МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение информационных систем и технологий»

Курсовой проект

по дисциплине «Оптимизация проектных решений»

Тема: «Программная поддержка обработки результатов тестирования по учебным дисциплинам»

Выполнила: ст. гр. 10702416

Паулич Е.В.

Проверила: доц. Ковалёва И.Л.

Минск 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение информационных систем и технологий»

Пояснительная записка

к курсовому проекту

по дисциплине «Оптимизация проектных решений»

Тема: «Программная поддержка обработки результатов тестирования по учебным дисциплинам»

Выполнила: ст. гр. 10702416

Паулич Е.В.

Проверила: доц. Ковалёва И.Л.

Минск 2019

Содержание

[Введение 4](#_Toc27576045)

[1 Описание проблемной области 6](#_Toc27576046)

[1.1 Анализ аналогов систем тестирования 6](#_Toc27576047)

[1.2 Подходы к обработке результатов тестирования 10](#_Toc27576048)

[2 Постановка задач 12](#_Toc27576049)

[3 Адаптация алгоритмов и подходов 14](#_Toc27576050)

[4 Программная реализация 18](#_Toc27576051)

[4.1 Схема ПО 19](#_Toc27576052)

[4.2 Описание классов, методов, функций 20](#_Toc27576053)

[4.3 Тестовый пример и руководство пользователя 22](#_Toc27576054)

[Заключение 27](#_Toc27576055)

[Используемая литература 28](#_Toc27576056)

[Приложение 29](#_Toc27576057)

# Введение

Для подведения итогов подготовки, обучения, аттестации, сертификации учащихся, сотрудников существуют различные способы: контрольные работы, устные опросы, экзамены, тесты и т.п. Наиболее малозатратным по времени и удобным в использовании является тестирование. Тестирование является широко распространенным во всех сферах деятельности: в школах, больницах, на различного рода предприятиях и т.д.

Тестирование – это стандартизированный метод оценки знаний, умений, навыков, который помогает выявить пробелы в подготовке, показать сильные стороны.

Путь к созданию тестов – изучение теории и методики тестового контроля знаний. Здесь самое главное – осознать, что тест – это не просто проба или проверка с помощью традиционных вопросов, а научно-обоснованный метод, представляющий систему заданий специфической формы, возрастающей трудности, определенного содержания, позволяющий качественно оценить структуру знаний и эффективно измерить их уровень.

По средствам предъявления существуют тесты:

* с использованием тестовых тетрадей, в которых находятся тестовые задания и в которых испытуемый фиксирует результаты;
* бланковые, когда испытуемые отмечают или вписывают правильные ответы на специальных бланках. Бланки предъявляются отдельно от заданий:
* компьютерные.

В процессе выполнения курсового проекта будут созданы компьютерные онлайн-тесты.

Немаловажным пунктом в тестирование является обработка результатов пройденных тестов. Обработка результатов тестирования может проводиться различными способами. В курсовом проекте это происходит с использованием нейронной сети.

В первом разделе рассмотрим существующие аналоги систем тестирования, их основные возможности, достоинства и недостатки, а также различные подходы к обработке результатов тестирования.

Во втором разделе обоснуем актуальность выбранной темы и поставим задачи, которые должны быть выполнены в курсовом проекте.

В третьем разделе выполнен анализ алгоритма работы нейронной сети Кохонена и проведем адаптацию данного алгоритма для решения поставленных задач.

В четвертом разделе представлена программая реализация данного алгоритма, привено описание используемых классов и методов, а также показан тестовый пример работы алгоритма и руководство пользователя.

# 1 Описание проблемной области

В современном мире для проверки знаний часто используют тестирование, так как оно позволяет довольно быстро проверить знания и умения людей. Обработка полученных результатов, позволяет выявить слабые и сильные стороны тестируемого. Этап обработки результатов тестирования является одним из важнейших этапов, так как с помощью полученных данных можно не только выделить тех, кто справился лучше или хуже, но также и использовать эти данные для улучшения показателей тестируемых, например, давать дополнительную литературу, более тщательно рассматривать темы, которые вызывают затруднение.

## 1.1 Анализ аналогов систем тестирования

На сегодняшний день существует множество программ для тестирования, причем как бесплатных, так и платных. К ним, например, относятся TestTurn, SunRav TestOfficePro, Айрен, Indigo.

TestTurn-  – это простая и очень удобная в применении бесплатная программа для проведения тестирования от VeralSoft. Интерфейс системы представлен на Рис 1.1

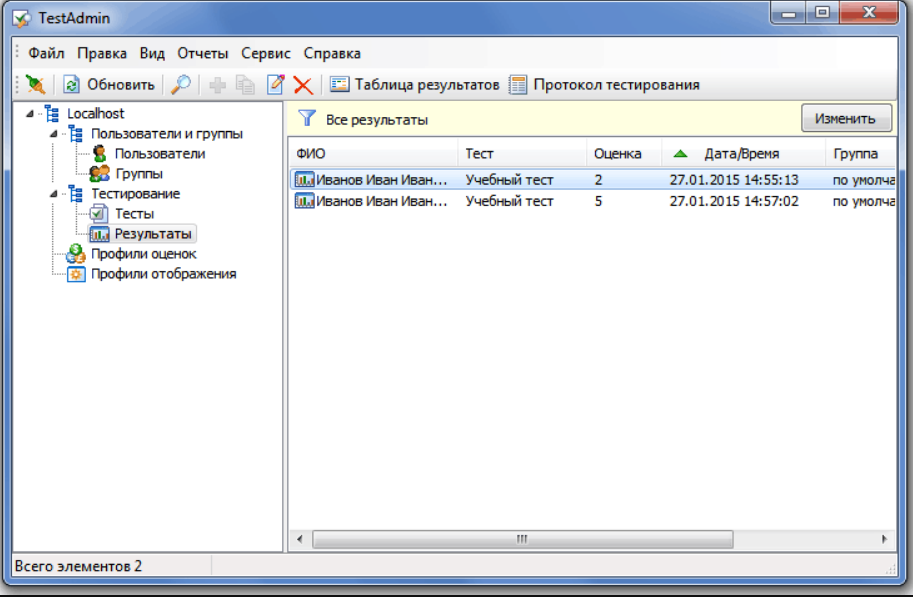
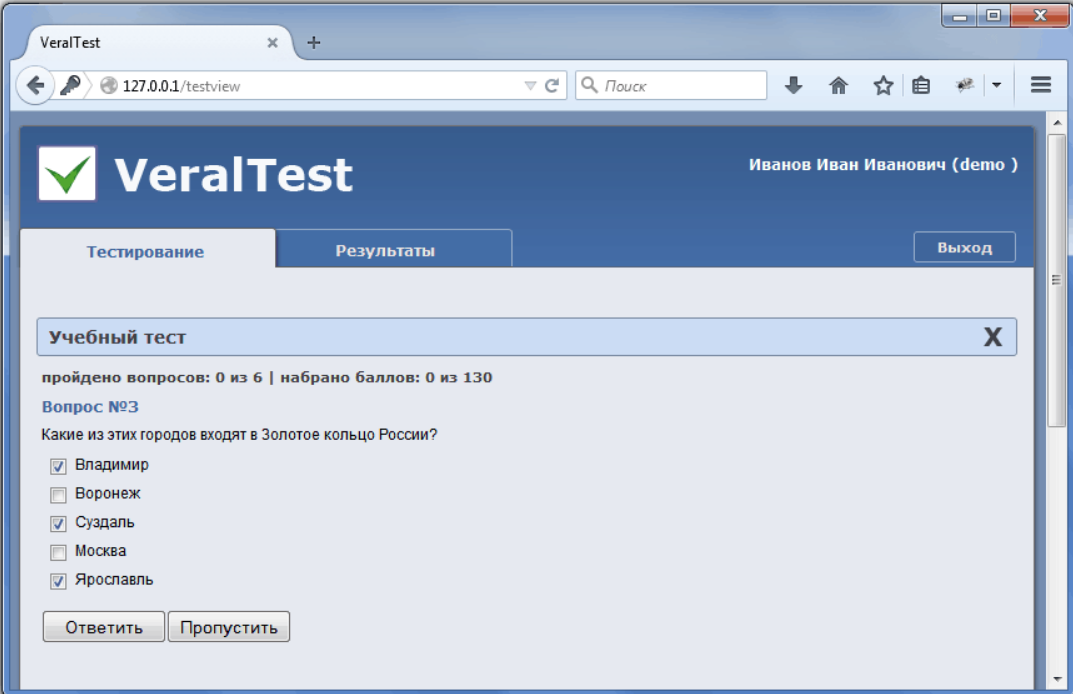


Рис 1.1 Интерфейс программы VeralSoft

Основные возможности:

* Простота и скорость развертывания. Все, что нужно для работы сервера тестирования уже присутствует в дистрибутиве. Никаких дополнительных программ вроде сервера баз данных или web сервера не требуется. Развернуть систему тестирования в компьютерном классе или локальной сети предприятия можно буквально на несколько минут.
* Многопользовательский режим. Проходить тестирование может неограниченное количество пользователей одновременно. В то же время, программа не требовательна к системным ресурсам и нормально работает на любом современном компьютере
* Самостоятельная регистрация пользователей. В программе предусмотрена возможность самостоятельной регистрации пользователей при первом входе в систему (если разрешено в настройках). Эта функция избавляет администратора, от рутинной процедуры ручного ведения списков тестируемых.
* Индивидуальный набор тестов. Тесты могут быть назначены как всем пользователям одновременно, так и каждому пользователю или группе индивидуально.
* Гибкие настройки тестирования. Для каждого назначенного теста можно установить собственные параметры выполнения, такие как: время тестирования, количество попыток прохождения, стиль отображения, шкалу оценок
* Подробные результаты тестирования. Для каждого выполненного теста сохраняется подробный протокол выполнения, в котором перечислены все заданные вопросы и ответы тестируемого.
* Возможность создания собственных отчетов. Встроенный в TestAdmin генератор отчетов позволяет в визуальном интерактивном режиме как редактировать существующие отчеты, так и создавать новые, используя любые данные из базы данных сервера тестирования.

Результаты тестирования обрабатываются путем сравнения ответа экзаменующегося с эталоном, хранящимся в памяти программы, при этом есть возможность указать способ расчета оценки, например "зачет/незачет" или по пятибалльной шкале.

SunRav TestOfficePro является платной программой. В пакет SunRav TestOfficePro входят программы для создания тестов, проведения тестирования и обработки результатов тестирования. С его помощью можно организовать и провести тестирования и экзамены в образовательных учреждениях (вузы, колледжи, школы), а предприятия и организации могут осуществлять аттестацию и сертификацию своих сотрудников. Пакет позволяет создавать тесты по предметам школьной программы, вузовским дисциплинам, тесты для профессионального тестирования, психологические тесты и т.д. Каждый вопрос может быть снабжен комментарием, содержащим информацию о правильном ответе и т.п. Пример некоторой функциональности системы изображен на Рис 1.2

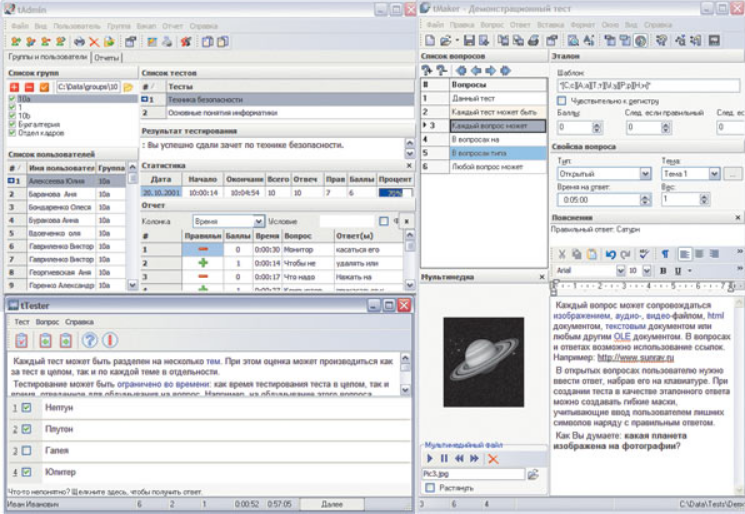


Рис 1.2 Интерфейс программы SunRav TestOfficePro

Обработка результатов тестирования выполняется с помощью сравнения ответа экзаменующегося с эталоном, хранящимся в памяти программы. Также есть возможность создания матрицы ответов.

Айрен — это бесплатная программа, позволяющая создавать тесты для проверки знаний и проводить тестирование в локальной сети, через интернет или на одиночных компьютерах. Тесты могут включать в себя задания различных типов: с выбором одного или нескольких верных ответов, с вводом ответа с клавиатуры, на установление соответствия, на упорядочение и на классификацию. При сетевом тестировании преподаватель видит на своем компьютере подробные сведения об успехах каждого из учащихся. По окончании работы эти данные сохраняются в архиве, где их в дальнейшем можно просматривать и анализировать с помощью встроенных в программу средств. Интерфейс системы является простым и удобным в использовании (Рис 1.3)

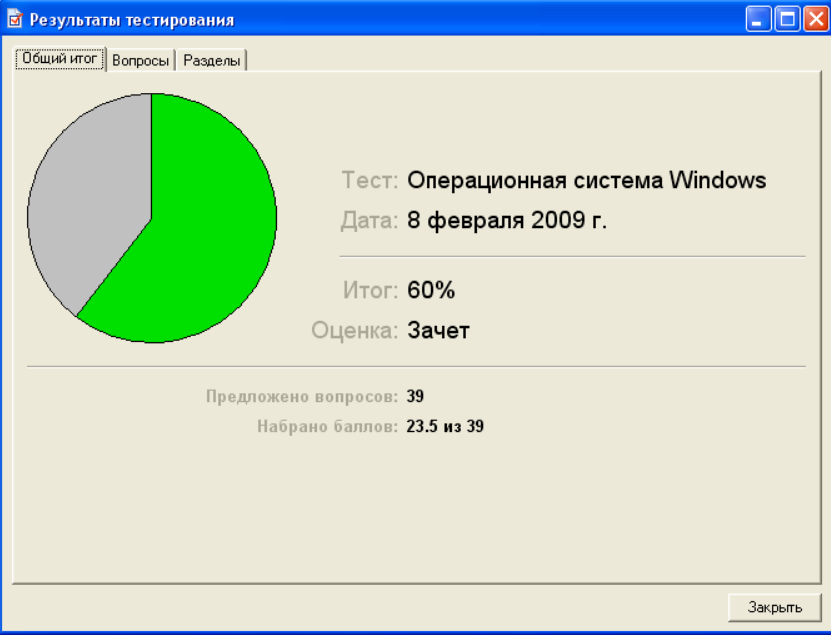
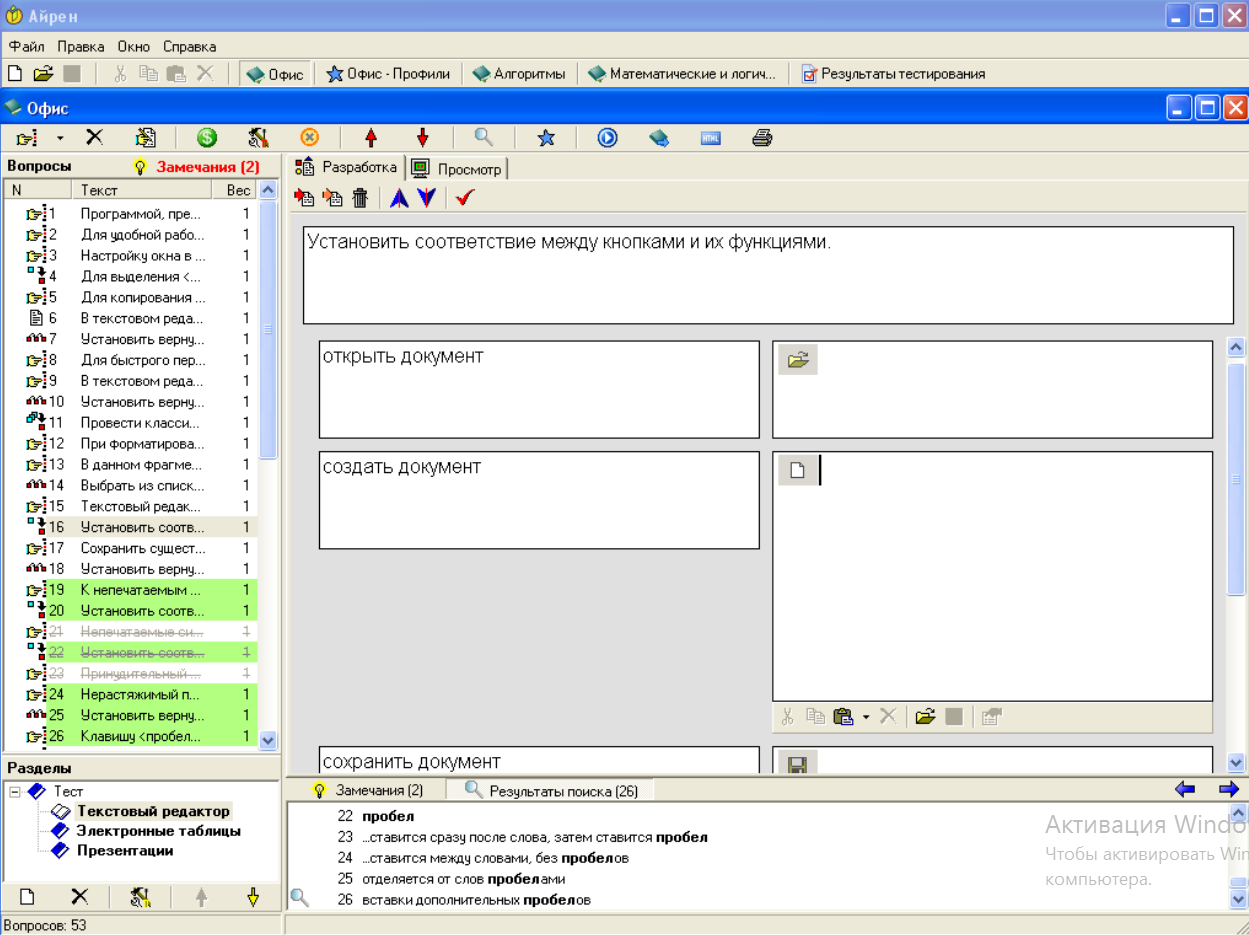


Рис 1.3 Интерфейс программы Айрен

Обработка результатов тестирования происходит в подсчете % правильных вариантов ответов на языке программирования Pascal.

Indigo  – это профессиональный инструмент автоматизации процесса тестирования и обработки результатов, который предназначен для решения широкого спектра задач, некоторые из них представлены на Рис 1.4:

* Тестирование и контроль знаний учащихся.
* Определение профессионального уровня сотрудников.
* Проведение психологического тестирования.
* Проведение опросов.
* Организация олимпиад и конкурсов.

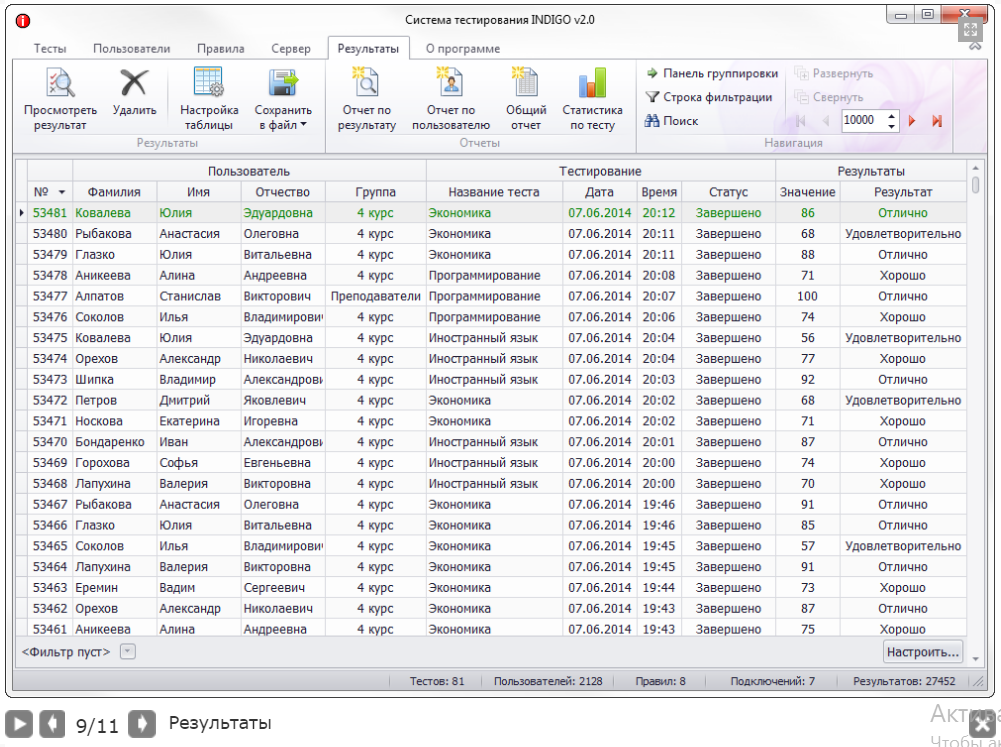
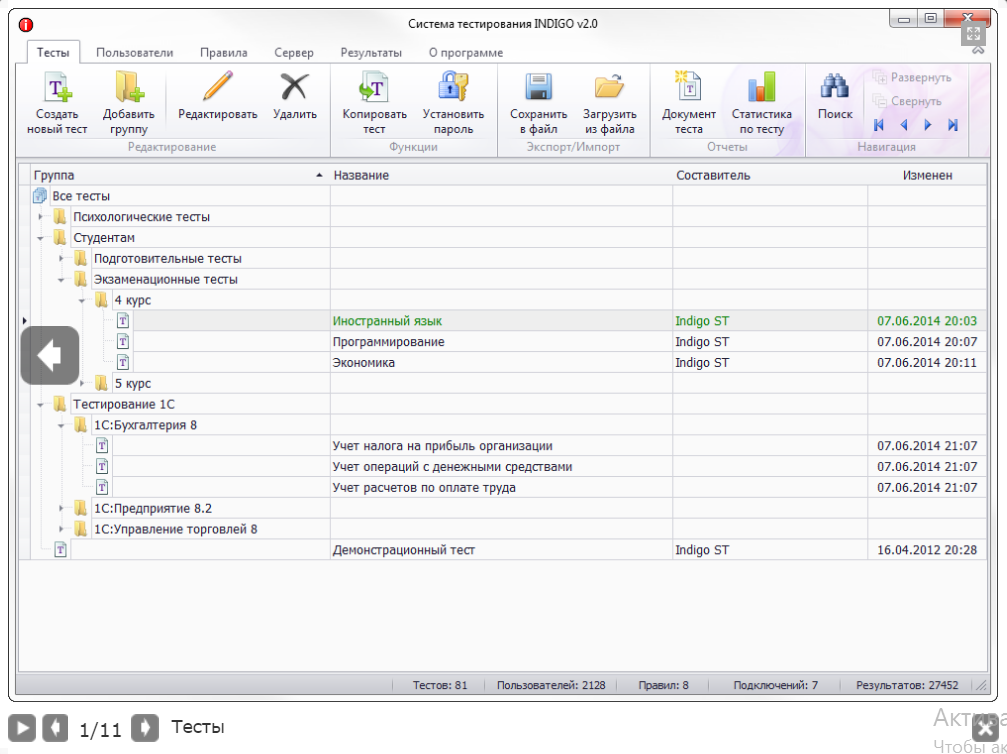


Рис 1.4 Интерфейс программы Indigo

Обработка результатов тестирования путем сравнения ответа экзаменующегося с эталоном, хранящимся в памяти программы. Оценка задается формулой:

Оценка = round(Балл(Корневая\_группа) / МаксБалл(Корневая\_группа) \* 100).

У каждой системы есть свои плюсы и минусы. Каждая из них, в зависимости от их функциональных задач использует определенный подход к подведению итогов тестирования. В большинстве случаев это сравнение полученных данных с данными, которые есть в системе.

1.2 Подходы к обработке результатов тестирования

Существуют различные подходы к обработке результатов тестирования. Перед обработкой результатов тестирования могут проводится также оценки заданий: математические и экспертные. Использование математической статистики является одним из эффективных средств познания объективных законов обучения, воспитания и развития. Экспертные методы позволяют не только оценить, в какой степени они измеряют именно те знания, умения, навыки, для которых разрабатывается тест, но и оценить задания с точки зрения их формы. Для интерпретации результатов тестирования может использоваться нормативно-ориентированный или критериально-ориентированный подходы. Первый позволяет сравнить учебные достижения отдельных учащихся друг с другом, а второй позволяет оценить, в какой степени учащиеся овладели необходимым учебным материалом.

В связи с развитием технологий и искусственного интеллекта, выигрышным по производительности и актуальным будет вариант использования нейронных сетей для получения результатов тестирования. На сегодняшнее время существует большое количество различных видов нейронных сетей, например, **нейронная сеть Хопфилда**, Кохонена, перцептрон Розенблатта.

Нейронная сеть Хопфилда (Hopfield network, HN) — это полносвязная нейронная сеть с симметричной матрицей связей. Во время получения входных данных каждый узел является входом, в процессе обучения он становится скрытым, а затем становится выходом. Сеть обучается так: значения нейронов устанавливаются в соответствии с желаемым шаблоном, после чего вычисляются веса, которые в дальнейшем не меняются. После того, как сеть обучилась на одном или нескольких шаблонах, она всегда будет сводиться к одному из них (но не всегда — к желаемому). Она стабилизируется в зависимости от общей “энергии” и “температуры” сети. У каждого нейрона есть свой порог активации, зависящий от температуры, при прохождении которого нейрон принимает одно из двух значений (обычно -1 или 1, иногда 0 или 1).  Такая сеть часто называется сетью с ассоциативной памятью; как человек, видя половину таблицы, может представить вторую половину таблицы, так и эта сеть, получая таблицу, наполовину зашумленную, восстанавливает её до полной.

Элементарный перцептрон состоит из элементов трёх типов: S-элементов, A-элементов и *одного* R-элемента. S-элементы — это слой сенсоров или рецепторов. В физическом воплощении они соответствуют, например, светочувствительным клеткам [сетчатки глаза](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D1%87%D0%B0%D1%82%D0%BA%D0%B0) или [фоторезисторам](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%82%D0%BE%D1%80%D0%B5%D0%B7%D0%B8%D1%81%D1%82%D0%BE%D1%80) матрицы камеры. Каждый рецептор может находиться в одном из двух состояний — *покоя* или *возбуждения*, и только в последнем случае он передаёт единичный сигнал в следующий слой, ассоциативным элементам.

В данном курсовом проекте будет использоваться нейронная сеть Кохонена. [Нейронные сети](https://neuronus.com/stat/1271-nejronnye-seti-iskusstvennyj-intellekt.html) Кохонена типичный пример нейросетевой архитектуры, обучающейся без учителя. Они используются для кластеризации данных или прогнозирования свойств. В сети Кохонена каждый [нейрон входного](https://wiki.loginom.ru/articles/input-neuron.html) слоя связан со всеми [нейронами выходного](https://wiki.loginom.ru/articles/output-neuron.html), а внутри слоев связей нет. Как и в обычной нейронной сети, входные нейроны не участвуют в процессе [обучения](https://wiki.loginom.ru/articles/training.html) и обработки данных, а просто распределяют входной сигнал по нейронам следующего слоя. Число входных нейронов равно размерности вектора признаков (т.е. числу признаков объекта).

2 Постановка задач

Современные подходы для обработки результатов тестирования позволяют учитывать накопленный опыт. Качество обработки может значительно повлиять на дальнейшую разработку и обработку тестов, поэтому тема данного курсового проекта является актуальной.

Целью данного курсового проекта является обработка результатов, полученных в результате тестирования, для улучшения их показателей.

Для выполнения поставленной цели необходимо решить следующие задачи:

* Выполнить адаптацию алгоритмов и подходов
* Разработать программу для реализации алгоритмов
  + Разработать и описать схему ПО
  + Разработать и описать классы, методы, функции
  + Разработать и описать тестовые примеры и руководство пользователя

Обработка результатов будет производиться по результатам статистики о прохождении тестов пользователями(Рис 2.1).

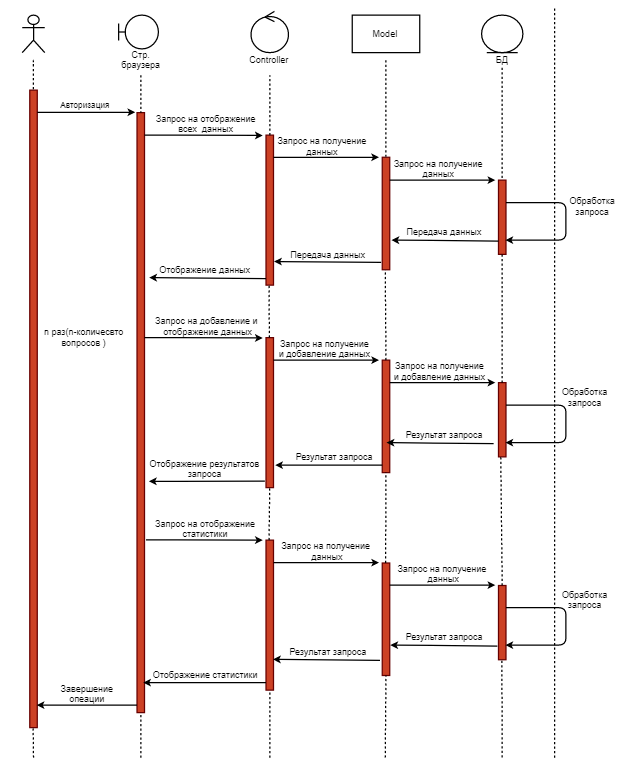


Рис 2.1 Диаграмма последовательности процесса прохождения тестов и получения статистики

# 3 Адаптация алгоритмов и подходов

[Нейронные сети](https://neuronus.com/stat/1271-nejronnye-seti-iskusstvennyj-intellekt.html)**Кохонена** типичный пример нейросетевой архитектуры, обучающейся без учителя. Отсюда и перечень решаемых ими задач: кластеризация данных или прогнозирование свойств. Кроме того, сети Кохонена могут использоваться с целью уменьшения размерности данных с минимальной потерей информации.

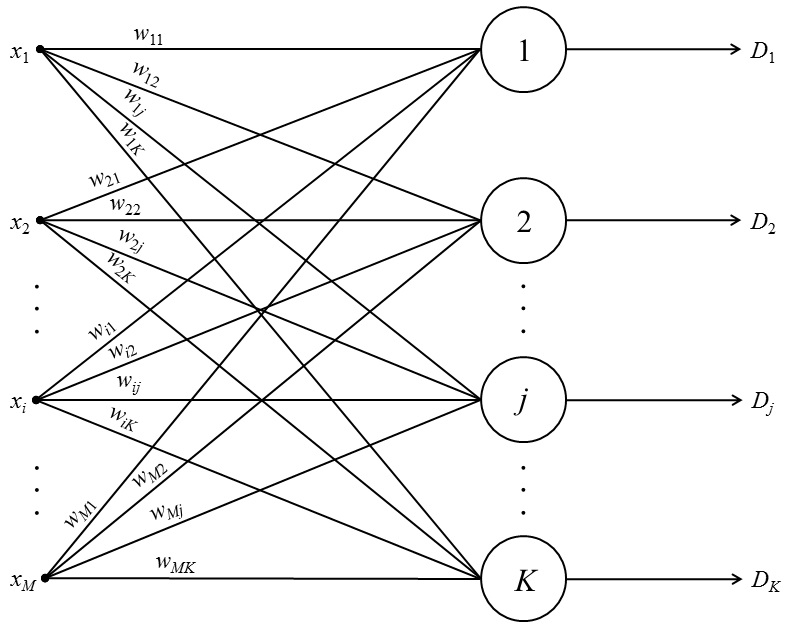
В нейронных сетях Кохонена выходные вектора в обучающей выборке могут быть, но могут и отсутствовать, и, в любом случае, они не принимают участия в процессе обучения. То есть выходы не используются в качестве ориентиров при коррекции синапсов. Именно поэтому данный принцип настройки нейронной сети называется самообучением.

В рассматриваемой архитектуре сигнал распространяется от входов к выходам в прямом направлении. Структура нейронной сети содержит единственный слой нейронов (слой Кохонена) без коэффициентов смещения (рис. 1). Общее количество весовых коэффициентов рассчитывается как произведение:

Общее количество весовых коэффициентов сети Кохонена

Количество нейронов равно количеству кластеров, среди которых происходит начальное распределение и последующее перераспределение обучающих примеров. Количество входных переменных нейронной сети равно числу признаков, характеризующих объект исследования и на основе которых происходит отнесение его к одному из кластеров.

Следует различать собственно самообучение и самоорганизацию нейронной сети Кохонена. При обычном **самообучении** сеть имеет строго фиксированную структуру, т. е. количество нейронов, не изменяющееся на протяжении всего жизненного цикла. При **самоорганизации** сеть, напротив, не имеет постоянной структуры. В зависимости от найденного расстояния до нейрона-победителя либо этот нейрон используется для кластеризации примера, либо для поданного на входы примера создается новый кластер с соответствующими ему весовыми коэффициентами. Кроме того, в процессе самоорганизации структуры сети Кохонена отдельные нейроны могут исключаться из нее. Общая структура нейронной сети Кохонена показана ниже(Рис 3.1)



**Рис 3.1** Общая структура нейронной сети Кохонена

Нормализация входных переменных выполняется в пределах [–1, 1] или [0, 1].

Для жизненного цикла нейронных сетей данной архитектуры характерны три основных стадии жизненного цикла: обучение, кластерный анализ и практическое использование.

Алгоритм **обучения**сети Кохонена включает этапы, состав которых зависит от типа структуры: постоянной (самообучающаяся сеть) или переменной (самоорганизующаяся сеть).

Для **самообучения**последовательно выполняются:

1. Задание структуры сети (количества нейронов слоя Кохонена) (K).

2. Случайная инициализация весовых коэффициентов значениями, удовлетворяющими одному из следующих ограничений:

– при нормализации исходной выборки в пределах [0, 1]:

https://neuronus.com/images/theory/ins/05062015/04.jpg,

где M – количество входных переменных сети – характеристических признаков объекта исследования.

В качестве признаков могут выступать: количественные, качественные и классификационные. Значение количественного признака соответствует величине какого-то свойства (например, 1,5,100). Качественные признаки никак не связаны с величиной какого-то свойства. Их труднее всего обрабатывать. Чтобы использовать их в распознавании, их приводят к количественным. К примеру, есть варианты да, нет, не знаю, результат их приведения к качественным – 1, 0, 0.5 соответственно.

3. Подача на входы сети случайного обучающего примера текущей эпохи обучения и расчет евклидовых расстояний от входного вектора до центров всех кластеров:

https://neuronus.com/images/theory/ins/05062015/05.jpg.

4. По наименьшему из значений Rj выбирается нейрон-победитель j, в наибольшей степени близкий по значениям с входным вектором. Для выбранного нейрона (и только для него) выполняется коррекция весовых коэффициентов:

https://neuronus.com/images/theory/ins/05062015/06.jpg,

где v – коэффициент скорости обучения.

5. Цикл повторяется с шага 3 до выполнения одного или нескольких условий окончания:

– исчерпано заданное предельное количество эпох обучения;

– не произошло значимого изменения весовых коэффициентов в пределах заданной точности на протяжении последней эпохи обучения;

– исчерпано заданное предельное физическое время обучения.

Коэффициент скорости обучения может задаваться постоянным из пределов (0, 1] или переменным значением, постепенно уменьшающимся от эпохи к эпохе.

К обученной нейронной сети применяется процедура **кластерного анализа** – процедуры описания свойств кластера на основе анализа количественного и качественного составов примеров, сформировавших его. Следует учитывать, что описание кластеров может базироваться не только на значениях входных переменных обучающей выборки, но и на значениях переменных, не участвовавших в формировании кластеров. В частности, в описание могут входить данные о средних значениях таких переменных среди всех примеров, сформировавших кластер. Кроме того, целесообразно для каждого кластера иметь данные о среднеквадратическом отклонении или дисперсии по каждой переменной.

При **практическом использовании** нейронной сети Кохонена новый пример подается на ее вход и относится к одному из существующих кластеров, либо делается вывод о невозможности такого отнесения (при большом расстоянии до центра ближайшего кластера). Если выбор кластера состоялся, его описание, полученное в результате кластерного анализа, и соответствующие кластеру решения должны распространяться в том числе на поданный пример.

# 4 Программная реализация

Для создания нейронной сети Кохонена для обработки результатов тестирования будут использоваться следующие технологии:

* Java – основной язык, используемый для написания программы. Java является строго типизированным объектно-ориентированным языком программирования, разработанным компанией Sun Microsystems.
* IntelliJ IDEA  – [интегрированная среда разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) для многих языков программирования, в частности [Java](https://ru.wikipedia.org/wiki/Java), [JavaScript](https://ru.wikipedia.org/wiki/JavaScript), [Python](https://ru.wikipedia.org/wiki/Python), разработанная компанией [JetBrains](https://ru.wikipedia.org/wiki/JetBrains).
* Apache Maven — [фреймворк](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) для [автоматизации сборки](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D1%81%D0%B1%D0%BE%D1%80%D0%BA%D0%B8) проектов на основе описания их структуры в файлах на языке [POM](https://ru.wikipedia.org/w/index.php?title=POM_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%B8_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%BE%D0%B2)&action=edit&redlink=1) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [*Project Object Model*](https://en.wikipedia.org/wiki/Project_Object_Model)), являющемся подмножеством [XML](https://ru.wikipedia.org/wiki/XML)[[5]](https://ru.wikipedia.org/wiki/Apache_Maven#cite_note-5). Проект Maven издаётся сообществом Apache Software Foundation, где формально является частью [Jakarta Project](https://ru.wikipedia.org/wiki/Apache_Jakarta_Project).
* Microsoft Excel (также иногда называется [Microsoft Office](https://ru.wikipedia.org/wiki/Microsoft_Office) Excel) — [программа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) для работы с [электронными таблицами](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0), созданная корпорацией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) для [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows), [Windows NT](https://ru.wikipedia.org/wiki/Windows_NT) и [Mac OS](https://ru.wikipedia.org/wiki/Mac_OS), а также [Android](https://ru.wikipedia.org/wiki/Android), [iOS](https://ru.wikipedia.org/wiki/IOS) и [Windows Phone](https://ru.wikipedia.org/wiki/Windows_Phone). Она предоставляет возможности экономико-статистических расчетов, графические инструменты и, за исключением Excel 2008 под [Mac OS X](https://ru.wikipedia.org/wiki/Mac_OS_X), язык макропрограммирования [VBA](https://ru.wikipedia.org/wiki/VBA) (*Visual Basic for Application*). Microsoft Excel входит в состав [Microsoft Office](https://ru.wikipedia.org/wiki/Microsoft_Office) и на сегодняшний день Excel является одним из наиболее популярных приложений в мире.

## 4.1 Схема ПО

Схема программного обеспечения для обработки результатов тестирования приведена ниже(Рис 4.1)

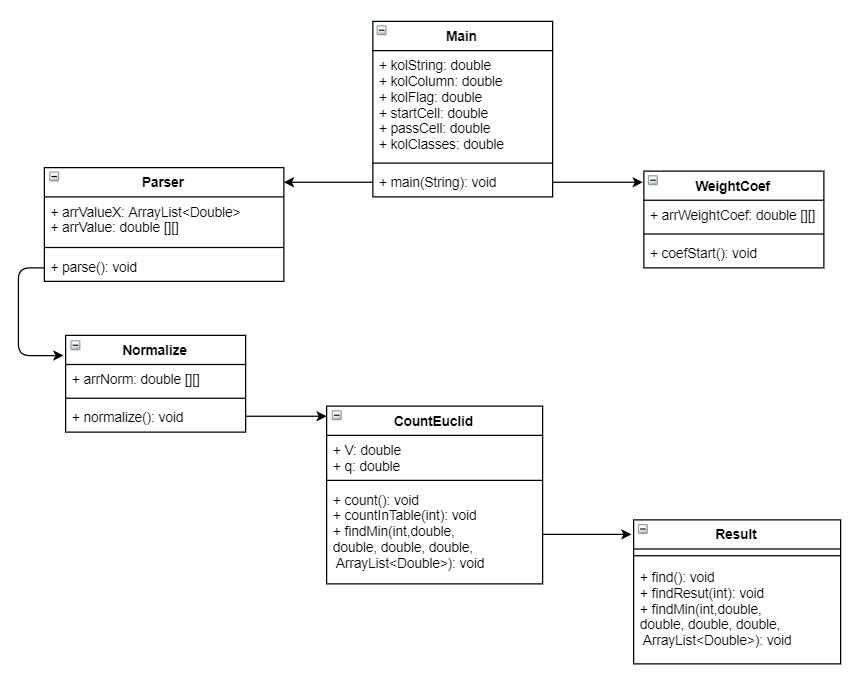


Рис 4.1 Схема ПО

## 4.2 Описание классов, методов, функций

Первым этапом для реализации нейронной сети необходимо указать параметры для считывания данных из excel файла: указать количество строк и столбцов, указать количество признаков, первоначальную ячейку, из которой будет формироваться нормализованная таблица, если необходимо, количество ячеек, которые необходимо пропустить, и количество классов(Рис 4.2)

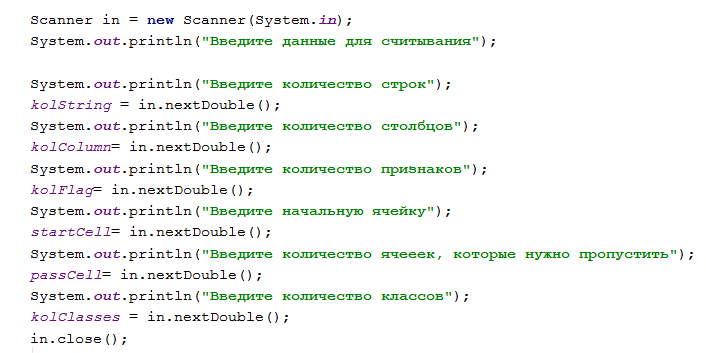


Рис 4.2 Параметры для считывания данных из excel

Для считывания данных из файла используется класс Parser, в котором реализован метод parse().

Далее нормализуем полученную таблицу при помощи метода normalize(), реализованного в классе Normalize.

В методе coefStart() класса WeightCoef реализуем рандомное заполнение весовых коэффициентов(Рис 4.2)

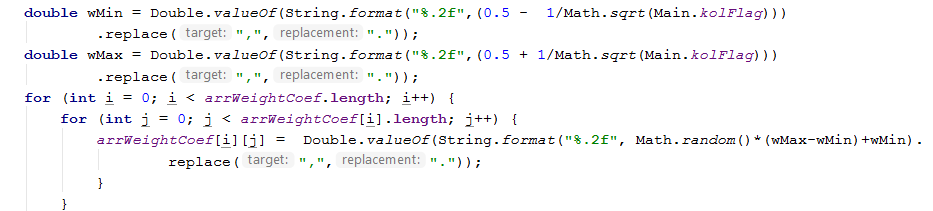


Рис 4.2 Метод coefStart() для задания весовых коэффициентов

Затем обучаем нашу сеть. Обучение происходит в методе count() класса CountEuclid. После обучения запускается метод find() в классе Result, после работы которого мы получаем результат разбития всех входных данных на три класса.

## 4.3 Тестовый пример и руководство пользователя

В данном проекте выполняется самообучение нейронной сети Кохонена для обработки результатов тестирования. Исходная выборка представлена на Рис 4.3

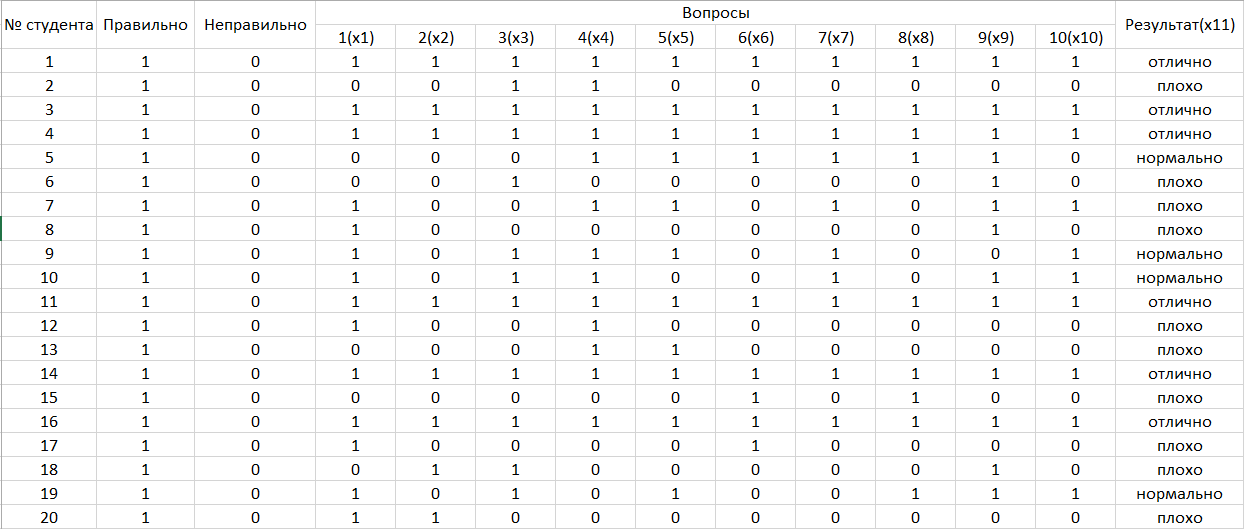


Рис 4.3 Исходная выборка данных об прохождении тестов

Данная выборка представляет собой результаты тестирование студентов одного из высших учебных заведений выполненных по дисциплине «Объектно-ориентированное программирование». Вопросы были следующие:

1. Что из перечисленного не относится к принципам ООП?
2. Выберите правильное определение сигнатуры метода.
3. Выберите правильное определение метода.
4. Может ли метод принимать разное количество параметров (аргументы переменной длины)?
5. Выберите правильное определение оператора instanceof.
6. Какие классы называются анонимными?
7. Что будет, если единственный конструктор класса объявлен как final?
8. Выберите правильное определение понятия “интерфейс”.
9. Что будет, если в static блоке кода возникнет исключительная ситуация?
10. Выберите правильное определение атрибута метода.

Каждый ответ на вопрос, может быть либо верным, либо неверным, поэтому можем точно сказать, что правильный ответ на вопрос – 1, а неправильный – 0.

Для того, чтобы начать самообучение нейронной сети, необходимо запустить приложение, выбрать нужный excel файл и ввести необходимые данные для считывания(Рис 4.4)

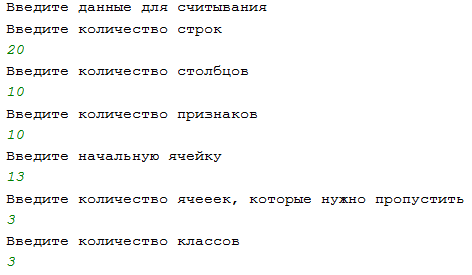


Рис 4.4 Ввод данных для считывания файла excel

Распределение примеров должно осуществляться строго по 3 кластерам. В качестве входных переменных используем x1 – x10, переменная x11 не будет использоваться для обучения, однако информация о ее значениях будет задействована в ходе кластерного анализа. Таким образом, структурно сеть будет состоять из единственного слоя нейронов, имеющего 10 входов и 3 выхода(Рис 4.5)

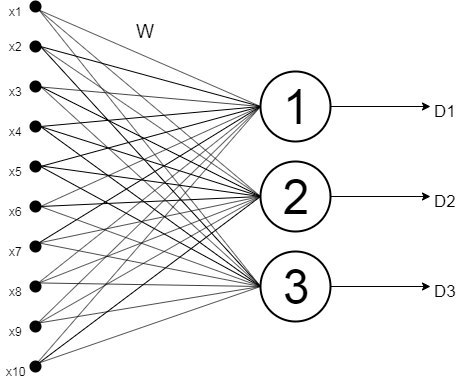


Рис 4.5  Структура нейронной сети Кохонена для кластеризации студентов

Выполним линейную нормализацию значений входных переменных выборки в пределах [0, 1] Результаты нормализации представлены на Рис 4.6

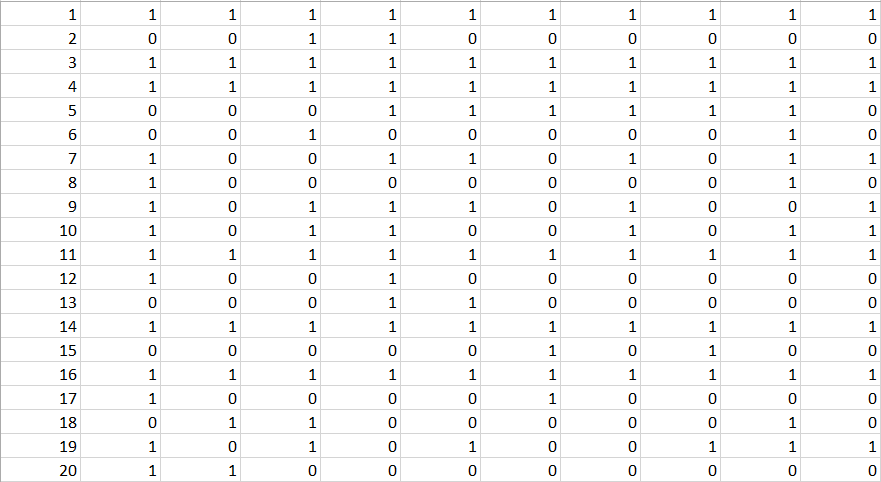


Рис 4.6 Нормализованная выборка данных об прохождении тестов

Проинициализируем все 30 весовых коэффициентов нейронной сети значениями, представленными на рисунке 4.7, с учетом ограничения.

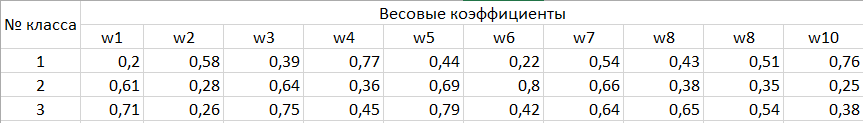


Рис 4.7 Первоначальные значения весовых коэффициентов

Алгоритм самообучения настроим следующим образом. Выберем коэффициент скорости обучения, равный 0,30, а количество эпох равно 1.

Изменяем значения весовых коэффициентов на каждом этапе в зависимости от того, какой нейрон был нейроном-победителем. Итоговые значения весовых коэффициентов приведены ниже(Рис 4.8)

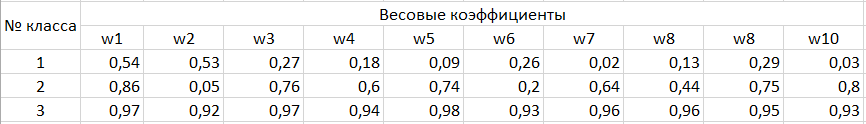


Рис 4.8 Итоговые значения весовых коэффициентов

Для обученной нейронной сети выполним процедуру кластерного анализа. Все три кластера оказались заполнены. Однако количественный состав их разный. В 1-й кластер, самый объемный, попали примеры № 2, 6, 8, 12,13, 15, 17,18,20. Во 2-м оказались примеры № 5,7, 9, 10,19. В 3-й кластер вошли примеры № 1,3,4,11,14,16. Качественные данные и описания кластеров, полученные в результате анализа, представлены на рисунке 4.9

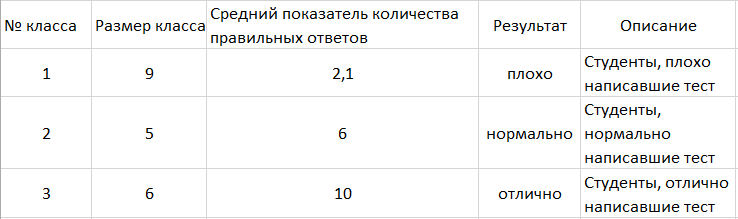


Рис 4.9 Результаты, полученные в результате работы нейронной сети

Выводы о результатах тестов в описаниях кластеров сделаны на основе анализа значений переменной *x*11, не участвовавшей в процессе обучения.

Проверим данную реализацию нейронной сети Кохонена на результатах диагностики, где исходная выборка имеет вид(Рис 4.10):

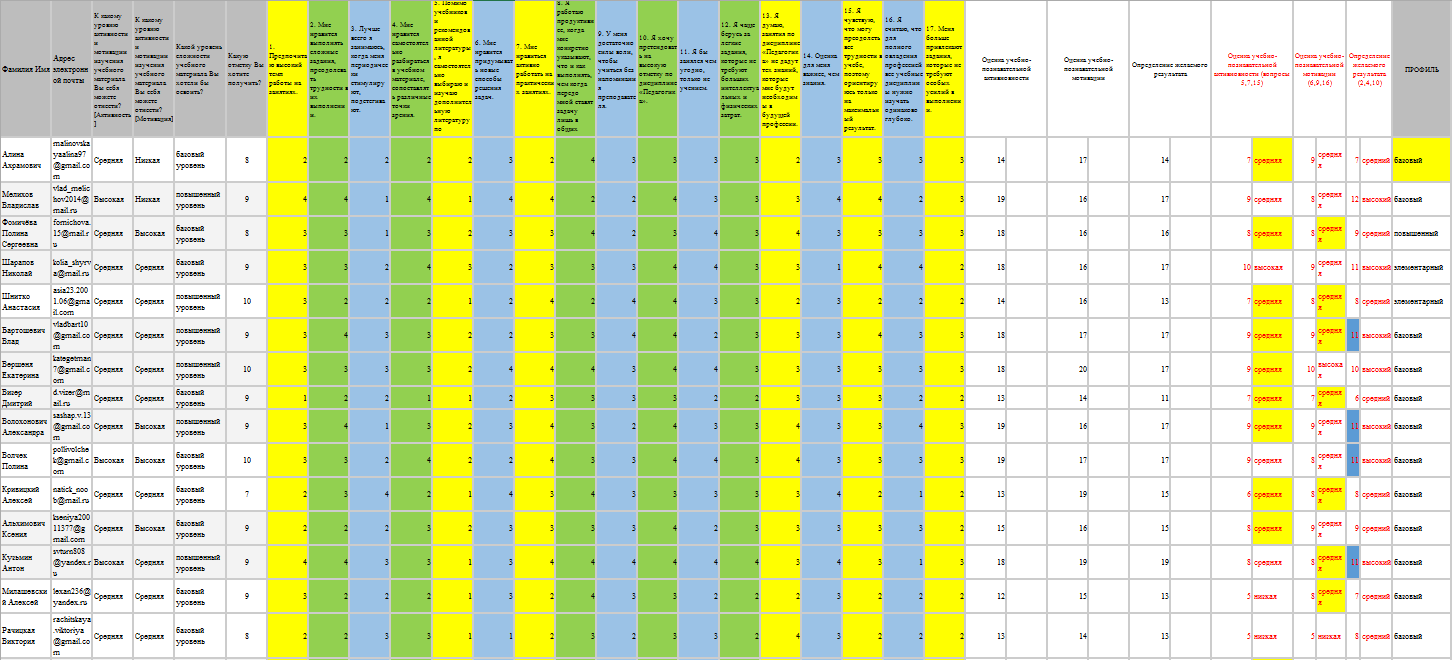


Рис 4.10 Исходная выборка данных результатов диагностики

В данной выборке имеется только три признака, которые содержатся в столбцах, выделенных красным, 56 строк, которые подвергаются нормализации и 3 класса. Проведя нормализацию данных, формирование начальных и итоговых весовых коэффициентов, получаем следующий результат, который представлен ниже (Рис 4.11)

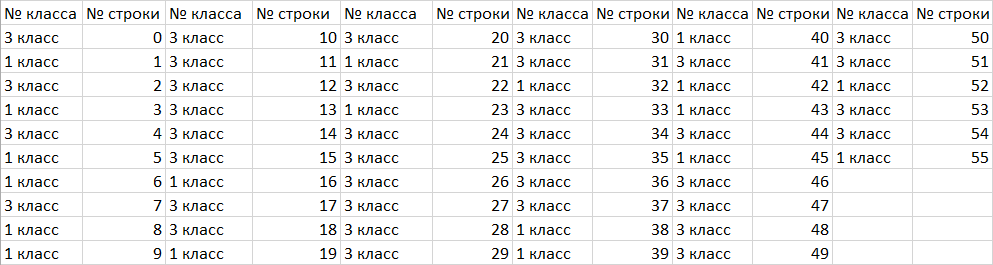


Рис 4.11 Результаты отнесения строк к различным классам

По полученным результатам можно оценить качество тестирования, так как известна класстеризация данной выборки экспертом. Полученные результаты нейронной сети и результаты эксперта совпадают, из этого следует высокое качество работы нейронной сети Кохонена.

# Заключение

При выполнение курсового проекта была реализована нейронная сеть Кохонена, которая позволяет обрабатывать полученные результаты тестирования по каждому пользователю. Качество работы нейронной сети довольно высоко, так как полученные данные сравнивались с экспертной оценкой и их результаты совпали.

Данные, полученные в результате работы нейронной сети Кохонена, можно использовать для дальнейшего составления тестов, для большего стимулирования студентов по некоторым темам, для поощрения студентов, которые лучше всех справились с выполнением тестов. Также с помощью данной сети можно проводить класстеризацию многих других результатов, например, диагностики, успеваемости, расходов.

# Используемая литература

1. Indigo[Электронный ресурс].- <https://indigotech.ru/>
2. Iren[Электронный ресурс].- <https://irenproject.ru/>
3. Habr[Электронный ресурс].- <https://habr.com/ru/>
4. TestTurn [Электронный ресурс].- <http://testturn.veralsoft.com/>
5. TestOfficePro[Электронный ресурс].- <https://compress.ru/article.aspx?id=15044#SunRav%20TestOfficePro>
6. Spring MVC[Электронный ресурс].- <https://habr.com/ru/post/336816/>
7. Википедия[Электронный ресурс].- <https://ru.wikipedia.org/wiki/>
8. Методика анализа и оценка результатов тестирования[Электронный ресурс].- <https://cyberleninka.ru/article/n/metodika-analiza-i-otsenka-rezultatov-testirovaniya/viewer>
9. Нейронные сети Кохонена[Электронный ресурс].- <https://neuronus.com/theory/nn/955-nejronnye-seti-kokhonena.html>
10. Тестирование как одна из форм эффективной учебной деятельности [Электронный ресурс].- <http://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/516934/>

# Приложение



