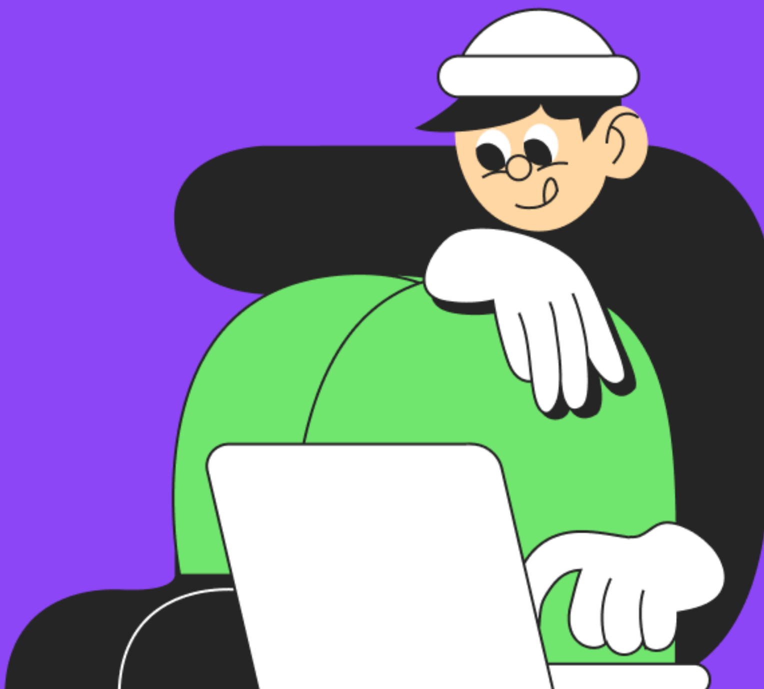
GeekBrains

# Основы логики для программистов



# Оглавление


[Приветствие]
[Что такое логика?]
- [Логика и компьютер: какая связь?]

[Теория множеств]
- [Разберём на примере преподавателей]
  - [Другие примеры множества преподавателей]
- [Пример факультета преподавания]
- [Универсальное множество]

**[00:01:36]**

# Приветствие

Друзья мои, приветствую вас на вводной лекции по математике!

Интересный факт в дополнение к тому, что вы уже слышали на предыдущей лекции: я, на самом деле, по образованию – учитель математики. И когда дело касается создания каких-то технических докладов или докладов по математике, то такой доклад обычно состоит из четырёх главных тезисов.

Первое – в процессе доклада должно прозвучать то, что точно будет понятно всем. Помимо этого, должно встречаться что-то похожее на формулы, а также слайды, которые непонятны вообще никому, причём в некоторых случаях даже лектору. Ну, и обязательно что-то милое. Наверное, вы встречали что-то подобное – это могут быть маленькие дети или котики, или что-то похожее.

**[00:02:20]**

# Что такое логика?

Попробуем разобраться с тем, что же такое логика. Удивительно, но логика является частью большого фундамента математики. На этом слайде представлены различные разделы. И всё это есть не что иное, как большая наука математика. В процессе изучения мы затронем такие разделы, например, как алгебра, добавим немного геометрии, линейную алгебру, численные методы в какой-то степени, математическую логику, теорию вероятностей. Но, опять же, по верхам пробежимся.

В чём основная идея введения в логику? Дело в том, что когда говорят «Давайте рассуждать логически», – это и является частью математики, хотя она находится на стыке с философией, которой мы с вами точно заниматься не будем. Но интересно то, что формально логика изучает высказывания. Основоположником формальной логики является Аристотель, о котором вы наверняка слышали. Но что такое высказывание? Высказывание, друзья – это повествовательное предложение, на которое однозначно можно дать ответ либо «Да», либо «Нет», то есть никакого третьего варианта быть не может. Например, «GeekBrains – площадка для изучения программирования». Является ли это высказыванием? Если вы ответили «Да», то технически – является. Когда речь идёт о высказываниях, мы также можем проверить истинность высказывания, то есть является ли истинным это высказывание. И ответ – «Нет», потому что GeekBrains изучает и обучает, например, также маркетингу, или управлению, или каким-либо другим специальностям.

Следующее: предложение «Сегодня 22 октября?». Обратите внимание на то, что высказывание – это всегда повествовательное предложение, а в этом случае – вопрос. Значит, это не высказывание.

«У треугольника четыре стороны». На это повествовательное предложение можно однозначно дать ответ «Нет», поэтому оно является высказыванием, причём истинность этого высказывания ложно.

«Илон Маск самый богатый человек в мире». Это, как вы можете наблюдать, высказывание, но, что интересно, если, например, на текущий момент он является самым богатым, и я могу ответить «Да», то, может быть, например, через год результат этого высказывания будет ложью.

Следующий тезис: «В этом городе точно живёт более полумиллиона человек». Является ли это высказыванием? Ответ «Нет», потому что непонятно, в каком именно городе. Если мы, например, будем говорить о городе Смоленск, то тогда станет понятно, что там нет полумиллиона жителей, а значит, во-первых, этот тезис будет являться высказыванием, и, во-вторых, истинность будет ложна. Если же мы будем говорить, о Москве, где точно живёт более полумиллиона человек, истинность будет «Да».

«Послушай этот трек». Очевидно, что этот тезис не является высказыванием.

### [00:05:29]

## Логика и компьютер: какая связь?

Что дальше? Каким образом можно связать логику и компьютер? Интересный момент. Как мы помним, компьютер обрабатывает только 0 и 1 и работает только с 0 и 1. Но что самое интересное – любое из высказываний отображается в этом самом 0 или 1. И первым, кто предложил так отображать всё что угодно, был английский учёный Джордж Буль. Дальше, кстати, будет раздел математики, названный в его честь.

# Теория множеств

Как связать булеву алгебру с математикой в целом? Для начала поговорим о теории множеств. Что такое множество? В общем случае определение звучит как «совокупность каких-либо элементов», абсолютно любых, но важным здесь является то, что во множестве встречаются только уникальные объекты. Например, если мы рассматриваем множество каких-то чисел, то мы нельзя сказать, что в нём есть «1111» и «2». То есть в этом случае множество состоит только из двух элементов: 1 и 2.

### [00:06:29]

## Разберём на примере преподавателей

В качестве примера рассмотрим множество всех преподавателей такой профессии, как «разработчик». И, опять же, как истинные математики обозначим его какой-то буквой. Пусть это будет «Т». Так как я веду один или несколько модулей, соответственно, являюсь преподавателем. То есть отношусь к множеству при всех преподавателях курса «Разработчик» и являюсь элементом этого множества. Чтобы упростить определение, в большинстве случаев математики всё обозначают каким-то одним значком, буквой или символом, поэтому обозначим меня как «Т». Таким образом, если у нас есть условный персонаж, например, Катя Молина, она не является преподавателем текущего курса, тогда мы можем сказать, что этот «элемент» к множеству не принадлежит. Чтобы это формально описать, мы можем записать так: t (малая) принадлежит T (большому) и q (малая) не принадлежит T. Для чего нам это нужно? Чтобы было удобнее и проще записывать дальнейшие наши выкладки и рассуждения.

Итак, рассмотрим как множество всех преподавателей; в этом случае вы пока что встречались с Ильнаром и со мной, после этого будут и другие преподаватели, но какие я, например, не знаю, поэтому я их условно обозначил как t3, t4 и t5, но их, конечно же, будет больше. И в этом случае элемент «Катя Молина» никакого отношения не имеет к множеству преподавателей курса программы «Разработчик». Как описать это ещё более формально? Т (большое), далее значок равенства и в фигурных скобках перечисляются те элементы, которые принадлежат этому множеству. Учитывая то, что два из этих элементов нам известны, то мы можем их записать.

Следующий этап. Рассмотрим, например, множество всех преподавателей GeekBrains, очевидно, что в него входят те преподаватели, которые являются лекторами – в данном случае на курсе «Разработчик». Но обратите внимание, наша Катя не попадает в них. Но в то же время не исключено, что где-то в мире может существовать такая барышня, которая как раз таки относится к множеству вообще всех преподавателей в мире. Чтобы это как-то формально описать, мы можем сказать так: пусть множество всех преподавателей обозначается Т (большое), множество всех преподавателей GeekBrains мы обозначим К. Тогда можно сказать, что множество Т включено во множество К. Или ещё говорят, множество Т является подмножеством множества К. Обратите внимание, каким значком это записывается.

Если продолжать логику, то становится очевидно, что все преподаватели GeekBrains являются элементами большого множества всех преподавателей в мире вообще. Надеюсь, что здесь ничего сложного нет.

Попробуем дать формальное определение подмножества. Если у нас есть некоторые множества с элементами t1, t2 и так далее, и каждый из элементов этого множества входит в другой или включён в другое множество, то, соответственно, первое множество Т является подмножеством множества U.

Теперь вам в качестве самостоятельной работы такой вот пример: попробуйте дать формальное определение равенству двух множеств.

## [00:12:06]

## Другие примеры множества преподавателей

Надеюсь, вы попробовали дать этому определение, и звучать оно будет примерно как «Если все элементы множества Т входят в множество U, а все элементы множества U содержатся во множестве Т, то это и означает, что два множества равны».

Вернёмся к нашему множеству всевозможных преподавателей. Мы пока что знакомы с Ильнаром, Сергеем и Катей, но технически могут существовать и другие элементы множества. Например, есть условный Андрей Рысев, который является преподавателем GeekBrains. Также у нас есть условная Галина Банару, а также есть Бенуа Мандельброт. Что дальше? Допустим, Сергей и Ильнар являются элементами множества Т. Андрей, Галина, и Бенуа являются элементами множества К, Александр и Катя являются элементами множества U. Но в то же время все элементы, которые включаются, точно также могут принадлежать или должны, или даже обязаны принадлежать своим, соответственно, более большим множествам. То есть в этом случае как Сергей, так и Ильнар принадлежат множеству К. Далее Сергей, Андрей, Галина и Бенуа принадлежат множеству U. Зная этот факт, мы можем попробовать определить различные операции, которые могут производиться со множествами. Допустим, мы сейчас разделим два множества на T, условно – только преподаватели группы направления «Разработчик» и все преподаватели GeekBrains. Обратите, пожалуйста, внимание, что если мы объединим эти два множества, такая операция справедлива для множеств, то в этом случае в то множество, которое получится, будут входить все элементы, как первого множества, так и второго. Объединение двух множеств обозначается вот таким значком.

### [00:13:55]

## Пример факультета преподавания

Следующий пример. Рассмотрим некоторый факультет преподавания и некоторый факультет разработки. То есть технически мы можем являться одновременно преподавателями, действующими практиками и в то же время есть те, кто только практикует, но не преподаёт. Если разделить их на такие множества, то есть множество преподавателей, оно будет включать Катю, Сергея и Ильнара, а вот факультет Разработки будет включать Андрея, Галину, Бенуа и также Сергея и Ильнара. Можно заметить, что одни и те же элементы есть в двух множествах – это есть не что иное, как пересечение двух множеств. В большинстве случаев обозначается это такими диаграммами, которые, кстати, называются диаграммами Эйлера или кругами Эйлера, или диаграммами Венна, или диаграммами Эйлера-Венна, их называют по-разному, но сути дела это не меняет. То есть мы можем сказать, что одновременно Сергей и Ильнар принадлежат ко множеству разработки факультета Разработки и множеству факультета Преподавание. Таким образом, пересечение двух множеств определяется вот таким значком.

Следующий этап. Теперь мы можем на такой картинке попробовать поискать те элементы, которые принадлежат множеству L, но не принадлежат множеству D. В данном случае таким элементом является только Катя. Как это формально описать? Называется эта операция разностью двух множеств, и в данном случае мы из множества L вычитаем множество D. Такая операция обозначается обратным слэшем. Таким же образом мы с вами можем найти разность множеств D и L. Помимо этого есть такая операция, которая называется симметрическая разность, то есть это те элементы, которые не являются общими. Обозначается эта операция треугольником, и в данном случае в него входят Андрей, Галина, Бенуа и Катя.

Здесь в качестве самостоятельной работы я дам вам такую идею: попробуйте доказать или описать данную операцию через ранее изученные. Небольшая подсказка: через разность и объединение.

### [00:20:08]

# Универсальное множество

Надеюсь, вы это сделали. А мы обязательно разберёмся задачу на семинарском занятии.

Следующая идея. Рассмотрим множество всех преподавателей. Интересно, что помимо преподавателей есть ещё множество «Другие». В частности, множество маркетологов или множество водителей или множество простых карандашей. Помимо этого, есть и другие. Например, под Q мы можем обозначить множество котиков. Здесь должен появиться котик, но его нет, к сожалению. Дальше может быть ещё какое-то множество, которое мы каким-то образом можем назвать. И в этом случае все множества включаются в какое-то множество, и это множество в большинстве случаев называется универсальным множеством.

Универсальное множество – это множество, содержащие в себе абсолютно все множества.

Если в качестве примера взять какое-то отдельное множество. Например, множество преподавателей L, и рассмотреть всё, что останется за его пределами, то это все элементы, не попавшие в L. Такая операция в контексте множеств называется инверсией или отрицанием, обозначается в математической логике и вообще в теории множеств в большинстве случаев вот таким значком. Программисты, кстати, чаще обозначают как восклицательный знак.

Следующая идея или следующие действия, которые могут производиться над множествами. Допустим, у нас снова есть множество преподавателей и универсальное множество. Отдельное внимание в теории множеств занимает такое множество, которое называется пустым, оно обозначается как перечёркнутый кружок.

Вопрос: как вы думаете, есть ли такое множество, в котором содержится пустое множество, помимо универсального? Дам вам 30 секунд на идею, может быть, даже кто-то успеет погуглить… Друзья мои, идея в том, что пустое множество является подмножеством любого множества. У кого-то может уже мозг сломаться после таких вот определений, но тем не менее. Если у кого-то вдруг возникают вопросы о том, где это может быть использовано то, как я уже не раз говорил, на семинарских занятиях мы обязательно об этом поговорим.

## [00:22:15]

## Увлекательная задача

Что дальше? Попробуем разобрать одну из практических задач, и в этом случае задачка начинается со слова «докажите». В большинстве случаев после этого народ встаёт и уходит. Но к сожалению или счастью, вам придётся остаться.

Докажите: $B \setminus A = B \cap (\neg A \cup \neg (A \cup B))$

Конечно, это можно попытаться прочитать, и в этом случае мы читать будем из скобочек. То есть объединение А и Б, далее нужно сделать отрицание этого и далее нужно объединить с не А. Звучит, согласитесь, так себе. Поэтому здесь проще показать.

Итак, каким образом это делается? Самое простое описание доказательства, которое можно придумать, – это круги Эйлера или диаграммы Эйлера-Венна, попробуем это сделать. Но чтобы начать что-то доказывать, нам нужно соблюсти маленький пункт. А именно выяснение порядка выполнения операций. Всегда, абсолютно всегда нам требуется выполнять в первую очередь действия в скобках. Следующее по приоритету – это операция инверсия или операция отрицания, затем делается всё остальное.

Для начала попробуем изобразить множество А и Б, потому что кроме них, больше ничего нет. Казалось бы, что нет. Далее, если взять левую часть, то мы вычтем из Б А, и получится что-то такое. Запомним это.

Теперь попробуем разобраться с правой частью. У нас есть множество А и Б, но, помимо этого, есть ещё и так называемое универсальное множество. В этом случае мы можем изобразить всё, что не относится к объединению множеств А и Б. Если вот это объединение множеств А и Б, то, соответственно, всё, не входящие в него, это инверсия А объединения Б. Запомним.

На следующем этапе нужно найти отрицание А. То есть все элементы, не относящиеся к А. В этом случае это множество А, и всё, что не включено в него. Теперь требуется найти объединение, запоминаемого на втором этапе с только что полученным. Если присмотреться, то это и будет то же самое. И теперь, если мы найдём пересечение того, что получилось с множеством Б, то у нас и получится вот этот кусочек. Это идентично тому, что запомнили изначально.

Примерно таким образом доказываются различные утверждения из теории множеств. В качестве самостоятельной работы предлагаю вам взять листочек, ручку и попробовать доказать то утверждение, которое представлено на слайде, либо убедиться в том, что оно несправедливо.

## [00:31:58]

## Ещё одна увлекательная задача

Надеюсь, что у вас получилось. Рассмотрим ещё одну интересную задачу, которая является простой математикой. Допустим, есть числовая прямая, на этой прямой мы можем отметить какие-то точки. Например, 1 и 10. И есть некоторая точка, которая находится между ними. Вопрос: каким образом можно записать тот факт, что F принадлежит множеству или отрезку от 1 до 10? Вспоминаем школьную математику, как это там записывалось? В большинстве случаев, как вы понимаете, мы должны будем указать все точки, принадлежащие отрезку. Или ещё, например, в школе это обозначали вот так. Мария Ивановна, преподавая математику, могла называть её ёлочкой. Как часто такие задачи применяются на практике, друзья мои? На самом деле в программировании, наверное, 80% тех условий, которые будут писаться именно при помощи таких условий, будут записываться так.

Попробуем рассмотреть ещё одну задачку с уклоном в геометрию. Если до текущего момента у нас была только числовая прямая, теперь пусть будет координатная плоскость или обычная классическая. Прямоугольная декартова система координат, так её называют в математике. И в рамках этой системы координат есть вот такой закрашенный прямоугольник. Каким образом мы можем описать тот факт, что какая-то точка принадлежит именно этому квадрату? Вспоминаете ли вы, как это делается? Если нет, то напоминаю, что вы должны были определить или провести так называемую нормаль, то есть перпендикуляры к соответствующим осям. Дальше каким-то образом дать названия полученным точкам и отдельно проверить координату по X и координату по Y. Но, что интересно, у нас сейчас достаточно простая фигура. Но фигуры ведь бывают гораздо сложнее. Например, если мы захотели бы с вами определить какой-то квадрат, внутри которого нет ничего. То есть получается такой вот контур. Или если фигура была более сложной и выглядела как буква П. Может показаться, что относительно предыдущей картинки задача стала проще, но от себя отмечу, на самом деле, она стала сложнее. А что, если внутри этой фигуры будет ещё что-то? И как раз таки всё это и нужно будет описывать, используя теорию множеств, и в этом случае классическую математику, которую, скорее всего, вы изучали где-то в районе 6–7 класса.

Что ещё хочу отметить. Не исключено, что множества, с которыми вы будете работать, не всегда будут представлены такими красивыми и простыми гранями. В некоторых случаях множества выглядит очень и очень забавно и интересно, особенно если их перевернуть.

Хорошо, пока что мы с вами работали только исключительно в двумерном представлении, но мы живём в трёхмерном пространстве. И теперь представьте, если к вам подойдёт пятиклассник и скажет: «Покажите нам, пожалуйста, то, каким образом математически можно описать чипсик?». Встречались ли вы когда-нибудь с такими пятиклассниками? Если нет, то теперь вы абсолютно налегке можете им ответить: «Да это же просто – это же гиперболический параболоид!». Кому интересно, обязательно погуглите формулу, и с использованием определённых ограничений вы можете получить формула того самого чипсика.

### [00:35:55]

# Введение в булеву алгебру

Ну что, друзья мои, теперь мы можем по полной заняться именно булевой алгеброй. Для начала вспомним, что высказывание – это некое повествовательное предложение, на которое однозначно можно дать ответ либо «Да», либо «Нет». Высказывания бывают простыми и сложными. В чём их отличие? Простое – это, допустим, «Сегодня 5 октября», а сложное – это ситуация, при которой мы используем так называемые логические связки. То есть, например, «Сегодня 5 октября и на улице лето». Или следующий пример: допустим, когда вы захотели сделать какой-то поисковый запрос и в Google пишете «Книги скачать pdf», или, например, «Книги скачать pdf Epub», условно. В большинстве случаев вы здесь не добавляете ту самую логическую связку «И», и гугл будет добавлять её автоматически.

Какие есть операции в булевой алгебре? Первая операция – это инверсия, или «НЕ», помимо этого, существует операция «И», операция «ИЛИ», «ИСКЛЮЧАЮЩЕЕ ИЛИ». Эти четыре операции наиболее часто используется в программировании. Также есть и другие, например, «ЕСЛИ, ТО», или ещё одна логическая связка, которая читается как «ТОГДА И ТОЛЬКО ТОГДА, КОГДА». Существуют и другие операции, но они, в свою очередь, не используются в контексте программирования.

### Как связать логику, булеву алгебру и компьютер?

Мы узнали, что такое множество, и теперь должны ввести ещё один пункт, который называется «функция». Вспомните, пожалуйста, как вам учителя в школе объясняли функции? Можете пока просто повспоминать или погуглить. Но дело в том, что в школе эта информация объясняется на примере каких-нибудь графиков функций, где говорят, что есть какой-то X, и что-то там соответствует игреку и так далее. В общем случае определение функции звучит абсолютно иначе. В качестве примера возьмём множество всех номеров телефонов, а в качестве второго множества мы возьмём множество людей. И то правило, по которому мы будем ставить в соответствие номер телефона какого-то конкретного человека и есть не что иное, как функция. Если кто-то хочет углубиться в теорию множеств и вообще разобраться о том, что же такое функция, погуглите такие определения, как «субъективное отображение», «объективное отображение» или «инъективное отображение». Напоминаю, что «простое отображение элементов одного множества на элементы другого множества». Можете ли вы теперь привести какие-то другие примеры функций? Если вы хотите сказать «f от x равно x в квадрате», да, технически это тоже подходит, но это слишком математически. Попробуйте посмотреть вокруг себя и придумать какие-то иные функции или иные отображения. У большинства людей, которые не связаны с математикой, это почему-то вызывает трудности. С точки зрения математики, она вообще всё любит формализовать, поэтому когда речь идёт о функции, то определить это простым значком можно как «f: N переходит в P[DP1] ». Это понимается так: элементы из N, в нашем случае – множество номеров телефонов, переходит в элементы P – множество людей.

## [00:39:26]

### Логика высказываний строится на 0 и 1

При чём здесь это всё? У вас может возникнуть вопрос, зачем нам какие-то множества, что там функции? А дело в том, что вся логика высказываний строится путём использования подобных функций и отображения их множеств в ноль и один. Ноль, как вы помните, это ложь, а единичка – истина.

Что дальше? Когда дело доходит до таких функций, которые могут быть отображением только исключительно как ноль и один, то она будет называться предикатом. В большинстве случаев в программировании вы будете встречаться с таким понятием. Например, если в учебнике или в каких-либо справочных материалах написано «функция-предикат», у вас сразу же должно сложиться понимание, что эта функция, будет что-то, пока неизвестное, принимать и возвращать в ответ либо «Да», либо «Нет». Если забегать вперёд в контекст программирования, то возвращать она будет логическое значение.

Каким образом нам перейти от, казалось бы, простых жизненных высказываний к тому, что действительно является предикатом? Рассмотрим такой пример. «2 – число чётное». Является ли этот тезис высказыванием? Наверное, у вас возникнет вопрос о том, что такое тезис. Итак, «2 – это число чётное». Как вы понимаете, это высказывание, потому что мы точно можем ответить «Да». Более того, это высказывание, являющееся истиной. Но дело в том, что если мы скажем «X – число чётное», будет ли этот пункт являться высказыванием? Конечно, нет, так как мы не знаем, какое число подразумевается под X. Если сюда подставить, например, число 2, тогда – да. Если мы поставим число 22, тогда тоже ответ правильный. А вот если просто X, тогда это не будет являться высказыванием, потому что в этом случае у нас не будет определённости. Помните, высказывание – это повествовательное предложение, с однозначным ответом «Да» или «Нет».

Если мы введём функцию, в качестве аргумента принимающую некоторый X, то это и будет называться предикатом. То есть у нас имеется некоторая функция, которой на вход подсовывают какой-то X в конкретном случае (определённое число), в этом случае мы получим конкретное высказывание и сможем проверить его истинность. Проверить истинность – это значит сказать однозначно либо «Да» (истина), либо «Нет» (ложь).

Попробуем рассмотреть такое высказывание: «Сумма чисел 2 и 6 – число чётное». Является ли оно истинным? В принципе, из уроков математики мы должны помнить, что сумма чётных чисел будет равна чётному числу. С одной стороны, это точно высказывание, с другой стороны, истинность этого высказывания – истина.

«Сумма двух чисел – число чётное». Такое предложение, которое, казалось бы, может являться утверждением, на самом деле не является высказыванием. Почему? Потому что прежде всего возникает вопрос: каких чисел? Если мы возьмём сумму двух нечётных чисел, тогда число получится чётное. Но если будет взято число, например, чётное и нечётное, будет очевидно, что ответом будет «Нет». В этом случае определённость отсутствует. Следовательно, тезис не является высказыванием.

Следующее. «Сумма двух чисел, А и В, – число чётное». Такой тезис не является высказыванием, потому что в этом конкретном случае мы не знаем, чем являются А и Б. Но в то же время мы можем перейти к функции, если быть точным, – к функции-предикату от двух аргументов, которая точно и гарантированно даёт отображение либо «Да», либо «Нет». В конкретном применении, если вместо А, например, будет число 2, а вместо Б будет число 10, то их сумма – число чётное. И истинность высказывания будет «истина». И если у нас будет, например, 1 и 10, в этом случае истинность высказывания будет «ложь».

В программировании предикаты бывают не только одноместными (один аргумент), или двухместными (два аргумента). Они могут быть n-местными. То есть таких аргументов может применяться сколько угодно.

Самый простой пример: нам требуется найти сумму 10-ти чисел, 20-ти чисел, 30-ти чисел и сделать, например, отображение того, что сумма этого набора чисел будет числом чётным или нечётным.

Итак, перейдём к тому, что точно применяется в программировании. Попробуем проверить истинность высказывания «2 – число чётное». Вы точно знаете о том, что это истина. Если мы перейдём к предикату P от X, где X – число чётное, то какие значения могут получиться в итоге? Значения могут быть либо «Да», если имеется конкретный X, либо «Нет», если, опять же, есть конкретный X. Но если бы было, например, число 2, – это чётное число, поэтому результатом будет единица или истина. И, соответственно, если бы было число 3, тогда это будет ноль, или ложь. Например, что получится. если попробовать сделать инверсию в контексте математики этого предиката, в контексте программирования этого выражения? Допустим, некоторое высказывание Q от X, которое звучит как «X – число нечётное». Подумайте, можно ли ещё каким-то образом сделать инверсию нашего высказывания P от X? Даю подсказку: да, можно. Например, X – нечётное число. Технически это также будет другим высказыванием, но суть его не поменяется.

Таким образом для того, чтобы не вводить отдельные буквы для практически одних и тех же высказываний, существует такая логическая операция – инверсия, или отрицание. Обозначается это по аналогии со множествами таким значком, и определяется она, как если бы P от X было ложно, то инверсия высказывания P от X будет истиной. Соответственно, если исходное высказывание было истинно, то инверсия этого высказывания будет ложно.

В качестве самостоятельной работы попробуйте сделать инверсию представленных на этом слайде высказываний. На это вам 3 минутки.

## [00:47:54]

## Логическая операция «И»

Хорошо. Надеюсь, у вас получилось. А для тех «жуликов», ожидающих, что я здесь и сейчас расскажу результаты, я скажу вам, друзья, что мы это разберём на семинаре.

Очень важное место занимает логическая операция «И». Ещё её называют логическим умножением или в разделе математики математическая логика, операция логическая «И» называется конъюнкцией. Каким образом дать определение этой операции? Допустим, у нас есть два высказывания. Первое высказывание: «Те, кто закончит курс, получают сертификат». Второе: «На южном полюсе холодно». Чтобы перейти от простых предложений к функциям, нам нужно ввести эту самую функцию. Напоминаю о том, что эта функция предикат. Допустим, P и Q. Для того, чтобы использовать логическую связку «И», мы можем записать её по-разному. Можно написать просто союз «и», или, как в некоторых языках программирования, слово «and», в некоторых случаях амперсанд, в некоторых случаях двойная амперсанда, или ещё что-то. Я, к сожалению или к счастью, всех языков программирования не знаю.

Каким образом звучит определение конъюнкции двух высказываний? Звучит оно так: «Высказывание истинное тогда и только тогда, когда оба исходных высказывания истинны». Для тех, кто слышит такое определение в первый раз, я рекомендую вдуматься в него и записать это как-то более просто. Строится такая табличка, где для некоторого высказывания P определяется его возможное значение. Как вы понимаете, исходное первое высказывание может быть либо ложным, либо истинным, так и запишем: P это либо нолик, либо единичка. Далее в эту же таблицу добавляется ещё один столбик, где мы указываем Q, и точно таким же образом это высказывание может быть либо ложным, либо истинным.

В случае если мы применим логическую связку «И», может получиться так, что первое и второе высказывание окажется ложно. Второй возможный вариант – если оба этих высказывания будут истинны. Всё ли это возможные вариации? Конечно же, нет, так как первое высказывание, например, может быть ложным, а второе истинным, и наоборот, первое – истина, а второе – ложь. Как это отобразить в таблице? В этом случае «ноль-один» означает, что первое ложно, а второе истинно.

Теперь добавим третий столбик, в котором указываем таким вот образом. Кстати, в математической логике операция конъюнкция определяется именно таким значком. И теперь, вспоминая определения, надеюсь, вы его запомнили. Можно отметить его как «истина тогда, когда оба истина» и в принципе этого хватит. Таким образом, если первое высказывание – истина и второе – истина, то получится истина. А дальше можно не думать и просто написать везде нолики, этого будет более чем достаточно.

## [00:51:04]

## Логическая операция «ИЛИ»

Следующая логическая операция — это так называемое логическое «ИЛИ», ещё его называют логическим сложением. В математической логике эта операция зовётся дизъюнкцией. Как звучит определение дизъюнкции: «Дизъюнкцией двух высказываний называется высказывание ложное тогда, когда оба ложные».

Если строить таблицу истинности, мы таким же образом должны будем выписать все возможные вариации высказываний, входящих в наше большое высказывание, и дальше «ложно тогда, когда оба ложные». Соответственно, где нолики – ставим нолик, а всё остальное априори истина.

Почему называется логическим сложением или логическим умножением? Вы могли заметить, что в этом случае «0 плюс 0 это 0», «0 плюс 1 это будет 1». «1 плюс 0 это тоже единичка», а «1 плюс 1 – точно не ноль». Поэтому, так как мы делаем отображение во множество ноль и один, и если это не ноль, то у нас вариантов больше не остаётся.

## [00:52:01]

## Логическая операция «ИСКЛЮЧАЮЩЕЕ ИЛИ»

Следующая логическая операция, которая имеет непосредственное отношение, например, в какой-то степени к теории шифрования, это так называемое «ИСКЛЮЧАЮЩЕЕ ИЛИ». В математической логике зовут эту операцию разделительная дизъюнкция. Предлагаю вам дать определение этой операции самостоятельно, а на семинаре мы разберём более подробно. Но я покажу получающаюся таблицу истинности. В частности, если первые два высказывания, входящие в большое высказывание, ложны, то мы определяем его как ложь. Также оно ложно если два исходных высказывания истинны. Получается, что истина у нас будет только в том случае, когда только одно из высказываний есть истина.

Теперь попробуйте проговорить это и, может быть, даже записать как-то более формально.Вот эти четыре логических операций покрывают абсолютно весь перечень условий, которые могут быть использованы при написании программ у разработчиков. Далее мы посмотрим на определения (логические операции), которые имеют непосредственное отношение к математической логике. Но к сожалению или счастью, практически не используется в программировании.

## [00:53:13]

## Логическая операция — эквиваленция

Следующая операция, которая имеет непосредственное отношение к математической логике, но практически не имеет отношения к программированию, – операция эквиваленция. Попробуйте таким же образом дать определение через ту операцию, которую вы только что узнали – разделительную дизъюнкцию. Здесь, если посмотреть, можно увидеть, что происходит инверсия ноликов и единичек. В качестве задания, разбираемого на семинаре, будет доказательство вот такого утверждения.

Каким образом оно доказывается? Нам потребуется построить таблицу истинности. Для начала все возможные вариации P и Q. Затем необходимо найти P исключающее или Q, затем сделать инверсию и далее, соответственно, проверить, будет ли это истиной.

## [00:54:00]

## Логическая операция — импликация

Итак, импликация — это последняя операция, рассмотренная в рамках этой лекции. В чём её особенность? Здесь можно простыми словами сказать, что, если есть ложь, из лжи может следовать истина. А может следовать и ложь. И из истины следует истина. В то же время из истины ложь следовать не может. В принципе, это то, что нужно помнить об этой операции. Будете ли вы использовать её на практике? Крайне маловероятно, если дело касается программирования. Если вдруг когда-нибудь вы захотите что-то доказывать или, например, занимаетесь какой-то теоретической математикой, то тогда, скорее всего, пригодится.

### [00:54:40]

# Докажите тождество

Далее в качестве простой задачи я предлагаю вам попробовать доказать вот это тождество. Но вы спросите, а как же его доказывать? Очень просто. Попробуем проверить истинность вот такого выражения. Каким образом это вообще делается? Есть какое-то очень большое выражение. В первую очередь, вам нужно изучить, способ и порядок выполнения операций. Сначала узнаём это. Самый большой приоритет или любой приоритет, ломает наличие скобок. Действие, указанное в скобках, должно выполняться в первую очередь. Следом идёт инверсия, затем идёт конъюнкция, дизъюнкция, исключающая дизъюнкция и все следующие операции.

Итак, зная это, мы теперь можем расставить в нужном порядке те действия, которые будут выполняться. Первое в этом случае – действия в скобках. Следующее – инверсия А, затем идёт инверсия Б, затем действие в скобках и последнюю операцию. После того как мы определили порядок операций, можно построить такую табличку. Шапка этой таблицы выглядит ровно так же, как мы расставляли операции. Здесь, обратите внимание, я написал в последнем столбике F, потому что не хочу писать достаточно большое выражение, то есть просто для красоты. На следующем этапе мы должны перебрать все возможные варианты, которые встречаются у наших высказываний А и Б. А это вариации: ложь – ложь, ложь – истина, истина – ложь и истина – истина. Далее мы будем выполнять каждое действие по порядку. Чтобы вам было чуть-чуть проще, я буду справа показывать таблицу истинности для выполняемого действия.

Итак, в этом случае, если мы выполняем «А импликация Б», то, соответственно, рабочими колонками для нас будут первая и вторая. Теперь здесь нет никаких проблем, просто нужно переписать «из лжи следует ложь, это истина», «из лжи следует истина, это истина» и так далее.

На следующем этапе нам нужно проделать инверсию А. В этом случае инверсия – это слишком простая операция, то есть, если была истина, будет ложь, если ложь, соответственно, будет истина. Как следствие, нашим рабочим столбиком здесь получается первый, и делаем инверсию. Аналогичным образом поступаем с инверсией Б. Следующим этапом нужно найти дизъюнкцию инверсии А и инверсии Б. Теперь рабочими столбиками будут являться четвёртый и пятый. Соответственно, по таблице истинности этой операции мы получаем такое вот заполнение.

Всё, что нам останется сделать, это всего лишь проверить истинность всего нашего высказывания, то есть рабочими столбиками будут третий и предпоследний. Зная нашу операцию, мы, соответственно, заполняем нужными значениями. Примерно вот так проверяется истинность высказывания.

Кстати, от себя ещё отмечу, что такое высказывание, которое истинно при любых наборах входящих в него предикатах, называется тавтологией. Что вам ещё может потребоваться в том случае, если вы захотите проверить истинность каких-то высказываний или какие-то математические утверждения, если быть точным, логические утверждения? В этом случае это те законы, которые имеются в рамках алгебры логики. Некоторые из них я привожу на экране. Но рассмотрим, например, закон двойного отрицания. Суть его очень простая: если берётся отрицание отрицания какого-то высказывания, то это высказывание получается тождественно равно тому, которое было в самом начале. Вы можете подумать, что несёт этот человек? На самом деле, друзья, всё просто. Представьте высказывание «Это молоко горячее», а теперь я скажу двойное отрицание этого же высказывания «Не молоко горячее», звучит странно, правда? Примерно таким же образом и мы в жизни не делаем таких двойных отрицаний, потому что в этом нет никакого смысла. Простое отрицание – да, конечно – «Это молоко не горячее», например, или «Не это молоко горячее», но двойное не особо нужно.

Каким образом проверить истинность такого тождества? Очень просто: мы снова строим с вами таблицу истинности, указываем изначальный набор для высказывания. А это, может быть, ложь и истина. В том случае, если у нас будет инверсия, это будет, как вы можете наблюдать во второй колонке, истина и ложь. И если мы снова сделаем инверсию инверсии, мы опять получим то, что было в первом столбике.

Кроме закона двойного отрицания в алгебре логики есть ещё куча других законов. Скорее всего, вы не будете их использовать постоянно, вам они не пригодятся в том случае, если вы не будете заниматься действительно доказательствами. Когда дело касается разработки или написания приложений, в большинстве случаев мы обходимся с вами инверсией, конъюнкцией и дизъюнкцией. В некоторых случаях используется разделяющая дизъюнкция. Импликация и эквиваленция, скорее всего, вам не понадобятся. Тем не менее, здесь я решил добавить некоторое количество этих законов, которые называются законами Де Моргана. Для чего они нужны? Для упрощения и не более, чем. Некоторые из них вы можете попробовать подоказывать. Попробуем проверить истинность первого закона. В этом случае мы должны взять высказывание, имеющиеся в левой части, и высказывание, находящиеся в правой части, чтобы построить таблицу истинности для каждой из этих частей и убедиться в том, что они будут соответствовать друг другу. Каждое значение будет соответствовать друг другу.

# Завершение урока

На этом, друзья, я с вами прощаюсь. Надеюсь, что из этой лекции вы узнали для себя что-то новое. И у вас не будет никаких проблем с логикой, и до встречи на семинарах!