МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский государственный университет аэрокосмического приборостроения»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

|  |  |
| --- | --- |
| на тему | Автоматизация программирования вычисляемых вопросов для системы |
| тестирования lms moodle | |
|  | |

|  |  |
| --- | --- |
| выполнена | Гохманом Ильей Михайловичем |
| фамилия, имя, отчество студента в творительном падеже | |

|  |  |  |  |
| --- | --- | --- | --- |
| по направлению подготовки/ специальности | 09.03.03 |  | Прикладная информатика |
|  | код |  | наименование направления подготовки/ специальности |
|  | | | |
| наименование направления подготовки/ специальности | | | |
| направленности | В экономике | | |
|  | наименование направленности | | |
|  | | | |
| наименование направленности | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент группы № | 8726 |  |  |  | И.М.Гохман |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский государственный университет аэрокосмического приборостроения»

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ   
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ БАКАЛАВРА

|  |  |  |  |
| --- | --- | --- | --- |
| студенту группы | 8726 |  | Гохман Илья Михайлович |
|  | номер |  | фамилия, имя, отчество |

|  |  |
| --- | --- |
| на тему | Автоматизация программирования вычисляемых вопросов для системы |
| тестирования lms moodle | |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| утвержденную приказом ГУАП от |  | № |  |

|  |  |
| --- | --- |
| Цель работы: | Заключается в практическом применении систем для повышения полезной |
| работы сотрудника | |
|  | |

|  |  |
| --- | --- |
| Задачи, подлежащие решению: |  |
|  | |
|  | |

|  |  |
| --- | --- |
| Содержание работы (основные разделы): |  |
|  | |
|  | |
|  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Срок сдачи работы « |  | » |  | 20 | 21 |

Задание принял(а) к исполнению

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| студент группы № | 8726 |  |  |  | И.М.Гохман |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Реферат

Актуальность темы.

Цель и задачи работы, которые определяются исходя из темы ВКР.

Объект и предмет исследования или разработки.

Содержание

[Введение 6](#_Toc68981859)

[1. Обзор области 7](#_Toc68981860)

[1.1. Moodle 7](#_Toc68981861)

[1.2. Допустимые формулы 13](#_Toc68981872)

[1.3. Python 15](#_Toc68981874)

[1.4. Pyramid 18](#_Toc68981879)

[1.5. Flask 19](#_Toc68981880)

[1.6. Tornado 20](#_Toc68981881)

[1.7. PHP-фреймворки 21](#_Toc68981882)

[1.8. Laravel 22](#_Toc68981883)

[1.9. Symphony 23](#_Toc68981884)

[1.10. CakePHP 24](#_Toc68981885)

[1.11. Причины использовать Django 25](#_Toc68981886)

[1.12. Обзор решений 26](#_Toc68981887)

[1.13. Перевод формул в код 27](#_Toc68981888)

[1.14. Обзор области применения 28](#_Toc68981889)

[2. Разработка формул для новой системы 29](#_Toc68981892)

[Заключение 48](#_Toc68981893)

[Приложение 1 49](#_Toc68981894)

[Тезариус 50](#_Toc68981895)

[Список литературы: 51](#_Toc68981896)

# Введение

# Обзор области

## Moodle

Что это?

Moodle - веб-приложение, позволяющее создать кастомизированную систему управления обучением.

Первые прототипы системы были созданы в 1999 году Мартином Дугиамасом (Martin Dougiamas), на тот момент студентом Университета Кертин, Австралия. Первый сайт на базе Moodle был создан в 2001 году, а сейчас число сайтов на Moodle перевалило за 1 миллион по всему миру. За это время вокруг Moodle сформировалось большое и сильное сообщество, которое разрабатывает плагины (модули) и делится ими в сети. C помощью плагинов можно создать уникальную систему управления обучением, которая по функционалу будет не хуже современных СДО.

Moodle - система управления курсами (электронное обучение), также известная как система управления обучением или виртуальная обучающая среда. Является аббревиатурой от англ. Modular Object-Oriented Dynamic Learning Environment (модульная объектно-ориентированная динамическая обучающая среда). Представляет собой свободное (распространяющееся по лицензии GNU GPL) веб-приложение, предоставляющее возможность создавать сайты для онлайн-обучения. Первая версия написана 20 августа 2002 года.

СДО Moodle занимает примерно 18 % рынка в США.

Платформа предоставляет пространство для совместной работы учителей и студентов. В Moodle доступны различные возможности для отслеживания успеваемости учащихся, а также есть поддержка массовой регистрации с безопасной аутентификацией.

Система имеет гибкий интерфейс с возможностью конфигурирования макетов и дизайна отдельных страниц. Платформу можно интегрировать с большим количеством программного обеспечения, включая инструменты для общения, совместной работы, управления документами и другие приложения для повышения производительности. Moodle имеет открытый исходный код.

### Возможности

### Что можно делать в системе Moodle

Став администратором в Moodle, вы можете выполнять следующие функции:

1. создавать курс и управлять им;
2. добавлять учеников;
3. оценивать успеваемость;
4. отслеживать прогресс учащихся;
5. разграничивать доступ к учебным материалам и блокам курса.

Рассмотрим основные возможности Moodle подробнее.

### Создание и управление курсом

Учебный курс в Moodle, который создается через встроенный редактор, может состоять из нескольких элементов. К примеру, самый простой и распространенный – это лекции, задания и тесты. В систему загружается обучающий контент, из которого формируются лекции. Это могут быть текстовые файлы в формате PDF, XLS, а также видео, фото, презентации и аудио.

Вот так, к примеру, выглядит форма создания курса:

### Форма создания курса Moodle

Для заданий администратор может использовать кейсы, в которых моделируется рабочая ситуация, а обучающийся дает развернутый ответ.

Курсы формируются в системе в режиме редактирования. В зависимости от задач они наполняются разными элементами, каждый из которых имеет свои настройки. Помимо лекций это могут быть вебинары, семинары, тренинги. Во время семинара, к примеру, можно организовать тестирование или дать обучающимся задание.

Курс в Moodle можно наполнить различными элементами: анкетами, опросами, семинарами, лекциями – «начинка» зависит от целей и задач обучения

Если у вас есть опыт работы с редакторами курсов (Articulate Storyline, Courselab, Easygenerator, Ispring Suite или QuizMaker), вы можете собрать и загрузить в Moodle SCORM-архив – готовый для изучения пакет файлов. Система поддерживает форматы SCORM, AICC и IMS, LTI, для xAPI нужен плагин (например, Logstore API).

Среди учебных элементов есть еще вики-статьи, глоссарий, форумы, чаты. Вики – коллекция веб-документов, которые ученики создают совместно. По сути, это веб-страница, которую каждый в вашем классе может создать прямо в браузере без необходимых знаний HTML.

Глоссарий в Moodle можно создавать, искать или просматривать в разных форматах. Все записи разбиваются по категориям. А в лекциях термины, которые находятся в глоссарии, будут автоматически выделены ссылками.

### Анкетирование и тестирование

Провести аттестацию, собрать обратную связь по обучению или узнать мнение большинства по какому-то решению можно в системе с помощью опросов и анкетирования. Оба элемента встроены в систему – анкеты собираются с готовыми вопросами, а опросы составляет администратор.

### Панель тестов в Moodle

### Тестирование в Moodle

Moodle позволяет организовать различные педагогические сценарии дифференцированного обучения. Как правило, администраторы пользуются двумя распространенными сценариями:

1. выделение группы – создается контент для каждой группы;
2. ограничение доступа к элементу – если вы не прошли тест по теме 1, вам не дадут доступ к материалам темы 2.

### Тестирование

Какую бы форму обучения Вы не практиковали - очную или онлайн-обучение, периодически Вам необходимо осуществлять контроль усвоения знаний. Одним из наиболее распространенных методов контроля является тестирование — метод, состоящий в предоставлении испытуемому системы заданий, позволяющих эффективно измерить уровень и структуру его знаний.

Электронное тестирование качественно отличается от тестирования, выполняемого на бумажном носителе, благодаря:

1. Возможности мгновенной автоматической проверки и оценки заданий (с закрытым ответом).
2. Автоматическому статистическому анализу теста и его элементов.

Если первое в одинаковой степени интересно и проверяющему, и тестируемому (в случае полностью самостоятельного обучения это может быть один и тот же человек), то второе крайне важно для проверяющего, точнее, для разработчика теста, так как позволяет понять насколько тест эффективен и предпринять действия по его доработке.

Помимо контроля знаний, с помощью электронных тестов может быть решена задача развития того или иного умения, перехода умения в навык. Для этого можно использовать тесты-тренинги, предполагающие несколько попыток ответа на вопрос. В таких тестах можно дать возможность обучающемуся осуществлять своеобразный «поиск знаний». Делается это посредством добавления отзывов-комментариев, отображаемых после выбора варианта ответа. При этом может использоваться система начисления штрафов за каждую дополнительную попытку (вычет из итогового балла).

На сегодняшний день имеется множество средств разработки электронных тестов. Такой функционал заложен и в авторские средства (Articulate Storyline, iSpring Suite и др.), и, непосредственно, в системы управления обучением (Moodle, eFront, eLearning 4g и др.).

Используя Moodle для организации электронного тестирования, Вы получаете действительно мощный инструментарий для создания тестов, одновременно с хорошим анализатором качества теста и его составляющих — тестовых заданий. Едва ли объема данной статьи может хватить для того, чтобы описать все возможности системы тестирования в Moodle, поэтому мы попробуем дать лишь краткий обзор.

Управление тестовыми вопросами в Moodle осуществляется через «Банк вопросов». Если Вы (возможно, совместно с коллегами) создаете курс или комплекс курсов в Moodle, то, наверняка, там будут тесты, и, скорее всего, их будет много. Тесты могут решать задачи входного контроля, текущего, итогового контроля или это могут быть тесты-тренажёры. При этом, тестовые вопросы могут быть общими для некоторых тестов, а также выбираться случайным образом из некоторого набора вопросов — оба эти условия могут быть реализованы благодаря «Банку вопросов». Кроме того, используя «Банк вопросов» легко организовать совместную работу над созданием тестовых вопросов и быстро найти нужный вопрос для теста.

### Банк вопросов

Базовая сборка Moodle включает возможность создавать следующие типы тестовых вопросов:

1. выбор одного/нескольких из нескольких;
2. верно/не верно;
3. краткий открытый ответ (или «вопрос с пропуском»);
4. числовой ответ;
5. вычисляемый (по формуле) ответ;
6. установление соответствия;
7. эссе (проверяется вручную);
8. вложенные ответы (комбинированный).

Множество плагинов для Moodle позволяют реализовать вопросы drag&drop, выбор области на изображении, открытые вопросы с механизмом проверки, использующим регулярные выражения, и другие типы вопросов.

При разработке теста в Moodle необходимо знать следующие основные возможности системы:

Тест может быть ограничен по времени и по числу попыток его прохождения;

День и время доступности теста могут быть жестко заданы, либо тест может быть открыт всегда;

Порядок представления вопросов в тесте и вариантов ответов в вопросах может быть, как заданным, так и случайным;

Тест может быть запущен в различных режимах, например, в адаптивном режиме тестируемому предоставляется неограниченное количество попыток ответа на каждый из вопросов теста (предполагается использование подсказок), при этом за неправильные ответы могут начисляться штрафные баллы, которые вычитаются из итоговой оценки за прохождение теста;

Отчет о результатах теста для тестируемого может быть гибко настроен;

Итоговая оценка за тест складывается из баллов, полученных за каждый вопрос теста, с учетом веса вопроса. Оценка выражается в процентах (доля баллов от максимально возможного). Настройки теста - режим немедленного отзыва

Завершив разработку теста, следует обязательно проверить его работу, чтобы внести возможные корректировки как в настройки теста, так и в содержание вопросов, прежде, чем тест будет пройден испытуемыми. После того, как все предварительные работы завершены, тест можно открыть.

### Вычисляемые вопросы

## Допустимые формулы

### Математические формулы

## Python

Краткая история Python

Python - это универсальный язык программирования высокого уровня, широко используемый для различных проектов разработки. Он был разработан и создан Гвидо ван Россумом и выпущен в 1991 году после разработки в конце 80-х годов. Первоначально язык был создан как ответ на язык программирования ABC из Нидерландов.

Гвидо ван Россум создал язык, который делает упор на удобочитаемость кода. Это позволяет инженерам выражать концепции и строить идеи в меньшем количестве строк кода по сравнению с другими языками программирования. Это значительно упрощает разработку проектов на Python.

Его многочисленные программные парадигмы делают Python излюбленным для использования в крупных организациях или организациях корпоративного уровня, но его любят разработчики в организациях любого масштаба и бизнеса. Он предлагает объектно-ориентированное, императивное и функциональное программирование, а также большую стандартную библиотеку, динамические функции и полезный функционал управления памятью.

### Почему разработчики выбирают Python?

Python является языком программирования для большинства разработчиков в сегодняшнем технологическом ландшафте и существует с начала 90-х годов. Его многочисленные преимущества и оптимизированные функции делают его главным конкурентом для проектов разработки программного обеспечения.

Ниже приведены лишь несколько причин, по которым многие разработчики тяготеют к Python при выборе языка программирования:

Python бесплатен и всегда будет бесплатным. Python, его набор вспомогательных инструментов расширения, библиотек и модулей - абсолютно бесплатны. Большинство популярных интегрированных сред разработки (IDE), включая PTVS, Pydev, Eclipse и Spyder Python, можно загрузить бесплатно. Это язык с открытым исходным кодом, и является таковым для всех благодаря поддержке Python Software Foundation.

Он используется крупнейшими технологическими корпорациями. Python является предпочтительным языком программирования для большинства гигантов в мире информационных технологий. К ним относятся: Google, Dropbox, Instagram и Spotify, и это лишь некоторые из них. За пределами мира ИТ многие другие огромные организации также используют этот мощный язык, включая Disney, NASA и Electronic Arts. Также язык Python является предпочтительным во многих технологических стартапах.

С ним легко работать, что сокращает время разработки. Написанный в удобочитаемом формате, Python делает процесс разработки программного обеспечения быстрым, удобным и максимально упрощенным. Если разработчик имеет базовые знания любого другого языка программирования, он может легко изучить Python и внедрить его в свои проекты.

Python является одним из самых популярных языков программирования сегодня - и на это есть веские причины. Он бесплатный, с открытым исходным кодом, надежный и безопасный, но при этом обеспечивает легкую масштабируемость и гибкость при необходимости. Он также поддерживается большим и полезным онлайн-сообществом.

Python также сокращает время и затраты, связанные с разработкой программных проектов. Многие преимущества языка уже сделали его фаворитом среди разработчиков, и он будет оставаться фаворитом в обозримом будущем, так как технология продолжает развиваться.

### Django

### Альтернативы

Фреймворки ускоряют разработку веб-приложений. И в экосистеме Python есть несколько фреймворков, которые можно рассматривать в качестве альтернативы Django. В их число входят Pyramid, Flask и Tornado. Все три инструмента подходят для работы как с масштабными проектами, так и с небольшими приложениями.

Также в качестве альтернативы можно рассматривать PHP-фреймворки, например, Laravel, Symphony или CakePHP. Например, Laravel подходит для работы с масштабными и маленькими проектами, и в этом фреймворке есть мощный движок шаблонов.

Если для вашего проекта критически важна безопасность, Laravel можно считать инструментом первого выбора. В нём реализованы мощные инструменты шифрования. Этот PHP-фреймворк работает с базами данных MySQL, PostgreSQL, SQL Server, SQ Lite.

Ещё одна достойная альтернатива — фреймворк Ruby on Rails. Этот инструмент ускоряет создание веб-приложений и позволяет разработчикам сфокусироваться скорее на бизнес-логике, чем на коде. Кстати, наш сайт создан на Ruby on Rails.

Достойные альтернативы Django существуют, а выбор инструмента всегда зависит от особенностей проекта, языка программирования, с которым вы работаете, и других факторов.

### Python-фреймворки

## Pyramid

## Flask

## Tornado

## PHP-фреймворки

## Laravel

## Symphony

## CakePHP

## Причины использовать Django

## Обзор решений

## Перевод формул в код

## Обзор области применения

### {рассмотреть какие задачи будут решаться (логические, программные циклы, if)}

### Какие системы использовались ранее

# 

# **Разработка формул для новой системы**

1. Условия

Любая логика программы задается условиями, выполнение или невыполнение которых приводит к одному из возможных действий. Прежде, чем мы перейдем к конкретным конструкциям, стоит рассмотреть работу логических операторов в математическом исполнении. Первая из поставленных задач, это поиск функции, которую можно использовать для преобразования чисел в один из двух ответов: 0 и 1. Если условие выполняется, то ответ равен 1, если нет, то ответ 0. Другое название такой функции - это предикат. Предикат - это функция, возвращающая булевое значение. Булевое значение это значение в виде истины или лжи.

Суть условия в программировании – это выполнение действия, которое закреплено за условием. В математике же, подобное действие можно выполнить следующим образом:

Формула 1

Результатом действий может быть последовательность математических операций, вычисляемых как ответ конкретного условия. Так как условие при истинности обращается в 1, а в другом случае 0, то в зависимости от условия результат умножается на 1 и выводится в ответ, или умножается на 0 и не учитывается.

Теперь найдем функцию, которая бы удовлетворяла нашим требованиям. Начать поиск стоит с определения области поиска, так как платформа moodle может использовать только математические формулы, при этом используемые в языке программирования PHP. В ходе проверки работоспособности формул PHP в системе moodle было установлено, что не все математические функции языка программирования корректно работают в moodle. В ходе поиска надлежащей функции по графикам была выдела функия гиперболического тангенса - сокращенно tanh.

Другие полезные функции, поддерживаемые moodle: abc – модуль числа и floor – округление в меньшую сторону.

Преимущество гиперболических функций тригонометрии над негиперболическими в том, что гиперболические не являются периодическими. Это говорит о том, что функция не повторяет сама себя и у функции есть два значения абсциссы, после которых при увеличении модуля x значение ординаты неизменно. Рассмотрим график функции tan и tanh на рисунке1*.*

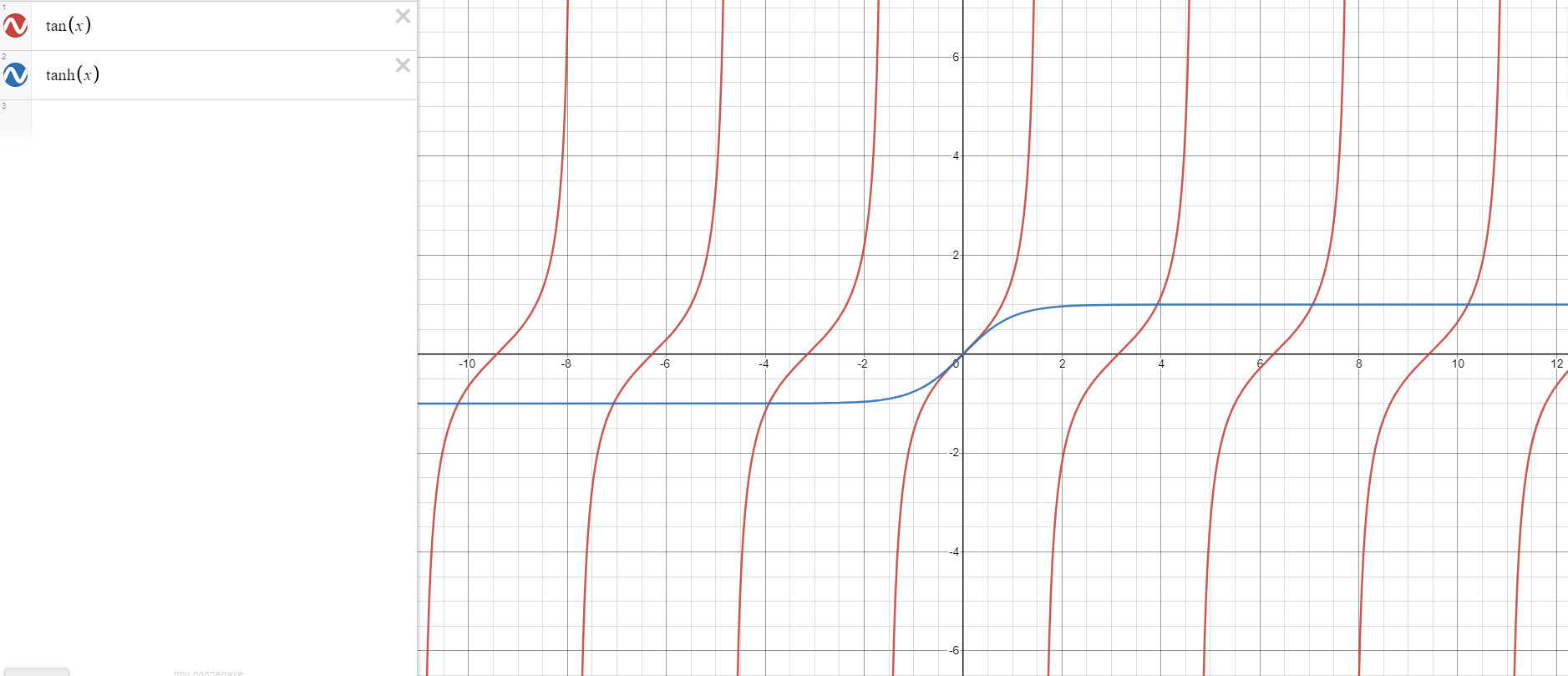


Рисунок 1.

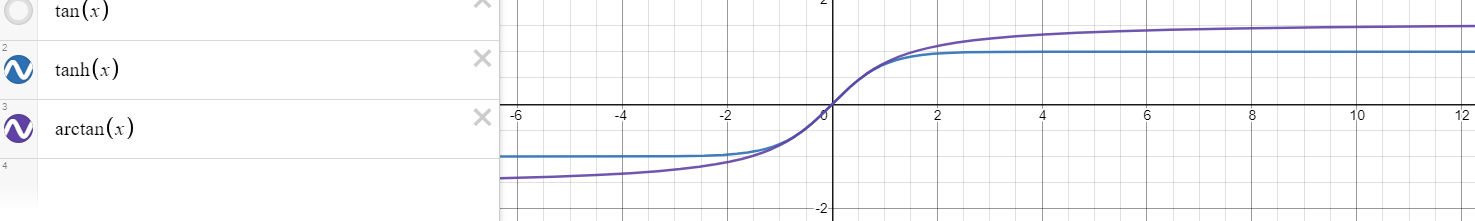
Красным изображена функция tan со свойственной ей периодичностью. Синим показана функция tanh, меняющая свое значение ординаты только в промежутке примерно от -4 до 4, далее сохраняя значение до бесконечности. При использовании функции tan нам бы пришлось учитывать её периодичность и подгонять значения в промежутки одного периода. А при выходе за пределы периода значения расчетов были бы уже не верны. Допустим мы работаем с двумя числами 1000 и 1. Если взять эти числа за X, то совершенно очевидно, что они находятся в разных периодах функции и для расчетов требуется приводить эти два числа в пределы одного периода. Если же взять за Y то нужно использовать обратную функции tan, arctan. Функции arctan очень похожа на tanh, это можно наблюдать на рисунке 2.

Рисунок 2.

Функция tanh, в сравнении с arctan, лучше тем, что промежуток изменений ординаты в первой занимает около 8 единиц, во второй растягивается на тысячи. Промежуток между максимальным значением функции и минимальным в tanh равен двум. В функции arctan идет привязка к числу пи и разница равна пи. Минимальное значение функции tanh равно -1, а максимальное 1. Те же параметры функции arctan:

От -. Так как в наших вычислениях не требудется тригонометрия, а значения π будут только создавать нецелые числа, воспользуется функцией tanh. Рассмотрим график функции поближе к значению нуля.

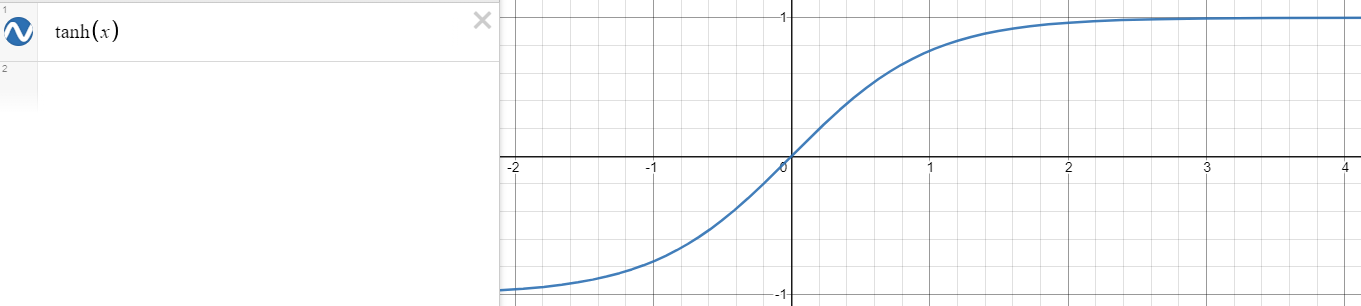


Рисунок 3.

График проходит через 0 под углом 45 градусов. Он является по модулю симметричным. Эти свойства будут очень полезны в использовании. Также значения от стабильного -1 относительно резко переходят в значение 1, такое поведение лучше всего видно на промежутке от -5 тысяч до 5 тысяч, наблюдает на рисунке 3*.*, мне очень напомнило упрощенный график работы биполярного транзистора, представленный на рисунке 0. Условие в своем роде будет служить транзистором, то есть в нужный момент передаст значение 0 или 1.

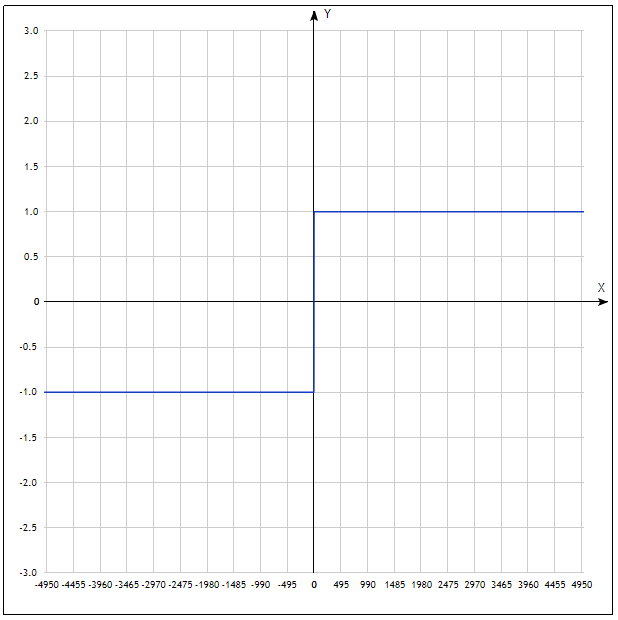


Рисунок 4.

Промежуток в 10 тысяч

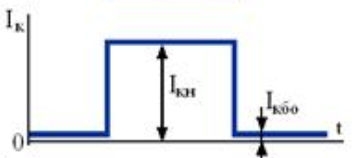


Рисунок 5. - транзистор

Осталось изменить функцию таким образом, чтобы можно было назвать её предикатом. Для начала нужно сделать так, чтобы основная часть положительной стороны по x возвращала 1, при этом основная часть отрицательной стороны возвращала 0. Для этого, мы будем использовать функцию округления floor, упомянутую ранее. Данная функция округляет число в сторону –inf. В следствии этого, значения положительной половины по большей части уже округляются в 1, а значения отрицательной половины округляются в -1. Нужно сдвинуть график таким образом, чтобы округление происходило в 0, значит левая сторона графика должна находиться в промежутке [0;1). Данный сдвиг проще всего сделать прибавлением единицы после функции. Но мы знаем, что верхняя граница тоже сместится на 1 вверх, и тогда значение 2 будет округляться до 2, что является неприемлемым поведением. На рисунке6*.* красным показано округление синей функции.

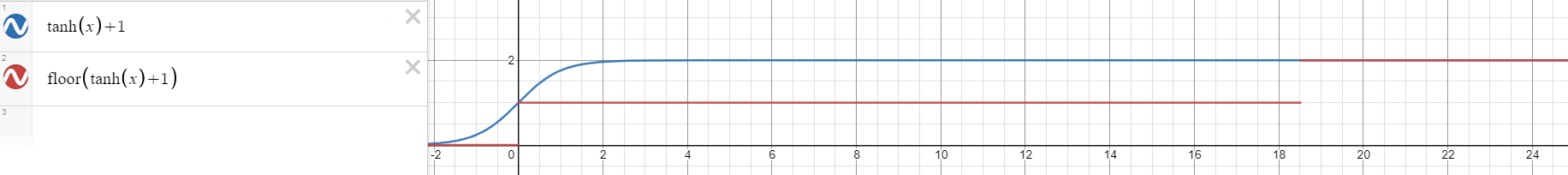


Рисунок 6.

1. Округление

Так как размах функции по Y в две единицы ровно, а для округления в 0 и 1 требуется расстояние меньше двух, то нужно найти способ сделать расстояние немного меньше. На помощь в данном случае придет функция abc, которая может отрицательные числа преобразовывать в положительные. Сместим график относительно прошлого положения на минимально возможно расстояние меньше двух и применим модуль к функции, для того, чтобы все числа, которые от нуля сдвинулись вниз по Y, сдвинулись вверх на тоже расстояние и, соответственно, стали положительными. На рисунке7*.* видно единственное пересечение точки с прямой абсцисс. Все значения являются положительными и минимум данной функции находится в точке [-6.103;0]. Округление теперь работает как мы и задумывали.

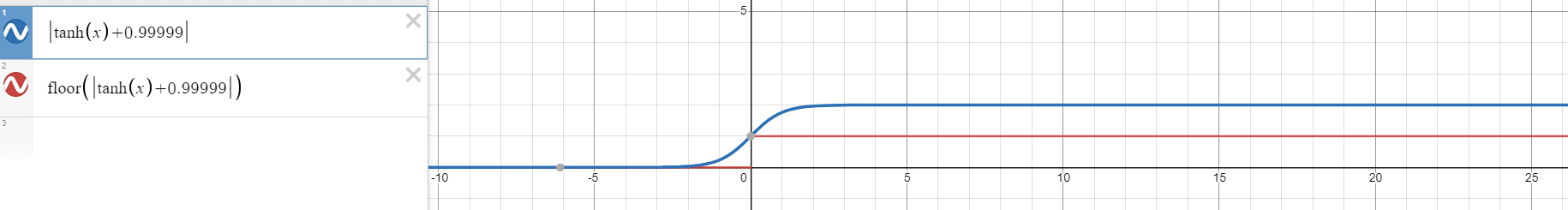


Рисунок 7.

Зададим точность, с которой планируем работать и выразим её из предыдущей формулы во всех случаях где она не явно использовалась или влияет на вычисления. Точность расчетов на графике установлена константой “а”, которая равна 10-7. Вынесение её отдельно несет практическое значение – быстрое изменение точности в дальнейшем, не изменяя формулы.

Формула 2

Точность важна при сдвиге вверх и вниз, так как от этого напрямую зависит точка, в которой график пересечет 1 на оси Y. Также точность важна для самого X, так как из-за сдвига графика немного вниз, сам график проходит чуть правее единицы по Y. Прибавление к X точности вычисления сдвигает график левее, тем самым снова возвращая в положение, когда происходит пересечение в точке [0;1].

Синим цветом показана основная функция, а зеленым и красным её округленные наследники. Эти две функции округляют значение функции и устанавливают промежуток, в котором идет пересечение с осью ординат с заданной точностью.

Формулы почти идентичны, за исключением значений около точки 0. При X = 0 первая функция вернет 1, а вторая 0. С помощью такого поведения можно имитировать знаки больше и больше равно и им противоположные, которые будут включать равенство или игнорировать его.

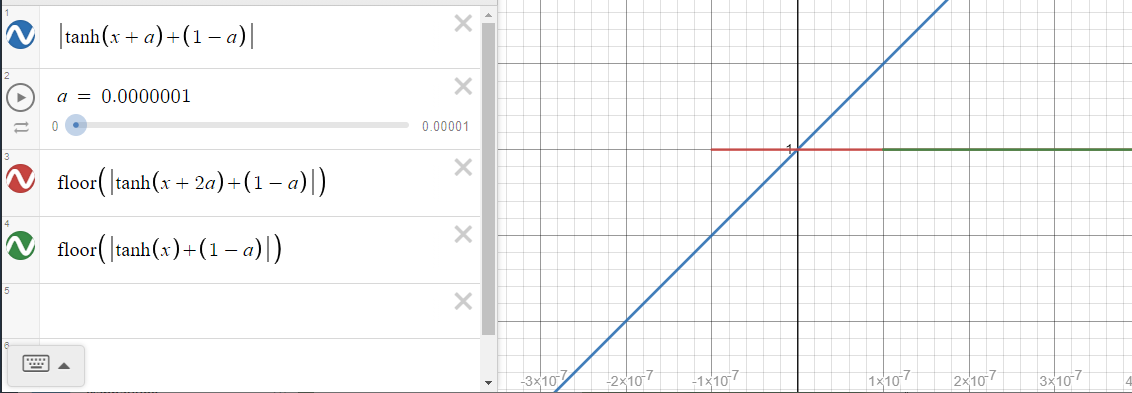


Рисунок 8.

Формула 3

Формула 4

Таким образом мы создали две функции, Промежутки X которых при Y = 1 равны [0; inf) и (0; inf) соответственно. Также задана точность вычислений, которую можно легко повысить или понизить.

Разберем формулы на примере сравнения двух чисел A и B.

Обратная функция

Часто в математичке встречаются функции обратные другим функциям. В данной разработке мы будем иметь дело только с функциями возвращающими 0 или 1. Обратная функция возвращает 0 там, где функция равна 1, и 1 там где функция равна 0. Аналог такой операции в булевой алгебре называется отрицание и обозначается отрицаемое A так .

Что бы получить отрицаемую функцию достаточно от исходной вычесть 1, тогда все единицы станут нулями, а нули минус единицами. Что превратить минус единицы в положительные единицы нужно взять модуль от функции.

Формула 5

Данный способ применим только к функциям предикатам.

1. Больше

Для данного сравнения уже есть готовая формула, обосную её применение. Из математики мы знаем, что, если из большего вычесть меньшее получится положительный ответ. Так как мы имеем функцию, которая при положительном числе дает 1, а при отрицательном 0, можно поместить разность вместо x. Осталось определиться какую из двух формула применять, так как условие у нас исключает равенство, то используем формулу, которая не включает ноль. А именно зеленую.

Формула 6

1. Больше равно

Данное сравнение похоже на предыдущее, за исключением равенства. Требуется что бы при равенстве чисел ответ функции был 1. Если вычесть равные числа друг из друга, то получится 0. Следовательно нужно использовать функцию, которая возвращает включает 0. То есть красную.

Формула 7

1. Меньше

Сравнение на меньше является обратным от больше равно. По описаной выше технологии достаточно применить отрицание функции.

Формула 8

1. Меньше равно

Сравнение на меньше равно является обратным от больше. Применим отрицание к функции больше.

Формула 9

1. Не равно

При равенстве чисел их разность равна нулю, соответственно нужна функция, которая будет возвращать 1 если разность не 0, и 0 если разность равна 0. Если взять модуль всего выражения, то произойдет отражение по оси X. А если взять модуль от аргумента X, то отражение будет относительно оси Y. Сделав отражение по оси Y от зеленой функции построенной ранее, то на всей прямой за исключением промежутка длинной в две точности измерения с центром в точке 0, функция будет равна 1. Только в точке 0 функция вернет 0. Также, так как левая часть относительно нуля не угрожает стать отрицательной, модуль всей функции можно убрать, оставив модуль только аргумента.

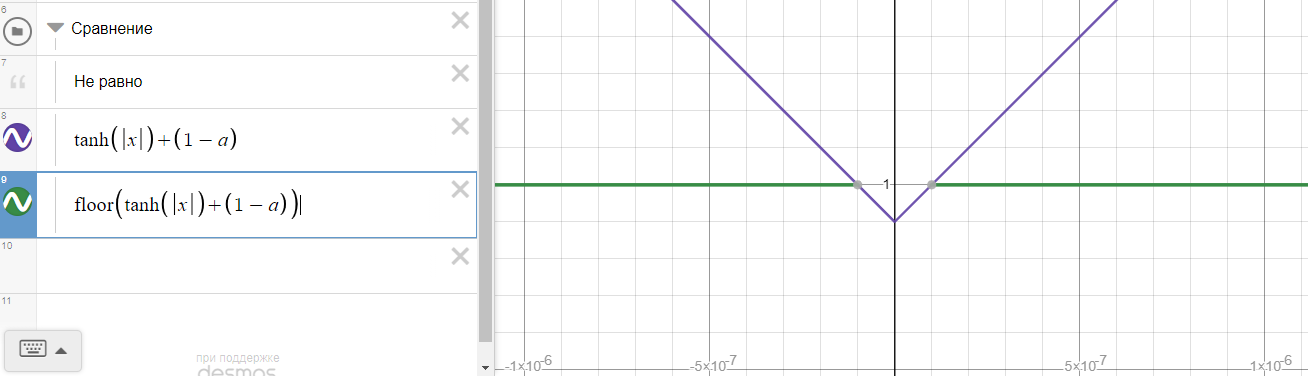


Рисунок 9.

Формула 10

1. Равно

Значение равенства можно получить как обратное неравенству. График функции полностью показать не получится так как используются функции, нарушающие целостность. Фиолетовый график пересекает ноль в **промежутке точности** (промежуток, образованный числом точности измерений +- от нуля). Значения ниже нуля округляются в меньшую сторону и становятся -1. Затем от данного числа получен модуль, который равен 1. Таким образом вся функция за исключением 0 возвращает 1, а в точке 0 она равна 1.

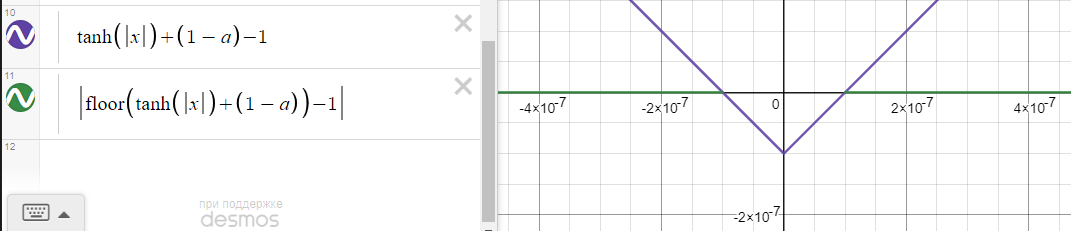


Рисунок 10.

Формула 11

1. Комбинирование условий

Так как все условные функции, которые мы создали являются предикатами, для них актуальна и Булева алгебра.

Затем

If.

If/else

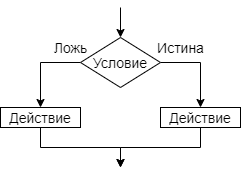


Рисунок 11.

**Вложенность**

**Тернарный оператор**

**Вложенность**

**Switch**

**Циклы**

1. **Switch case:**
   1. **Case.**
   2. **Default.**
2. **Вложенность.**
   1. **If**
   2. **Циклы**
3. **Циклы (массивы)**
   1. **For**
   2. **While do**
   3. **Do while**
   4. **Foreach**
   5. **GOTO: case**
   6. **Вложенность**
      1. **If**
      2. **Switch**
4. **\*try catch**
5. **\*exception**
6. **Функции**
   1. **параметр по значению**
   2. **параметр по ссылке**
   3. **вложенность**
7. **\*Классы/структуры**
   1. **Конструктор**
   2. **Деструктор**
   3. **Наследование**
   4. **Полиморфизм**
   5. **Инкапсуляция**
8. \*(утечка памяти)
10. Внешний вид

Разделы меню включают раздел с добавлением конструкций, раздел настроек, раздел добавления данных, а также раздел с завершением работы.

1. Применение формул на примере системы Moodle
2. Проектирование программы
3. Тестирование программы

## Заключение

## Приложение 1

## Тезариус

## Список литературы:

