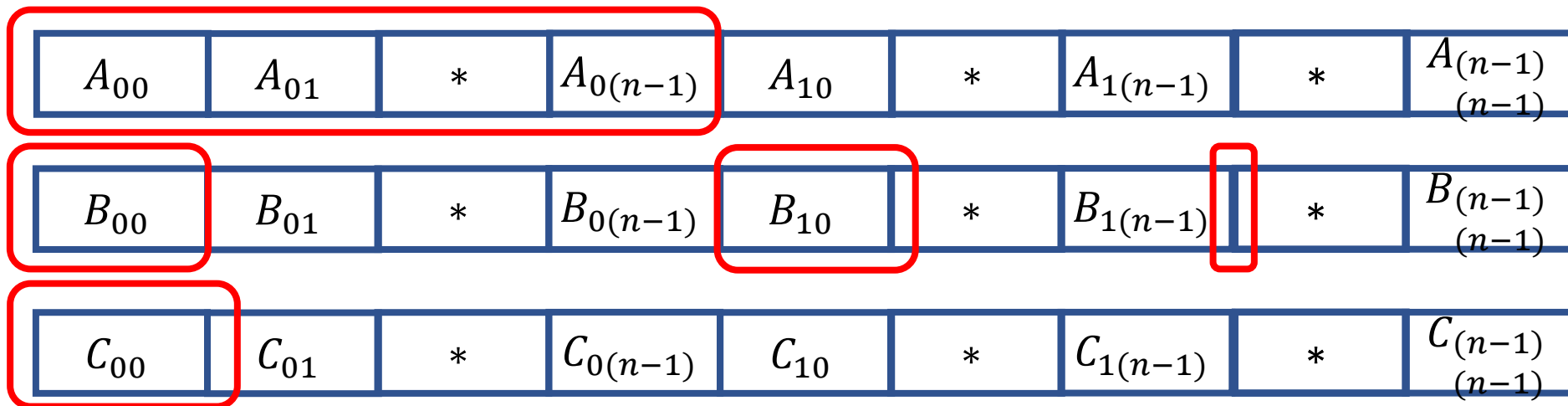
# Вместо введения





# Вместо введения

## Оперативная память

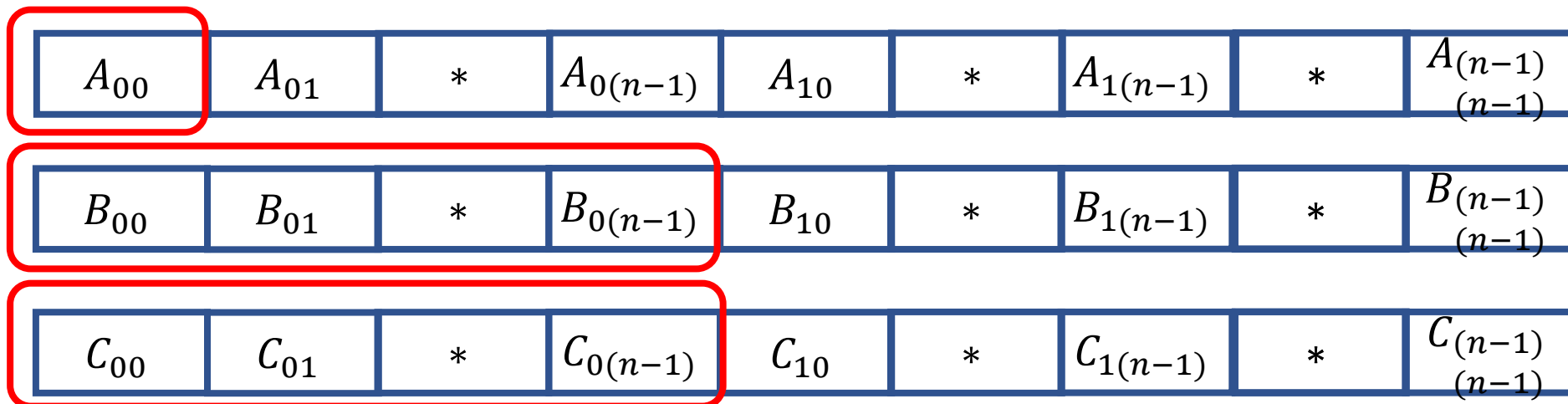


$$\begin{pmatrix} A_{00} & * & A_{0(n-1)} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} B_{00} & * & B_{0(n-1)} \\ * & * & * \\ B_{(n-1)0} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} A_{00}B_{00} + \dots + A_{(n-1)0}B_{0(n-1)} & * & * \\ * & * & * \\ ? & * & ? \end{pmatrix}$$



# Вместо введения

## Оперативная память



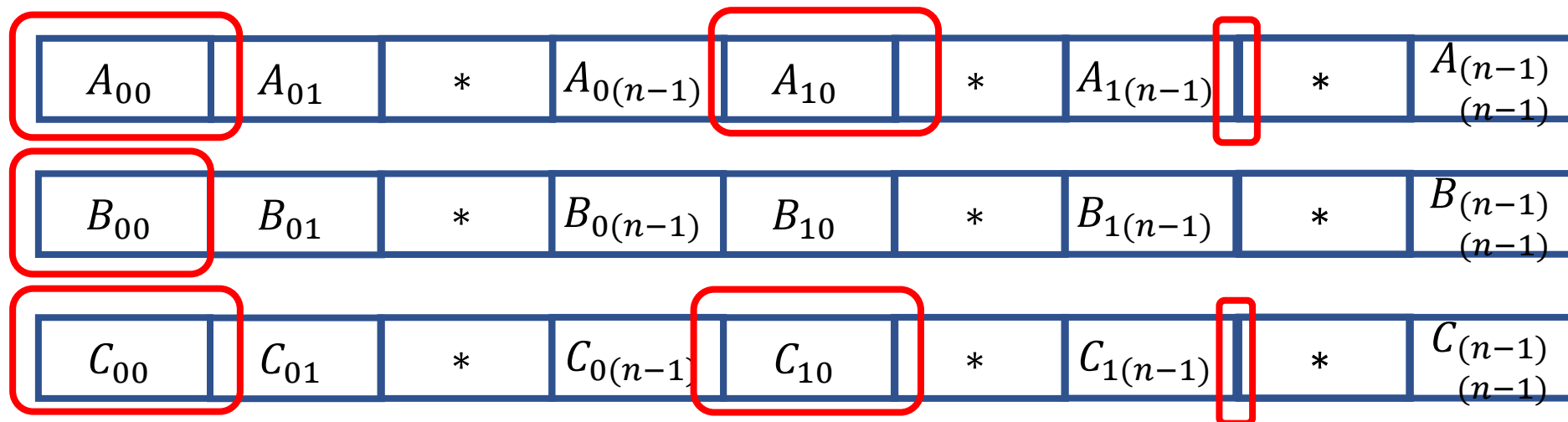
$$\begin{pmatrix} A_{00} & * & A_{0(n-1)} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} B_{00} & * & B_{0(n-1)} \\ * & * & * \\ B_{(n-1)0} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} A_{00}B_{00} & * & A_{00}B_{0(n-1)} \\ * & * & * \\ ? & * & ? \end{pmatrix}$$





# Вместо введения

## Оперативная память



$$\begin{pmatrix} A_{00} & * & A_{0(n-1)} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} B_{00} & * & B_{0(n-1)} \\ * & * & * \\ B_{(n-1)0} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} A_{00}B_{00} & * & ? \\ * & * & * \\ A_{(n-1)0}B_{00} & * & ? \end{pmatrix}$$



# Вместо введения

$$1. \begin{pmatrix} \boxed{A_{00}} & * & \boxed{A_{0(n-1)}} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} \boxed{B_{00}} & * & B_{0(n-1)} \\ * & * & * \\ \boxed{B_{(n-1)0}} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} \boxed{A_{00}B_{00} + \dots + A_{(n-1)0}B_{0(n-1)}} & * & * \\ * & * & * \\ ? & * & ? \end{pmatrix}$$

$$2. \begin{pmatrix} \boxed{A_{00}} & * & A_{0(n-1)} \\ * & * & * \\ A_{(n-1)0} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} \boxed{B_{00}} & * & \boxed{B_{0(n-1)}} \\ * & * & * \\ B_{(n-1)0} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} \boxed{A_{00}B_{00}} & * & \boxed{A_{00}B_{0(n-1)}} \\ * & * & * \\ ? & * & ? \end{pmatrix}$$

$$3. \begin{pmatrix} \boxed{A_{00}} & * & A_{0(n-1)} \\ * & * & * \\ \boxed{A_{(n-1)0}} & * & A_{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} \boxed{B_{00}} & * & B_{0(n-1)} \\ * & * & * \\ B_{(n-1)0} & * & B_{(n-1)(n-1)} \end{pmatrix} = \begin{pmatrix} \boxed{A_{00}B_{00}} & * & ? \\ * & * & * \\ \boxed{A_{(n-1)0}B_{00}} & * & ? \end{pmatrix}$$



# Вместо введения

## **Вывод!**

Кроме понимания алгоритмов нужно знать то «железо», под которое пишешь программу













# 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%

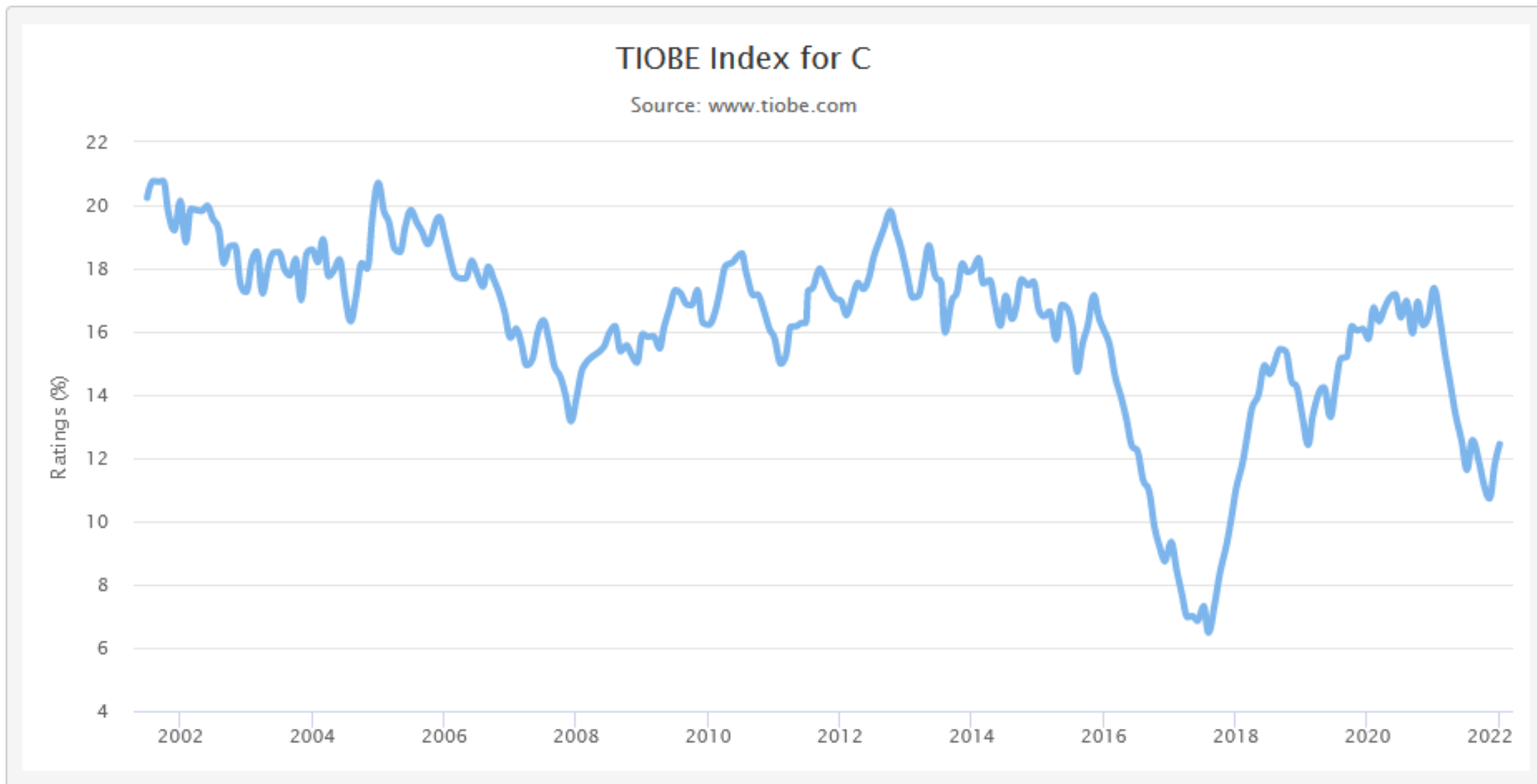


# Почему мы учим именно язык «С»

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%

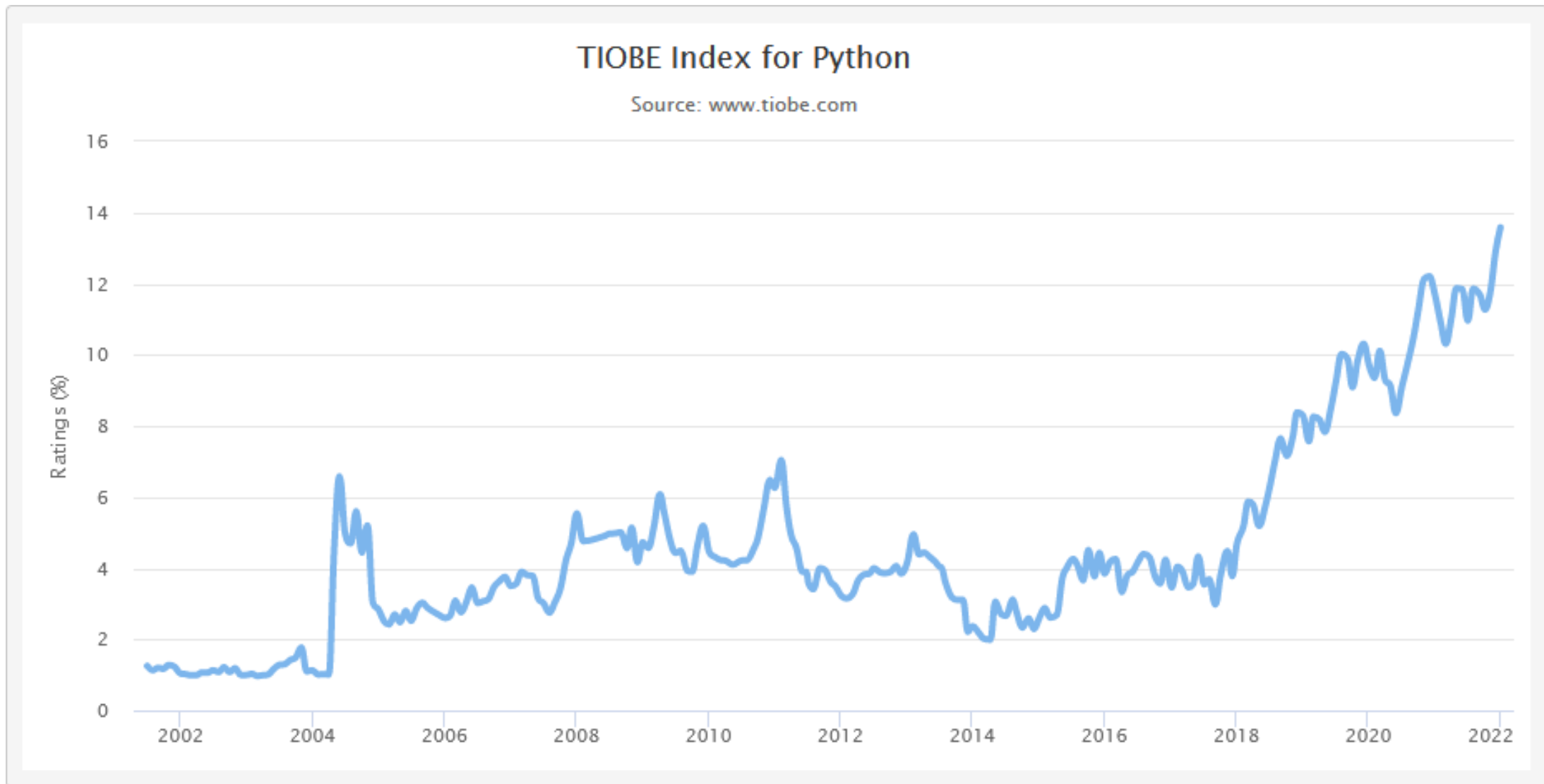


# Почему мы учим именно язык «С»



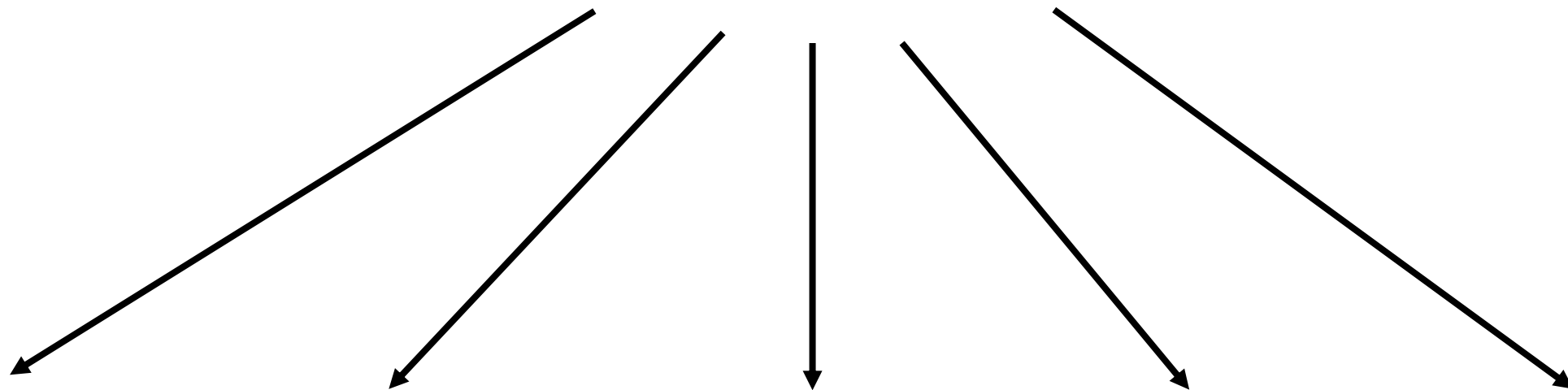


# Почему не Python?





# Области применения языков программирования



Программирование  
микроконтроллеров



- C
- Assembler

Веб разработка



- PHP
- C#
- JavaScript

Приложения для  
смартфонов



- Java
- Kotlin

Научные  
исследования



- Python
- Matlab
- C++

Разработка  
приложений



- C++
- Java
- Python



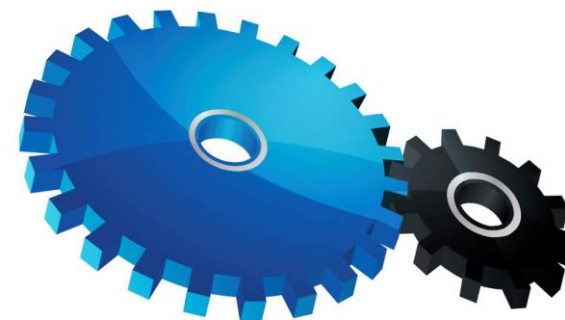
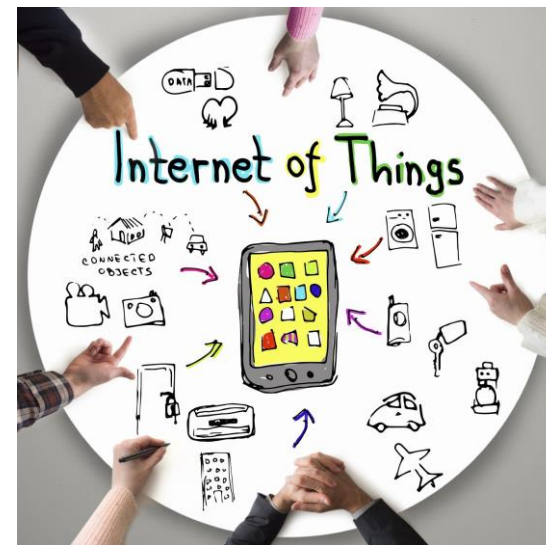


# Почему мы учим именно язык «С»



Язык С применяется:

- Микроконтроллеры
- Интернет вещей
- Операционные системы
- Драйверы



## Drivers



# Парадигмы программирования

## Парадигмы программирования

← Императивная

- Структурная
- Процедурная
- Модульная
- ООП

→ Декларативная

- Функциональная



# Парадигмы программирования

main

Настройка порта  
Чтение данных  
Запись данных в буфер  
Чтение данных из буфера  
Анализ данных  
Открытие текстового файла  
Запись данных  
Закрытие текстового файла  
Отправка сигнала завершения в  
порт



# Парадигмы программирования

Настройка порта

...

Чтение данных

...

Запись данных в буфер

...

Чтение данных из буфера

...

main

Отправка сигнала завершения

...

Анализ данных

...

Открытие текстового файла

...

Запись данных

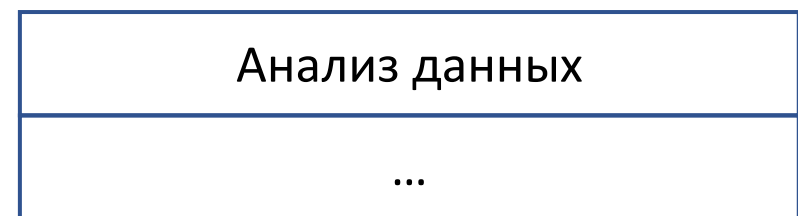
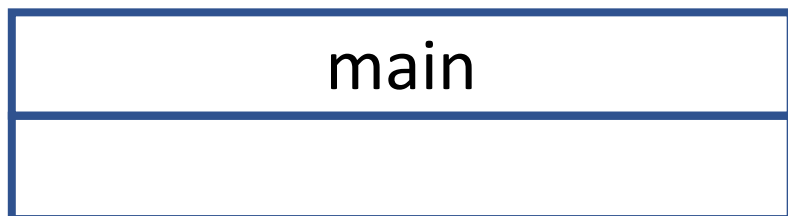
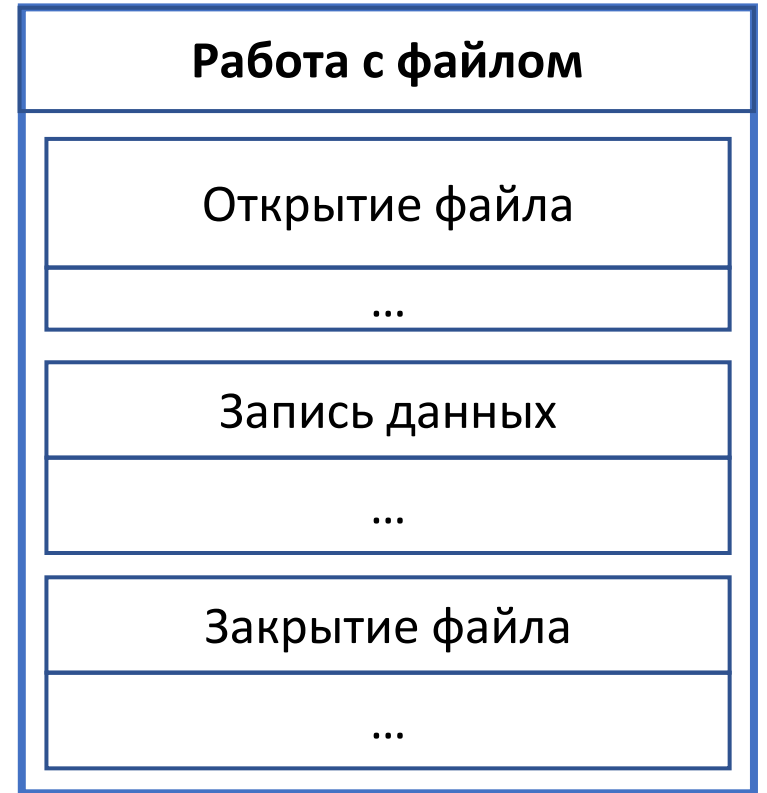
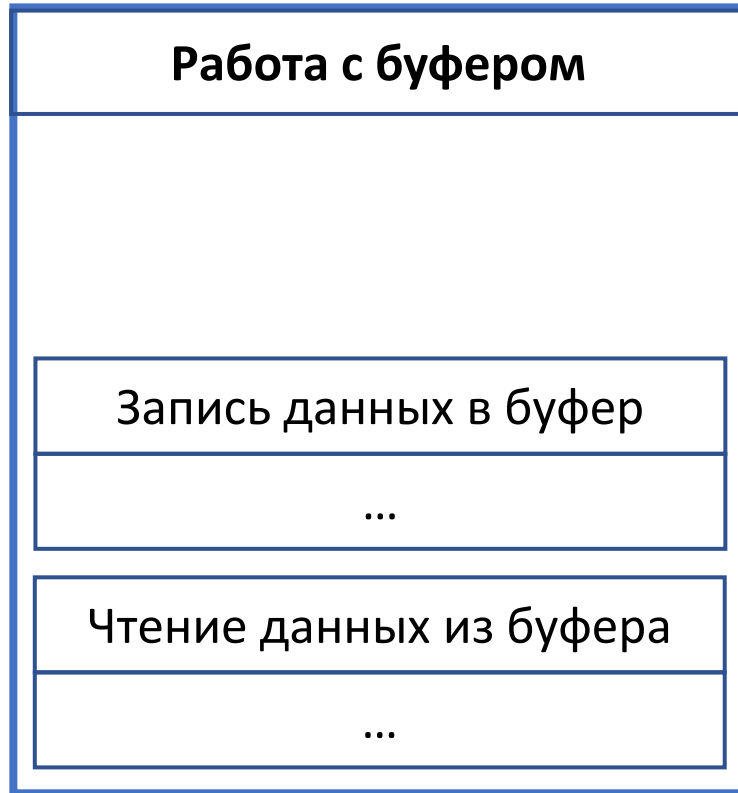
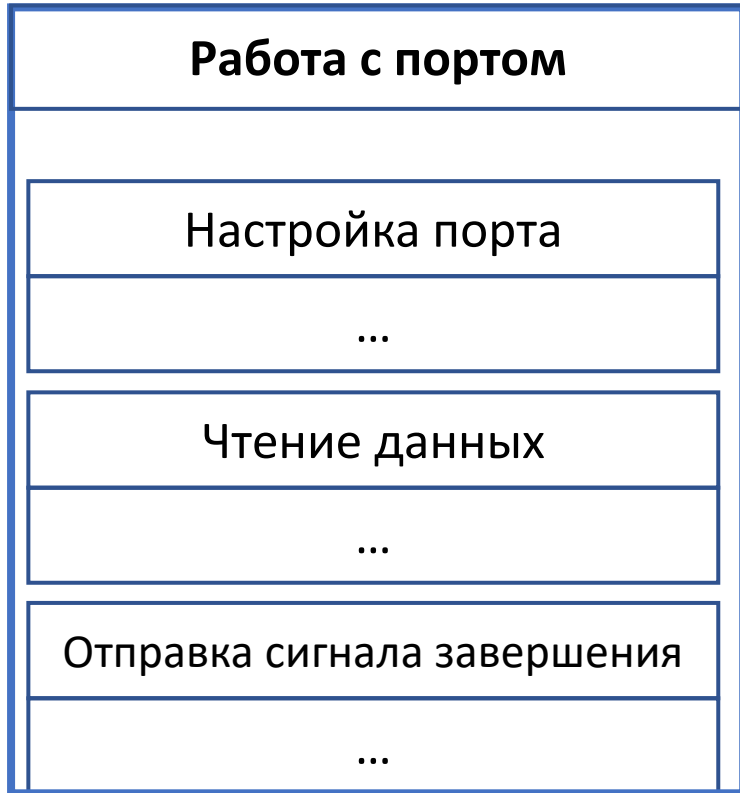
...

Закрытие текстового файла

...

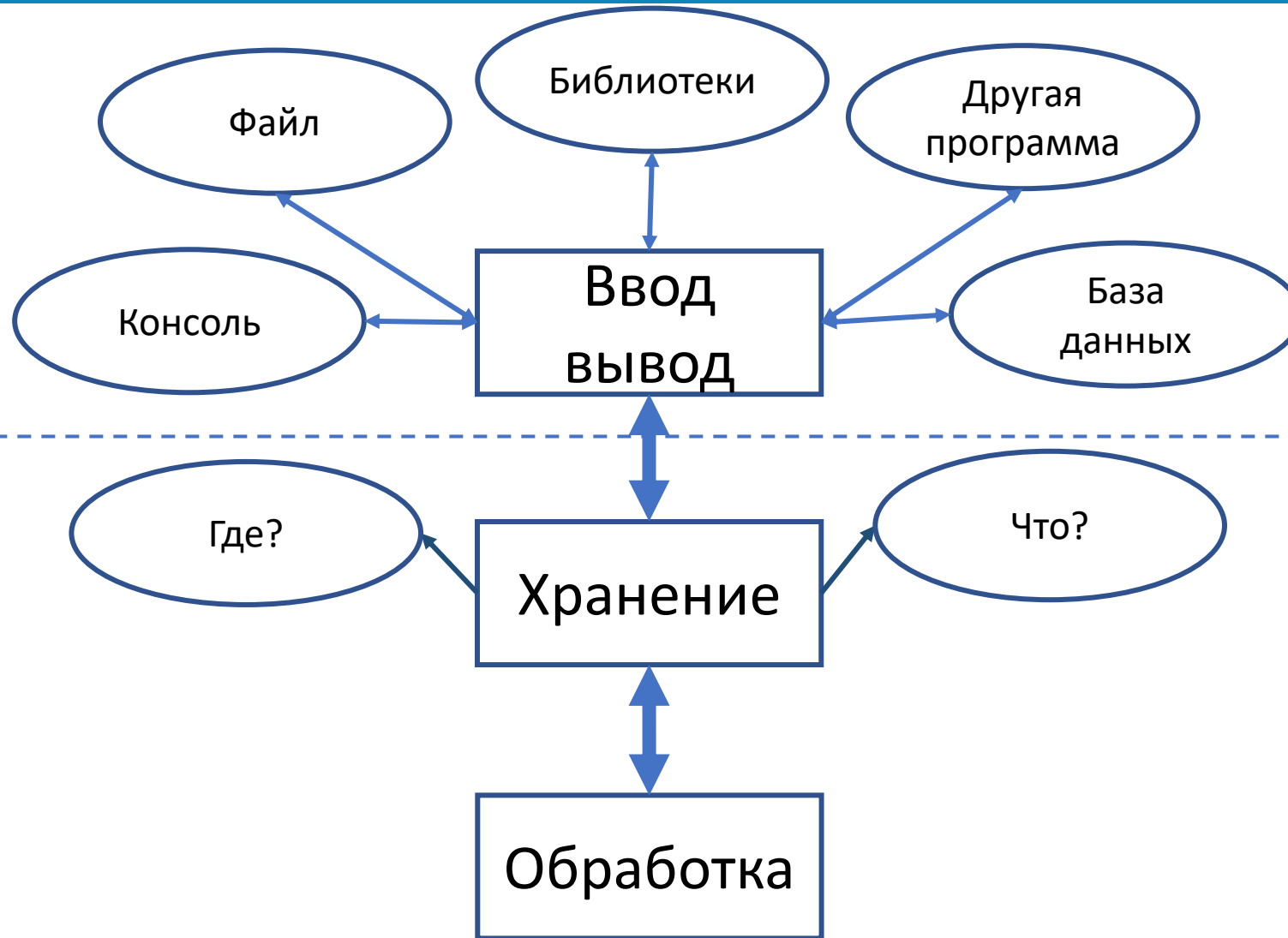


# Парадигмы программирования



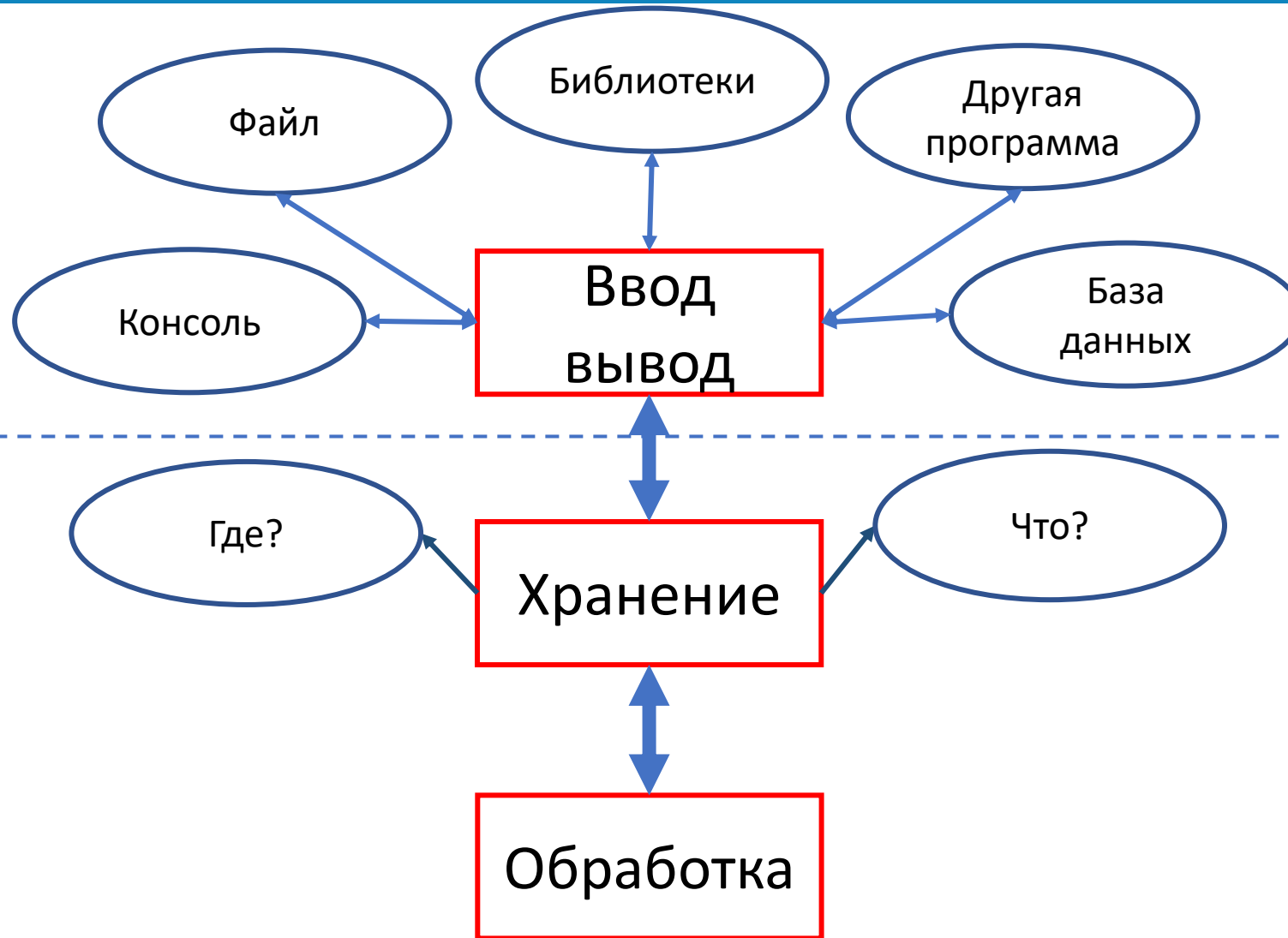


# Дерево языка





# Дерево языка





# Практическая часть





# Работа с командной строкой

```
sab@SAB: .../Lesson1$
```

№	Команда	Описание	Пример
0	man	Описание работы команды	man ls
1	pwd	Показать текущее местонахождение	~/SAB\$ pwd /home/user/SAB
2	ls	Позволяет просмотреть содержимое текущего каталога	~/SAB\$ ls 1 1.txt
3	cd <путь к директории>	Перейти в другую директорию	~\$ cd ~ /SAB/2 (полный путь) или ~/SAB\$ cd 2 (короткий путь)
4	mkdir <название директории>	Создание директории	~/SAB\$ mkdir 1
5	touch <название файла>	Создание файла	~/SAB\$ touch 1.txt
6	nano <название файла>	Редактирование файла	~/SAB\$ nano 1.txt
7	cp <что_копировать куда_копировать>	Копирование файла	~/SAB/1\$ cp 1.txt ~/SAB/2
8	cp -r <путь_к_папке путь_к_новому_месту>	Копирование директории	~/SAB/1\$ cp -r 1 ~/SAB/2
9	mv <что_переместить куда_переместить>	Переместить файл	~/SAB/1\$ mv 1.txt ~/SAB/2
10	rm <название файла>	Удалить файл	~/SAB/1\$ rm 1.txt
11	rm -r <название файла>	Удалить директорию	~/SAB/1\$ rm -r 1



# Потренируемся

1. Откройте терминал Linux
2. Создайте директорию. Назовите ее «Task1»
3. Войдите внутрь директории
4. Создайте внутри нее еще две директории – «Task1\_1» и «Task1\_2»
5. Войдите внутрь директории Task1\_1
6. Создайте внутри ее файл с названием «File\_1» и еще одну директорию «Task1\_1\_1»
7. Откройте созданный файл и запишите туда любую информацию. Закройте его
8. Скопируйте «File\_1» и «Task1\_1\_1» в директорию «Task1\_2»
9. Опуститесь на уровень ниже и удалите директорию Task1\_1
10. Из данной директории переместите «File\_1» в директорию «Task1»



# Перенаправление ввода и вывода

Ввод и вывод распределяется между тремя стандартными потоками:

- **stdin** — стандартный ввод (клавиатура), - 0
- **stdout** — стандартный вывод (экран), - 1
- **stderr** — стандартная ошибка (вывод ошибок на экран). - 2

**< file** — использовать файл как источник данных для стандартного потока ввода.

**> file** — направить стандартный поток вывода в файл (перезапись)

**2> file** — направить стандартный поток ошибок в файл (перезапись)

**>>file** — направить стандартный поток вывода в файл (добавление)

**2>>file** — направить стандартный поток ошибок в файл. (добавление)

**&>file** или **>&file** — направить с.п. вывода и с.п. ошибок в файл.



# Перенаправление ввода и вывода

- < file** — использовать файл как источник данных для стандартного потока ввода.
- > file** — направить стандартный поток вывода в файл (перезапись)
- 2> file** — направить стандартный поток ошибок в файл (перезапись)
- >>file** — направить стандартный поток вывода в файл (добавление)
- 2>>file** — направить стандартный поток ошибок в файл. (добавление)
- &>file** или **>&file** — направить с.п. вывода и с.п. ошибок в файл.

```
sab@SAB: /$ ps > 1.txt  
sab@SAB: /$ cat 1.txt  
sab@SAB: /$ ps >> 1.txt  
sab@SAB: /$ ps qq > 1.txt  
sab@SAB: /$ ps qq 2> 1.txt
```



# Pipes (Каналы)

Каналы используются для перенаправления потока из одной программы в другую.

```
sab@SAB: /$ ps | grep p
```

```
sab@SAB: /$ ps > 1.txt; ls >> 1.txt; cat 1.txt | grep a
```



# Несколько полезных команд Linux

**hexdump** — показывает шестнадцатеричное представление данных, поступающих на стандартный поток ввода.

**cat** — считывает данные со стандартного потока ввода и передает их на стандартный поток вывода.

**grep** — возвращает только строки, содержащие (или не содержащие) заданное регулярное выражение.

**sudo** - запуск программы от имени других пользователей, а также от имени суперпользователя.

**bc** — калькулятор

**python3** — среда разработки Python

```
sab@SAB: /$ echo "10*10" | bc
```

```
sab@SAB: /$ hexdump 1.txt
```



**Спасибо за внимание!**