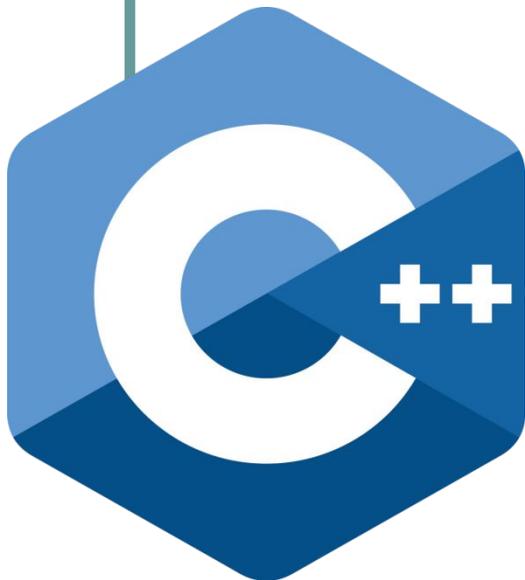


Programming Introduction



Преподаватель

Александр Загоруйко
zagorujko@itstep.academy



Первое задание

- Добавиться в телегу: 063 03 000 35

Telegram



План презентации

- Программа курса ПП
- Литература и полезные сайты
- Создание проекта в Visual Studio 2017
- Запуск HelloWorld-приложения
- Вывод данных, типы данных
- Переменные и константы
- Ввод данных
- Практика



Программа курса

- Введение в процедурное программирование
- Вывод и ввод данных в консоль
- Условные операторы, битовые операции
- Циклы, псевдографика, работа с отладчиком
- Массивы, случайные числа, двумерные игры
- Алгоритмы поиска и сортировки
- Функции, шаблоны функций, рекурсия
- Указатели, ссылки, динамическая память
- Строки, регулярные выражения, структуры
- Устное собеседование с заведующим кафедры

Структура занятия

- Ответы на вопросы по ДЗ
- Краткое повторение прошлой темы
- Новая тема (презентация)
- Перемена по желанию 10 минут
- Продолжение презентации
- Практика (пишем код)
- Выдача домашнего задания



Примеры практики и ДЗ

1. Написать программу, которая находит среднее арифметическое значение трёх вещественных чисел.
2. Написать программу, которая находит корень линейного уравнения $ax + b = 0$.
3. Пользователь вводит число и степень. Программа вычисляет указанную степень этого числа (пригодится функция `pow`).
4. Написать программу, которая предлагает пользователю ввести радиус окружности, а затем считает площадь и длину этой окружности. Число Pi задать в программе как вещественную константу.
5. Написать программу, которая предоставляет пользователю возможность ввести с клавиатуры количество гривен, и переводит это количество в доллары, евро и биткоины.
6. Написать программу, которая переводит километры в сухопутные и морские мили.
7. Написать программу, которая находит процент P от числа N .
8. Написать программу, которая переводит указанное количество градусов по Цельсию в градусы по шкале Фаренгейта, Кельвина, Реомюра и Делиля. Затем реализовать перевод из градусов по Фаренгейту в градусы по Цельсию.

В каждом блоке в среднем по 12 заданий разных уровней сложности.

Файлы заданий

 0. cout	10/30/2017 3:19 PM	Документ Micros...	132 КБ
 1. cin	12/2/2017 10:06 PM	Документ Micros...	30 КБ
 2. casting	10/30/2017 2:53 PM	Документ Micros...	33 КБ
 3. reminder	10/30/2017 2:53 PM	Документ Micros...	32 КБ
 4. conditions part 1	10/30/2017 2:52 PM	Документ Micros...	30 КБ
 5. conditions part 2	10/30/2017 2:53 PM	Документ Micros...	33 КБ
 6. bit operations	10/30/2017 2:51 PM	Документ Micros...	28 КБ
 7. loops part 1	10/30/2017 2:54 PM	Документ Micros...	34 КБ
 8. loops part 2	10/30/2017 3:51 PM	Документ Micros...	197 КБ
 9. arrays part 1	10/30/2017 2:53 PM	Документ Micros...	32 КБ
 10. arrays part 2	10/30/2017 3:15 PM	Документ Micros...	123 КБ
 11. arrays final task	11/8/2017 6:34 PM	Документ Micros...	37 КБ
 12. strings	10/30/2017 2:57 PM	Документ Micros...	49 КБ
 13. functions	10/30/2017 2:54 PM	Документ Micros...	34 КБ
 14. recursion	10/30/2017 2:57 PM	Документ Micros...	45 КБ
 15. structures	10/30/2017 4:24 PM	Документ Micros...	273 КБ

Файлы презентаций

 0. hello world	12/2/2017 10:08 PM	Презентация Міс...	4,002 КБ
 1. data types, variables	11/1/2017 11:11 PM	Презентация Міс...	507 КБ
 2. algorithms	11/2/2017 12:13 AM	Презентация Міс...	1,639 КБ
 3. casting	11/1/2017 10:13 PM	Презентация Міс...	314 КБ
 4. conditions	11/1/2017 10:33 PM	Презентация Міс...	349 КБ
 5. loops	12/2/2017 9:34 PM	Презентация Міс...	710 КБ
 6. nested loops	12/2/2017 9:34 PM	Презентация Міс...	599 КБ
 7. arrays	11/8/2017 8:23 PM	Презентация Міс...	654 КБ
 8. sorting	11/8/2017 8:28 PM	Презентация Міс...	7,815 КБ
 9. 2D-arrays	11/8/2017 6:35 PM	Презентация Міс...	1,595 КБ
 10. strings	11/14/2017 3:35 PM	Презентация Міс...	633 КБ
 11. regex	11/14/2017 3:36 PM	Презентация Міс...	2,494 КБ
 12. functions	11/1/2017 10:30 PM	Презентация Міс...	342 КБ
 13. recursion	11/2/2017 12:34 AM	Презентация Міс...	3,763 КБ
 14. structures	11/2/2017 12:34 AM	Презентация Міс...	3,763 КБ

Mystat

ВОЙТИ

ПОЛУЧЕНИЕ

Войти

Alex

Пароль

.....

Город

Одесса

ВХОД

ЗАБЫЛИ ПАРОЛЬ?

RU EN BG PT UA RO GE AZ CS

SK ES PL

Сдача ДЗ

	ФИО студента	31.10.17	17.10.17	13.10.17	10.10.17	06.10.17	29.09.17	29.09.17	26.09.17
1	 Абдул Муджиб Джабарович (Допуск с разрешения бухгалтерии)	- ▾	+ 8	+ 12	+ 12	+ 12	+ 11	+ 12	+ 12
2	 Куцаков Кирило Владиславович	- ▾	+ 12	+ 12	+ 12	+ 12	+ 12	+ 12	+ 11
3	 Луценко Антон Олегович	- ▾	- ▾	- ▾	- ▾	- ▾	- ▾	- ▾	- ▾
4	 Михайленко Сергій Леонідович	- ▾	- ▾	- ▾	- ▾	- ▾	+ 12	+ 12	+ 12
5	 Соловей Михайло Васильович	- ▾	+ 12	- ▾	+ 12	+ 12	+ 12	+ 12	- ▾
6	 Сторожева Наталія Григорівна	- ▾	+ 12	+ 12	+ 12	+ 12	+ 12	+ 11	+ 12
7	 Тимчишин Вадим Станиславович	- ▾	- ▾	- ▾	- ▾	- ▾	+ 12	- ▾	+ 12
8	 Хорзова Лілія Сергіївна	- ▾	- ▾	- ▾	- ▾	+ 12	+ 12	+ 12	+ 12

Файловый сервер

\\fs \ Преподаватели
\ Загоруйко А \ Public

Литература по C++

<https://drive.google.com/open?id=17wEY2Uv7GA3Wor-2waLPqVD-CN9wE9lw>

- Прата
- Шилдт



https://www.youtube.com/playlist?list=PL_TCIVgAqn3qMPCYSjOgpzjMDstkNZOLb

Литература must read

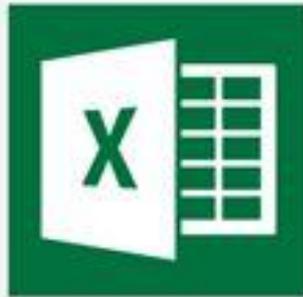
<https://drive.google.com/open?id=1eDH3NpXujoyqr5PdKCwZKegvyLU3G0tu2>

- Для (будущих) тимлидов
- Про компиляторы
- Личностное развитие
- Проектирование
- Публичные выступления
- Собеседования
- Совершенный код

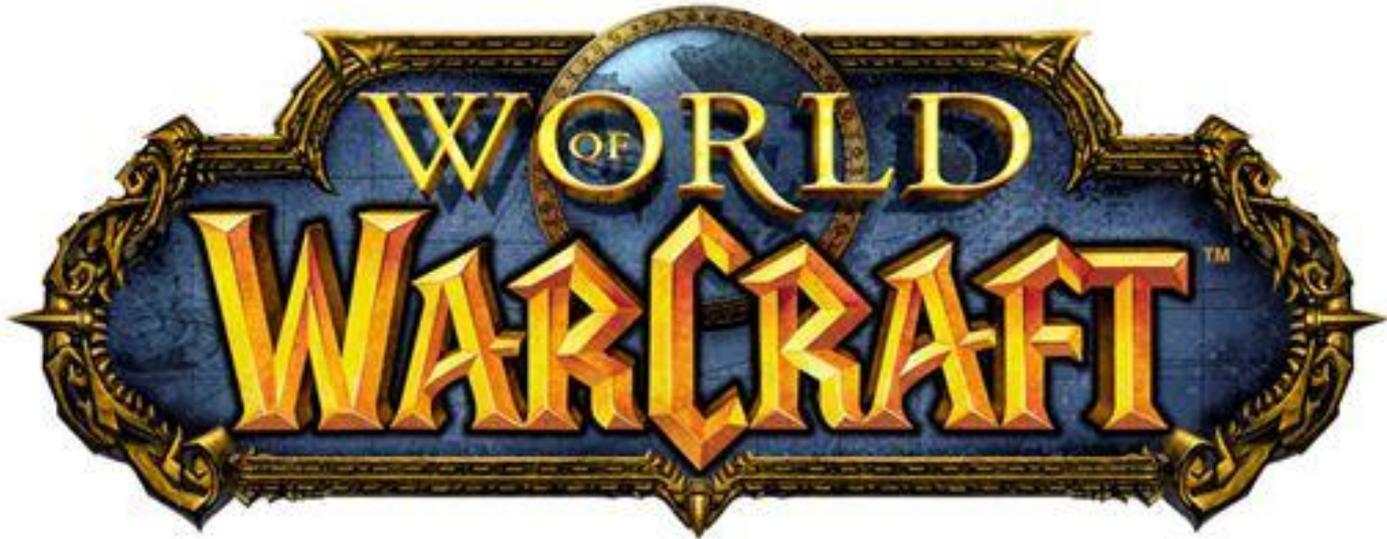
Что пишут на C и C++



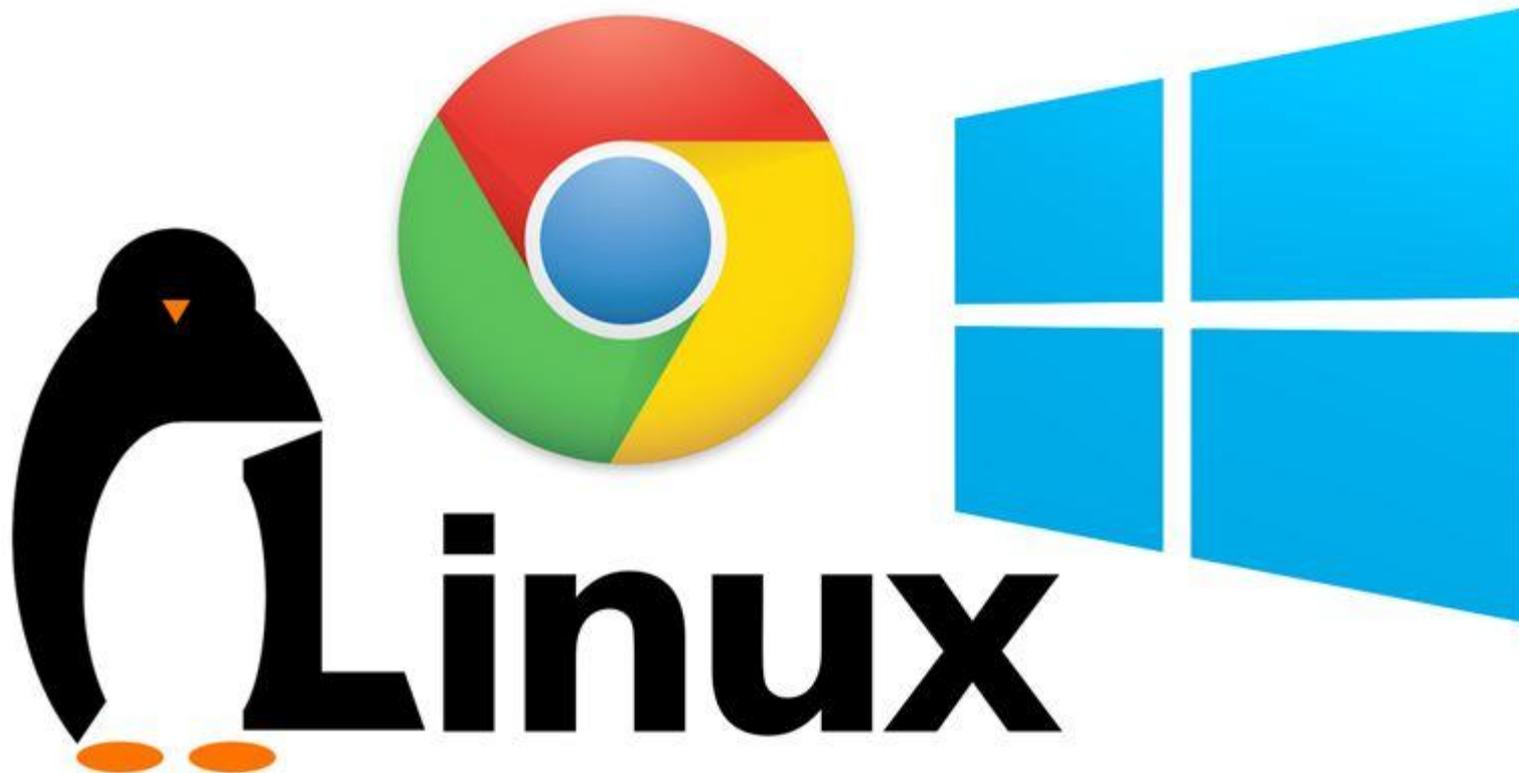
Что пишут на C и C++



Что пишут на C и C++



Что пишут на C и C++



Вакансии для программистов

← → ↻ Надежный | <https://jobs.dou.ua/vacancies/?city=Одесса&category=C%2B%2B>

23 вакансии в категории C++ в Одессе [RSS](#)

 **Lead 3D Developer** в  3DLOOK до \$5000  Киев, Одесса

3DLOOK develops an instant human body scanning technology that works on any mobile device and doesn't require any additional hardware.

 **Middle C++ Developer** в  DataArt  Киев, Харьков, Львов, Днепр, Одесса, Херсон, Люблин (Польша), Вроцлав (Польша)

DataArt has been engaged in software development and IT consulting since 1997. Companies that create interesting products and services have complex business processes, they want to beat the competition with the help of technology, and usually they turn to us.

https://t.me/junior_dou_ua <<< !!!

C++ разработчик в  Infomir  Одесса

Компания-производитель клиентского IPTV оборудования, сетевых медиаплееров, светодиодного освещения и систем учета ресурсов открывает конкурс на замещение вакансии C++ разработчик.

Senior C++ Generalist, OLM STM в  Intellias  Киев, Львов, Одесса

Work on the project Here is an opportunity to work on the navigation systems of leading automotive manufacturers in the world.

C++ Game Developer (Odessa) в  Ubisoft  Одесса

Company Description Ubisoft, an industry leading developer of video games offers a unique environment where creativity, teamwork and cutting-edge technology bring to life critically acclaimed video games and iconic AAA franchises.

C++ Qt engineer for ITSM project в  Comodo Group  Одесса

ITSM (IT & Security Manager) is online system designed for managing devices (Windows, Android, iOS). It gives company a possibility to put proper restrictions to devices, to lock device if it has been stolen, to configure wifi, vpn and other sections from the portal.

Город

Одесса

В других городах:

[Киев](#) 68

[Одесса](#) 23

[Львов](#) 21

[Харьков](#) 15

[Днепр](#) 6

[Пало-Альто \(США\)](#) 2

[Черновцы](#) 2

[Вроцлав \(Польша\)](#) 1

[Херсон](#) 1

[Ровно](#) 1

[Хмельницкий](#) 1

[Ивано-Франковск](#) 1

[Винница](#) 1

[Люблин \(Польша\)](#) 1

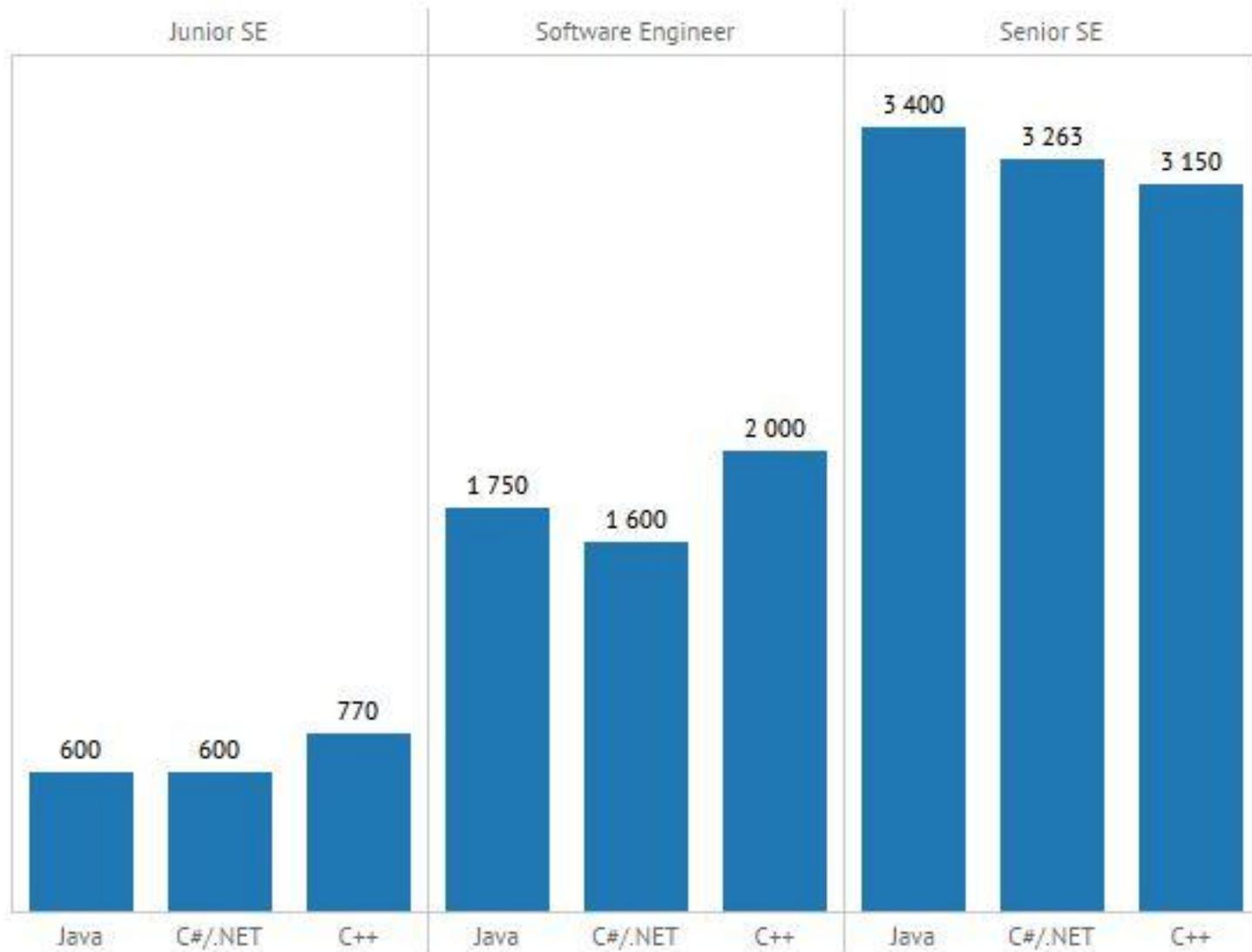
[Сумы](#) 1

[удаленная работа](#) 5

[работа за рубежом](#) 7

<https://jobs.dou.ua/vacancies/?city=%D0%B4%D0%B5%D1%81%D1%81>

Зарплаты (Java, .Net, C++)



Много букв про язык Си

Си — компилируемый статически типизированный язык программирования, разработанный в 1969 — 1973 годах сотрудником Bell Labs Деннисом Ритчи как развитие языка Би. Первоначально был разработан для реализации операционной системы UNIX, но впоследствии был перенесён на множество других платформ. Согласно дизайну языка, его конструкции близко сопоставляются типичным машинным инструкциям, благодаря чему он нашёл применение в проектах, для которых был свойственен язык ассемблера, в том числе как в операционных системах, так и в различном прикладном программном обеспечении для множества устройств — от суперкомпьютеров до встраиваемых систем. Язык программирования Си оказал существенное влияние на развитие индустрии программного обеспечения, а его синтаксис стал основой для таких языков программирования, как C++, C#, Java и Objective-C.

Идея языка Си

Си используют для того, чтобы понять как работает ПК, так как он не скрывает многое «закулисье».

Си написали поверх ассемблера, чтобы было проще писать, но к сожалению не думали про удобство, поэтому можно считать, что этот язык создан «монстрами для монстров» 😊

Деннис Ритчи и Кен Томпсон

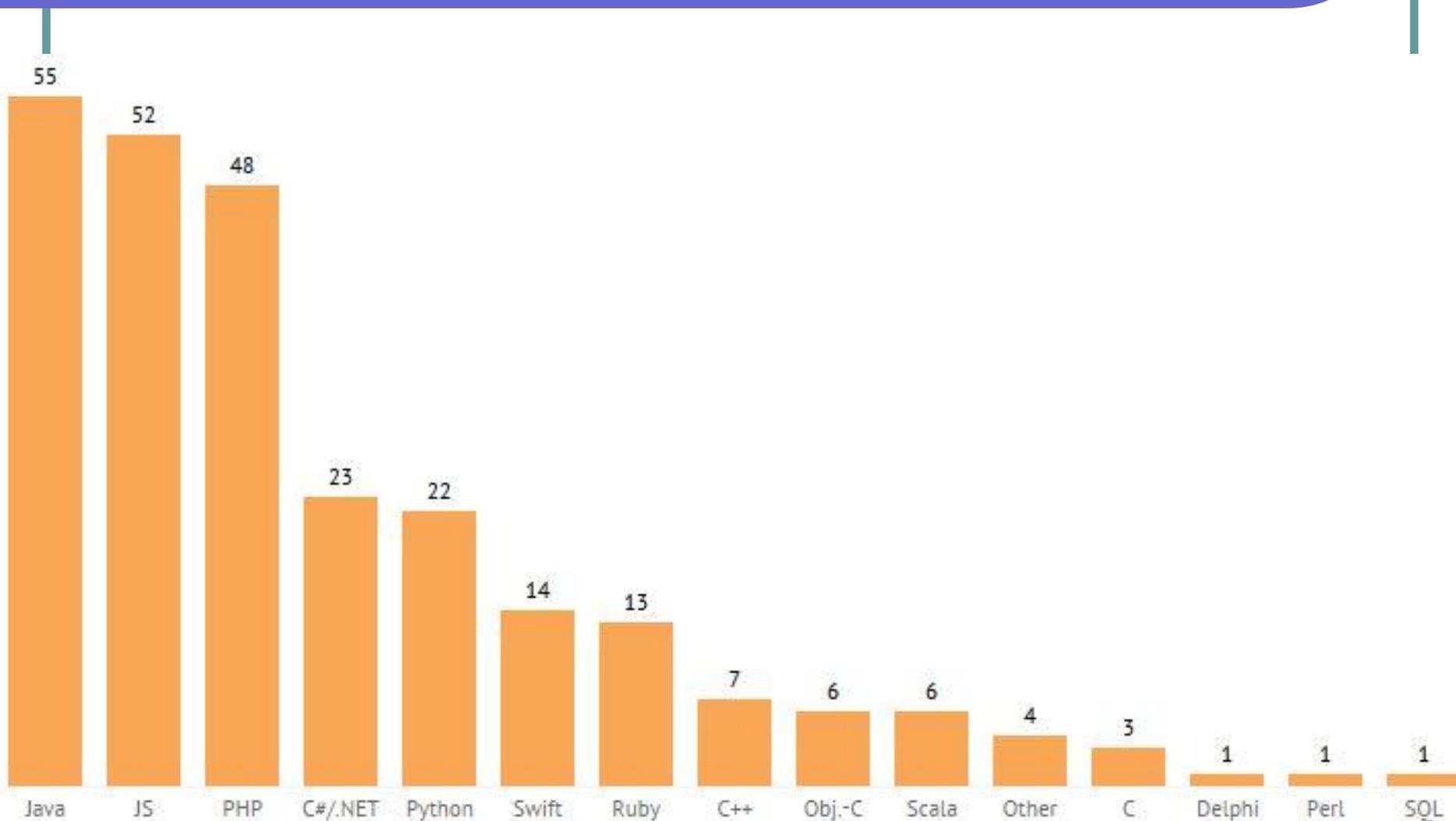


Рейтинг языков

Nov 2017	Nov 2016	Change	Programming Language	Ratings	Change
1	1		Java	13.231%	-5.52%
2	2		C	9.293%	+0.09%
3	3		C++	5.343%	-0.07%
4	5	▲	Python	4.482%	+0.91%
5	4	▼	C#	3.012%	-0.65%
6	8	▲	JavaScript	2.972%	+0.27%
7	6	▼	Visual Basic .NET	2.909%	-0.26%
8	7	▼	PHP	1.897%	-1.23%
9	16	▲▲	Delphi/Object Pascal	1.744%	-0.21%
10	9	▼	Assembly language	1.722%	-0.72%
11	19	▲▲	R	1.605%	-0.11%
12	15	▲	MATLAB	1.604%	-0.36%
13	14	▲	Ruby	1.593%	-0.39%

<https://www.tiobe.com/tiobe-index/>

В стартапах программируют на:



Какой язык выбрать?

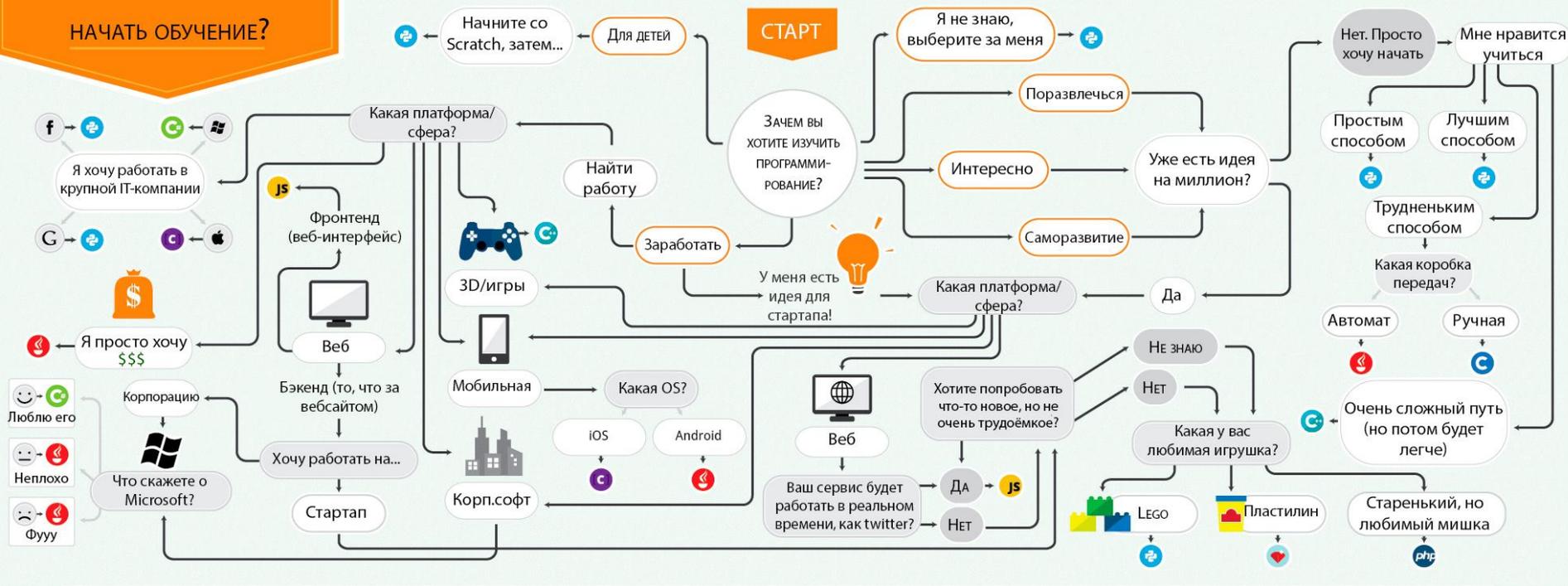
С КАКОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ НАЧАТЬ ОБУЧЕНИЕ?

ЧТО ТАКОЕ ПРОГРАММИРОВАНИЕ?

Написание крайне специфичных инструкций глупой, но очень послушной машине



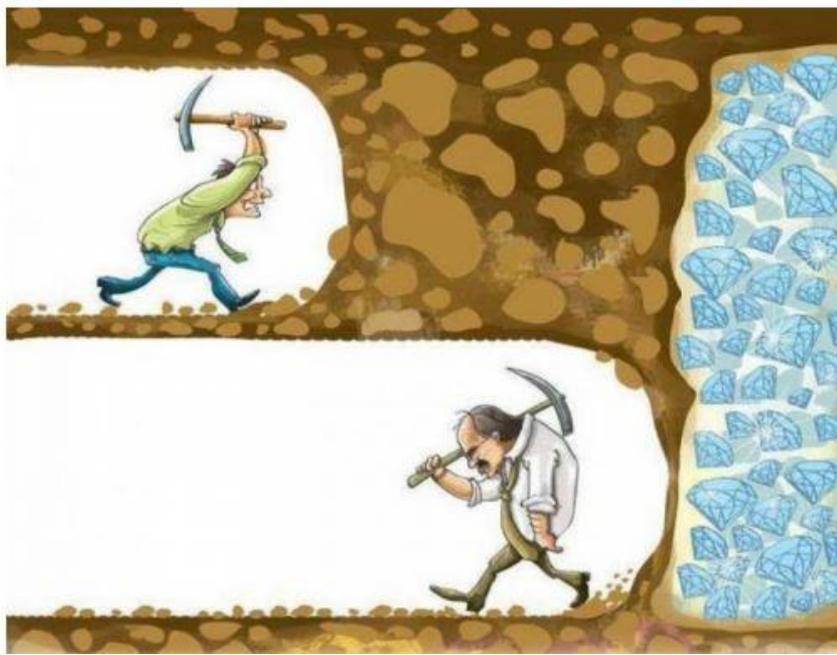
ЯЗЫКИ



Советы для начинающих

- <https://proglib.io/p/some-advice-for-newbie/>
- <https://proglib.io/p/how-to-google-as-a-programmer/>

Я хотел бы знать массу всего, когда только начинал, но эти 27 вещей первыми приходят в голову.



Что такое программирование?

Программирование – это процесс разработки и поддержки программного обеспечения.

В узком смысле программированием называют процесс написания программы на конкретном языке программирования (этап кодирования).

Этапы разработки ПО

- **Анализ задачи** – этап формализации задачи и определения методов её решения
- **Проектирование** – этап разработки структуры системы, определения способов взаимодействия компонентов, выбора структур для хранения данных, выбор алгоритма и т.д.
- **Кодирование** – этап написания программного кода компонентов системы
- **Тестирование** – этап исследования и испытания программного продукта
- **Документирование** – этап составления руководств и документации к программному продукту
- **Сопровождение** – этап улучшения, оптимизации ПО после передачи в эксплуатацию

Что такое программа?

Программа — представленная в объективной форме совокупность данных и команд, предназначенных для выполнения ЭВМ с целью получения определённого результата.

Данные – это информация, которую обрабатывает программа.

Что такое алгоритм?

Алгоритм – это точный набор инструкций, описывающий порядок действий исполнителя для достижения цели – решения задачи.



Свойства алгоритма

- **Понятность** – алгоритм должен включать только те инструкции, которые доступны и понятны исполнителю
- **Определённость** – в любой момент времени следующий шаг работы однозначно определяется состоянием системы
- **Конечность** – алгоритм должен состоять из конечной последовательности шагов
- **Результативность** – алгоритм должен завершиться с определённым результатом – решением поставленной задачи
- **Массовость** – алгоритм может применяться к некоторому набору входных данных
- **Эффективность** – алгоритм должен минимально расходовать ресурсы

Способы записи алгоритмов

- Словесная или вербальная форма
- Псевдокод
- Блок-схема – схематическая запись алгоритма в виде последовательности блоков и соединительных линий, СВЯЗЫВАЮЩИХ ИХ

<https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC>

Пример вербального алгоритма

Алгоритм перехода дороги пешеходом, с соблюдением ПДД.

1. Подходим к дороге;
2. Смотрим налево;
3. Если видим автомобиль, переходим к пункту 2;
4. Если дорога пуста, переходим дорогу до середины;
5. Смотрим направо;
6. Если видим автомобиль, переходим к пункту 5;
7. Если дорога пуста, переходим дорогу до конца.

Алгоритм выбора одежды, перед выходом на улицу, в зависимости от погодных условий.

1. Подходим к окну;
2. Если идёт дождь, тогда надеваем плащ, обуваем резиновые сапоги, берём зонт;
3. Иначе, если идёт снег, тогда надеваем шапку ушанку, шубу, валенки;
4. Иначе, если светит солнце и тепло, тогда одеваем шорты, футболку, сандалии;
5. Иначе, надеваем пальто и ботинки;
6. Выходим на улицу.

Задания на алгоритмы

- Алгоритм приготовления шаурмы
- Алгоритм телефонного разговора
- Алгоритм получения отличной оценки на экзамене по С
- Алгоритм лечения простуды

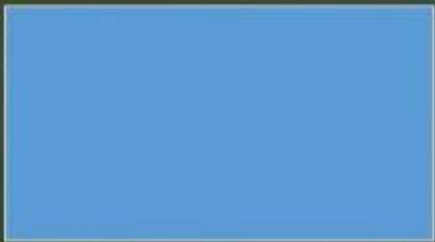
Основные типы блоков



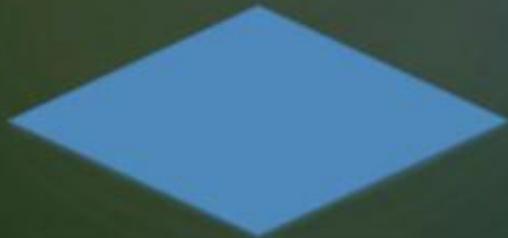
- **Начало/Конец алгоритма**



- **Ввод/вывод данных**



- **Блок действия**



- **Блок условия**

Начало и конец алгоритма

Элемент отображает
вход в программу и
выход из неё



(наиболее частое применение – начало
и конец программы). Внутри фигуры
записывается соответствующее
действие.

Блок действия

Выполнение одной или нескольких операций, обработка данных любого вида (изменение значения данных, формы представления, расположения). Внутри фигуры записывают непосредственно сами операции, например, операцию присваивания: $a = 10 * b + c$;



Данные (ввод-вывод)

Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). Данный символ не определяет тип носителя данных (для указания типа носителя данных используются специфические символы).



Логический блок (блок условия)

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.

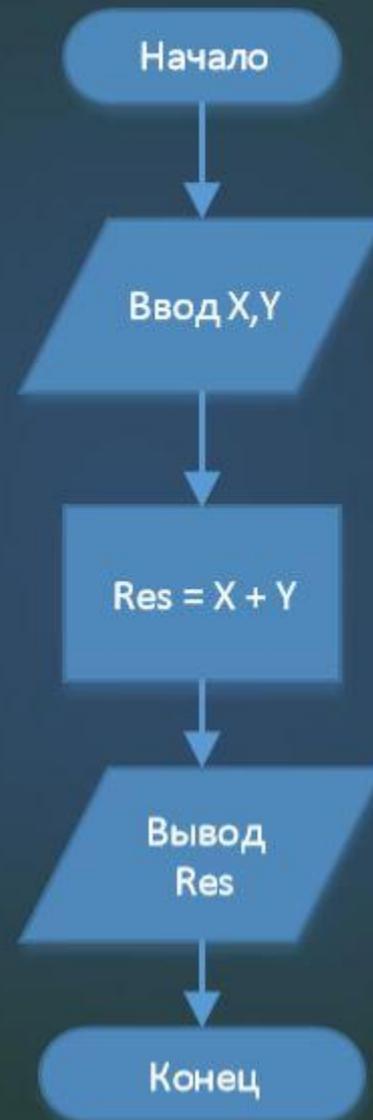


Структура алгоритма

- **Линейная** – инструкции выполняются последовательно друг за другом
- **Разветвлённая** – алгоритм, который содержит условие, в результате проверки которого может произойти разделение на несколько ветвей алгоритма
- **Циклическая** – алгоритм, который предусматривает многократное повторение одних и тех же инструкций

Линейная структура

На примере алгоритма суммирования двух чисел



Пример линейного алгоритма

Алгоритм посадки дерева

Алгоритм в виде
блок-схемы

Словесный
алгоритм

1. Выкопать в земле ямку
2. Опустить в ямку саженец
3. Засыпать ямку с саженцем землёй
4. Полить саженец водой



Начало

Выкопать в земле
ямку

Опустить в ямку
саженец

Засыпать ямку с
саженцем землёй

Полить саженец
водой

Конец



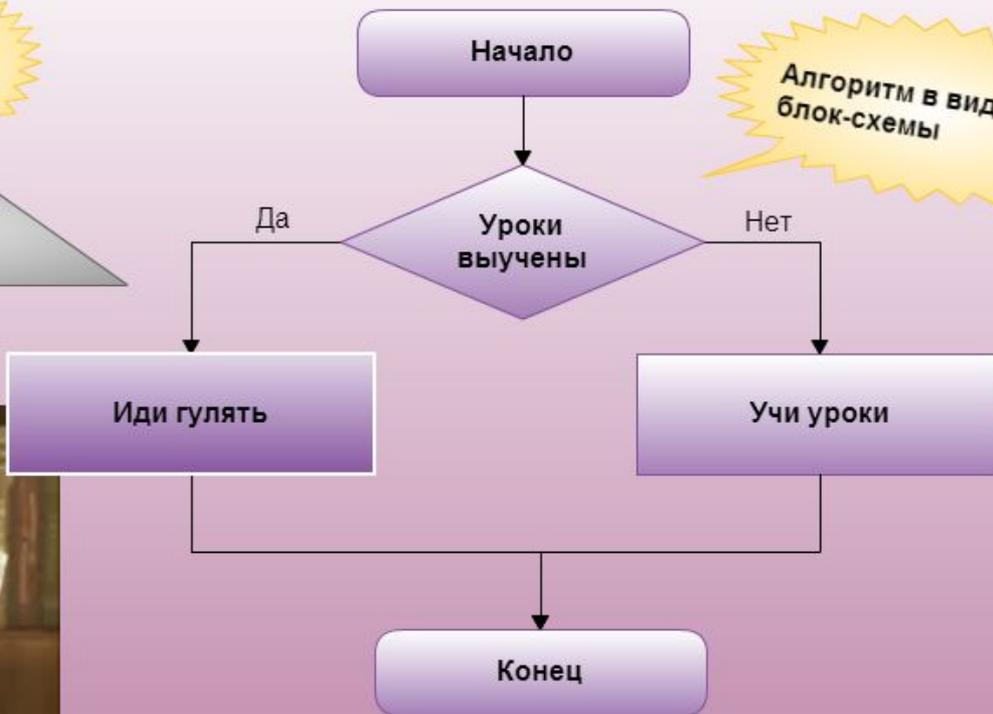
Разветвлённый алгоритм

Алгоритм выполнения домашнего задания

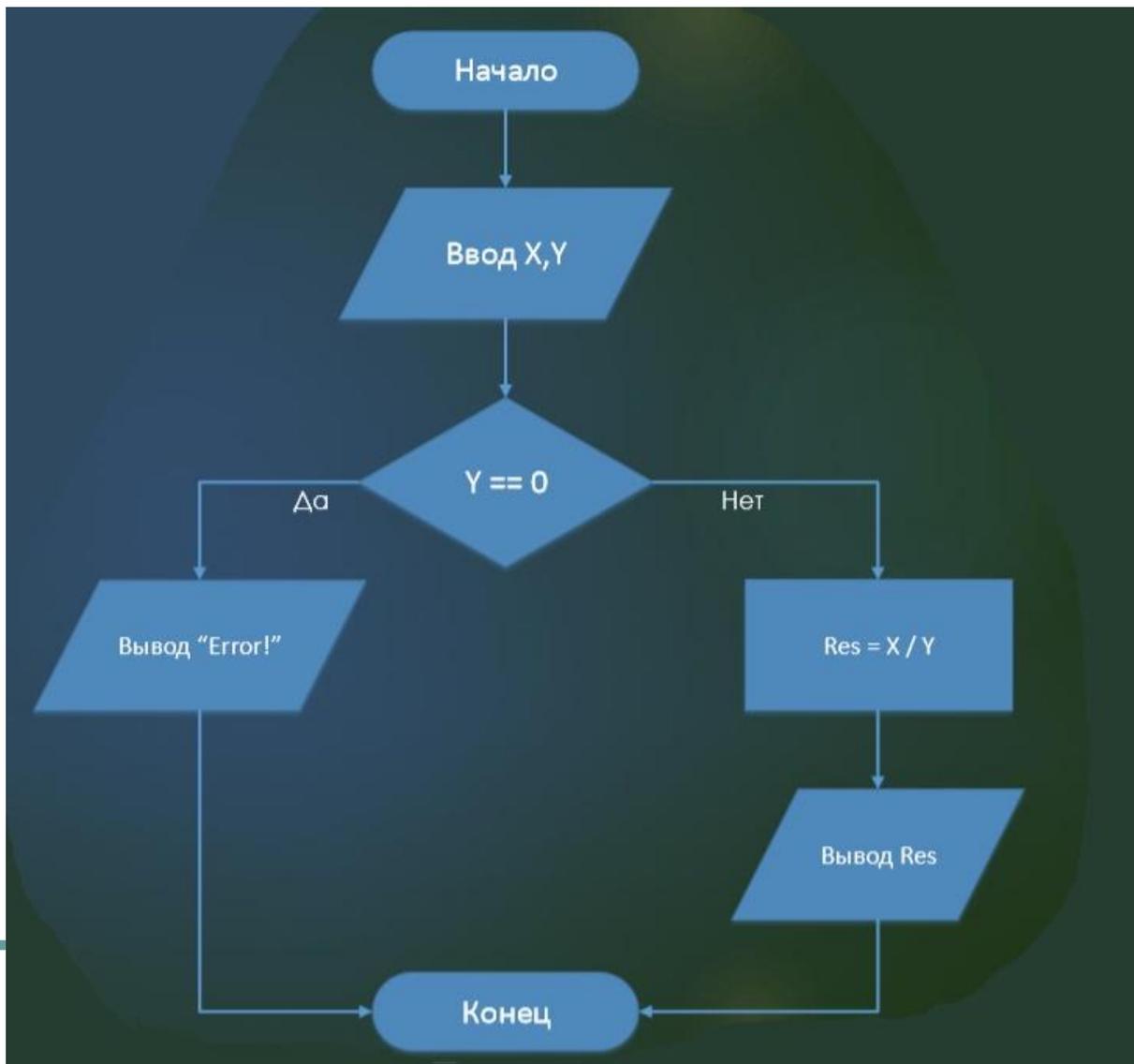
Словесный алгоритм

Если уроки выучены, то иди гулять, иначе учи уроки

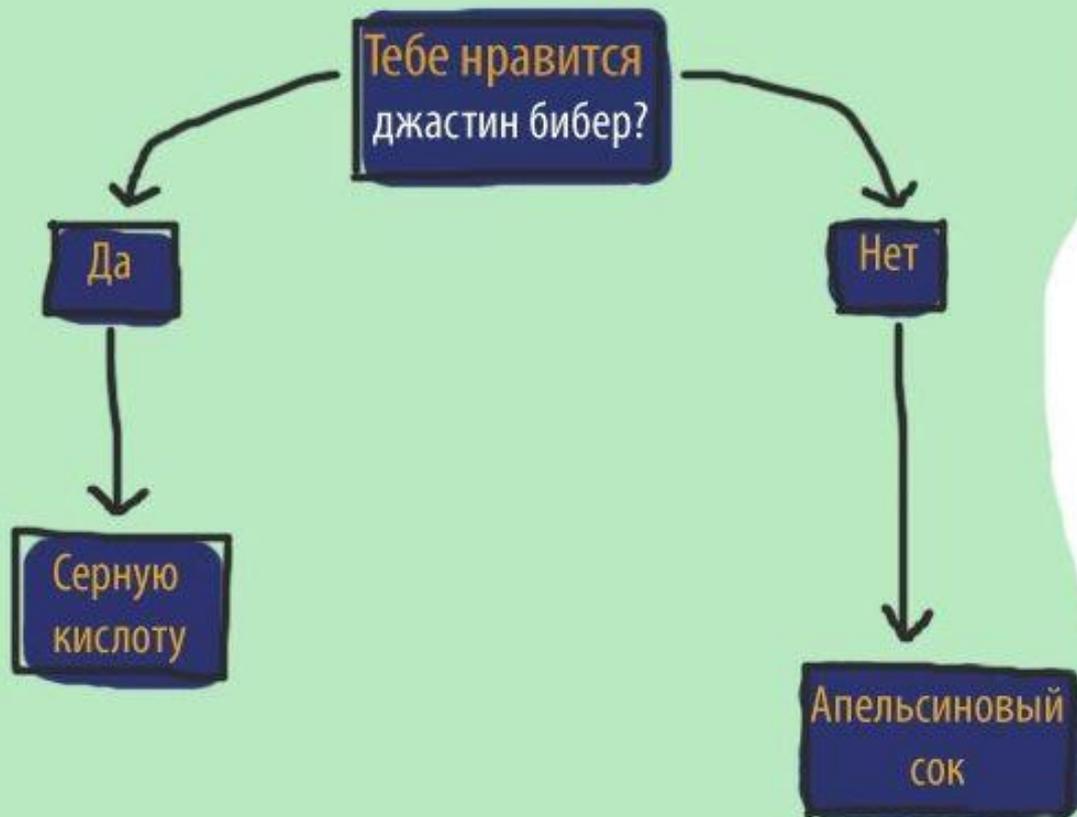
Алгоритм в виде блок-схемы



Ещё пример (деление)

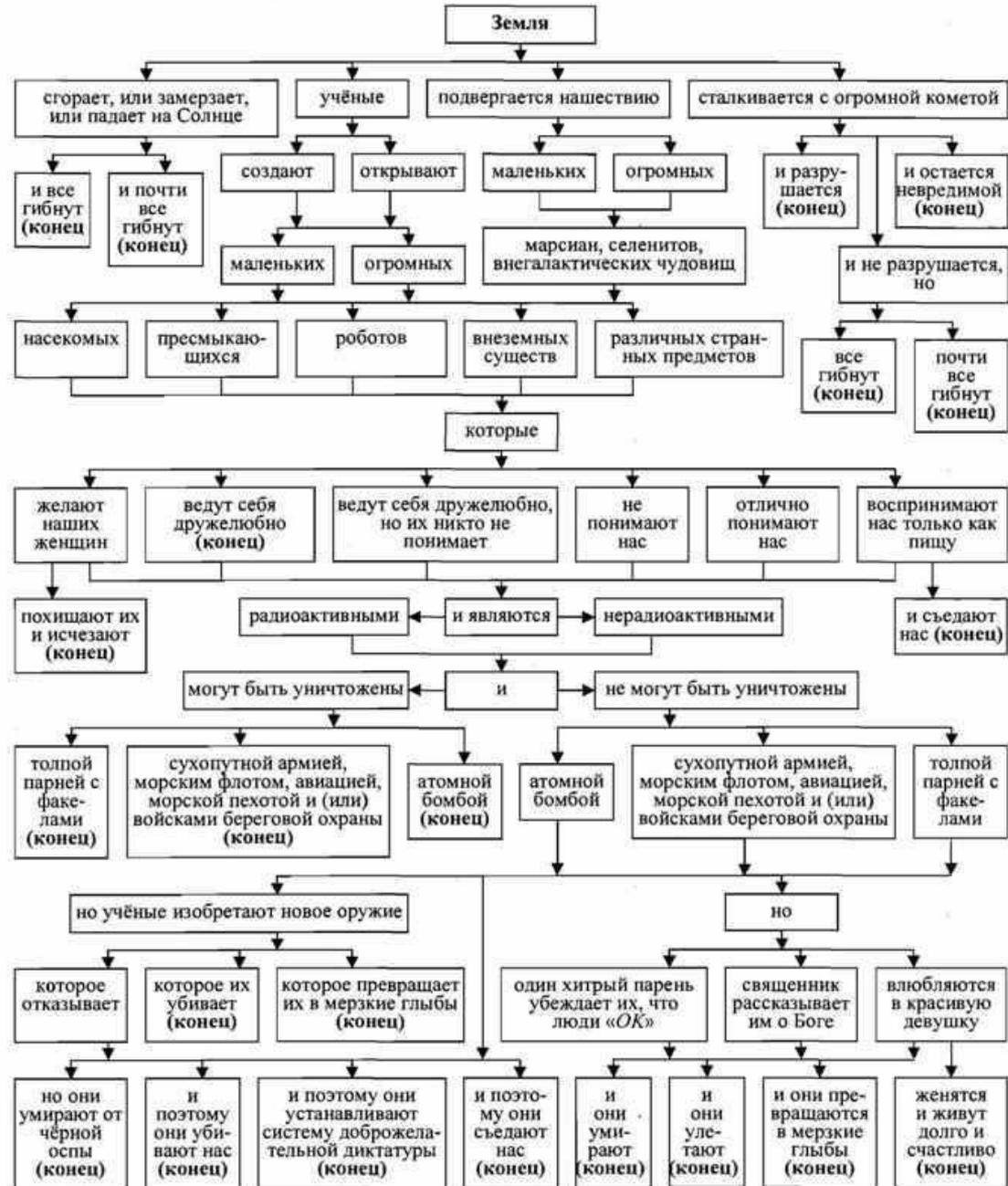


Что следует выпить на завтрак?



Станислав Лем

Карманный компьютер фанатов научной фантастики

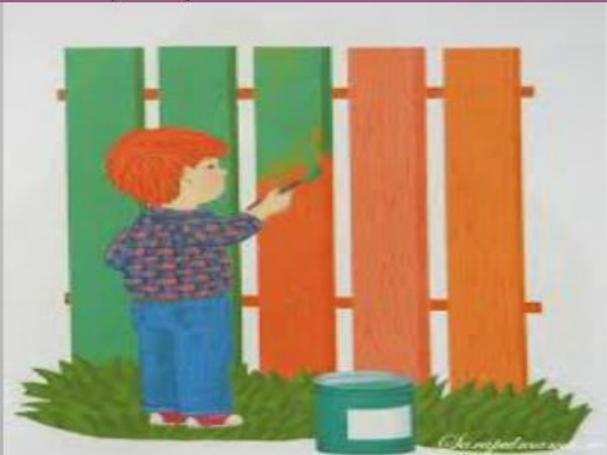


Циклический алгоритм

Алгоритм покраски досок

Словесный алгоритм

1. покрасим доску
2. если есть еще доска, то перейти к ней и выполнить действие 1
3. если доски закончились, то завершаем работу

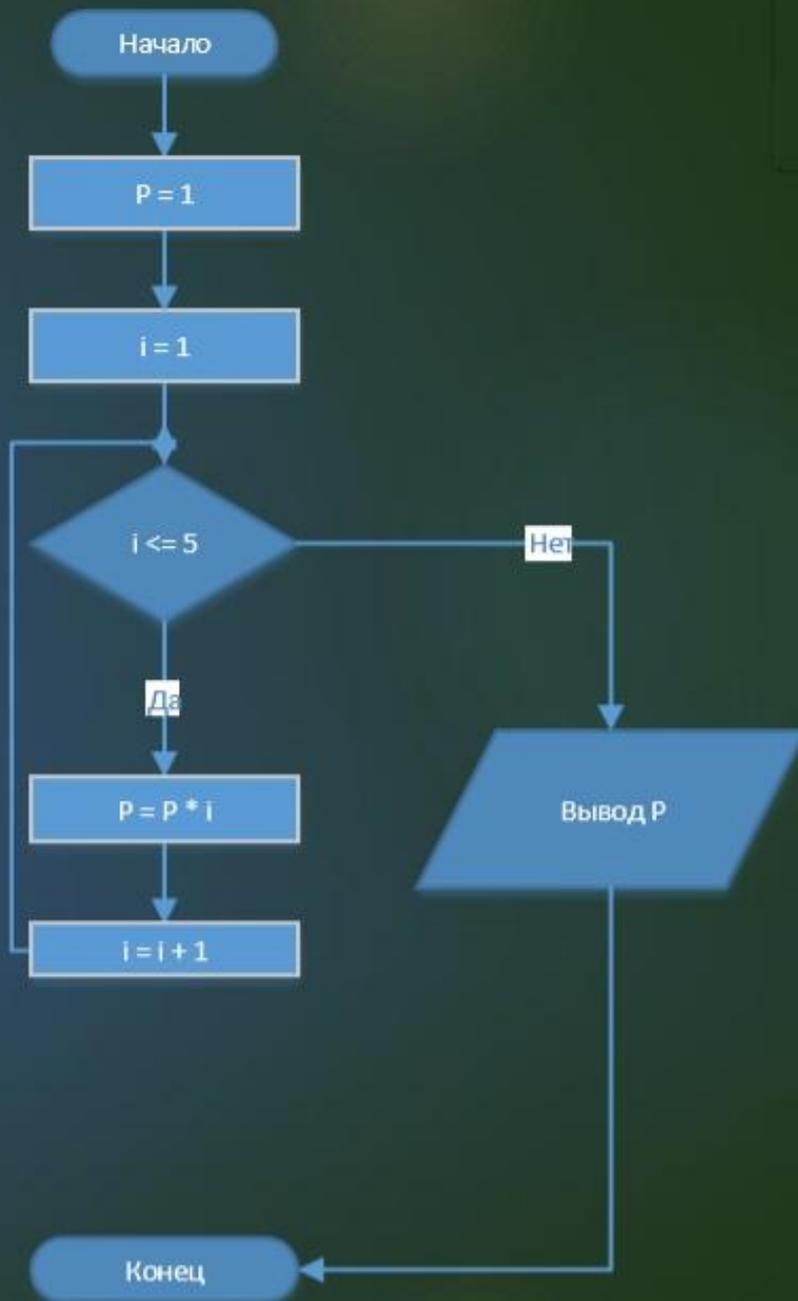


Алгоритм в виде блок-схемы



Ещё пример

Алгоритм вычисления произведения всех чисел от 1 до 5 включительно.



Задания на блок-схемы

- Алгоритм покупки футболки
- Алгоритм знакомства с девушкой
- Как отличить кошку от собаки
- Любая популярная пословица или поговорка

Язык низкого уровня

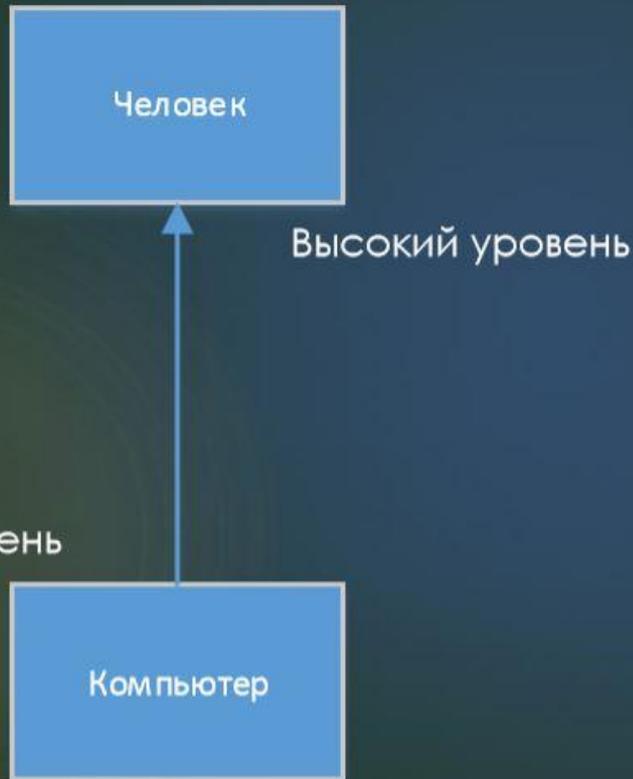
Код на языке ассемблера

Assembler является
языком **низкого уровня**

В языке ассемблера команды процессору представлены в виде **мнемонических последовательностей**.

Адрес	Команды	Комментарии
Вызывающая программа		
2000	PUSH PARAM2	Помещение параметров в стек
2006	PUSH PARAM1	
2012	CALL SUB1	
2017	POP RESULT ADD ESP,4	
Первая подпрограмма		
2100	SUB1 PUSH EBR	Сохранение регистра указателя на фрейм
	MOV EBR,ESP	
	PUSH EAX	Загрузка указателя на фрейм
	PUSH EBX	
	PUSH ECX	Сохранение регистров
	PUSH EDX	
	MOV EAX,[EBR+8]	Загрузка первого параметра
	MOV EBX,[EBR+12]	Загрузка второго параметра
	...	
2160	PUSH PARAM3	Помещение параметра в стек
2165	CALL SUB2	
	POP ECX	Выталкивание результата работы SUB2 в ECX
	MOV [EBR+8],EDX	Помещение результата в стек
	POP EDX	Восстановление регистров
	POP ECX	
	POP EBX	
	POP EAX	
	POP EBP	Восстановление регистра указателя на фрейм
	RET	Возврат в главную программу
Вторая подпрограмма		
3000	SUB2: PUSH EBR	Сохранение регистра указателя на фрейм
	MOV EBP,ESP	
	PUSH EAX	Загрузка указателя на фрейм
	PUSH EBX	
	MOV EAX,[EBP+8]	Сохранение регистров
	...	Загрузка параметров
	MOV [EBP+8],EBX	Помещение результата SUB2 в стек
	POP EBX	Восстановление регистров
	POP EAX	
	POP EBP	Восстановление регистра указателя на фрейм
	RET	Возврат в первую подпрограмму

Уровни языков



Чем выше уровень языка, тем больше мы **абстрагируемся** от деталей устройства машины.

Сравнение уровней

Языки высокого уровня	Языки низкого уровня
Нет прямого доступа к аппаратным ресурсам	Прямой доступ к аппаратным ресурсам
Скорость выполнения программы ниже	Высочайшая скорость выполнения программы
Размер исполняемого файла больше	Минимальный размер исполняемого файла
Возможно неоптимальное использование платформы	Максимальное использование специфики платформы
Высокая скорость и низкая сложность разработки	Огромная сложность разработки

Трансляция программ

Для выполнения программы на компьютере её необходимо транслировать – перевести в машинный (байтовый) код.

Существует 2 метода – трансляции:

- Компиляция
- Интерпретация

Компиляция

- Исходный код программы переводится с определённого языка программирования на машинный код специальной программой – компилятором
- Машинный код может многократно выполняться процессором
- Машинный код является непереносимым между разными аппаратными и программными платформами

Интерпретация

- Программа, команда за командой, переводится в машинный код, и каждая инструкция сразу же выполняется специальной программой – интерпретатором
- На компьютере, который запускает такую программу, обязательно должен быть установлен интерпретатор
- Это позволяет добиться переносимости между платформами

SDK

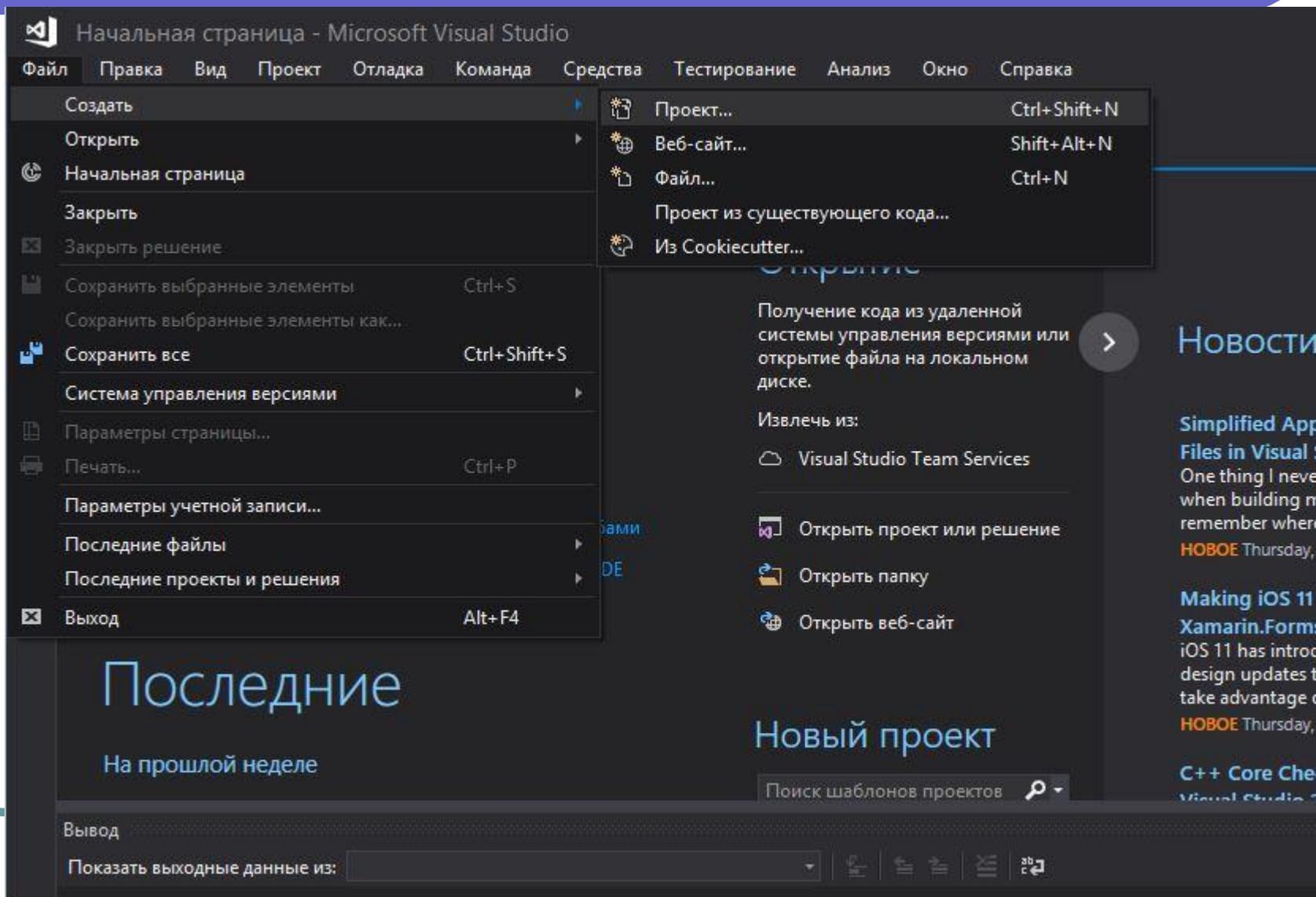
SDK (от англ. *software development kit*) — комплект средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.

Скачать Visual Studio 2019

<https://www.visualstudio.com/ru/downloads/>

(Community Version!)

Создание проекта



Создание проекта

Создание проекта

Последние файлы: .NET Framework 4.6.1 | Сортировка: По умолчанию | Поиск (Ctrl+E)

Установленные

- Visual C#
- Visual Basic
- Visual C++
 - Классическое приложение Windows
 - Общий
 - Библиотека ATL
 - Тест
- Python
- Другие типы проектов

В сети

Иконка	Название проекта	Тип
	Пустой проект	Visual C++
	Проект Makefile	Visual C++
	Проект общих элементов	Visual C++

Тип: Visual C++
Пустой проект по созданию локального приложения

Не нашли то, что искали?
[Открыть Visual Studio Installer](#)

Имя: Project2

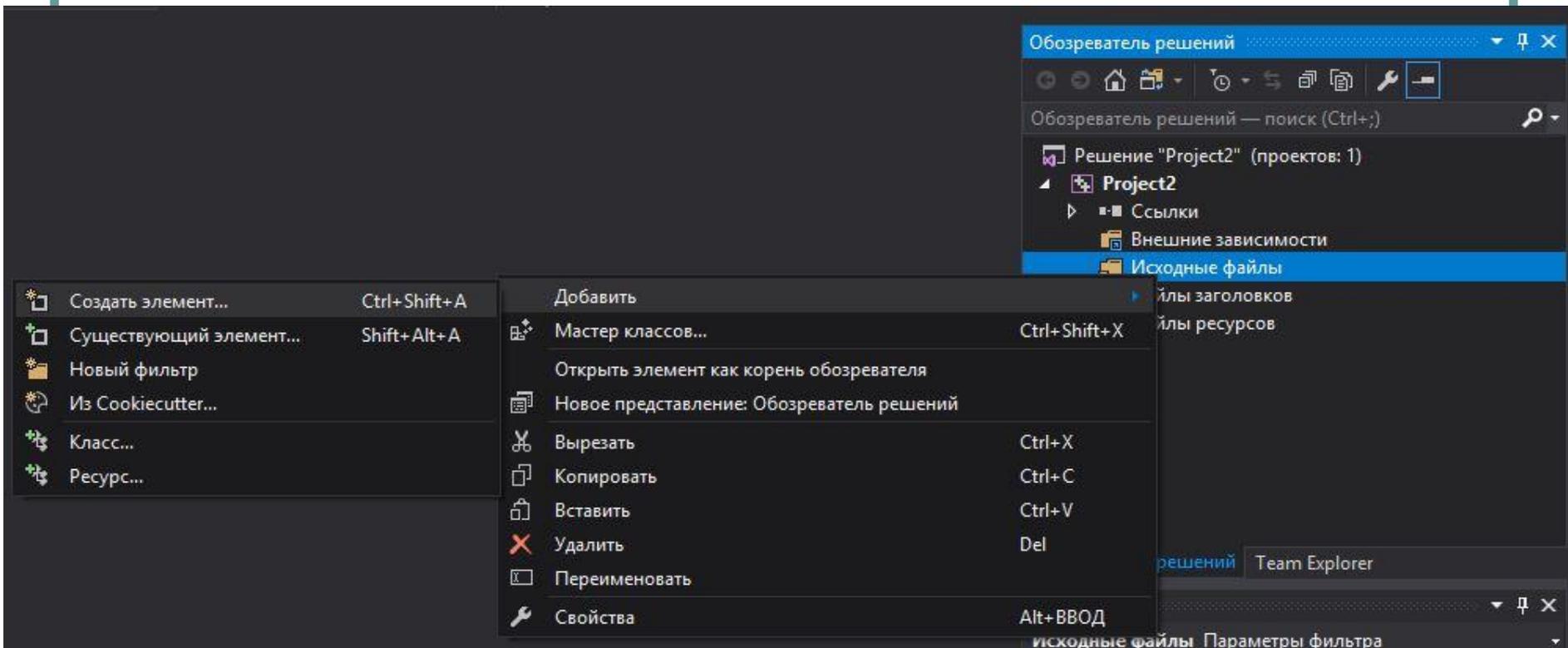
Расположение: C:\Users\Alex.HOME.000\source\repos

Имя решения: Project2

Создать каталог для решения
 Добавить в систему управления версиями

Обзор... | OK | Отмена

Создание проекта



Создание проекта

Добавление нового элемента - Project2

Установленные

Visual C++

- Код
- UI
- Библиотека ATL
- Данные
- Ресурсы
- Web
- Служебные программы
- Таблицы свойств
- HLSL
- Графика

В сети

Сортировка: По умолчанию

	Файл C++ (.cpp)	Visual C++
	Файл заголовка (.h)	Visual C++
	Класс C++	Visual C++

Поиск (Ctrl+E)

Тип: Visual C++

Создает файл, содержащий исходный код C++.

Имя: Source.cpp

Расположение: C:\Users\Alex.HOME.000\source\repos\Project2\Project2\

Обзор...

Добавить

Отмена

Первая программа

```
(Глобальная область) main()
1  #include <iostream>
2  using namespace std;
3
4  void main()
5  {
6      cout << "Hello, world!\n";
7  }
```

Обозреватель решений — поиск (Ctrl+;)

- Решение "Project2" (проектов: 1)
 - Project2
 - Ссылки
 - Внешние зависимости
 - Исходные файлы
 - Source.cpp
 - Файлы заголовков
 - Файлы ресурсов

Обозреватель решений Team Explorer

Свойства

Создание проекта

Запуск проекта – Ctrl + F5.

Обычный
программист



- Всегда занят
- Ничего не умеет
- Требуется деньги
- Разговаривает непонятными словами

ТЫЖ
программист



- Всегда свободен
- Всё бесплатно
- Взлом странички ВК
- Установка Windows
- Починка утюга

```
#include <iostream>
```

Данная строка подключает в программу библиотеку под названием `iostream`. Библиотека - файл, в котором содержатся описания различных функций, реализованных другими программистами. Данная программа получила возможность использовать функции находящиеся в библиотеке `iostream`.

```
using namespace std;
```

В языке C++ существует понятие **пространство имён**. Это пространство определяет некую область, на которую приходится действия оператора или функции. Для того, чтобы использовать оператор, находящийся в определенном пространстве, необходимо подключить это пространство в свою программу.

Пространство имён

Механизм пространств имён позволяет избежать конфликта имён (т.е. их совпадения) при командной разработке проекта. Когда-нибудь, мы создадим не одно своё пространство имён, а пока будем пользоваться стандартным.



Понятие «main»

main – название основной функции приложения, т.н. **точка входа** – отсюда программа стартует, и все команды будут выполняться построчно одна за другой, сверху вниз. Пока что договоримся все команды писать между фигурными скобками этой функции. Придёт время, и мы научимся делать дополнительные функции, кроме мэйна.

Вывод данных на экран

```
cout << “Hello world!\n”;
```

Существуют символы, которые имеют специальное назначение. Они могут быть представлены с помощью т.н.

Escape-последовательностей

(управляющих последовательностей).

Escape-последовательность

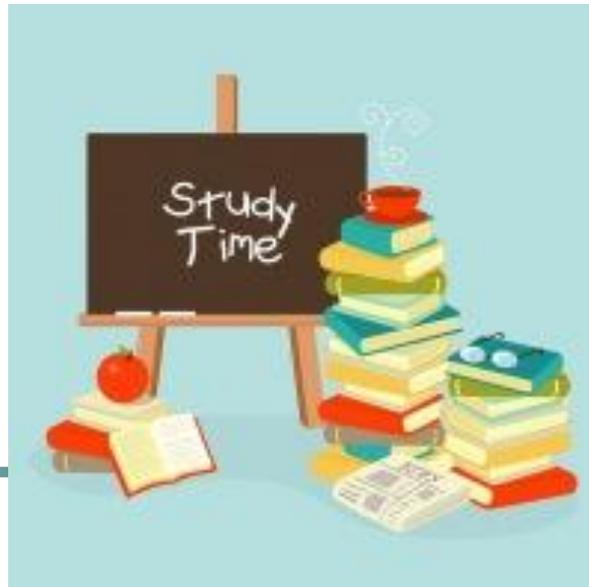
Escape-последовательность – это специальная комбинация символов, которая начинается с символа обратного слэша (\) и заменяется транслятором на определенный символ, который имеет специальное значение, либо его нельзя использовать в данном контексте, либо он отсутствует на клавиатуре.

Основные последовательности

- `\n` – переход на новую строку
- `\t` – табуляция (выравнивание)
- `\\` - просто слэш
- `\”` – кавычка в тексте
- `\0` – завершение строки
- `\r` – перевод каретки
- `\a` – звуковой сигнал

Домашнее задание

- **Добавиться в телеграм!**
- **Установить Visual Studio**
- **Составить формальный алгоритм на одну из предложенных тем**
- **Выполнить блок-схему (нарисовать на листочке бумаги и потом сфотографировать, либо выполнить в графическом редакторе) на одну из предоставленных тем**
- **Выполнить все задания на cout**
- **Прочитать 1-2 начальные главы рекомендованных учебников по С**



Полезные сайты по C++

- <https://techrocks.ru/2018/03/31/free-online-games-to-learn-programming/>
- <https://www.codecademy.com/learn/learn-c-plus-plus>

Пример работы студентов 😊

🎮 Растения против Александра



Спасибо за внимание!

**Ответы на любые
ваши вопросы**



Спасибо за внимание!

