

# Введение в C\C++

Встреча 1 (2 пары)



# Алексей Меркотан

Full stack Developer.  
Teamlead компании MIKO  
Преподаватель академии “ШАГ”

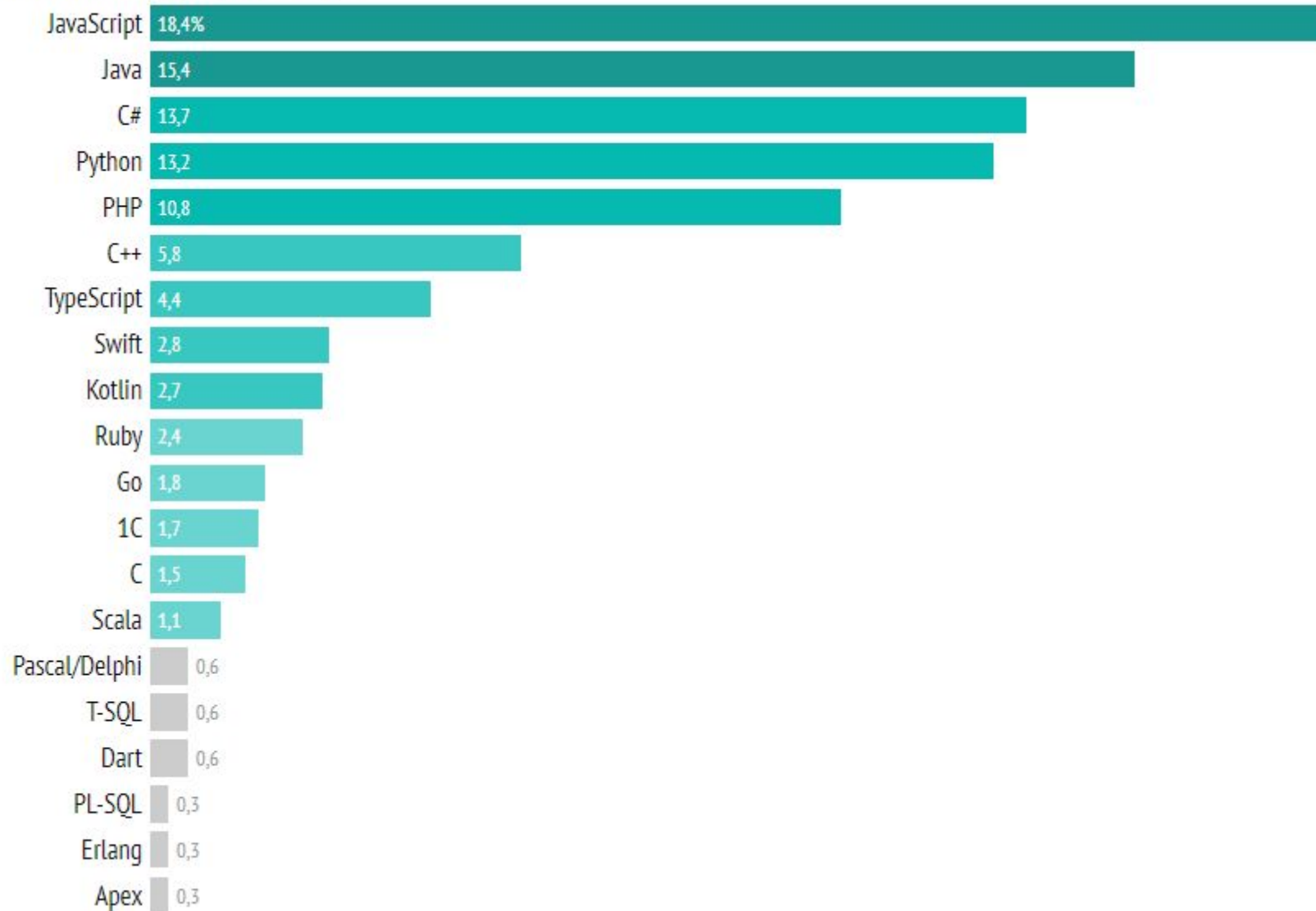
## Стек технологий

Html  
Css  
JavaScript  
jQuery  
Bootstrap  
ReactJS  
Vue.js

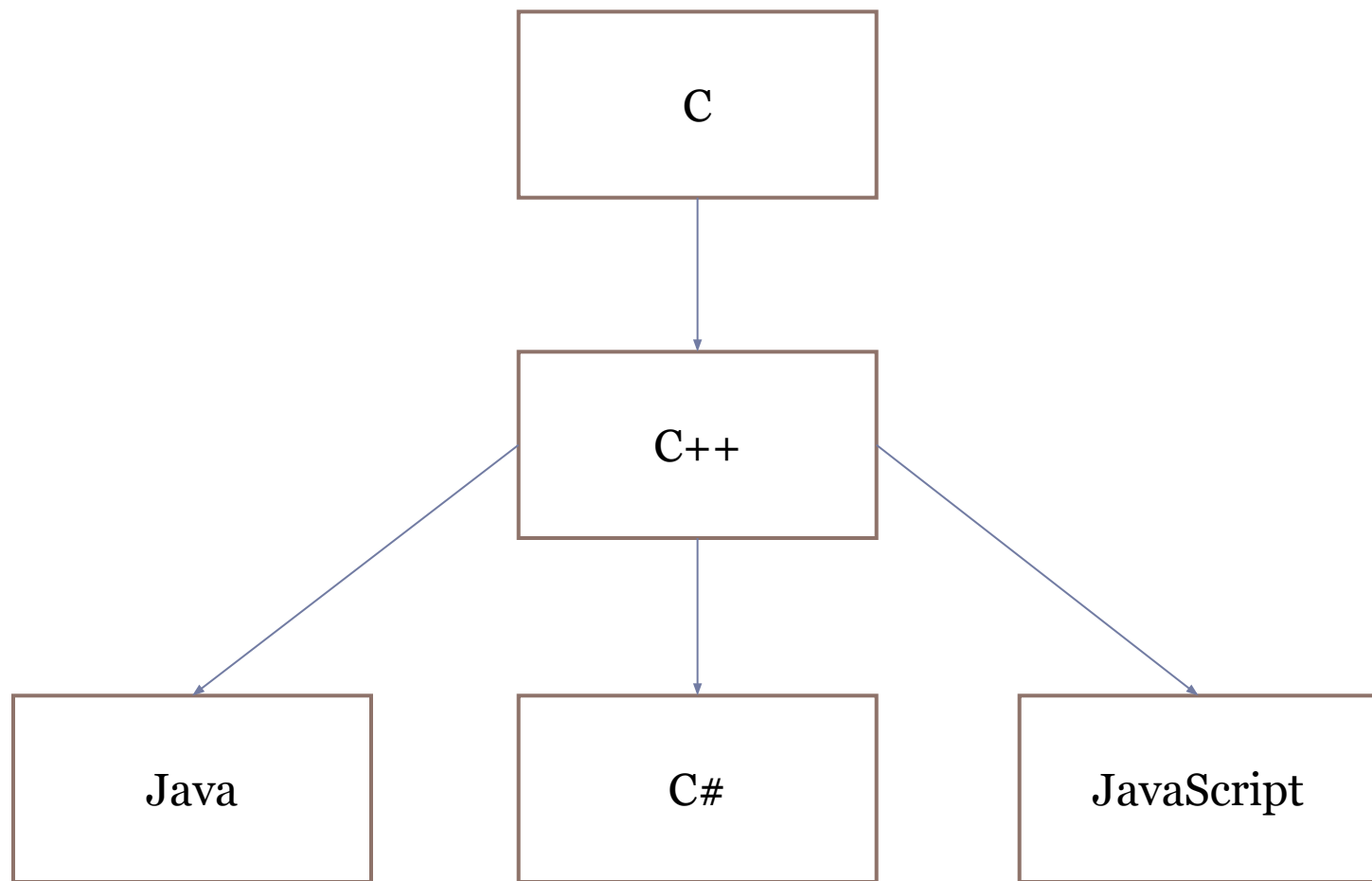
PHP,  
Yii1.x, Yii2.0  
Laravel  
Slim  
C / C++  
C#  
MySQL

# Рейтинг языков программирования 2020

Якою мовою пишете для роботи зараз

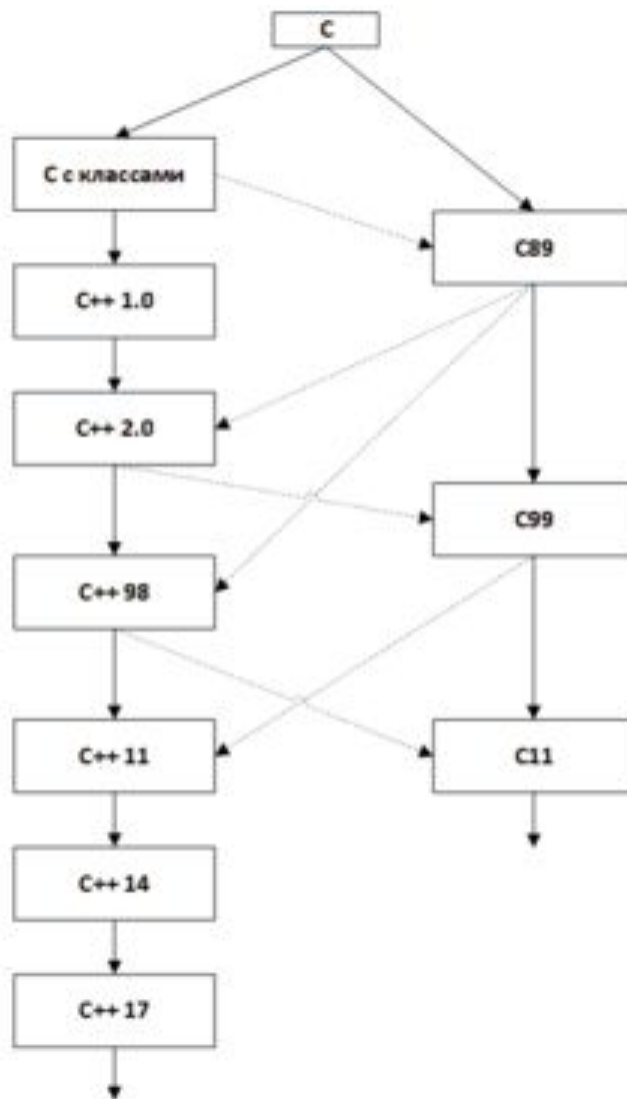


# Генеологическое дерево Си подобных языков



# Развитие C++

Язык программирования C++ был создан в начале 1980-х годов, его создатель сотрудник фирмы Bell Laboratories — Бьёрн Страуструп.

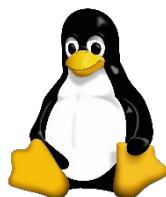


# Области применения

## 1. Операционные системы

- Быстрый и энергоэффективный, он достаточно близок к аппаратному слою, что позволяет работать с кодом низкого уровня.

Вот почему он идеален для разработки операционных систем.



## 2. Программирование встроенных систем

- Язык программирования C просто необходим при проектировании встроенных систем.



### 3. Разработка игр и игрового движка

- Являясь самым быстрым на сегодняшний день языком программирования, С++ оказывается одним из лучших для игр в 3D, многопользовательских и других.



Факт:

В ядре игрового движка Unity — самого популярного движка для создания видеоигр под несколько операционных систем одновременно — также использовался С++.

# Сколько зарабатывают Junior C++

## Software engineer (C++, Qt)

CHM software

**30 000 грн**  Харьков

Уважаемый коллега - разработчик, в нашей компании открылась вакансия Software engineer C++. Мы в поиске человека, который не первый год разрабатывает на...

6 дней назад

 Не интересуется



# Сколько зарабатывают Senior C++

## Embedded C++/Qt Software Engineer

Крива Н.В., ФЛП

**60 000 грн** Киев

Бесплатные обеды

Бонусы / премии

Корпоративные мероприятия

Офис возле метро

Холодильник с едой

Мы - небольшая продуктовая компания, которая работает в сфере теле-радио коммуникаций. У нас есть как программные, так и аппаратные продукты. Создаем...

3 дня назад

Не интересуется

## C++ Developer (relocation to Berlin)

IT Recruitment Solutions (ITRS.ua) ®

**280 000 грн** Другие страны



Бесплатное обучение

Курсы иностранного языка

Медицинская страховка

Корпоративные мероприятия

REQUIREMENTS: Deep knowledge of C++ programming; Expertise with Windows and MacOS will be a plus; Degree in technical sciences; English: Upper-Intermediate...

## Какие требования к Junior C++

- ◆ 1+ год опыта программирования на языке C++.
- ◆ Опыт работы с STL/Boost/Qt.
- ◆ Понимание принципов ООП и шаблонов проектирования.
- ◆ Опыт работы с базами данных и знание языка SQL.
- ◆ Понимание концепции многопоточного программирования.
- ◆ А также высокая степень самостоятельности и ответственности в работе.

Будет плюсом:

- ◆ Знание стандарта C++11 и выше.
- ◆ QML.
- ◆ Опыт работы с оборудованием или написание драйверов.
- ◆ Опыт разработки серверных решений.
- ◆ Написание unit тестов.
- ◆ Умение оптимизировать код.

Условия работы следующие:

- ◆ В офисе на полную занятость.
- ◆ Удобное месторасположение офиса.
- ◆ Комфортная рабочая обстановка.
- ◆ Дружный и позитивный коллектив.
- ◆ Корпоративные мероприятия.
- ◆ Чай, кофе.

Ждём твое резюме и пример кода (портфолио, если есть). Пожелания по уровню заработной платы мы будем обсуждать до и после нашего с тобой собеседования. Надеемся, что именно ты станешь частью нашей команды!

# Программа обучения по курсу

Модуль 1. Введение в язык программирования C++

Модуль 2. Переменные и типы данных

Модуль 3. Логические операторы и операторы ветвлений

Модуль 4. Циклы, использование отладчика

Модуль 5. Массивы: одномерные и многомерные

Модуль 6. Функции

Модуль 7. Указатели

Модуль 8. Строки

Модуль 9. Многомерные динамические массивы

Модуль 10. Структуры

Модуль 11. Препроцессор в приложениях

Модуль 12. Использование файловой системы

Модуль 13. Экзамен

## По окончании курса Вы будете:

- понимать, что такое алгоритм;
- уметь разрабатывать алгоритмы;
- уметь строить блок-схемы;
- использовать циклы, условия и другие базовые конструкции;
- оперировать основами языка программирования С;
- понимать и использовать отладчик;
- создавать функции;
- использовать одномерные и многомерные массивы;

## Этапы решения задачи на компьютере

1. Постановка задачи
2. Построение математической модели
3. Разработка алгоритма
4. Составление программы
5. Компиляция программы
6. Компоновка программы
7. Настройка программы
8. Эксплуатация программы

# Определение понятия алгоритма. Свойства алгоритма. Типы алгоритмов.

**Алгоритм** - это словесный, рукописный или схематично описание этапов работы (деятельности) любого объекта с определенной степенью детализации.

## Свойства алгоритма:

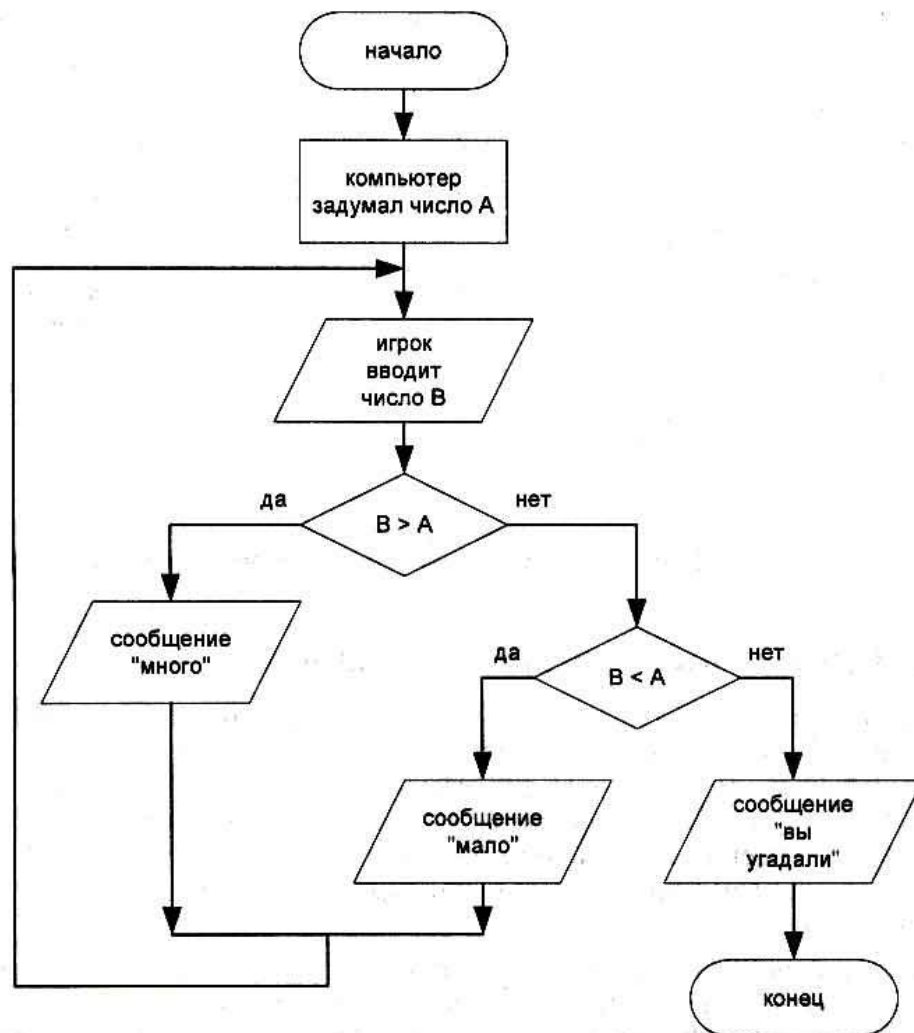
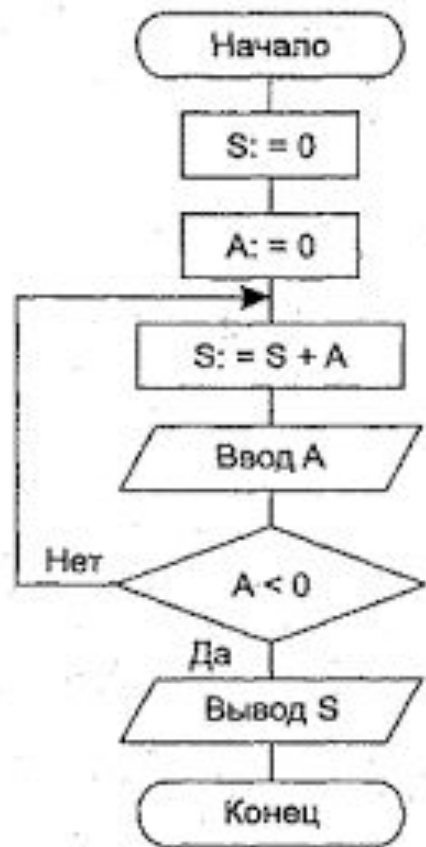
- Определенность (детерминированность)
- Массовость
- Дискретность
- Результативность
- Формальность
- Конечность

## Типы алгоритмов:

- линейный алгоритм ;
- разветвленный алгоритм;
- циклический алгоритм.

# Что такое блок-схема?

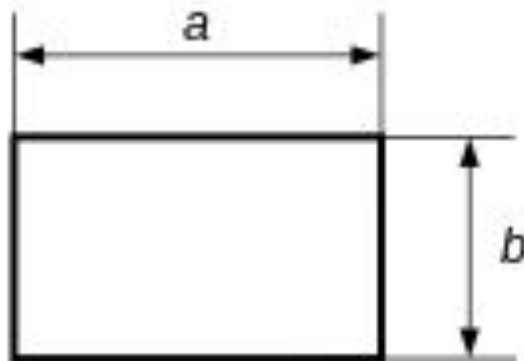
**Блок-схема** — распространённый тип схем (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности.



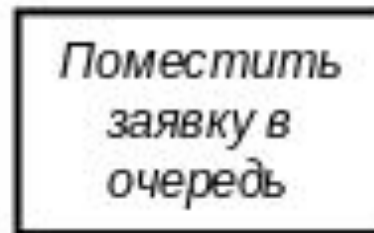
# Действие

Символ отображает функцию обработки данных любого вида (выполнение определённой операции или группы операций, приводящее к изменению значения, формы или размещения информации или к определению, по которому из нескольких направлений потока следует двигаться).

*Начертание*



*Пример*

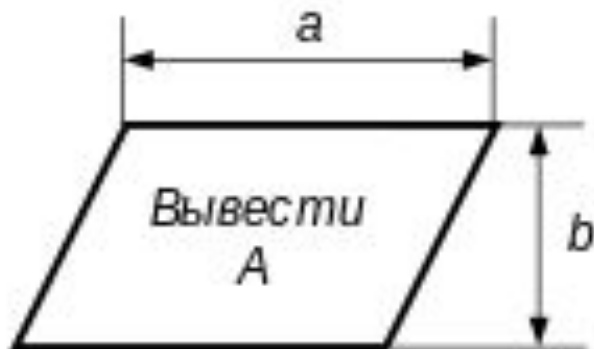




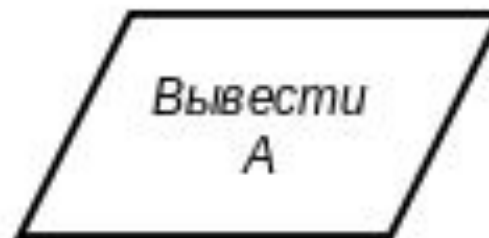
# Данные (ввод-вывод)

Символ отображает данные, носитель данных не определён.

*Начертание*



*Пример*



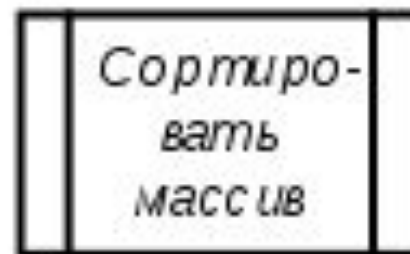
## Предопределённый процесс (функция)

Символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле). Например, в программировании – вызов процедуры или функции.

*Начертание*

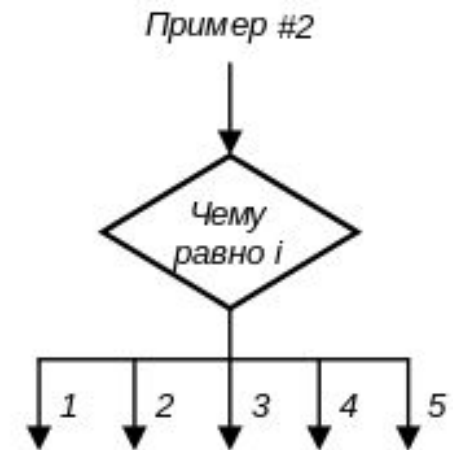
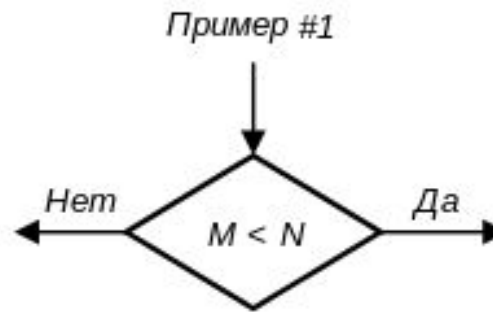
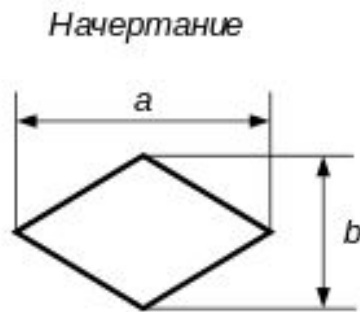


*Пример*



# Вопрос (условие или решение)

Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определённых внутри этого символа. Соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути.

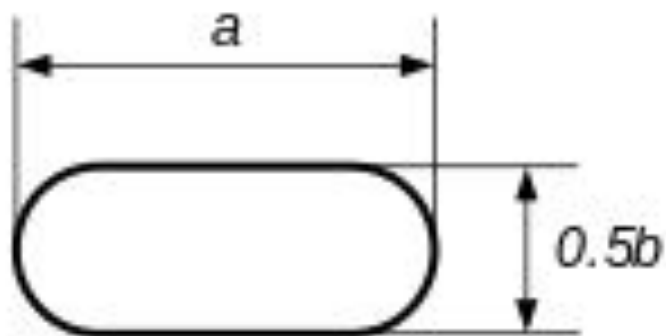


Отображает решение или функцию переключательного типа с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран после вычисления условий, определённых внутри этого элемента. Вход в элемент обозначается линией, входящей обычно в верхнюю вершину элемента. Если выходов два или три, то обычно каждый выход обозначается линией, выходящей из оставшихся вершин (боковых и нижней). Если выходов больше трех, то их следует показывать одной линией, выходящей из вершины (чаще нижней) элемента, которая затем разветвляется. Соответствующие результаты вычислений могут записываться рядом с линиями, отображающими эти пути. Примеры решения: в общем случае – сравнение (три выхода:  $>$ ,  $<$ ,  $=$ ); в программировании – условные операторы *if* (два выхода: *true*, *false*) и *case* (множество выходов).

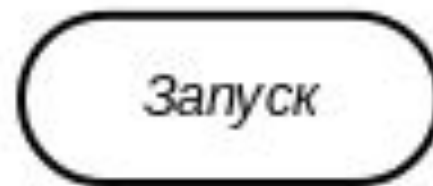
# Ограничитель

Символ отображает вход из внешней среды и выход во внешнюю среду (начало или конец схемы программы, внешнее использование и источник или пункт назначения данных).

*Начертание*

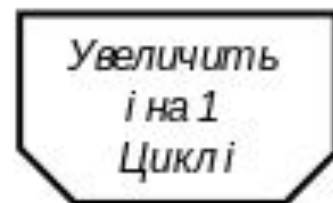


*Пример*

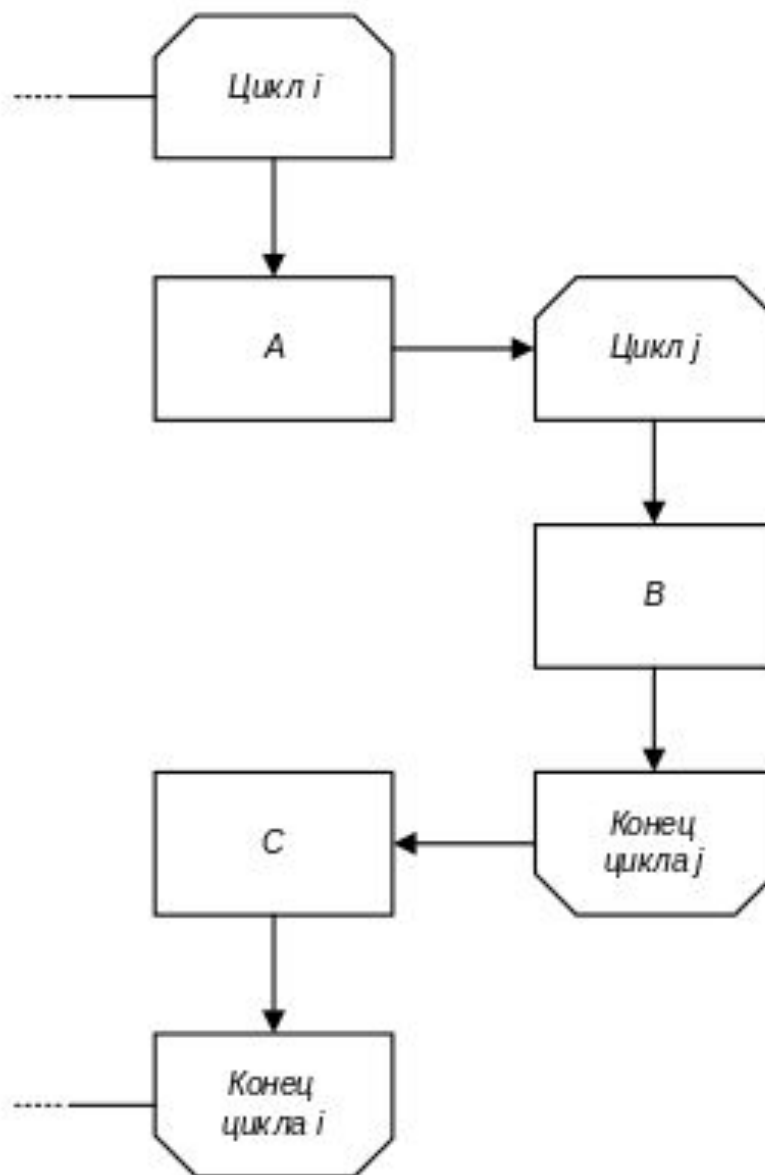


# Цикл

Символ, состоящий из двух частей, отображает начало и конец цикла. Обе части символа имеют один и тот же идентификатор. Условия для инициализации, приращения, завершения и т. д. помещаются внутри символа в начале или в конце в зависимости от расположения операции, проверяющей условие.



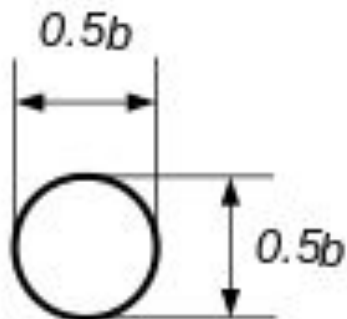
# Пример вложенных циклов



# Соединитель

Символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения её в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение

Начертание



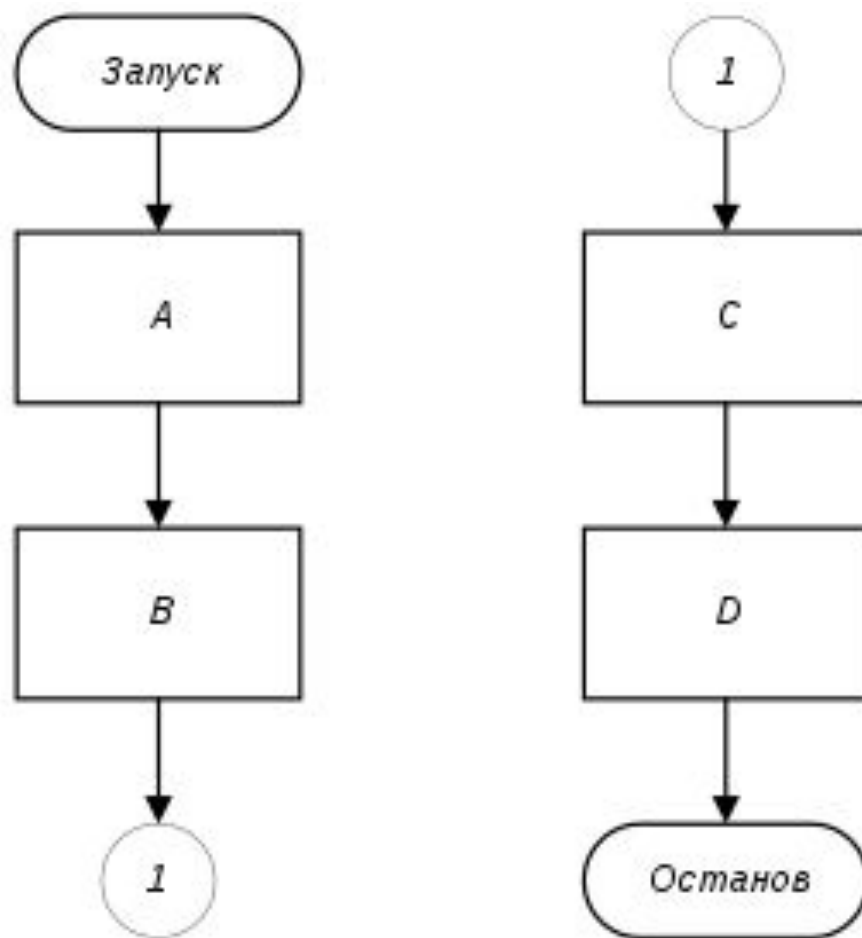
Пример #1



Пример #2



# Разделение алгоритма на две части с использованием соединителей

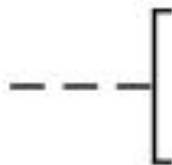




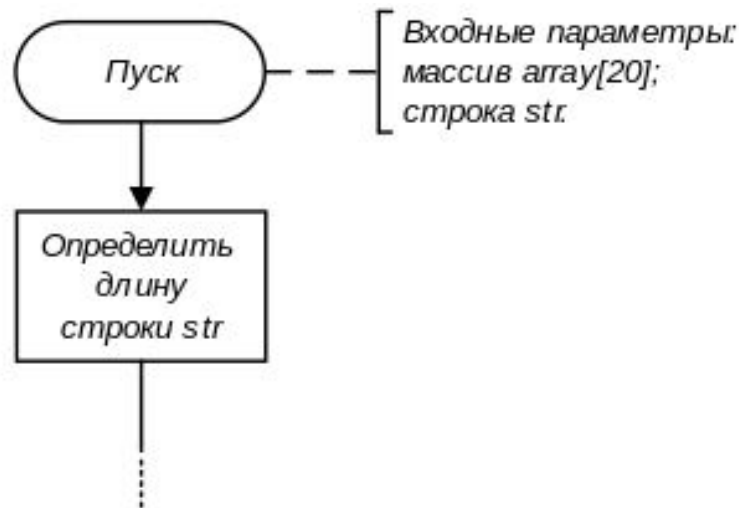
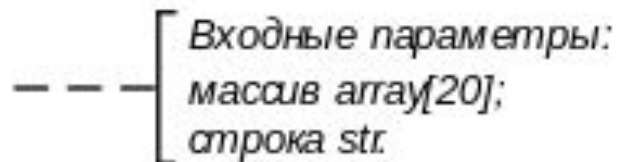
# Комментарий



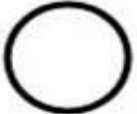
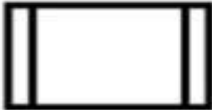

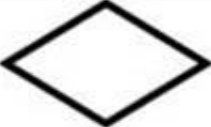
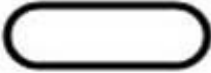
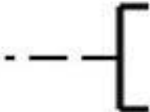
Символ используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обводить группу символов. Текст комментариев или примечаний должен быть помещен около ограничивающей фигуры.

*Начертание*



*Пример*



Символ	Название	Назначение
	Данные	Общее обозначение ввода или вывода данных
	Процесс	Обработка данных, операция или группа операций
	Соединитель	Соединение прерванных линий потока
	Предопределенный процесс	Вычисления по подпрограмме (модулю)
	Подготовка	Осуществляет задание изменений параметров цикла
	Решение	Проверка условия
	Терминатор	Вход или выход во внешнюю среду
	Комментарий	Для записи пояснений к алгоритму

**Спасибо за внимание.  
Вопросы?**