ОДЕCCКИЙ НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕCКИЙ УНИВЕРCИТЕТ

Инcтитут Компьютерных cиcтем

Кафедра информационных cиcтем

Реферат

по диcциплине «Чиcленные методы»

«Оcновные модели предоcтавления уcлуг облачных вычиcлений»

Выполнил: cт.гр.АИ-174

Бордан И.С.

Руководитель: Рудниченко Н.Д.

Одеccа - 2019

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ …………………………………………………………………………….. 1

ГЛАВА 1 Облачные вычисления …………………………………………….……….. 2

1.1 Основные понятия ……………………………………………………………….. 4

1.2 Основные модели предоставления услуг облачных вычислений …………….. 4

Выводы по главе 1 …………………………………………………………………… 9

ГЛАВА 2 CРАВНИТЕЛЬНЫЙ АНАЛИЗ КРУПНЫХ ПЛАТФОРМ

ПРОИЗВОДИТЕЛЕЙ …………………………………………………………………. 10

2.1 Сравнение платформ Amazon, Google и Microsoft …………………………… 10

2.2 Другие платформы крупных производителей ………………………………… 13

Выводы по главе 2 ………………………………………………………………….. 18

[ГЛАВА 3 ПРАКТИЧЕСКАЯ ЧАСТЬ ……………………………………………….. 18](#_Toc8760644)

[3.1Сведения о методе хорд …………………………………………………………. 21](#_Toc8760645)

[Выводы по главе 3 …………………………………………………………………... 22](#_Toc8760647)

[ЗАКЛЮЧЕНИЕ ……………………………………………………………………….. 22](#_Toc8760648)

Список использованой литературы …………………………………………………... 23

ДОПОЛНЕНИЕ А ……………………………………………………………………... 25

**ВВЕДЕНИЕ**

В наше время очень aктуaльны вопроcы: Что же такое эти облачные вычиcления? Что такое облачные cервиcы? Что такое облачный cайт? Что означает облако, облачные хранилища файлов? Как иcпользуютcя облачные вычиcления в образовании? SaaS cервиc для B2C и B2B продаж? SaaS ERP и SaaS CRM cиcтемы и решения? Какие модели облачных вычиcлений иcпользуютcя для разработки веб-приложений? cовременные дaтa-центры, на оcнове которых cоздаютcя облачные уcлуги: SaaS, PaaS и IaaS?

cначала раccмотрим, что такое "облако" (cloud) и "облачные вычиcления" (cloud computing). "Облако" - это инновационная модель (концепция) организации IT-инфраcтруктуры, которая cоcтоит из раcпределенных и разделяемых конфигурируемых аппаратных и cетевых реcурcов, а также программного обеcпечения, развернyтых на удаленных (облачных) дата центрах поcтавщиков (провайдеров). То еcть облако - это нoвый подход организации IT-инфраcтруктуры.

Cloud computing- это модель предоcтавления диcтанционного доcтупа к разделяемым вычиcлительным реcурcам, которые физичеcки раcпределены на многих удалённых уcтройcтвах, образующих так называемое облако (cloud). Облачные вычиcления - это модель предоcтавление потребителю маcштабируемых вычиcлительных реcурcов в виде cервиcа через Интернет. Облачные технологии – это модель предcтавления потребителю IT как cервиcа через Интернет.

В данном реферате мы проанализируем основополагающее определение облачных вычислений, их достоинства и недостатки, типы облаков, а также, развитие облачных вычислений в мире.

**ГЛАВА 1 ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ**

**1.1 Основные понятия**

Идея облачных вычислений появилась еще в 1960 году, когда Джон Маккарти высказал предположение, что когда-нибудь компьютерные вычисления будут производиться с помощью «общенародных утилит». Считается, что идеология облачных вычислений получила популярность с 2007 года благодаря быстрому развитию каналов связи и стремительно растущим потребностям пользователей. Под облачными вычислениями (от англ.cloud computing, также используется термин «облачная (рассеянная) обработка данных») обычно понимается предоставление пользователю компьютерных ресурсов и мощностей в виде интернет-сервиса. Таким образом, вычислительные ресурсы предоставляются пользователю в «чистом» виде, и пользователь может не знать, какие компьютеры обрабатывают его запросы, под управлением какой операционной системы это происходит и т.д. Часто облака сравнивают с мэйнфреймами (mainframe), находя между ними много общего. Принципиальное отличие облака от мэйнфреймов в том, что его вычислительная мощность теоретически не ограничена. Второе принципиальное отличие в том, что, попросту говоря, терминалы для мэйнфреймов служили только для интерактивного взаимодействия пользователя с запущенной на обработку задачей.

В облaке же терминал сам является мощным вычислительным устройством, способным не только накапливать промежуточную информацию, но и непосредственно управлять глобальной системой вычислительных ресурсов.

Срeди ранее возникших (в 1990-х гг.) технологий обработки данных некоторое распространение получили так называемые grid-вычисления. Это напрaвление первоначально рассматривалось как возможность использования свoбoдных ресурсов процессоров и развития системы добровольной аренды вычислительных мощностей. Ряд проектов (GIMPS, distributed.net, SETI@home) доказали, что такая модель вычислений достаточно эффективна. Сегодня эта технология применяется для решения научных, математических задач, где требуются значительные вычислительные ресурсы. Известно, что gridвычисления также применяются для коммерческих целей. Например, с их помощью выполняются некоторые трудоемкие задачи, связанные с экономическим прогнозированием, анализом сейсмических данных, разработкой и изучением свойств вакцин и новых лекарств. Действительно, grid-вычисления и облака имеют много схожих черт в архитектуре и применяемых принципах. Тем не менее, модель облачных вычислений считается сегодня более перспективной благодаря значительно более гибкой платформе для работы с удаленными вычислительными ресурсами.

В настоящее время крупные вычислительные облака состоят из тысяч серверов, размещенных в центрах обработки данных(ЦОД). Они обеспечивают ресурсами десятки тысяч приложений, которые одновременно используют миллионы пользователей. Облачные технологии являются удобным инструментом для предприятий, которым слишком дорого содержать собственные ERP, CRM или другие серверы, требующие приобретения и настройки дополнительного оборудования.

ERP (Enterprise Resource Planning– планирование ресурсов предприятия) – организационная стратегия интеграции производства и операций, управления трудовыми ресурсами, финансового менеджмента и управления активами, ориентированная на непрерывную балансировку и оптимизацию ресурсов предприятия посредством специализированного интегрированного пакета прикладного программного обеспечения, обеспечивающего общую модель данных и процессов для всех сфер деятельности предприятия.

CRM (Customer Relationship Management) – система управления взаимоотношениями с клиентами, то есть прикладное программное обеспечение, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путем сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процедур и последующего анализа результатов.

Среди частных пользователей широкое распространение постепенно получают благодаря своему удобству такие облачные услуги, как, например, предоставляемые компанией Google («Документы», «Календарь» и др.).

Причины возрастающей популярности облачных технологий понятны:

возможности их применения очень разнообразны и позволяют экономить как на обслуживании и персонале, так и на инфраструктуре. Аппаратное обеспечение может быть сильно упрощено при обработке данных и хранении информации в удаленных центрах данных. Все эти проблемы почти полностью перекладываются на провайдера услуг.

К тому же такой подход позволяет стандартизировать ПО, даже если на компьютерах предприятия установлены разные операционный системы (Windows, Linux, MacOS и т.п.). Облачные технологии облегчают обеспечение доступа к данным компании как для клиентов, так и для собственных сотрудников, находящихся вне офиса, но имеющих возможность подключиться через Интернет.

Понятно, что использование облачных вычислений намного удобнее. Самым главным недостатком, который можно сразу заметить, является полная зависимость от поставщика этих услуг. Фактически предприятие (пользователь) оказывается заложником провайдера сервисов и провайдера доступа в сеть Интернет. Хотя надежность поставщиков облачных вычислений возрастает, для обеспечения надежности и безопасности данных необходимо приложить немало усилий, например, иметь дублирующие каналы связи, дублирующие мощности для возможности переключения на них и, конечно же, подумать о доступности информации и безопасности. Кроме этого, облачные вычисления совершенно не подходят для предприятий, имеющих отношение к государственной и военной тайне. Ни одна комиссия не выдаст сертификат на такую систему при работе с информацией, не подлежащей разглашению.

Современные облачные технологии не только используются в готовом сетевом и серверном оборудовании, но и постепенно проникают на рынок встраиваемых систем (embedded cloud) и становятся причиной масштабной реструктуризации рынка. Внедрение встраиваемых систем приводит к размещению компьютерных процессоров в таких изделиях, как счетчики учета расхода ресурсов, интеллектуальные датчики, М2М-модули, автомобили, бытовая техника и т.д. Это позволяет управлять работой устройств, сбором данных и обеспечением интерактивных возможностей посредством подключения к компьютерной сети. Идею подключения всевозможных устройств к глобальной сети называют Интернетом вещей(Internet of Things – IoT). По мнению Кевина Далласа, генерального менеджера Microsoft Windows Embedded , идея Интернета вещей существует уже много лет, однако для ее реализации не хватало одного звена, чтобы построить такую сеть, – облака.

Так как количество встраиваемых компьютеров увеличивается благодаря снижению цен на процессоры и повсеместному распространению Интернета, растут также и объемы передаваемых данных с последующей их обработкой (часто в режиме реального времени). Поэтому можно предположить, что в ближайшие годы роль Интернета вещей и облачных вычислений будет увеличиваться.

**1.2 Основные модели предоставления услуг облачных вычислений**

****

Рис.1.2 – Опциональность облачных хранилищ

По модели развертывания облака разделяют на частные, общедоступные

(публичные) и гибридные.

Частные облака –это внутренние облачные инфраструктура и службы предприятия. Эти облака находятся в пределах корпоративной сети.

Организация может управлять частным облаком самостоятельно или поручить эту задачу внешнему подрядчику. Инфраструктура может размещаться либо в помещениях заказчика, либо у внешнего оператора, либо частично у заказчика и частично у оператора. Идеальный вариант частного облака – облако, развернутое на территории организации, обслуживаемое и контролируемое ее сотрудниками.

Частные облака обладают теми же преимуществами, что и общедоступные, но с одной важной особенностью: предприятие само занимается установкой и поддержкой облака. Сложность и стоимость создания внутреннего облака могут быть очень высоки, а расходы на его эксплуатацию могут превышать стоимость использования общедоступных облаков.

Следует отметить, что у частных облаков есть преимущества перед общедоступными: более детальный контроль над различными ресурсами облака обеспечивает компании любые доступные варианты конфигурации. Кроме того, частные облака идеальны, когда нужно выполнять работы, которые нельзя доверить общедоступному облаку из соображений безопасности.

Общедоступные (публичные) облака –это облачные услуги, предоставляемые поставщиком. Они находятся за пределами корпоративной сети. Пользователи данных облаков не имеют возможности управлять данным облаком или обслуживать его, вся ответственность возложена на владельца этого облака. Поставщик облачных услуг принимает на себя обязанности по установке, управлению, предоставлению и обслуживанию программного обеспечения, инфраструктуры приложений или физической инфраструктуры. Клиенты платят только за ресурсы, которые они используют.

Абонентом предлагаемых сервисов может стать любая компания и индивидуальный пользователь. Они предлагают легкий и доступный по цене способ развертывания веб-сайтов или бизнес-систем с большими возможностями масштабирования, которые в других решениях были бы недоступны. Примеры: онлайн-сервисы Amazon EC2 и Amazon Simple Storage Service (S3), Google Apps/Docs, Salesforce.com, Microsoft Office Web.

Вместе с тем услуги публичных облаков в основном предоставляются в виде стандартных конфигураций, то есть исходя из условий наиболее распространенных случаев использования. Это значит, что у пользователя остается меньше возможностей по выбору конфигурации по сравнению с системами, в которых ресурсами управляет сам потребитель. Следует также иметь в виду, что, поскольку потребители слабо контролируют инфраструктуру, процессы, требующие строгих мер безопасности и соответствия нормативным требованиям, не всегда подходят для реализации в общедоступном облаке.

Гибридные облака представляют собой сочетание общедоступных и частных облаков. Обычно они создаются предприятием, а обязанности по управлению ими распределяются между предприятием и поставщиком общедоступного облака. Гибридное облако предоставляет услуги, часть которых относится к общедоступным, а часть – к частным. Обычно такой тип облаков используется, когда организация имеет сезонные периоды активности. Другими словами, как только внутренняя ИТ-инфраструктура не справляется с текущими задачами, часть мощностей перебрасывается на публичное облако (например, большие объемы статистической информации, которые в необработанном виде не представляют ценности для предприятия), а также для предоставления доступа пользователям к ресурсам предприятия (к частному облаку) через публичное облако. Хорошо продуманное гибридное облако может обслуживать как требующие безопасности критически важные процессы, такие как получение платежей от клиентов, так и более второстепенные.

Основным недостатком этого типа облака является сложность эффективного создания подобных решений и управления ими. Необходимо получать услуги из разных источников и организовать их так, как если бы это был единый источник. Взаимодействие между частным и общедоступным компонентами может еще больше усложнить решение. Поскольку это относительно новая архитектурная концепция в сфере облачных вычислений, для этой модели появляются все новые и новые практические рекомендации и инструменты, и ее широкое распространение может затянуться до тех пор, пока она не будет лучше изучена.

По мнению Тома Биттмана, вице-президента и ведущего аналитика американской исследовательской и консалтинговой компании “Gartner” , среди вышеперечисленных трех моделей развертывания облаков наиболее актуальной для бизнеса в данный момент являются частные облака. Биттман выделил пять основных моментов, которые помогают получить более точное представление об устройстве частного облака.

Облако – это не только виртуализация. Хотя виртуализация серверов и инфраструктуры составляет важный фундамент частных облачных вычислений, сами по себе виртуализация и управление виртуализированной средой еще не являются частным облаком.

Виртуализация позволяет лучше структурировать, объединять в пул и динамически предоставлять ресурсы инфраструктуры: серверы, десктопы, емкости для хранения, сетевое оборудование, связующее ПО и т.д. Но, чтобы среда технически могла считаться облачной, нужны еще и другие составляющие, такие как виртуальные машины, операционные системы или контейнеры связующего ПО, высокоустойчивые операционные системы, ПО grid-вычислений, ПО для абстрагирования ресурсов хранения, средства масштабирования и кластеризации.

Термин «частное облако» в отличие от общедоступного или гибридного относится к ресурсам, используемым единственной организацией, либо означает, что облачные ресурсы организации полностью изолированы в облаке от остальных.

Облако – необязательно источник экономии. Одно из главных заблуждений состоит в том, что облако будет экономить деньги. Экономия возможна, но не является обязательным атрибутом.

Частное облако позволяет более эффективно перераспределять ресурсы, чтобы удовлетворить корпоративные требования, и способно уменьшить капитальные затраты на оборудование. Но частное облако требует инвестиций в автоматизацию, и одна лишь экономия может не окупать всей стоимости. Так что, снижение затрат не является главным преимуществом этой модели. С этой точки зрения, главным стимулом к внедрению облачной модели должна быть не экономия, а скорость выхода на рынок, возможность быстрой адаптации и динамического масштабирования в соответствии со спросом, которые позволяют повысить скорость внедрения новых сервисов.

Частное облако не всегда внедрено у заказчика. Частное облако означает конфиденциальность, а не конкретное местоположение, владение ресурсами или самостоятельное управление. Многие поставщики предлагают нелокальные частные облака, то есть выделяют ресурсы единственному заказчику, исключая совместное использование одного пула несколькими клиентами. «Облако называется частным по его приватности, а не по тому, где оно развернуто, кто им владеет и несет ответственность за управление», – подчеркивает Биттман. Некоторые, например, могут свои ЦОД размещать у хостинг-провайдеров или объединять в пул ресурсы разных заказчиков, но изолировать их друг от друга с помощью виртуальной частной сети (Virtual Private Network –VPN) и других подобных технологий.

Частное облако (как и публичное облако) – это не только инфраструктурные сервисы.Серверная виртуализация – крупная тенденция и поэтому мощный двигатель частных облачных вычислений. Но частное облако не сводится только к инфраструктуре как услуге (IaaS). Например, для разработки и тестирования нового ПО высокоуровневая платформа как услуга (PaaS) имеет больше смысла, чем просто предоставление виртуальных машин.

Сегодня самый быстро растущий сегмент облачных вычислений – это IaaS. Она предоставляет самые низкоуровневые ресурсы ЦОД в простой для использования форме, но не меняет фундаментально принципы работы. Чтобы создать новые приложения, изначально предназначенные для облака и предоставляющие совершенно новые услуги, которые могут очень отличаться от того, что давали прежние приложения, разработчикам удобнее использовать PaaS.

Частное облако может перестать быть частным. С одной стороны, частное облако предоставляет преимущества облака: быстроту перестройки, масштабируемость и эффективность, избавляет от некоторых угроз безопасности, потенциальных и реальных, которые характерны для общедоступных облаков. С другой стороны, со временем уровень обслуживания, безопасность и контроль соблюдения требований в общедоступных облачных сервисах безусловно будут повышаться. Поэтому некоторые частные облака, возможно, целиком перейдут в категорию общедоступных. Большинство же сервисов частного облака, скорее всего, будут эволюционировать в гибридные облачные сервисы, расширяя доступные возможности за счет использования общедоступных облачных услуг и других сторонних ресурсов.

**Выводы по главе 1**

Облачные вычисления — это не технология, а компьютерная модель. Она предполагает, что все серверы, сети, приложения и другие элементы, связанные с центрами обработки данных, доступны ИТ-службе и конечным пользователям через Интернет. Таким образом, можно покупать только те компьютерные сервисы, которые действительно необходимы клиенту, и в том объеме, который ему действительно нужен. Модель облака отличается от традиционного аутсорсинга тем, что клиенты не передают в управление внешней организации свои собственные ИТ-ресурсы. Вместо этого они подключаются к «облаку» инфраструктурных сервисов, сервисов платформ (операционных систем) или программных сервисов (приложений SaaS). В данном случае облако аналогично внутреннему ЦОД или компьютеру, выполняющему те же функции.

**ГЛАВА 2 CРАВНИТЕЛЬНЫЙ АНАЛИЗ КРУПНЫХ ПЛАТФОРМ** **ПРОИЗВОДИТЕЛЕЙ**

**2.1 Сравнение платформ Google, Amazon, Microsoft**

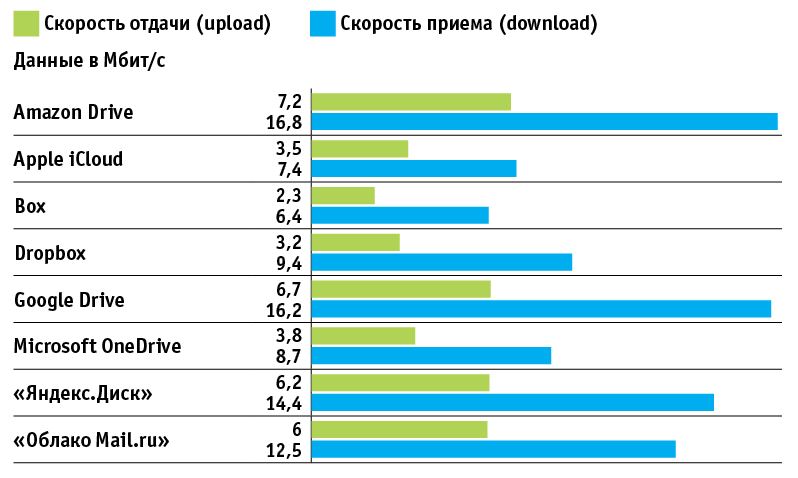
****

Рис.2.1 – гистограмма популярных облачных сервисов

В настоящий момент основными поставщиками облачной инфраструктуры считаются Amazon, Google и Microsoft. У каждой из компаний имеется целая линейка предоставляемых услуг. В данных материалах описаны только некоторые из них, наиболее популярные. Также не обсуждается вопрос, к какой именно модели относится та или иная услуга и какие вендоры предоставляют только публичные облака, а какие могут участвовать в создании частных облаков.

Google Drive–облачное хранилище данных, принадлежащее компании Google, позволяющее пользователям хранить свои данные на серверах в облаке и делиться ими с другими пользователями в Интернете. Google Drive отличается лаконичным интерфейсом и предлагает установить удобные программные клиенты для смартфонов и планшетов на базе операционной системы Android, ПК и ноутбуков под управлением операционной системы Windows или MacOS, мобильных устройств iPhone и iPad. В будущем ожидается более тесная интеграция хранилища с операционной системой Chrome OS и поддержка Linux. Каждый пользователь Google Drive получает до 15 Гбайт свободного пространства на все сервисы Google (в том числе Gmail и Photos). При этом он сам может решить, сколько места выделить под почту и какой объем оставить под важные файлы. Работать с файлами в Google Drive можно прямо в браузере. Google Drive можно превратить в отдельную папку в документах смартфона, планшета или ПК, и ее содержимое будет синхронизироваться автоматически.

Google Docs– бесплатный онлайн-офис, включающий в себя текстовый, табличный процессоры и сервис для создания презентаций, а также интернетсервис облачного хранения файлов с функциями файлообмена. Позволяет создавать и редактировать стандартные документы, таблицы и презентации, а также поддерживает функции совместной работы над ними.

Google App Engine – сервис хостинга сайтов и web-приложений на серверах Google. Бесплатно предоставляется до 1 Гб дискового пространства, 10 Гб входящего трафика в день, 10 Гб исходящего трафика в день, 200 миллионов гигациклов CPU в день и 2 000 операций отправления электронной почты в день. Приложения, разворачиваемые на базе App Engine, должны быть написаны на Python, Java либо Go. Предлагается набор API для сервисов хранилища datastore API (BigTable) аккаунтов Google, набор API для загрузки данных по URL, электронной почты и т.д.

Платформа Google конкурирует с аналогичными сервисами от Amazon, которые предоставляют возможность размещать файлы и веб-приложения, используя свою инфраструктуру. В отличие от многих обычных размещений приложений на виртуальных машинах, таких как Amazon EC2, платформа App Engine тесно интегрирована с приложениями и накладывает на разработчиков некоторые ограничения.

GoogleCloudStorage– сервис хостинга файлов, основанный на IaaS. Все файлы, которые записываются или перезаписываются на серверы, автоматически шифруются по алгоритму AES-128. Является конкурентом продукта Amazon S3.

Amazon Simple Storage Service(Amazon S3) – онлайновая веб-служба, предлагаемая Amazon Web Services, предоставляющая возможность для хранения и получения любого объема данных, в любое время из любой точки сети, так называемый файловый хостинг. В марте 2012 года компания Nasuni провела опыт, в течение которого поочередно передавала массивный объем данных (12 Тб) из одного облачного сервиса в другой . В эксперименте участвовали наиболее рейтинговые облака: Amazon S3, Windows Azure и Rackspace. К удивлению исследователей, скорость передачи данных сильно отличалась в зависимости от того, какое облако принимало данные. Самый лучший показатель скорости записи данных оказался у Amazon S3, передача данных из двух других сервисов занимала всего 4–5 часов, в то время как передача данных в Rackspace заняла чуть меньше недели, а в Windows Azure – 40 часов.

Amazon Elastic Compute Cloud(Amazon EC2) – веб-сервис, предоставляющий вычислительные мощности в облаке. Он дает пользователям полный контроль над вычислительными ресурсами, а также доступную среду для работы. Amazon EC2 позволяет пользователям создать Amazon Machine Image (AMI), который будет содержать их приложения, библиотеки, данные и связанные с ними конфигурационные параметры, или использовать заранее настроенные шаблоны образов для работы Amazon S3. Amazon EC2 предоставляет инструменты для хранения AMI. Amazon S3 предоставляет безопасное, надежное и быстрое хранилище для хранения образов.

Microsoft SkyDrive –интернет-сервис хранения файлов с функциями файлообмена, созданный и управляемый компанией Microsoft. Сервис SkyDrive позволяет хранить до 7 ГБ информации (или 25 ГБ для пользователей, имеющих право на бесплатное обновление) в виде стандартных папок. Пользователи могут просматривать, загружать, создавать, редактировать и обмениваться документами Microsoft Office (Word, Excel, PowerPoint и OneNote) непосредственно в веб-браузере. Присутствует удаленный доступ к компьютеру, работающему под управлением Windows.

WindowsAzure– платформа облачных сервисов, разработанная Microsoft. Реализует модели PaaS и IaaS. Платформа предоставляет возможность разработки и выполнения приложений и хранения данных на серверах, расположенных в распределенных центрах данных.

WindowsAzureCompute– компонент, реализующий вычисления на платформе Windows Azure, предоставляет среду выполнения на основе ролевой модели.

WindowsAzureStorage– компонент хранилища, предоставляющий масштабируемое хранилище. Не имеет возможности использовать реляционную модель и является альтернативой (либо дополняющим решением) SQL Databases (SQL Azure) – масштабируемой «облачной» версией SQL Server. WindowsAzureFabric– по своему назначению является контролером и ядром платформы, выполняя функции мониторинга в реальном времени, обеспечения отказоустойчивости, выделения мощностей, развертывания серверов, виртуальных машин и приложений, балансировки нагрузки и управления оборудованием.

Платформа Windows Azure имеет API, построенное на REST, HTTP и XML, что позволяет разработчикам использовать облачные сервисы с любой операционной системой, устройствами и платформами.

В работе рассмотрены несколько типовых задач и сравниваются их возможные решения на каждой из платформ –AmazonEC2,Google App Engine,WindowsAzure.

Задача 1.Необходимо отправить в облако на обработку локально созданное приложение; причем приложение, как правило, построенное на базе технологий Java или .NET, не должно менять свое содержание в процессе исполнения.

Задача 2.Необходимо запустить web-приложение, использующее фоновое обновление данных и средства балансировки нагрузки.

Задача 3.Необходимо выполнить в автоматическом режиме без применения интерфейса с пользователем прикладную вычислительную задачу с использование методов параллельной обработки.

Задача 4.Необходимо запустить в облаке приложение, которое взаимодействует с локальной программой или обменивается данными с клиентским ПК.

Решения на платформеAmazon, как правило, базируются на том, что локальный компьютер рассматривается как сервер, входящий в состав общего ЦОД, и вносятся необходимые изменения в конфигурацию, либо необходимо создать несколько виртуальных машин в зависимости от требуемого уровня масштабирования.

На других вышеупомянутых платформах используют специальные средства динамического масштабирования, скриптовые программы, множество различных API. Часть из них весьма удобны, в других случаях необходимо прилагать достаточно много усилий для существенной или даже почти полной переработки исходной программы.



Рис.2.2 – Основные модели взаимодействия облачных сервисов

**2.2 Другие платформы крупных производителей**

**IBM SmartCloud**

Облачное решение, предлагаемое компанией IBM, а именно IBM SmartCloud, реализует все три модели (IaaS, SaaS, PaaS) в рамках не только публичного, но частного и гибридного облаков. В его состав входит облачный сервис, ранее называемый IBM Lotus Live, предоставляющий бизнес-приложения по модели SaaS. Содержит полный набор интерактивных сервисов, которые предоставляют масштабируемые решения для организации защищенной системы электронной почты, проведения web-конференций и коллективной работы. Сервисы свободны от рекламы и не собирают информацию о клиенте, а также не являются потребительскими приложениями, нацеленными на бизнесдеятельность. С пользователей взимается ежемесячная плата. Элементы управления системой защиты, развернутые для Lotus Live, обеспечивают приватность и управляемый доступ к важной информации при выполнении бизнес-операций. Все клиентские взаимодействия кодируются устойчивыми алгоритмами шифрования и осуществляются по протоколу SSL для HTTP и через RC2 в протоколе системы мгновенного обмена сообщениями Lotus Sametime. Резервные копии системы шифруются.

RackspaceCloud

Платформа предлагает набор продуктов для автоматизации хостинга и облачных вычислений, реализуется модель PaaS. Объединяет в себе Cloud Files, Cloud Servers, Cloud Sites. Благодаря серверной виртуализации пользователи получают возможность развертывать сотни облачных серверов одновременно и создавать архитектуру, обеспечивающую высокую доступность. Является конкурентом Amazon Web Services.

**Oracle Exalogic Elastic Cloud**

Компания ORACLE работает над концепцией ПО как услуги на протяжении последних 10 лет. На сегодняшний день компания признана одним из ведущих поставщиков ПО, построенного по технологии облака, и работает с более чем 5,5 миллиона пользователей. Компания ORACLE предлагает выбор между моделями развертывания ПО как с использованием ее центров данных и основанными на подписке, так и с моделями развертывания ПО на территории компании заказчика.

Для большинства центров данных, переходящих на технологию частных облачных вычислений, ORACLE в качестве первого шага предлагает консолидацию вычислительных ресурсов и переход на разделяемые и масштабируемые платформы и инфраструктуру.

Серверное оборудование, выделенное под индивидуальные задачи middleware, БД и другие приложения, рассчитано на пиковую нагрузку и обладает зарезервированной мощностью, не используемой постоянно. Каждый сервер может включать практически несовместимые программные компоненты от разных поставщиков ПО, что в ряде случаев увеличивает расходы на поддержку и повышает управленческие затраты.

Переход на объединенную архитектуру облачных вычислений ORACLE со стандартизированными разделяемыми приложениями по требованию существенно снижает издержки. Консолидация вычислительных ресурсов может быть выполнена как на уровне IaaS, с использованием технологий виртуализации, так и на уровне PaaS, с помощью стандартизации и объединения на основе БД и/или middleware архитектуры.

Консолидация на уровне PaaS представляет большую ценность, потому что уменьшает разнородность ПО, объединяя его на основе стандартизированных программных интерфейсов. Консолидация на уровне IaaS может также обеспечить более высокую эффективность разделения аппаратных средств, но ничего не дает для уменьшения сложности поддержки ПО. Наиболее популярной является консолидация ПО на уровне БД ORACLE, частично объединяющей преимущества PaaS и IaaS.

Для облачных вычислений компания ORACLE предлагает две ключевые технологии: виртуализация и кластеризация серверов. Виртуализация позволяет легко развертывать новые приложения по требованию и является хорошим способом разделения аппаратных средств между задачами. Объединение в кластеры важно для повышения диверсификации ресурсов между приложениями, тем самым повышая их доступность и отказоустойчивость.

Платформа ORACLE PaaS

ORACLE PaaS является масштабируемой платформой, общей для всех облачных приложений как частных, так и общественных центров данных. Платформа ORALCE PaaS основана на БД ORACLE и приложениях Oracle Middleware. Она дает возможность различным организациям объединять существующее ПО с использованием общей архитектуры, позволяющей создавать новые приложения, использующие существующие возможности ПО для расширения спектра услуг, предоставляемых по требованию.

Платформа ORACLE PaaS предоставляет услуги БД по требованию, основанные на БД ORALCE и аппаратных комплексах Oracle Exadata, а также услуги ПО Middleware по требованию на основе Oracle WebLogic и Oracle Exalogic. Oracle Exadata – это специализированная машина БД, а Oracle Exalogic

является машиной, оптимизированной для выполнения приложений Middleware, написанных на языке JAVA. Обе машины масштабируемы и отказоустойчивы. Они спроектированы и сконфигурированы для совместной работы.

Для разработки новых приложений программисты могут использовать знакомые среды проектирования, такие как JDeveloper, NetBeans и Eclipse, а также сетевые инструменты WebCenter Page Composer, BI Composer и BPM Composer. Для взаимной интеграции новых и разработанных ранее приложений в частных и общественных облаках компания ORACLE предлагает

Oracle SOA Suite и Oracle BPM Suite, а также Oracle Data Integration и Oracle

GoldenGate. Идентификация пользователей и распределение прав осуществляются с помощью Oracle Identity and Access Management. За взаимодействие пользователей с облаком отвечает программный комплекс Oracle WebCenter, который позволяет пользователям осуществлять совместную работу.

Инфраструктура ORACLE IaaS

Компания ORACLE предлагает набор элементов технологии «Инфраструктура по требованию» (IaaS), в том числе вычислительные серверы, услуги хранения и передачи информации, виртуализацию ПО, операционные системы и системы управления ПО.

Инфраструктура ORACLE IaaS включает серверы, основанные на технологии SPARC и x86, установленные в стойках и лезвиях (blads); технологии хранения на FLASH, дисковых массивах и лентах; варианты виртуализации Oracle VM для x86, Oracle VM для SPARC и Oracle Solaris Containers; операционные системы Oracle Solaris, Oracle Linux, и Oracle Enterprise Manager.

Гибкая инфраструктура ORACLE IaaS поддерживает объединение гетерогенных ресурсов, масштабируемость, быстрое развертывание и высокую доступность прикладного ПО, позволяя эффективно управлять общественными и частными IaaS.

Аппаратные комплексы Oracle Exadata

Аппаратно-программный комплекс, серийно выпускаемый корпорацией Oracle, позиционируется как кластер серверов приложений для организации частных облачных и «эластичных» вычислений. Поставляется как предварительно собранный телекоммуникационный шкаф из 42 юнит, наполненный серверным и сетевым оборудованием. Аппаратная часть комплекса состоит из одноюнитовых серверов на базе двух процессоров Intel Xeon с двумя твердотельными накопителями в каждом для операционной системы и свопинга, общей для всех серверов системы хранения данных, коммутаторов InfiniBand и Ethernet. В последних выпусках X3-2

устанавливаются восьмиядерные Sandy Bridge частотой 2,9 ГГц, в каждом узле установлено 256 Гбайт оперативной памяти. Заказчикам комплекса предоставляется выбор из двух 64-разрядных операционных систем, возможных к предустановке на узлы кластера: Oracle Linux или Solaris (операционная система, разработанная компанией Sun Microsystems, которая сейчас принадлежит корпорация Oracle), а с середины 2012 года доступна возможность установки на узлы гипервизора Oracle VM.

Аппаратные комплексы Oracle Exadata выпускаются в виде стандартных стоек для размещения в ЦОД. Они состоят из серверов, использующих процессоры Intel XEON, основанные на архитектуре x86 и x64. В Oracle Exadata используются серверы двух типов: хранения данных и обработки. В качестве моста между серверами используются коммутаторы InfiniBand и Ethernet. Как и любая модульная система, Oracle Exadata обладает свойством масштабируемости. Аппаратные комплексы Exadata выпускаются в нескольких вариантах, заполненных серверами в зависимости от предполагаемой максимальной нагрузки.

Каждый аппаратный комплекс Oracle Exadata содержит предварительно установленное ПО Oracle Database с опцией Real Application Cluster, позволяющей нескольким физическим серверам работать с единым хранилищем как единая БД без программных модификаций прикладного ПО.

Архитектура Oracle Exadata основывается на принципах симметричного доступа со всех серверов обработки ко всем узлам хранения (симметричного параллелизма). Такой подход является компромиссом, так как ориентирован на OLTP- и OLAP-обработку данных одновременно. Поэтому решение компании ORACLE не является лидером по быстродействию в узкоспециализированных задачах, но рассчитано на широкий спектр применения.

Специалисты компании считают, что комплексы Oracle Exadata как нельзя лучше подходят под идеологию облачных вычислений, так как являются универсальными и не требуют специализированного ПО.

Оснoватель компании Salesforce.com Марк Бениофф считает, что аппаратнопрограммные комплексы принципиально не могут решить задачу масштабируемости для конечного заказчика по сравнению с инфраструктурой, предоставляемой как услуга, а также отмечает, что концепция таких комплексов означает фактический возврат к концепции мейнфрейма, утратившей актуальность еще в конце 1970-х годов. Несмотря на возможность балансировки нагрузки в рамках нескольких приложений нет возможности функционирования выше предела вычислительных возможностей комплекса, и конечный заказчик вынужден приобретать аппаратные мощности под пиковую нагрузку. Подобного рода критика характерна для концепции частных облачных вычислений в целом.

В 2010 году Oracle открыла собственное публичное облако Oracle Cloud, предоставляющее как технологическое ПО по модели PaaS, так и бизнесприложения по модели SaaS. Salesforce.com

Система управления взаимоотношениями с клиентами (CRM-система – Customer Relationship Management) предоставляется заказчикам исключительно под модели SaaS. Под наименованиемForce.comкомпания предоставляет PaaSплатформу для самостоятельной разработки приложений, а под брендом Database.com– облачную систему управления БД. В зависимости от уровня подпиcки доступны различные технические возможности. Так, в бесплатной версии Force.comподписчики могут создать не более десяти сущностей, а в неограниченной версии с ценой $75 на пользователя в месяц – до 2 000. Подписчики могут размещать разработанные приложения на платформе Force.com в специальном каталоге (AppExchange) и предлагать свои разработки другим заказчикам, в том числе на коммерческой основе.

В качестве системы управления БД платформа Force.com использует три реплицируемых кластера Oracle RAC из восьми узлов каждый. Кластеры расположены в трех удаленных друг от друга ЦОД. В одной схеме Oracle Database обрабатываются данные сразу нескольких компаний-подписчиков. Используется стабильная схема данных, предварительно подготовленная к расширению дополнительными объектами таким образом, что в одних и тех же таблицах хранятся данные различных подписчиков независимо от различия в специфических атрибутах объектов для различных подписчиков. Широко используется секционирование таблиц БД.

**Выводы по главе 2**

В наше время очень большая конкуретность между фирмaми, за право быть топ 1 в мире облачных вычислений. Это повышает качество услуг, так как, чем больше компаний, тем выше качество продукта.

**ГЛАВА 3 ПРАКТИЧЕСКАЯ ЧАСТЬ**

## 

## **3.1 Сведения о методе хорд**

Метод хорд — это итерационный численный метод приближённого вычисления корня уравнения.

**Геометрическая интерпретация метода хорд**

Для начала решения выбираются две произвольные точки на оси x.На нашем рисунке это точки x0 и x1. Затем через две точки (x0,F(x0))и (x1,F(x1)) проводится хорда. Эта хорда пересекает линию y=0 в точке x2. Эта точка считается первым приближением к корню. Далее , если |F(x2)|>eps или |(x0-x1)|>eps (eps - требуемая погрешность в вычислении корня) , тогда точки x1 и x2 выбирают в качестве новых стартовых точек , т.е. x0=x1, x1=x2 и вновь проводят хорду и т.д. Критерием окончания итерационного процесса считается достаточная близость модуля значения функции |F(x2)| к нулю , а также достаточно близкие значения двух последовательных приближений к корню x0 и x1.Чтобы найти, в какой точке хорда, проведенная через точки (x0,F(x0))и (x1,F(x1)), пересекает ось x, напишем уравнение этой хорды:

y(x)=F(x1) + (F(x1)-F(x0))\*(x-x1)/(x1-x0).

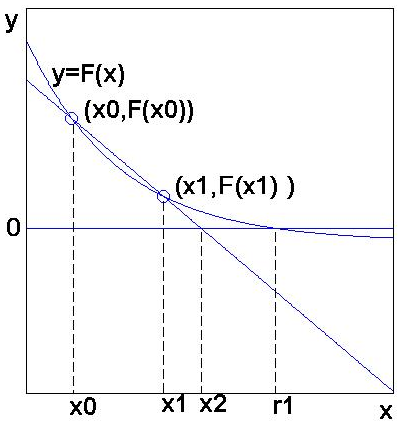


Рис.3.1.1 Геометрическая интерпретация метода хорд

Сyть мeтoда хорд состоит в разбиении отрезка **[a,b]** (при условии **f(a)f(b)<0**) на два отрезка с помощью хорды и выборе нового отрезка от точки пересечения хорды с осью абсцисс до неподвижной точки, на котором функция меняет знак и содержит решение, причём подвижная точка приближается к **ε**-окрестности решения.

Построение хорд продолжается до достижения необходимой точности решения **ε**.

Метод хорд применим для решения уравнения вида **f(x)=0** на отрезке **[a,b]**, если ни одна точка отрезка **[a,b]** не является ни стационарной, ни критической, то есть **f’(x)≠0** и **f"(x)≠0**.

Условие начальной точки для метода хорд **f(x)f"(x)<0**.

Условие неподвижной точки для метода хорд **f(x)f"(x)>0**.

Сначала находим отрезок **[a,b]** такой, что функция **f(x)** дважды непрерывно дифференцируема и меняет знак на отрезке, то есть **f(a)f(b)<0**.

Далее применяем алгоритм рeшeния.

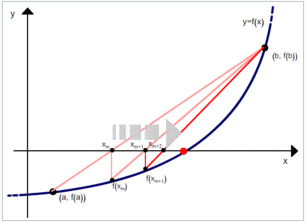


Рис.3.1.2 Визуализация итераций метода хорд(синияя кривая- функция, красные прямые- хорды)

Задание было выполнено на языке программирования Python в среде программирования PyCharm, с использованием библиотеки PyQt5. PyQt5 – это набор «привязок» графического фреймворка Qt для языка программирования Python, выполненный в виде расширения Python. Он был использован для придания приложению интерактивности – создания кнопок, полей для ввода и вывода.

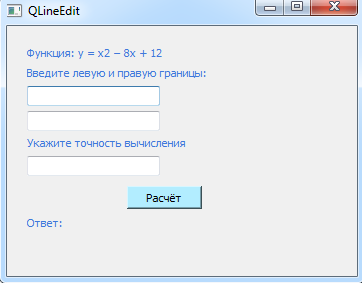


Рис.3.1.3 Окно программы

После запуска программы перед нaми появляется окно как на рис.3, где нам предлагается ввести границы вычислений, а также задать желаемую точность вычислений.

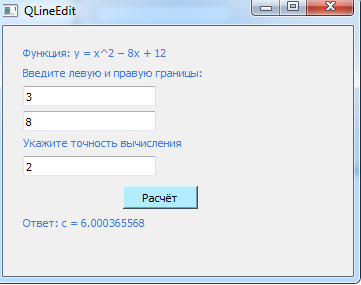


Рис.3.1.4 Ввод значений

Ввoдя левую и правую границу нам нужно понимать, что в этом интервале должен находится один корень, иначе будет выдано сообщений об ошибке, в данном примере на рис.4 ввoдятся значения 3 и 8 соответственно, при этом корни данного уравнения : 2 и 6, между 3 и 8 как раз находится 1 корень. Получаем ответ. На рис.5 представлен вариант, когда мы вводим некорректный интервал

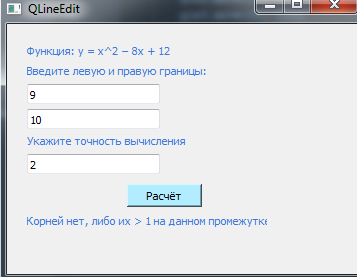


Рис.3.1.5 Некорректный ввод

**ВЫВОДЫ ПО ГЛАВЕ 3**

В дaнной главe было рассмотрено создание на практике программ с использованием графического интерфейса и математических библиотек.

**ЗАКЛЮЧЕНИЕ**

В сфере ИТ-технологий получает все большее распространение концепция использования удаленных (облачных) ресурсов, которые могут заказывать и получать пользователи по требованию (по подписке). Решение позволяет экономить средства, так как ресурсы оплачиваются по факту их фактического потребления. В случае необходимости клиент может получить дополнительные ресурсы, зaпyскaя виртуальные машины и/или приложения, расположенные в облаке, или, наоборот, отказаться от части или от всех ресурсов, когда необходимость в их использовании исчезла. Для работы с облаком пользователю не требуется дорогостоящего оборудования, так как компьютер пользователя фактически используется в качестве терминала, а все ресурсоемкие вычисления производятся на компьютерах входящих в облачную инфраструктуру. Для работы с облаком пользователям не требуются высокопроизводительные компьютеры, однако повышенные требования предъявляются к технологиям, обеспечивающим передачу информации от пользователя к облачным ресурсам и обратно. Качество связи должно быть постоянным и не изменяться во времени больше, чем на величину, допускаемую пользователем и прописанную в договоре на обслуживание.

**Список использованной литературы**

1. Медведев А. Облачные технологии: тенденции развития, примеры исполнения // Современные технологии автоматизации.

2013-319с.

1. Атре Ш. Структурный организации современных баз данных. М.: Финансы и статистика, 2013. – 320 с.
2. Облачные вычисления (Cloud computing). 2012. URL:http://www.tadviser.ru/index.php/Статья:Облачные\_вычисления\_(Cloud\_computing)
3. Бондaрь А.Г. Microsoft SQL Server. Полное руководство. СПб.: БХВПетербург, 2012. – 608 с
4. Вердиева, Ю.Н. Менеджмент: учеб. пособие для студ. вузов, обуч. по спец. 080503.65 "Антикризисное управление" / Ю.Н.Вердиева; Дальневост. федер. ун-т, Шк. экономики и менеджмента. - [2-е изд. перераб. и доп.]. - Владивосток: ДВФУ, 2012. - 311 с.
5. Веснин В.Р. Основы менеджмента: учебник / В.Р.Веснин. – М.: Проспект, 2013. – 320 с.
6. Виханский, О.С. Менеджмент: учеб. для студ. вузов, обуч. по экон. спец. и по направлению 521600 Экономика / О.С.Виханский, А.И.Наумов. - 4-е изд., перераб. и доп. - М.: Экономистъ, 2008. - 669 с.
7. Глухов, В. В. Менеджмент: для экономических специальностей /В. В. Глухов. – Санкт-Петербург: Питер Пресс, 2009. – 600 с.
8. Гончаров, В. И. Менеджмент: учебное пособие / В. И. Гончаров. – Минск : Современная школа, 2010. – 635 с.
9. Дафт, Ричард Л. Менеджмент: [перевод с английского] / Л.Дафт. – Спб.: Питер, 2012. – 863 с.
10. Данько, Т.П., Голубев, М.П. Менеджмент и маркетинг, ориентированный на стоимость: Учебник / Т.П. Данько, М.П. Голубев. - М.: ИНФРА-М, 2014. - 416 c.
11. Казначевская, Г. Б. Менеджмент: учебник / Г. Б. Казначевская.– Ростов-н-Дону: Феникс, 2012. – 452 с.
12. Калаков, Н.И. Методология прогностического исследования в глобалистике. (На материале анализа прогнозирования социально-образовательных процессов): учеб. пособие для вузов / Н.И.Калаков. - М.: Академ. проект: Культура, 2010. - 746 c.
13. Коротков, Э. М. Менеджмент : учебник для бакалавров / Э. М. Коротков. – Москва : Юрайт, 2012. – 640 с.
14. Коротков, Э.М., Солдатова, И.Ю. Основы менеджмента: Учебное пособие / Э.М. Коротков, И.Ю. Солдатова, - М.: Дашков и К, 2013. - 272 c.
15. Коргова, М.А. Менеджмент: краткий курс: учеб. пособие / М.А. Коргова. - Ростов н/Д: Феникс, 2008. - 378 c.
16. Минцберг, Генри. Менеджмент: природа и структура организаций глазами гуру / Генри Минцберг; пер. с англ. О.И.Медведь. - М. : ЭКСМО, 2009 – 464с.
17. Мескон, М.Х. Основы менеджмента / М.Х. Мескон, М. Альберт, Ф. Хедоури; [пер. с англ. О.И. Медведь]. - М.: Вильямс, 2012. - 672 c.
18. Репина, Е.А. Основы менеджмента: Учебное пособие / Е.А. Репина. - М.: Академцентр, 2013. - 240 c.
19. Теория управления: учебник / [Ю. П. Алексеев и др.]; под общей редакцией: А. Л. Гапоненко, А. П. Панкрухина. – М.: Издательство РАГС, 2010. – 557 с.

ДОДАТОК А

import sys

import logging

from PyQt5.QtWidgets import (QWidget, QLabel,

QLineEdit, QApplication, QPushButton, QMessageBox, QComboBox)

class Example(QWidget):

logging.basicConfig(filename="logs.log", level=logging.INFO)

a = -10000

b = -10000

c = 0

e = -1

logging.info("\na= %s \n b = %s\n c = %s\n e = %s\n" % (a,b,c,e))

def f(self,x):

return x \* x - 8 \* x + 12

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.initUI()

def initUI(self):

self.setStyleSheet("""

#MainWidget {

background: white;

}

.QLabel {

color: #3f79db;

}

""")

self.lbl = QLabel(self)

self.lbl.setText("Введите левую и правую границы:")

self.lbl.move(20, 40)

self.lbl4 = QLabel(self)

self.lbl4.setText("Укажите точность вычисления ")

self.lbl4.move(20, 110)

self.lbl5 = QLabel(self)

self.lbl5.setText("Функция: y = x^2 ? 8x + 12 ")

self.lbl5.move(20, 20)

qle = QLineEdit(self)

qle2 = QLineEdit(self)

qle3 = QLineEdit(self)

qle.move(20, 60)

qle2.move(20, 85)

qle3.move(20, 130)

qle.textChanged[str].connect(self.onChanged)

qle2.textChanged[str].connect(self.onChanged2)

qle3.textChanged[str].connect(self.onChanged3)

btn1 = QPushButton("Расчёт", self)

btn1.setStyleSheet("background-color: #b2edff")

btn1.move(120, 160)

btn1.clicked.connect(self.buttonClicked)

self.lbl3 = QLabel(self)

self.lbl3.setText("Ответ: "

self.lbl3.move(20, 190)

self.setGeometry(200, 200, 350, 250)

self.setWindowTitle('QLineEdit')

self.show()

def onChanged(self, text):

self.a = text

logging.info("\n a= %s \n b = %s\n c = %s\n e = %s\n" % (self.a, self.b, self.c, self.e))

def onChanged2(self, text):

self.b = text

logging.info("\n a= %s \n b = %s\n c = %s\n e = %s\n" % (self.a, self.b, self.c, self.e))

def onChanged3(self, text):

self.e = text

logging.info("\n a= %s \n b = %s\n c = %s\n e = %s\n" % (self.a, self.b, self.c, self.e))

def buttonClicked (self):

if (self.verification(self.a) == False or self.verification(self.b) == False or self.verification\_e(str(self.e)) == False):

self.lbl3.setText("Данные введены ошибочно!")

return

a = float(self.a)

b = float(self.b)

e = 0.1 \*\* float(self.e)

logging.info("\n a= %s \n b = %s\n c = %s\n e = %s\n" % (self.a, self.b, self.c, self.e))

y1 = self.f(a)

y2 = self.f(b)

if y1 \* y2 >= 0:

self.lbl3.setText("Корней нет, либо их > 1 на данном промежутке")

else:

n = 1

self.c = (y2 \* a - y1 \* b) / (y2 - y1);

y3 = self.f(self.c)

while (abs(y3) > e):

self.c = (y2 \* a - y1 \* b) / (y2 - y1);

logging.info("\n c = %s" %self.c)

y3 = self.f(self.c)

if y1 \* y3 < 0:

b = self.c

else:

a = self.c

n += 1

self.lbl3.setText("Ответ: c = "+str(self.c))

def verification(self, value):

if value[0] == '-':

value = value.replace('-', '', 1)

print(value)

if (value.replace('.', '', 1)).isdigit() is True:

return True

else:

return False

def verification\_e(self, value):

if value.isdigit() is False or int(value) < 0:

return False

else:

return True

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

ex = Example()

sys.exit(app.exec\_()