

РЕФЕРАТ

Выпускная квалификационная работа 44 с., 33 рис, 16 источн.

ПРОЦЕСС, СЛУЧАЙНЫЙ ПРОЦЕСС, СТОХАСТИЧЕСКИЙ ПРОЦЕСС, БЛУЖДЕНИЕ, ВЕРОЯТНОСТЬ, ОБЪЕКТ, DISCORD, СЛУЧАЙНОЕ БЛУЖДЕНИЕ, ПРИЛОЖЕНИЕ, ПОМОЩНИК

Объектом выпускной квалификационной работы является чат-бот, кроссплатформенной проприетарной системы мгновенного обмена сообщениями – Discord.

Целью выпускной квалификационной работы является разработка помощника для генерации стохастических процессов в формате Discord-бота.

Для достижения поставленной цели необходимо изучить язык программирования C#, также умение работать в интегрированной среде разработки – Visual studio. Знать документацию и уметь работать с библиотеками Discord и QuickChart.

Итог проделанной работы – работающее приложение для генерации стохастических процессов, в формате Discord-бота.

СОДЕРЖАНИЕ

Введение.....	3
1 Анализ предметной области	5
1.1 Анализ платформы Discord	5
1.2 Актуальность чат-ботов.....	6
1.3 Анализ стохастических процессов	7
1.4 Классификация стохастических процессов	9
1.5 Основные понятия Случайных блужданий и их примеры.....	11
1.6 Основные понятия Пуассоновского стохастического процесса и его примеры.....	19
1.7 Основные понятия броуновского движения и его примеры.....	25
2 Программная реализация	29
2.1 Характеристика инструментальных средств разработки	29
2.2 Тестирование и отладка программного обеспечения	30
2.3 Руководство системного программиста.....	31
2.4 Результаты работы приложения.....	36
Заключение	40
Список использованных источников	41

ВВЕДЕНИЕ

В настоящее время приходится постоянно встречать стохастические процессы и явления, а также статистические закономерности. Существует достаточно примеров того, как игнорирование стохастической природы явлений приводит к невообразимым материальным потерям в энергоснабжении, транспорте и других областях. Интерес к развитию всех отраслей науки требует адекватного изучения элементов стохастического мышления. Исследования становятся все более математическими, и роль стохастических методов быстро возрастает во всех областях человеческой деятельности.

Сегодня понимание взаимосвязи между случайностью и необходимостью, статистическими и динамическими закономерностями является обязательной частью общего образования современного человека. Концепция случайности становится все более фундаментальной в науке, и идея принятия оптимальных, научно обоснованных решений получает все большее распространение. Стохастика – это язык, на котором можно по-разному обсуждать ряд проблем. В последние годы этот язык находит все большее применение в науке. Чтобы сделать этот предмет более доступным для студентов, используется платформа Discord, один из лучших мессенджеров в мире.

В роли объекта данной работы выступает помощник в формате discord-бота.

Предметом являются стохастические процессы

Цель данной работы - разработка помощника для генерации стохастических процессов в формате Discord-бота.

Итоговым результатом станет автоматизированный помощник в формате discord-бота, генерирующий стохастические процессы.

Помощник представляет собой специальную программу, имитирующую речевое поведение человека и выступающую в качестве виртуального собеседника в интернет-общении с одним или несколькими пользователями

Чат-бот должен выполнять поставленные задачи, а именно:

- 1) Создание и демонстрация графиков;
- 2) Доступность для всех пользователей интернета;
- 3) Автоматизированная работа;
- 4) Простота использования;
- 5) Возможность безграничного улучшения функционала и работоспособности.

1 Анализ предметной области

1.1 Анализ платформы Discord

Одним из сервисов, который имеет огромный потенциал для использования в удаленном формате, является Discord. Это приложение представляет собой платформу VoIP, мгновенного обмена сообщениями и цифрового распространения информации для создания сообщества. Эта программа появилась на рынке не так давно, но за короткое время завоевала большую популярность благодаря своей универсальности.

Каждый день люди используют Discord для самых разных целей - от обсуждения, художественных проектов и семейных прогулок до проверки домашних заданий и психологических консультаций. Discord открыт для сообществ любого размера, но в основном используется небольшими, активными группами для повседневного общения. Каждый месяц в Discord заходят около 2 миллионов пользователей со всего мира. Большинство серверов являются приватными и предназначены только для приглашенных, где друзья и члены сообщества могут общаться и проводить время вместе. Существуют также более крупные и открытые сообщества, которые обычно сосредоточены на конкретных темах, будь то образовательный или развлекательный контент.

Пользователи общаются по различным каналам, обмениваясь текстовыми сообщениями в различных чатах или в сообществах, где встречаются люди со схожими интересами.

Из Основных функций можно выделить:

Организация голосовых конференций с настройкой канала связи.

- 1) Создание приватных и общих чатов для обмена СМС;
- 2) Голосовое общение;
- 3) Возможность отправки файлов, картинок и ссылок в текстовом чате;

- 4) Личное и групповое общение;
- 5) Трансляция дисплея;
- 6) Возможность использования и создания чат-ботов.

Первый запуск программы состоялся 6 марта 2015 года и с тех пор пользуется популярностью у геймеров. Позже создатель Discord Джейсон Цитрон заявил, что платформа больше не ориентирована только на геймеров, а привлекает людей с совершенно разными вкусами и интересами. Если провести открытый опрос среди студентов колледжей и университетов о том, какую платформу они используют для общения и знают ли они о платформе Discord, результаты будут очевидны для большинства.

Люди любят Discord, потому что это общий дом для их друзей и сообществ, к которым они принадлежат. Это место, где можно быть самим собой и проводить время с людьми, разделяющими те же интересы и увлечения. Здесь нет алгоритмов, определяющих, что они должны увидеть, нет бесконечной прокрутки и ленты новостей. Общение в Discord основано именно на общих интересах. [11]

1.2 Актуальность чат-ботов

В последние годы интерес к использованию чат-ботов возрос в связи с их потенциальным применением в различных областях. Чат-боты стали очень популярны, особенно после того, как разработчикам удалось интегрировать их в различные мессенджеры и социальные сети, такие как Discord, Telegram, Facebook и другие.

В данной статье чат-боты — это специальные программы, которые имитируют речевое поведение человека и выступают в качестве виртуальных собеседников в интернет-общении с одним или несколькими собеседниками.

Чат-боты могут использоваться для развлечения или простого общения; существуют также функциональные чат-боты, которые могут предоставлять информацию о погоде, курсах валют и т.д. Чат-боты используются в

компаниях в рекламных и маркетинговых целях для привлечения внимания потребителей к компании с помощью современного метода общения.

Это программное обеспечение предназначено для облегчения жизни студентов или обычных людей путем предоставления дополнительных функций, которые по тем или иным причинам недоступны в основной программе.

1.3 Анализ стохастических процессов

Стохастический означает случайный. Стохастический процесс — это процесс, поведение которого не является детерминированным, а последующее состояние такой системы описывается величинами, которые могут быть либо предсказанными, либо случайными. Согласно М. Катцу и Э. Нельсону, любой процесс, развивающийся во времени (детерминированный или вероятностный), является стохастическим процессом при анализе в терминах вероятностей. [5]

Стохастический или случайный процесс можно определить как набор случайных величин, индексированных математическим множеством, что означает, что каждая случайная величина случайного процесса однозначно связана с элементом этого множества. Множество, используемое для индексации случайных величин, называется индексным множеством. Каждая случайная величина в коллекции принимает значения из одного и того же математического пространства, называемого пространством состояний. Это пространство состояний может состоять, например, из целых чисел, вещественных чисел или одномерного евклидова пространства. Приращение — это величина, на которую изменяется случайный процесс между двумя значениями индекса, часто интерпретируемыми как две точки во времени. Стохастический процесс может иметь множество исходов в силу своей случайности, а один исход стохастического процесса называется функцией выборки или реализации. Использование термина стохастический в

математике восходит к работе Владислава Борцкевича, который понимал под ним формулирование гипотез, что в свою очередь восходит к древнегреческим философам и работе И. Бернулли "Ars Conjectandi" (лат. "искусство строить гипотезы").

В области искусственного интеллекта стохастические программы используют вероятностные методы. Примерами таких алгоритмов являются: алгоритм моделированного отжига, стохастические нейронные сети, стохастическая оптимизация, генетические алгоритмы. В этом случае стохастичность может быть включена в саму задачу или в планирование в условиях неопределенности.

Метод Монте-Карло получил широкое распространение благодаря физикам Станиславу Уламу, Энрико Ферми, Джону фон Нейману и Николасу Метрополису. Название происходит от названия казино в Монте-Карло, Монако, где дядя Улама одолжил деньги для азартных игр. Использование природы случайности и повторения для изучения процессов похоже на деятельность, происходящую в казино. Использование методов Монте-Карло требует большого числа случайных величин, что привело к разработке генераторов псевдослучайных чисел, которые были намного быстрее, чем табличные методы генерации, ранее использовавшиеся для статистической выборки. [2]

Понятие "стохастический шум" было введено в биологических системах для усиления сигнала внутренней обратной связи. Оно используется для контроля метаболизма у пациентов с диабетом.

Стохастическая теория кроветворения используется в медицине. Рак является примером таких стохастических эффектов.

Нейрокриптография — это подраздел криптографии, которое занимается применением стохастических алгоритмов, особенно искусственных нейронных сетей, для шифрования и криптоанализа. Идеи взаимного обучения, самообучения и стохастического поведения нейронных сетей и подобных алгоритмов могут быть использованы для различных

аспектов криптографии, таких как криптография с открытым ключом, решение проблем распределения ключей путем взаимной синхронизации нейронных сетей, хэширование или генерация псевдослучайных чисел.

1.4 Классификация стохастических процессов

Случайный процесс можно классифицировать различными способами, например, по его пространству состояний, его индексному множеству или связи между случайными величинами, в то время как стохастические процессы классифицируются одним способом, а именно по мощности индексного множества и пространства состояний.

С точки зрения времени, случайный процесс считается дискретным во времени, если его индексное множество содержит конечное или счетное число элементов, например, множество конечных чисел, множество целых или натуральных чисел. Считается, что время непрерывно, если индексное множество является интервалом вещественной линии. Стохастические процессы дискретного времени и стохастические процессы непрерывного времени — это два типа стохастических процессов. Если индексное множество состоит из целых чисел или их подмножества, стохастический процесс также называется случайной последовательностью.

Если пространство состояний состоит из целых или натуральных чисел, стохастический процесс называется дискретным или целочисленным стохастическим процессом. Если пространство состояний является вещественной прямой, то стохастический процесс называется вещественным стохастическим процессом или непрерывным процессом пространства состояний. Если пространство состояний является η -мерным евклидовым пространством, то стохастический процесс называется η -мерным векторным процессом или η -векторным процессом.

В дискретном времени значения переменных рассматриваются как происходящие в разные, отличные друг от друга "моменты времени", или, что

одно и то же, как неизменные в каждый ненулевой интервал времени ("период времени"), т.е. время рассматривается как дискретная переменная. Не временная переменная, таким образом, изменяется от одного значения к другому при переходе от одного временного периода к другому. Такое представление о времени соответствует цифровым часам, которые показывают фиксированное показание 10:37 в течение некоторого времени, а затем переходят к новому фиксированному показанию 10:38 и так далее. D. В этой структуре каждая переменная, представляющая интерес, измеряется один раз в каждом временном периоде. Количество измерений между любыми двумя временными периодами ограничено. Измерения обычно проводятся на последовательных целочисленных значениях переменной "время".

Дискретно-временной сигнал или сигнал дискретного времени — это временной ряд, состоящий из последовательности величин.

В отличие от сигнала непрерывного времени, сигнал дискретного времени не является функцией непрерывного аргумента. Если сигнал дискретного времени получен путем дискретизации последовательности равномерно распределенных временных точек, то он имеет соответствующую частоту дискретизации.

Сигналы дискретного времени могут иметь несколько источников, но в целом их можно разделить на две группы:

Определение значений аналогового сигнала с постоянной или переменной скоростью. Этот процесс называется дискретизацией. Наблюдение за процессом, который по своей природе дискретен во времени, например, недельный пик определенного экономического показателя.

В непрерывном времени переменные рассматриваются так, как будто они имеют конкретное значение, которое может быть актуально только в течение бесконечно короткого периода времени. Между любыми двумя точками во времени существует бесконечное количество других точек во времени. Переменная "время" включает в себя весь ряд действительных чисел или, в зависимости от контекста, их подмножество, например,

неотрицательные числа. Таким образом, время рассматривается как непрерывная переменная.

Непрерывный сигнал или сигнал непрерывного времени — это изменяющаяся величина (сигнал), диапазон, который часто занимает длительное время - континуум (например, интервал непрерывных чисел). То есть, диапазон функции является бесконечной величиной. Сама функция не обязательно должна быть непрерывной. Напротив, сигнал дискретного времени имеет счетный диапазон, подобно натуральным числам.

Сигнал с непрерывной амплитудой и временем называется сигналом непрерывного времени или аналоговым сигналом. Он (сигнал) будет иметь определенное значение в любой момент времени. Электрические сигналы, полученные в отношении физических величин, таких как температура, давление, звук и т.д. и т.п., обычно являются непрерывными сигналами. Другими примерами непрерывных сигналов являются синусоидальные волны, косинусоидальные волны, треугольные волны и т.д.

Сигнал определяется в области, которая может быть или не быть конечной, и существует функциональное отображение области на значение сигнала. Непрерывность временной переменной, обусловленная законом плотности вещественных чисел, означает, что значение сигнала может быть найдено в любой момент времени. [9]

1.5 Основные понятия Случайных блужданий и их примеры

Случайную прогулку можно определить как серию дискретных шагов, которые объект совершает в определенном направлении. А также вероятностно определяется направление и движение объекта на каждом шаге. В математике и теории вероятностей случайная прогулка — это случайный процесс.

В случайном блуждании будущее положение совершенно не зависит от текущего положения объекта. Это также пример марковского процесса. Из

заданного положения объект может двигаться в любом направлении. Каждый шаг объекта в любом направлении связан с вероятностью. Таким образом, конечная позиция полностью зависит от начальной точки.

В концепции случайных операций наиболее важной задачей является нахождение функции распределения вероятности, которую можно использовать для оценки вероятности текущего положения объекта после случайной прогулки в течение определенного периода времени.

Случайные блуждания — это случайные процессы, которые обычно определяются как суммы случайных величин iid или случайных векторов в евклидовом пространстве, что означает, что они являются процессами дискретного времени. Однако некоторые люди используют этот термин для процессов, которые изменяются в реальном времени, например, процесс Винера, используемый в финансах, что привело к некоторой путанице и критике. Сторонники этой теории предполагают, что цены на ценные бумаги на фондовом рынке развиваются случайным образом.

Существует два типа случайных блужданий, основанных на положении объекта: повторяющиеся и временные.

Повторяющиеся случайные прогулки гарантируют возврат в исходное положение, начиная со случайной позиции. Одномерные и двумерные случайные блуждания относятся к этой категории. Это видно из приведенных примеров. Большое количество шагов гарантирует, что объект вернется в исходное положение.

В отличие от повторяющихся случайных блужданий, временные случайные блуждания не гарантируют возвращения в исходное положение. В большинстве случаев существует даже положительная вероятность того, что прогулка никогда не вернется в исходное положение.

Случайные блуждания в 3 и более измерениях попадают в категорию переходных процессов. Огромное количество вариаций в движении объекта на каждом шаге делает невозможным его возвращение в исходное положение.

Существует множество приложений случайного блуждания в математике, информатике, биологии, химии и физике. В случае генетического дрейфа в биологии случайные блуждания могут дать нам общее представление об участвующих статистических процессах. В физике есть возможность использовать их для описания идеальной цепи в физике полимеров. [1]

Концепция случайного блуждания также очень важна и используется в различных областях, таких как психология, финансы и экология. Кроме того, имеется возможность описать колебания на фондовом рынке с помощью концепции случайного блуждания. Кроме того, они также используются алгоритмами поисковой системы Google.

Рассмотрим примеры случайных блужданий:

Одномерное случайное блуждание является классическим примером случайного блуждания. Это стохастический процесс в дискретном времени с целыми числами в качестве пространства состояний, основанный на процессе Бернулли, где каждая переменная Бернулли принимает либо положительное, либо отрицательное значение. Другими словами, простое случайное блуждание происходит с целыми числами, и его значение увеличивается на единицу с вероятностью p или уменьшается на единицу с вероятностью $1-p$, так что индексное множество этого случайного блуждания - натуральные числа, а его пространство состояний - целые числа. Если $p=0,5$, то такое случайное блуждание называется симметричным случайным блужданием.

Самая простая и базовая случайная прогулка — это одномерная прогулка. Рассмотрим на Рисунок 1 случайную прогулку по целым числам:



Рисунок 1 – Случайная прогулка по целым числам (первый этап)

Итак, перед нами объект, который находится в точке 0 и может двигаться в двух направлениях: вперед и назад. Теперь начнется определение направления каждого шага объекта, подбрасывая монету. Если это головы, объект движется вперед. Если это хвосты, объект движется назад. Здесь подбрасывается монета, объект перемещается на один шаг в соответствии с правилом, и монета подбрасывается снова.

На Рисунок 2 изображен первый этап:

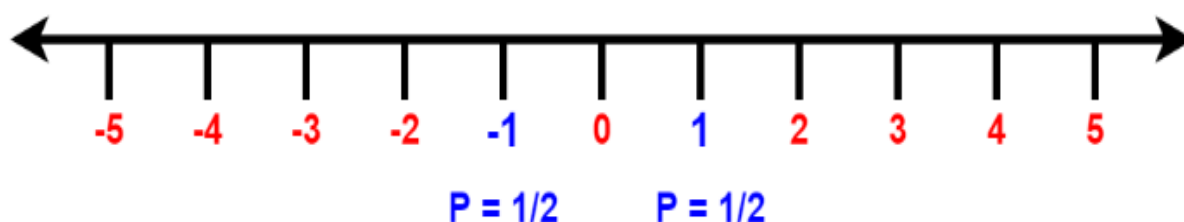


Рисунок 2 – Случайная прогулка по целым числам (2 этап)

После первого хода объект может либо перейти в положение +1, либо -1 с равной вероятностью $\frac{1}{2}$. Рассмотрим второй этап на Рисунок 3:

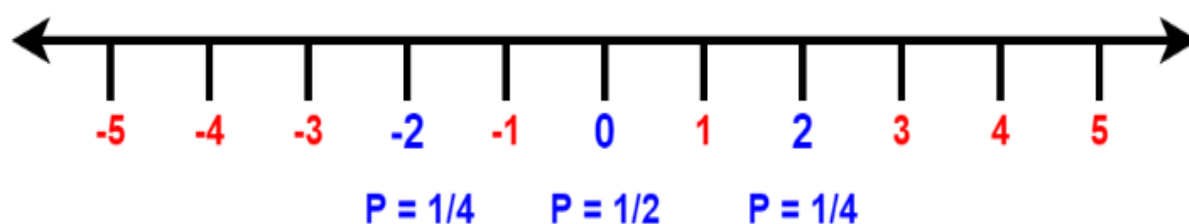


Рисунок 3 – Случайная прогулка по целым числам (3 этап)

После второго шага имеется возможность найти объект в трех позициях: -2, 0 или 2. Здесь вероятности для позиций 2 и -2 одинаковы. С вероятностью $1/2$ объект может находиться в положении 0. Аналогично, рассмотрим вероятности на третьем этапе на Рисунок 4:

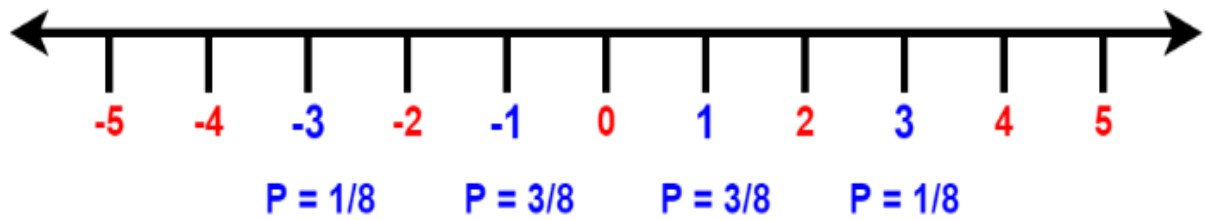


Рисунок 4 – Случайная прогулка по целым числам (4 этап)

Здесь объект находится в позициях -3, -1, 1, 3. Вероятность найти объект в позиции 3 или -3 равна $3/8$, а вероятность найти объект в позиции 1 или -1 равна $1/8$.

Теперь легко понять, что если число вращений нечетное, то все возможные положения, в которые может переместиться объект, нечетные. Аналогично, если число вращений четное, то все возможные позиции четные.

Теперь, чтобы получить больше информации от одномерного случайного блуждания, предположим, что S_j — это результат броска j -й монеты в случайном процессе. Таким образом, S_1 — это результат броска первой монеты в первом раунде. Кроме того, известно, что переменная S_j является случайной величиной. [4]

В одномерном случайном блуждании ожидаемое или среднее значение случайной величины S_j всегда равно 0.

На Рисунок 5 рассматривается случайное блуждание в двумерной целочисленной сетке:

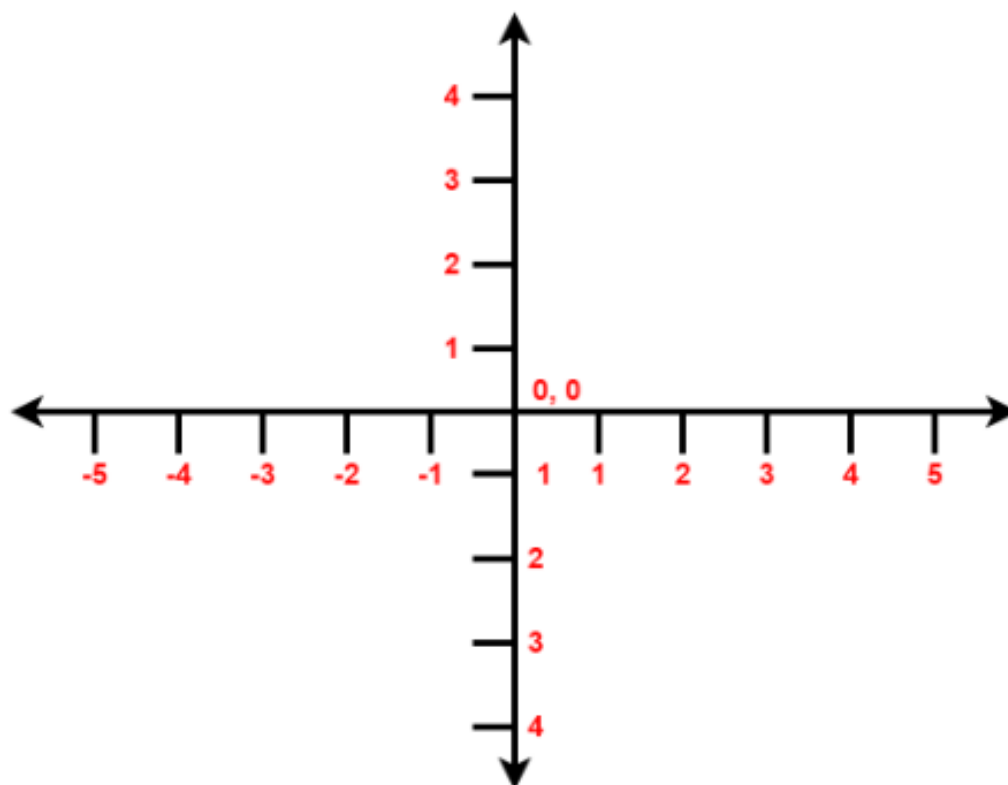


Рисунок 5 – Случайное блуждание в двумерной целочисленной сетке

В двумерной случайной прогулке объект может двигаться в четырех различных направлениях: вперед, назад, влево, вправо. Следовательно, в этой среде, чтобы переместить объект, необходимо подбрасывать монету дважды на каждом шаге. При первом повороте есть возможность решить, движется ли объект вперед или назад. Второй бросок решает, куда он полетит – направо или налево.

На Рисунок 6 движение начинается в начальной точке Н, которая обычно совпадает с началом координат ($x=0$; $y=0$), и заканчивается через М шагов в точке К. В этом случае часто требуется определить расстояние точки К, полученной за М шагов, от начала координат.

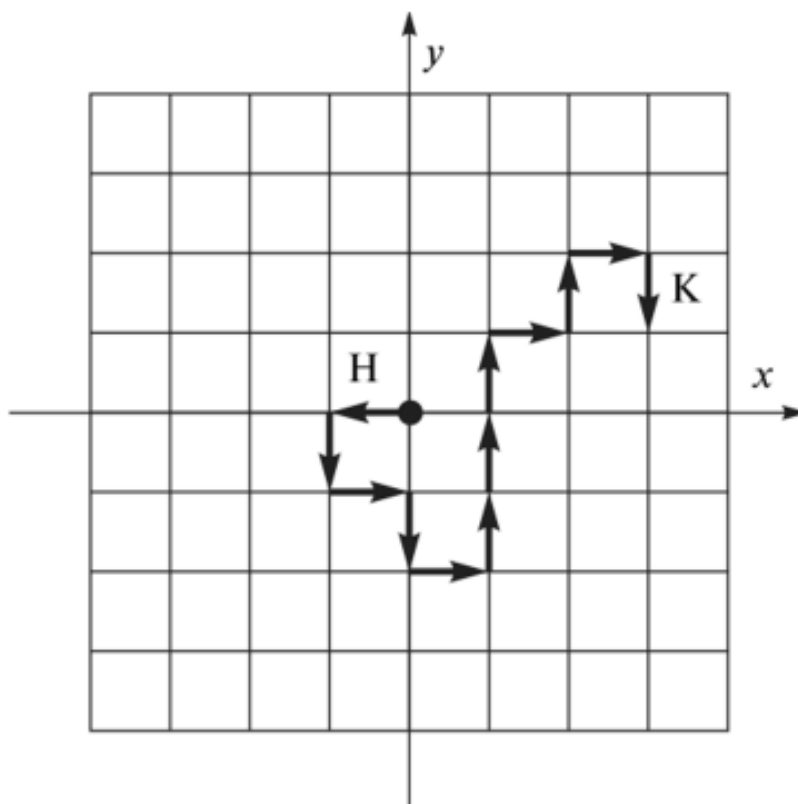


Рисунок 6 – Случайное блуждание в двумерной целочисленной сетке

Обратите внимание, что сетки могут иметь и другую форму. Например, на Рисунок 7 изображено случайное блуждание по треугольной сетке. При такой ходьбе движение осуществляется с равной вероятностью в одном из шести возможных направлений на каждом шаге

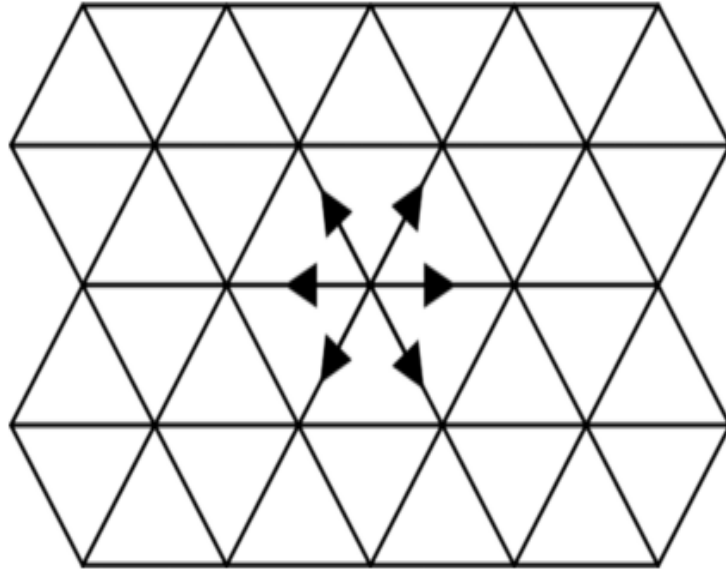


Рисунок 7 – Случайное блуждание по треугольной сетке

На Рисунок 8 проиллюстрирована проблема случайной миграции на треугольной сетке

Проблема случайной миграции на сотах также хорошо известна. В такой миграции движение осуществляется с равной вероятностью в одном из трех возможных направлений на каждом шаге

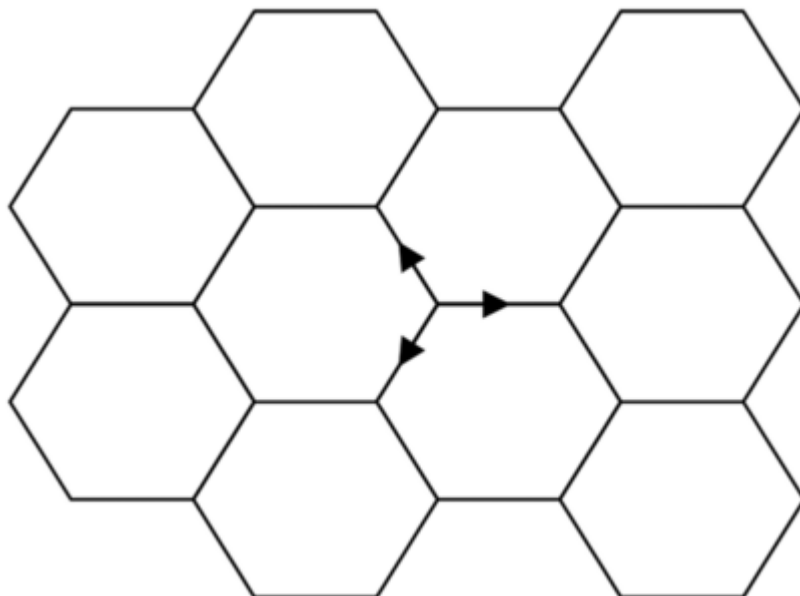


Рисунок 8 – Случайная миграция на сотах

Другим вариантом модели ограниченного случайного блуждания является модель дождевых капель. Случай дождевой капли, подверженной воздействию случайных легких порывов ветра, можно смоделировать с помощью случайной прогулки по квадратной сетке (См. Рисунок 9). Движение начинается в узле, расположенном на расстоянии h от горизонтальной линии (над поверхностью земли). В этом случае частица достигает поверхности земли после определенного количества шагов. Необходимо оценить среднее время падения капли. [3]

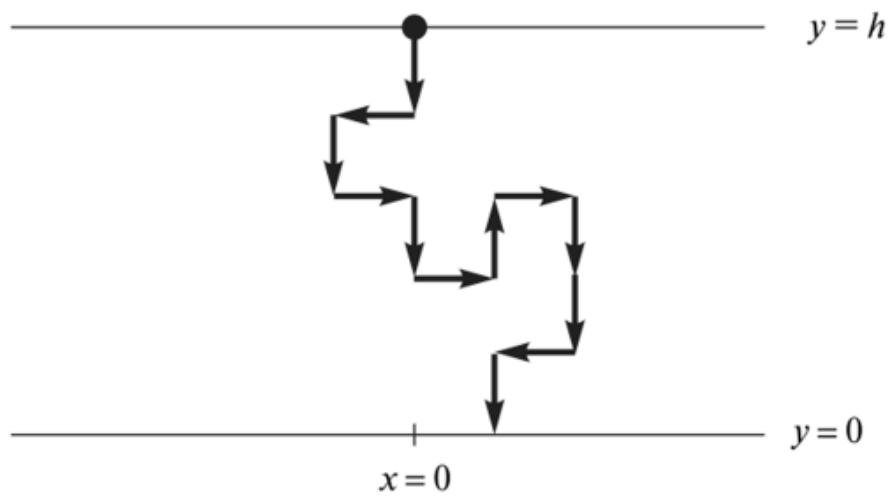


Рисунок 9 – Модель дождевых капель

Интересно, что при случайном блуждании вероятность достижения любой точки в 2D-сетке равна 1, когда устанавливается количество шагов равное бесконечности. [8]

1.6 Основные понятия Пуассоновского стохастического процесса и его примеры

В теории вероятностей, статистике и смежных областях пуассоновский точечный процесс — это тип математического случайного объекта, состоящего из точек, которые случайно расположены в математическом пространстве. Пуассоновский точечный процесс часто называют просто

пуассоновским процессом, но его также называют пуассоновской случайной мерой, пуассоновским случайным точечным полем или пуассоновским точечным полем. Этот точечный процесс обладает практическими математическими свойствами, благодаря которым он часто определяется в евклидовом пространстве и используется в качестве математической модели для кажущихся случайными процессов во многих дисциплинах, таких как астрономия, биология, экология, геология, сейсмология, физика, экономика, визуализация и телекоммуникации. [6]

Процесс Пуассона можно использовать для моделирования количества событий, например, пациентов, поступающих в отделение неотложной помощи, в течение определенного периода времени.

Пуассоновский процесс обладает следующими свойствами:

1) Он состоит из последовательности случайных переменных $X_1, X_2, X_3, \dots, X_k$, таких, что каждая переменная представляет собой количество раз, когда событие, например, прибытие пациентов в отделение неотложной помощи, происходит за определенный период времени.

2) Это случайный процесс. Каждый раз, когда вы запускаете процесс Пуассона, он будет производить различную последовательность случайных исходов в соответствии с распределением вероятности, которое будет показано через некоторое время.

3) Это дискретный процесс. Результатом процесса Пуассона является число повторений события в течение заданного периода времени, которое является однозначно целым числом, т.е. дискретным.

4) Он имеет независимые инкременты. Это означает, что количество событий, которые процесс предсказывает в данном интервале, не зависит от количества событий в другом, не перекрывающемся интервале.

5) Составляющие переменные пуассоновского процесса X_1, X_2, X_3, X_k имеют одинаковое распределение.

б) Составляющие переменные пуассоновского процесса $X_1, X_2, X_3, \dots, X_k$ имеют распределение Пуассона, задаваемое функцией массы вероятности (1):

$$P_x(k) = \frac{e^{-\lambda} \cdot \lambda^k}{k!} \quad (1)$$

Из приведенной выше формулы следует, что вероятность появления k событий в единицу времени получается, когда средняя частота появления составляет λ событий в единицу времени.

График изображенный на Рисунок 10 показывает функцию массы вероятностей для различных значений λ :

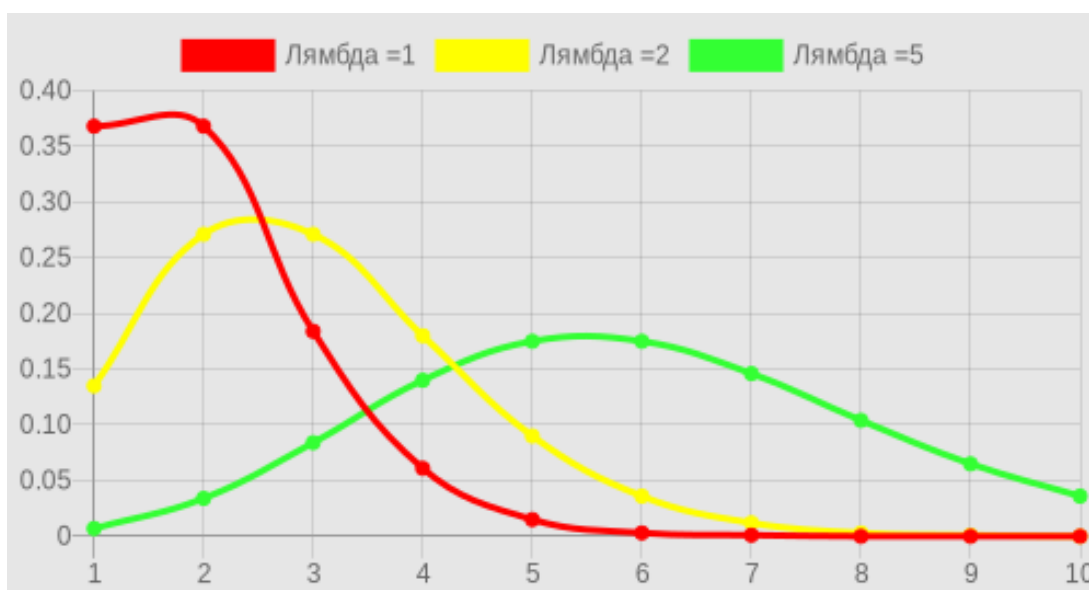


Рисунок 10 – Функция массы вероятностей

Каждая диаграмма показывает, что вероятность достигает максимума при соответствующем значении λ и уменьшается по обе стороны от этого значения.

В каждой диаграмме сумма вероятностей для всех возможных значений k всегда равна 1,0, т.е. существует уверенность в том, что один из исходов произойдет (2).

$$\sum_{k=0}^{\infty} P_x(k) = \sum_{k=0}^{\infty} \frac{(\lambda)^k \cdot e^{-\lambda}}{k!} = 1 \quad (2)$$

Давайте подробнее рассмотрим ситуацию, когда $\lambda = 5$. Как вы можете видеть, вероятность достигает максимума при $k = 5$.

Чтобы определить вероятность того, что k пациентов попадут в отделение неотложной помощи через t часов, происходит моделирование как процесс Пуассона со скоростью (λt) . Соответствующая формула (3) для массовой функции вероятности для k случаев в момент времени t выглядит следующим образом:

$$P_x(k) = \frac{(\lambda t)^k \cdot e^{-(\lambda t)}}{k!} \quad (3)$$

Пуассоновский процесс имеет замечательную подструктуру. В то время как количество событий моделируется дискретным распределением Пуассона, временной интервал между последовательными событиями может быть смоделирован экспоненциальным распределением, которое является непрерывным распределением.

Пусть X_1, X_2, X_3 и т.д. - являются такими случайными величинами, что:
 X_1 = интервал времени между началом процесса и первым событием, т.е. первым прибытием,
 X_2 = интервал времени между первым прибытием и вторым прибытием,
 X_3 = интервал времени между вторым приходом и третьим и т.д.

Распределение случайной переменной X_k , которая представляет собой время между $(k-1)$ -м и (k) -м прибытием, имеет вид (4):

$$X_k = \text{Exponential}(\lambda) \quad (4)$$

Функция плотности вероятности случайной величины X_k выглядит следующим образом (5):

$$P_x(t) = \lambda e^{-\lambda t} \quad (5)$$

И Кумулятивная функция распределения (КФР) (6):

$$1 - e^{-\lambda t} \quad (6)$$

Напомним, что кумулятивная функция распределения X показывает вероятность того, что временной интервал между последовательными поступлениями меньше или равен некоторому значению t .

Теперь у нас достаточно информации, чтобы определить время между прибытиями в процессе Пуассона. Используя метод обратного КФР, где буквально строится обратная функция КФР и вводятся различные значения вероятности из равномерного распределения (0,1). Это дает соответствующие временные интервалы между прибытиями для соответствующих вероятностей.

Обратная функция КФР от времени между прибытиями такая же (7):

$$F_x^{-1}(t) = -\frac{\ln(1-t)}{\lambda} \quad (7)$$

Теперь, когда есть информация, как определить время между прибытиями, легко определить время прибытия пациента.

Время прибытия первого пациента = $x_1 = 0,431257556$

Время прибытия второго пациента = $x_1 +$ время между прибытием первого и второго пациента = $x_1 + x_2 = 0,431257556 + 0,264141966 = 0,6954$

Время прибытия третьего пациента = $x_1 + x_2 + x_3 = 0,431257556 + 0,264141966 + 0,190045932 = 0,885445 \dots$ и так далее.

Помня, что X_1, X_2, X_3, X_k — это время между поступлениями, определяем $T_1, T_2, T_3, \dots, T_k$ как переменные, представляющие время прибытия пациента в отделение неотложной помощи, видно, что:

$$T_1 = X_1$$

$$T_2 = X_1 + X_2$$

$$T_3 = X_1 + X_2 + X_3$$

...

$$T_k = X_1 + X_2 + X_3 + \dots + X_k$$

Поскольку $T_1, T_2, T_3, \dots, T_k$ определены как линейные комбинации случайных величин X_1, X_2, X_3, X_k , переменные T_1, T_2, T_3, T_k также являются случайными величинами.

Теперь имеется возможность перейти к моделированию всего пуассоновского процесса.

Для этого нам необходимо применить простую двухэтапную процедуру:

- 1) Для заданной средней частоты событий λ используйте обратный метод КФР, чтобы найти время между прибытиями;
- 2) Сформируйте фактическое время прибытия, построив график текущей суммы интервальных времен прибытия (См. Рисунок 11).

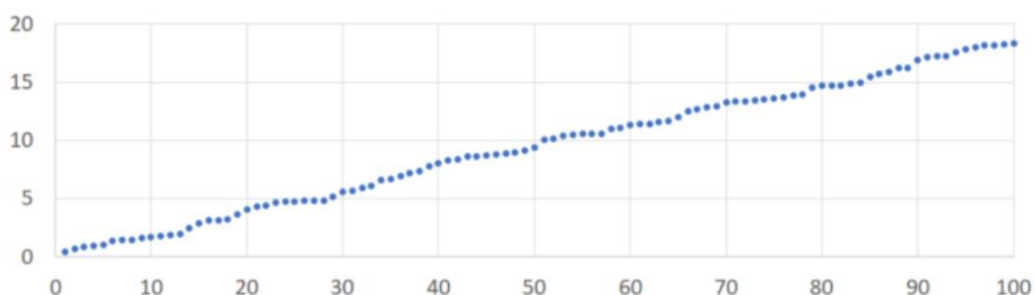


Рисунок 11 – Пуассоновский стохастический процесс

1.7 Основные понятия броуновского движения и его примеры

Броуновское движение — это хаотическое движение частиц молекулярного размера, взвешенных в жидкости. Это происходит из-за случайных столкновений частиц с быстро движущимися молекулами в жидкости (возбуждаемой внутренней тепловой энергией). Считается, что вышеупомянутая жидкость находится в так называемом тепловом равновесии, где нет предпочтительного направления потока/

Объяснение Венского процесса, с точки зрения непрофессионала, является обязательной концепцией для прогнозирования количественных финансов.

Взаимодействия многих тел, которые приводят к сложной, но красивой броуновской модели движения, не могут быть решены с помощью модели первого принципа, учитывающей детальное движение молекул. Таким образом, для их описания могут использоваться только вероятностные макромодели, применяемые к молекулярным популяциям.

Это рассуждение основано на описании броуновского движения, по существу, как чисто случайного процесса в его современной форме. Почти во всех практических приложениях также используется этот подход.

Математические свойства одномерного броуновского движения были впервые проанализированы американским математиком Норбертом Винером. Полученный формализм представляет собой реальный стохастический процесс непрерывного времени, называемый Венским процессом.

Это один из самых известных стохастических процессов с такими привлекательными характеристиками, как неподвижность и независимые шаги. Поэтому он часто находит применение в самых разных областях, которые включают чистую и прикладную математику, количественные финансы, экономическое моделирование, квантовую физику и даже эволюционную биологию.

Броуновское движение — это концепция, которую нужно знать. Они широко используются в ряде областей, таких как моделирование фондового рынка, физика, биология, химия, квантовые вычисления и многие другие. Кроме того, этот процесс лежит в основе большинства моделей моделирования.

В квантовой физике явления диффузии, связанные с уравнениями Фоккера-Планка и Лангьюина, изучаются с использованием броуновского движения. Это также является основой для формирования строгой формулировки интеграла по траекториям квантовой механики. Например, используя формулу Фейнмана-Каца, можно представить себе решение знаменитого уравнения Шредингера по отношению к Венскому процессу. Модель вечной инфляции в физической космологии вдохновлена динамикой броуновского движения.

В мире финансов и эконометрического моделирования броуновское движение имеет мифический статус. Он занимает видное место почти во всех основных математических теориях финансов.

Броуновское движение также набирает популярность в проектах прогнозирования в области науки о данных и закладывает основу для моделирования Монте-Карло.

Итак, броуновское движение — это тема, с которой все должны быть знакомы.

Основное уравнение (8), лежащее в основе генерации точек данных в соответствии с динамикой броуновского движения, довольно простое

$$W\left(\frac{i}{n}\right) = W\left(\frac{i-1}{n}\right) + \frac{Y_i}{\sqrt{n}} \quad (8)$$

где Y_i может быть базовым стохастическим процессом, таким как случайное блуждание или выборка из нормального распределения.

Имеется возможность использовать базовый стохастический процесс, такой как случайное блуждание, для генерации точек данных для броуновского движения, как на Рисунок 12.

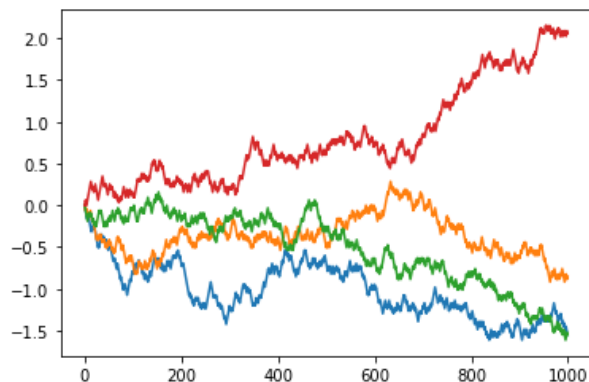


Рисунок 12 – Случайное блуждание

Процесс с начальным значением 20 и использованием нормального распределения

Теперь есть возможность генерировать данные о броуновском движении, отталкиваясь от нормального распределения (См. Рисунок 13).

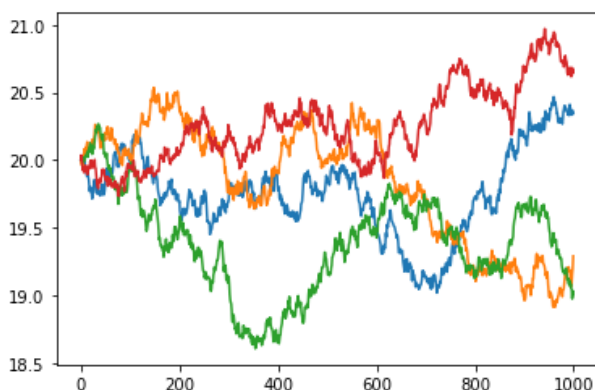


Рисунок 13 – Нормальное распределение

Была реализована геометрическую модель броуновского движения в классе в качестве метода (9).

$$S(t) = S(0) \exp \left\{ \left(\mu - \frac{\sigma^2}{2} \right) t + \sigma W(t) \right\}, t \geq 0. \quad (9)$$

Обратите внимание, что динамика контролируется средним значением и дисперсией базового нормального распределения. Это каким-то образом имитирует восходящий тренд и "волатильность" акций.

Например, акции с положительным восходящим трендом имеют положительное среднее значение. Для этого конкретного моделирования выбор среднего значения (μ) равен 0,2, а выбор стандартного отклонения (квадратный корень из дисперсии) равен 0,68.

Обратите внимание, что, хотя сценарии выглядят довольно стохастично, они имеют тенденцию к снижению. Это связано с тем, что даже при положительном среднем у нас немного высокая скорость или волатильность (См. Рисунок 14). [7]

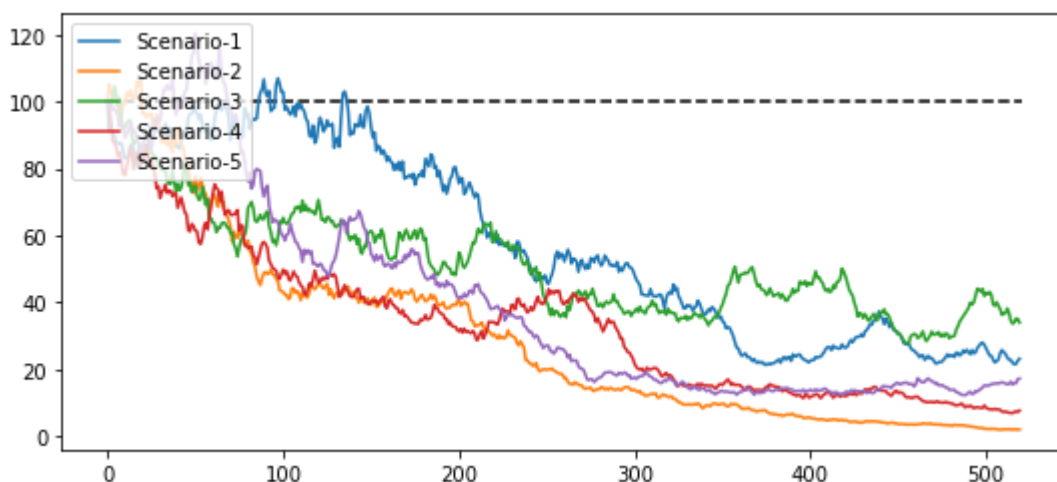


Рисунок 14 – Модель броуновского движения

2 Программная реализация

2.1 Характеристика инструментальных средств разработки

В данной работы был использован язык объектного программирования C# с помощью Visual Studio. NET — это разработка Microsoft.

Программирование было выполнено на C# с использованием Discord API, Discord.NET, QuickChart.

C# — это язык программирования, который сочетает в себе объектно-ориентированные и контекстно-ориентированные концепции. Разработан в 1998-2001 годах группой специалистов под руководством Андрея Хейлсберга в Microsoft в качестве основного языка программирования для платформы Microsoft.NET. Язык имеет строгую статическую типизацию, поддерживает полиморфизмы, перегрузки операторов, ссылки на функции класса-члена, атрибуты, события и свойства, исключения, комментарии в формате XML.

Discord.NET - это асинхронная мультиплатформенная платформа. Сетевые библиотеки, используемые для совместного взаимодействия с Discord API. Сам Discord API представляет собой набор инструментов, которые позволяют вам взаимодействовать с самим Discord API.

Quick chart – это веб-сервис, который генерирует изображения диаграмм "на лету". Эти изображения подходят для встраивания в электронную почту, SMS, чат-боты и другие форматы. Диаграммы отображаются с помощью Chart.js, популярная библиотека построения графиков с открытым исходным кодом.

Данное программное обеспечение разрабатывалось для запуска на сервере, поэтому полноценного пользовательского графического интерфейса она не имеет, вместо него присутствует примитивный консольный интерфейс, представленный на Рисунок 15. [10]

```
C:\Users\Pushka\source\repos\DiscordBot\bin\Debug\net5.0\DiscordBot.exe
08:05:28 Discord      Discord.Net v3.7.1 (API v9)
08:05:28 Gateway      Connecting
08:05:33 Gateway      Connected
08:05:34 Gateway      Ready
```

Рисунок 15 – Графический интерфейс

2.2 Тестирование и отладка программного обеспечения

В основном за устойчивость ПО отвечает обработчик команд. В него были добавлены проверки, от кого, в каком формате и как была послана команда. Первая проверка происходит на отправителя команды, проверяется от кого вообще исходит команда. В реальности может произойти такое, что команды могут исходить от других ботов, и что бы не было произвольного вызова функционала, производится проверка отправителя. Если команда исходит не от пользователя, то она игнорируется. Вторая проверка производится на наличие префикса «!». Так как все команды начинаются с этого префикса (это нужно что бы бот не воспринимал обычные слова пользователей из текстовых каналов), то соответственно отсутствие этого префикса ведет к игнорированию входных данных. Если эти проверки были пройдены, то тогда программа, инициализируя командный контекстный сокет, переходит к исполнению команды. Вызов обработчиков команд устроен следующим образом: производится асинхронный вызов обработчиков, и в случае какой-либо ошибки, вместо остановки исполнения программы они вернут саму ошибку, которая отразится в чате. Тем самым реализована устойчивость программы к внутренним ошибкам, которые не приведут к падению программы. На рисунках (Рисунок 16, Рисунок 17, Рисунок 18) представлены примеры вывода ошибок: [12]

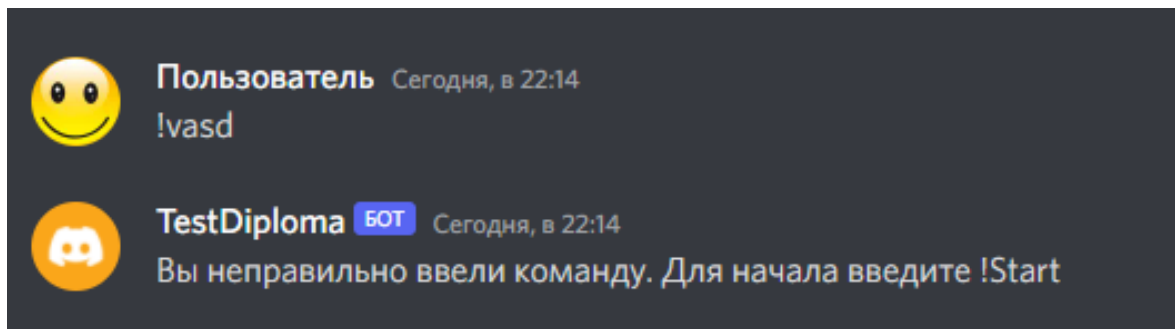


Рисунок 16 – Обработка ошибки

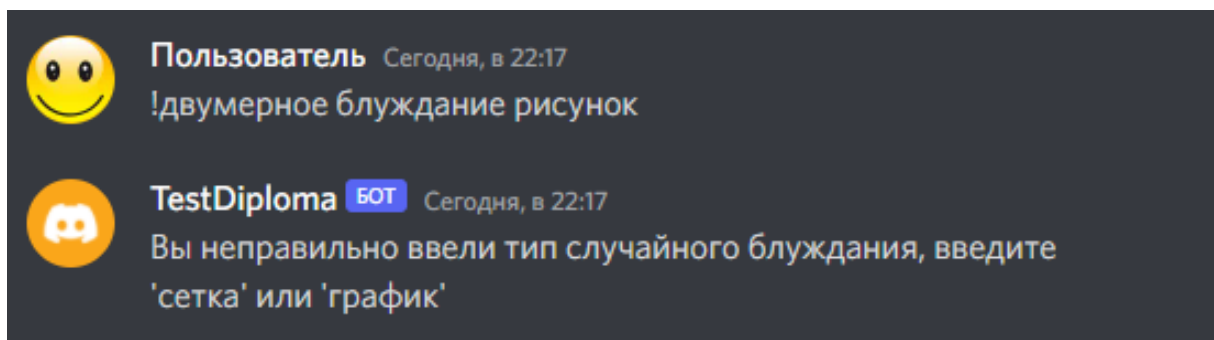


Рисунок 17 – Обработка ошибки

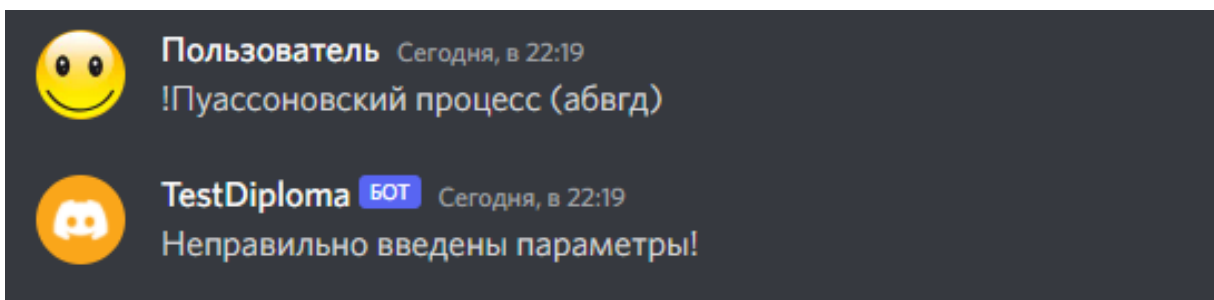


Рисунок 18 – Обработка ошибки

2.3 Руководство системного программиста

Прежде всего нужно создать объект своего бота для Discord, для этого необходимо перейти на сайт <https://discordapp.com/developers/applications/>, где производится авторизация, и изображен список приложений, как на Рисунок 19

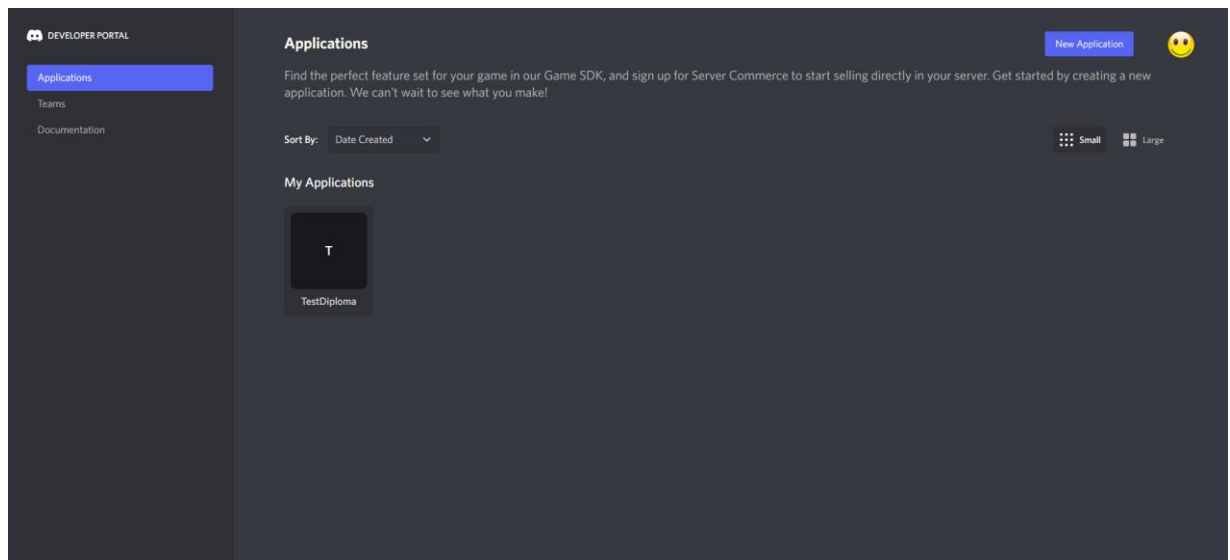


Рисунок 19 – Список приложений

Изначально, созданные приложения отсутствуют, пользователь нажимает на **New Application**, как показано на Рисунок 20:

The image shows a 'CREATE AN APPLICATION' form. It features a message box asking if the user is a game dev and suggesting they contact Dev Support. Below this is a 'NAME' input field with a red asterisk. A blue link for 'Discord API Terms of Service' is present. At the bottom are 'Cancel' and 'Create' buttons.

Рисунок 20 – Создание приложения

После этого нажимается кнопка **Create**, и происходит переход на страницу бота, где есть возможность загрузить аватарку для бота, добавить описание и название как показано на Рисунок 21:

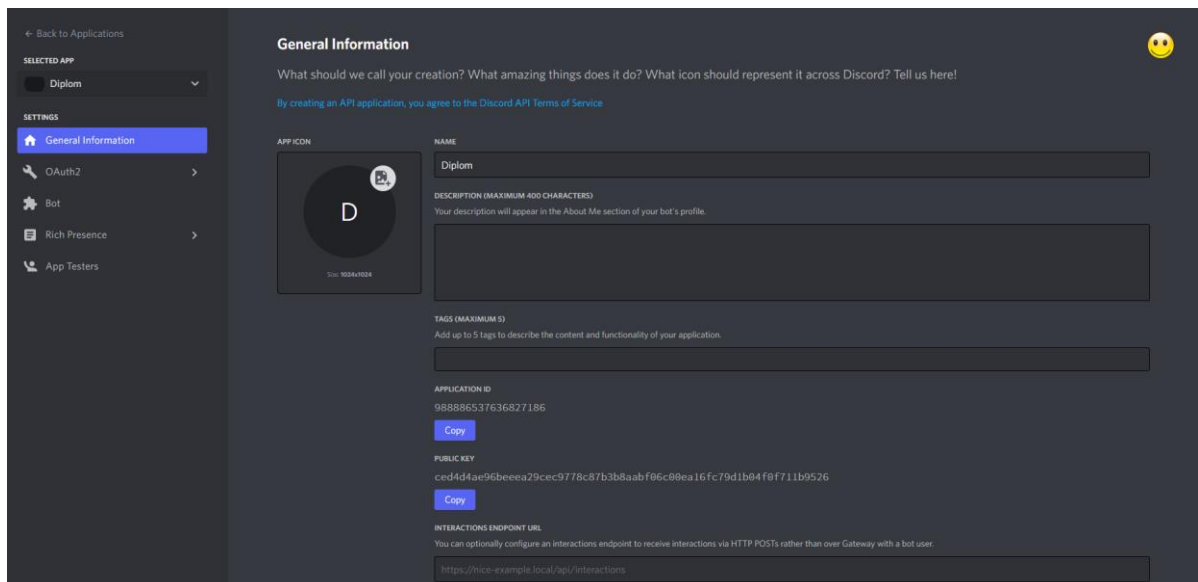


Рисунок 21 – Страница бота

Пользователю необходимо перейти во вкладку Bot и нажать кнопку Add Bot, как показано на Рисунок 22.

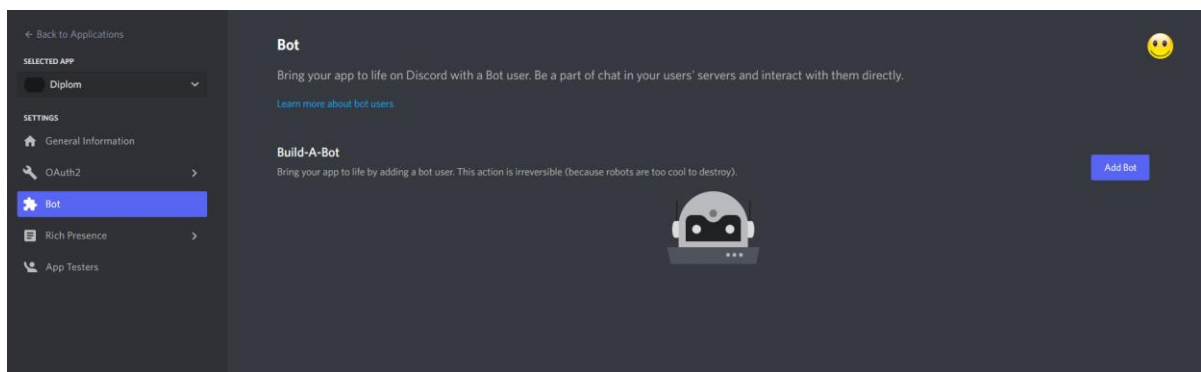


Рисунок 22 – Создание бота

Принимаем все соглашения, и после этого появиться раздел бота (будет сказано, что бот появился), как показано на Рисунок 23

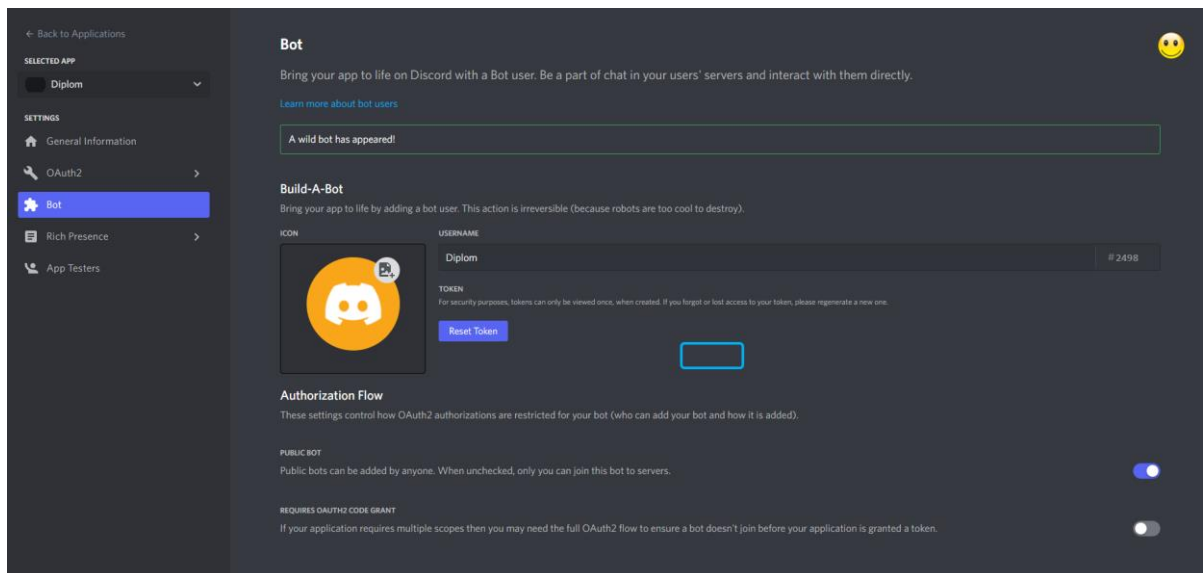


Рисунок 23 – Меню бота

Нужно запомнить, где находится раздел Токена и кнопка Copy (См. Рисунок 24), это необходимо в будущей настройке:

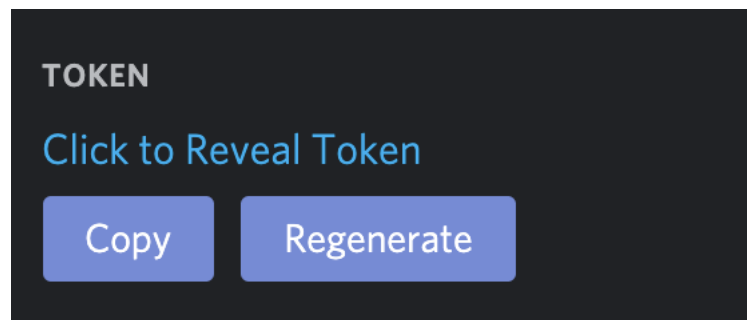


Рисунок 24 – Токен

Теперь необходимо добавить бота на сервер, для этого требуется зайти во вкладку OAuth2 и в секции Scopes выбрать bot, как показано на Рисунок 25.

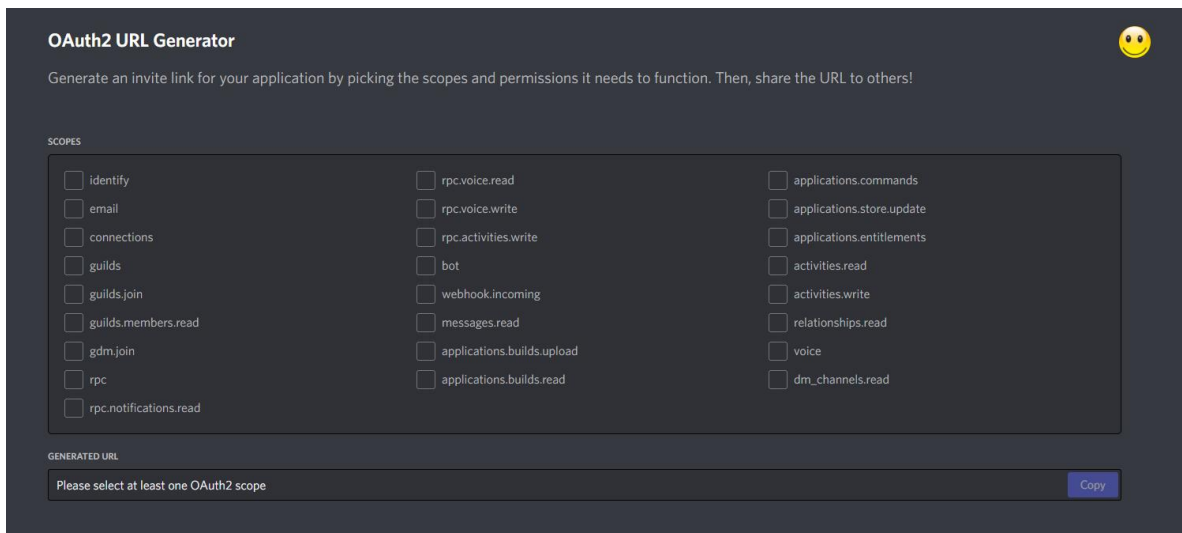


Рисунок 25 – Ссылка для добавления на сервер

После этого выбирается доступный сервер, и нажимается кнопка. Авторизовать как показано на Рисунок 26

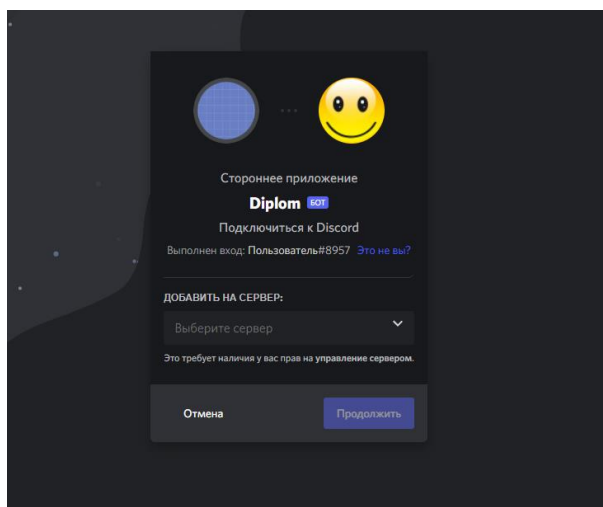


Рисунок 26 – Авторизация бота

Теперь пользователь может перейти на сервер, и увидеть бота в списке пользователей. Пример указан на Рисунок 27:

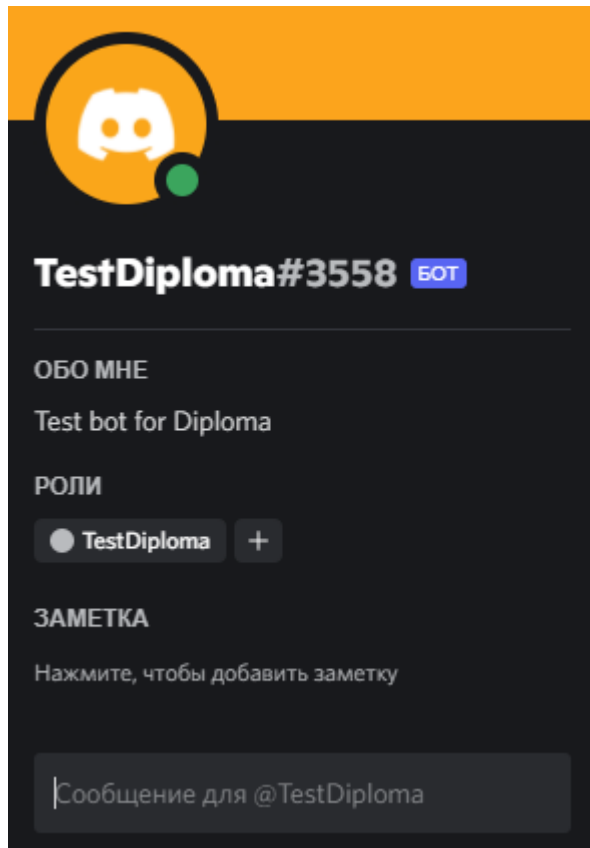


Рисунок 27 – Созданный бот

После успешного запуска (Gateway – Ready), можно перейти на канал в Сервере Discord и проверять бота.

2.4 Результаты работы приложения

После добавления бота на сервер, для познания команд можно использовать команду! Start, вывод данной команды представлен на Рисунок 28

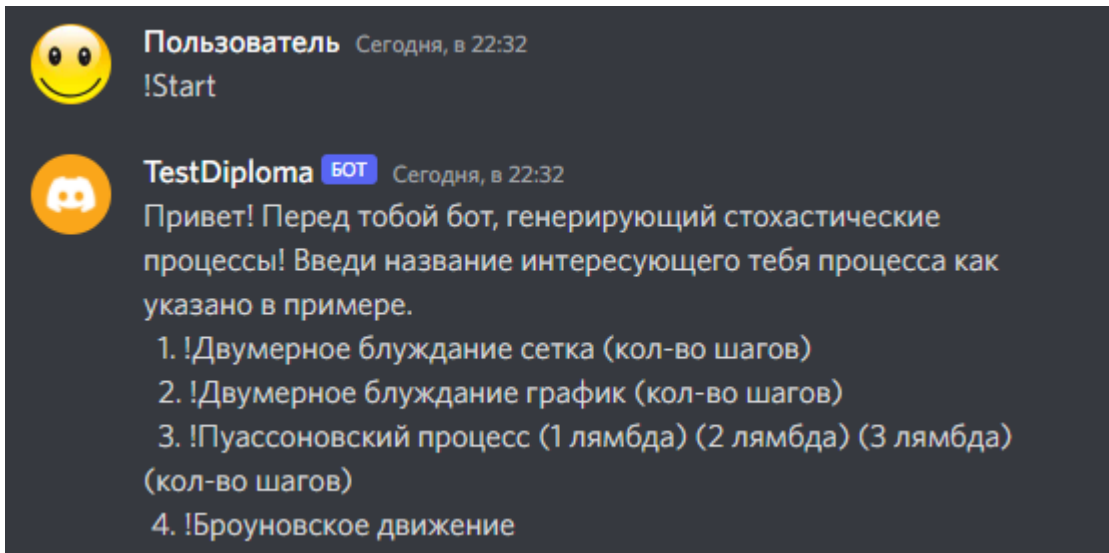


Рисунок 28 – Список команд

Выбирается нужная команда, и вводится в чат, результат будет как на Рисунок 29.

Для начала будет рассмотрено случайное блуждание на сетке (См. Рисунок 30):

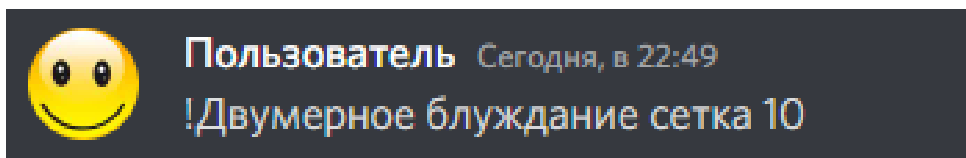


Рисунок 29 – Команда двумерного блуждания на сетке

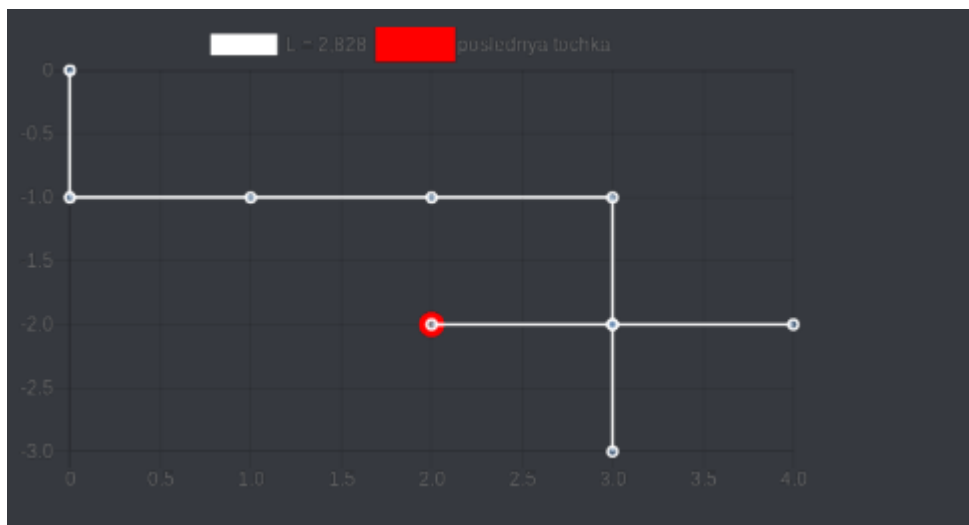


Рисунок 30 – Двумерное блуждание на сетке

Далее идет Пуассоновский процесс (См. Рисунок 31, Рисунок 32, Рисунок 33).



Рисунок 31 – Функция распределения

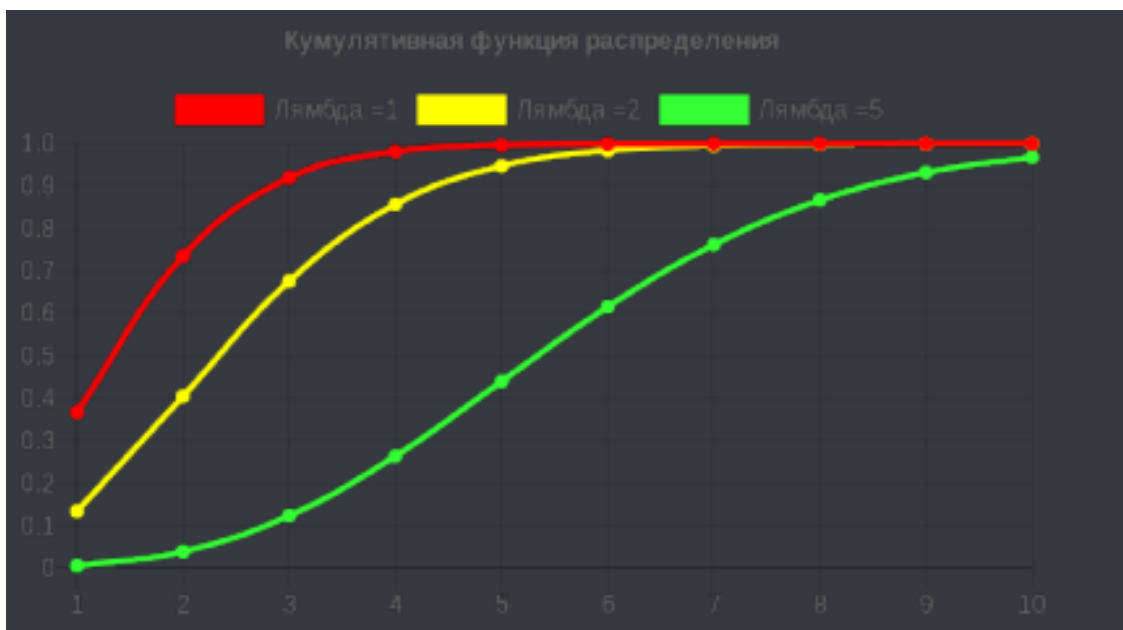


Рисунок 32 – Кумулятивная функция распределения

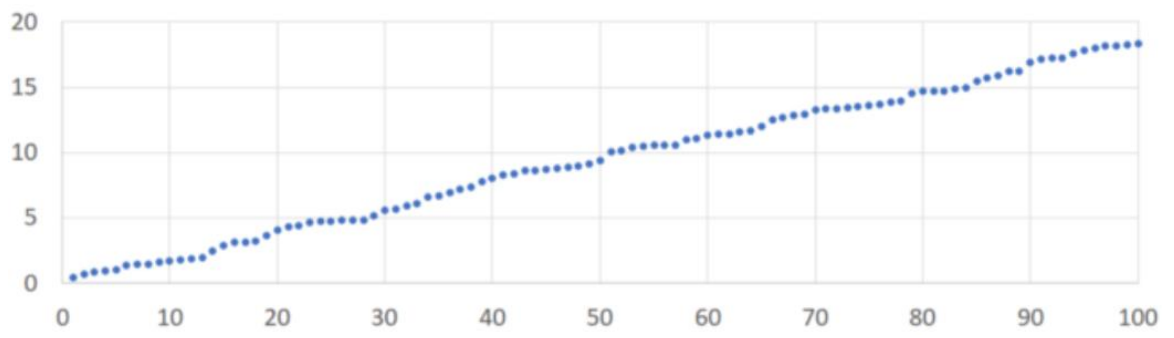


Рисунок 33 – Пуассоновский стохастический процесс

ЗАКЛЮЧЕНИЕ

В рамках дипломной квалификации была изучена теория случайных процессов: классификация, типы, формулы и различные математические примеры. На основе полученной информации были созданы и изучены необходимые графики и предложена практическая реализация в качестве помощника в формате бота Discord. Все задачи, поставленные для достижения этой цели, были решены, а именно:

- рассмотрена платформа Discord;
- изучена фундаментальная теория случайных процессов;
- было разработано автоматизированное приложение, которое генерирует стохастические процессы;
- была созданы и продемонстрированы необходимые графики;
- приложение доступно для всех пользователей Интернета;
- можно добавлять новые функции в неограниченном количестве;
- приложение было протестировано и работает без сбоев.

Такой помощник поможет любому, кто использует Интернет для изучения теории случайных процессов, и поможет вам создать примеры, необходимые для этого.

Это приложение актуально, потому что оно упрощает изучение и получение необходимой информации о случайных процессах и остается уникальным по сей день.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Волков, И. К. Случайные процессы / И. К. Волков, С. М. Зуев – М.: Периодика, Москва, 1999. – 265 с.
2. Кузнецов, Д. Ф. Численное моделирование стохастических дифференциальных уравнений и стохастических интегралов – Москва, 1999. – 157 с.
3. Винер, Н. Нелинейные задачи в теории случайных процессов; Архангельск: Северо-Западное книжное изд-во – Москва, 2004. – 678 с.
4. Колмогоров, А. Н. Основные понятия теории вероятностей. – М.: Наука, 1974. – 120 с.
5. Вентцель, Е. С. Теория случайных процессов и ее инженерные приложения / Е. С. Вентцель, Л. А. Овчаров – М.: Мнемозина, Москва, 2000. – 251 с.
6. Феллер, В. Введение в теорию вероятностей и ее приложения. – М.: Мир, 1984. – 528 с.
7. Боровков, А.А. Теория вероятностей. – М.: Наука, 1986. – 432 с.
8. Гнеденко, Б.В. Курс теории вероятностей. – М.: Наука, 1988. – 451 с.
9. Гардинер, К.В. Стохастические модели в естественных науках. – М.: Мир, 1986. – 591 с.
10. Полное руководство по языку программирования C# 7.0 и платформе .NET 4.7 URL: <http://metanit.com/sharp/tutorial/> (дата обращения 10.06.2022)
11. Discord.Net Doc URL: <https://discord.com/developers/docs/intro> (дата обращения 10.06.2022)
12. Stack Overflow URL: <https://stackoverflow.com/> (дата обращения 10.06.2022)

СПРАВКА

Кубанский Государственный университет

о результатах проверки текстового документа
на наличие заимствований

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Макаров Роман Александрович
Самоцитирование
рассчитано для: Макаров Роман Александрович
Название работы: РЕАЛИЗАЦИЯ ПОМОЩНИКА ДЛЯ ГЕНЕРАЦИИ СТОХАСТИЧЕСКИХ ПРОЦЕССОВ В ФОРМЕ DISCORD БОТА
Тип работы: Выпускная квалификационная работа
Подразделение: ФКТИПМ, кафедра анализа данных и искусственного интеллекта

РЕЗУЛЬТАТЫ

■ ОТЧЕТ О ПРОВЕРКЕ КОРРЕКТИРОВАЛСЯ: НИЖЕ ПРЕДСТАВЛЕНЫ РЕЗУЛЬТАТЫ ПРОВЕРКИ ДО КОРРЕКТИРОВКИ

ЗАИМСТВОВАНИЯ	22.13%	ЗАИМСТВОВАНИЯ	22.13%
ОРИГИНАЛЬНОСТЬ	73.15%	ОРИГИНАЛЬНОСТЬ	73.15%
ЦИТИРОВАНИЯ	4.72%	ЦИТИРОВАНИЯ	4.72%
САМОЦИТИРОВАНИЯ	0%	САМОЦИТИРОВАНИЯ	0%

ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 22.06.2022

ДАТА И ВРЕМЯ КОРРЕКТИРОВКИ: 22.06.2022 20:39

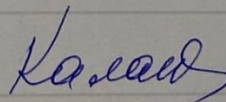
Модули поиска: ИПС Адилет; Библиография; Сводная коллекция ЭБС; Интернет Плюс; Сводная коллекция РГБ; Цитирование; Переводные заимствования (RuEn); Переводные заимствования по eLIBRARY.RU (EnRu); Переводные заимствования по Интернету (EnRu); Переводные заимствования издательства Wiley (RuEn); eLIBRARY.RU; СПС ГАРАНТ; Модуль поиска "КубГУ"; Медицина; Диссертации НББ; Перефразирования по eLIBRARY.RU; Перефразирования по Интернету; Перефразирования по коллекции издательства Wiley; Патенты СССР, РФ, СНГ; СМИ России и СНГ; Шаблонные фразы; Кольцо вузов; Издательство Wiley; Переводные заимствования

Работу проверил: Калайдина Галина Вениаминовна

ФИО проверяющего

Дата подписи:

23.06.2022



Подпись проверяющего



Чтобы убедиться
в подлинности справки, используйте QR-код,
который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование
корректным, система оставляет на усмотрение проверяющего.
Предоставленная информация не подлежит использованию
в коммерческих целях.

ОТЗЫВ

на бакалаврскую дипломную работу студента
ФГБОУ ВО «КубГУ» Корнийченко М.М. на тему:
«Разработка чат-бота для обучения backend разработчиков»

С развитием средств коммуникации наметился повышенный интерес к различным способам дистанционного обучения и к интернет-образованию. На рынке существует множество продуктов, которые позволяют организовать такой формат обучения – начиная электронной почтой и заканчивая Web конференциями, онлайн-курсами, и даже чат-ботами.

В связи с этим, тема дипломной работы, направленной на разработку чат-бота для обучения backend разработчиков, является актуальной.

Дипломная работа состоит из трех глав, введения, заключения и списка использованной литературы.

В главе 1 были рассмотрены основные понятия чат-ботов и рынок мессенджеров.

Глава 2 посвящена сравнительному анализу существующих подобных продуктов, созданию модели и проектированию приложения. Также в главе 2 рассмотрены инструменты разработки.

В главе 3 описан процесс разработки приложения, продемонстрирована работа приложения.

Таким образом, цель, поставленная в работе, достигнута, представлен необходимый иллюстративный материал, имеются ссылки на использованные источники. Результаты дипломной работы имеют практическую значимость.

Дипломная работа «Разработка чат-бота для обучения backend разработчиков» выполнена на высоком уровне и заслуживает оценки «отлично».

Научный руководитель дипломной работы,
доцент кафедры анализа данных и
искусственного интеллекта КубГУ



Г. А. Кесиян