

```
window.yaContextCb.push(=>{ Ya.Context.AdvManager.render({ "blockId":  
"R-A-7038977-3", "renderTo": "yandex_rtb_R-A-7038977-3" }) })
```



Что за комплект.

Электронная книга + видеокурс по программированию об использовании указателей и динамических структур данных.

Автор Андрей Поляков.

Что вошло в комплект.

Содержание.

- **Введение.**
- **1. Динамические структуры данных.** Общие сведения. Динамическая и статическая память. Структуры данных. Что такое адрес. Динамически распределяемая память ("куча").
 - **2. Что такое указатель.** Что такое указатель. Что такое ссылочный тип данных. Что хранят указатели. Как связываются указатели с памятью компьютера. Адреса.
 - **2.1. Типизированные и нетипизированные указатели.** Типизированные и нетипизированные указатели. Что такое типизированные и нетипизированные указатели. Зачем вообще этот урок.
 - **2.2. Объявление указателей.** Как объявляются указатели в языках Паскаль и C++. Чем отличаются объявления указателей в этих языках. Какие операции можно выполнять над указателями...
 - **2.3. Операторы для работы с указателями.** Как получить адрес переменной. Как получить значение из памяти, на которую указывает указатель. Как записать значение в нужный адрес памяти. Как вывести адрес на экран. Как изменять значения переменных, не используя их имена.
 - **2.4. Пример размещения массива в памяти.** Как располагаются элементы массива в памяти. Сколько памяти занимает каждый элемент массива. Как перебрать элементы массива в цикле, используя указатели. Как получить адрес элемента массива. Как работает функция `Inc` со ссылочными переменными. Почему при работе с указателями можно получить неожиданный результат. Почему адрес памяти является числом, но с ним нельзя работать как с числом.
 - **2.5. Указатель на массив и массив указателей.** Что такое указатель на массив и что такое массив указателей. Чем отличаются. Какие ошибки можно совершить при объявлении переменных. Как объявить массив указателей. Как объявить указатель на массив.
 - **2.6. Методы работы с памятью.** Особенности выделения памяти и возможные ошибки. Инициализация памяти. Процедуры для заполнения памяти. Освобождение памяти: как и для чего делается.
 - **2.7. Возможные проблемы с указателями.** Потеря указателя: как это

происходит и какие могут быть последствия. Как использование указателей может повлиять на скорость работы программы. В каких случаях указатели лучше не использовать. Примеры программ для понимания обозначенных проблем.

- **2.8. Передача указателей функциям и процедурам.** Передача аргументов по значению. Передача значений указателей. Возможные ошибки при передаче указателей в подпрограммы. Как изменить значение глобальной переменной в подпрограмме, не используя имя переменной.

- **2.9. Использование нетипизированных указателей.** Объявление нетипизированных указателей. Выделение и освобождение блока памяти, с которым связан указатель. Процедуры и функции для выделения и освобождения памяти. Определение размера блока памяти. Изменение поведения программы с помощью предопределённой переменной `ReturnNilIfGrowHeapFails`.

- **3. Списки, деревья, графы.** Общие сведения, введение. И напоминание о том, что такое записи (структуры).

- **3.1. Односвязный список.** Что такое односвязный (однонаправленный) список. В каких случаях лучше использовать списки вместо статических массивов. Из чего состоит элемент списка. Как вставить новый элемент в список. Как удалить элемент из списка. Как получить доступ к данным в списке.

- **3.2. Двусвязный список.** Что такое двусвязный (двунаправленный) список. Из чего состоит элемент списка. Как вставить новый элемент в список (в начале, в конце и в произвольном месте)...

- **3.3. Бинарные деревья.** 8 видеоуроков общей продолжительностью 1,5 часа. Что такое бинарные деревья. Зачем нужны бинарные деревья. Что такое рекурсия и как её использовать. Как создать новый узел в бинарном дереве. Что такое бинарное дерево поиска и как создать узел в таком дереве...

- **3.5. Стек.** Что такое стек. Что такое LIFO. Организация стека на простом примере.

- **3.6. Очередь.** Что такое очередь. Что такое FIFO. Организация очереди на простом примере.

- **4.1. Динамические массивы.** Общие сведения о том, где применяются динамические структуры данных. Пример использования указателей и динамических структур данных при создании динамического массива. Использование стандартных динамических массивов в Lazarus.

- **4.2. Стековый калькулятор и обратная польская запись.** Пример использования стека. Что такое польская запись (прямая и обратная). Стековый калькулятор. Пример создания простого стекового калькулятора (с исходными кодами).

- **4.3. Использование очереди.** Пример использования очереди.

Дополнительную информацию смотрите на сайте комплекта:
[подробности \(перейти на страницу комплекта\)](#)

[Узнать](#)

** Карточка курса:** * Название: Куда указывают указатели

Автор(ы): Поляков Андрей

Формат курса: Комплект

Цена: 960 руб.

Партнерская программа: Есть

*-данные карточки курса могут не соответствовать текущим данным о курсе, подробную информацию смотрите на сайте курса

** Другие курсы автора(ов), издателя:** Ссылки на страницы автора(ов),
издателя: Поляков Андрей

```
window.yaContextCb.push(()=>{ Ya.Context.AdvManager.render({ "blockId":  
"R-A-7038977-2", "renderTo": "yandex_rtb_R-A-7038977-2" }) })
```