Учреждение образования Федерации профсоюзов Беларуси

«Международный университет «МИТСО»

Рег. № \_\_\_\_\_\_\_\_\_\_ Кафедра информационных систем и технологий

Дата \_\_\_\_\_\_\_\_\_2020

**КУРСОВАЯ РАБОТА**

на тему \_\_\_\_Название\_\_\_Разработать программный модуль «Картотека недвижимости»\_\_\_\_\_\_\_\_

по дисциплине \_Название\_\_\_\_Основы конструирования программ\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Основные замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Отметка о допуске курсовой работы к  защите:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_  Подпись научного руководителя:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  Урбанович Владислав Сергеевич (Ф.И.О. – полностью)  Курс \_\_1\_\_, группа \_\_\_2020\_\_\_\_  Факультет \_\_экономический\_\_\_  Специальность \_\_Информационные системы и технологии\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Научный руководитель:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность, ученая степень, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Ф.И.О. – полностью) |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc41387618)

[1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc41387619)

[2.ТРЕБОВАНИЕ К ПРОГРАММЕ 6](#_Toc41387620)

[3.КОНСТРУИРОВАНИЕ ПРОГРАММЫ 7](#_Toc41387621)

[4.ОПИСАНИЕ ИНТЕРФЕЙСА 9](#_Toc41387622)

[5.ВЫБОР СПОСОБА ОРГАНИЗАЦИИ ДАННЫХ 11](#_Toc41387623)

[6.РАЗРАБОТКА ПЕРЕЧНЯ ПОЛЬЗОВАТЕЛЬСКИХ МЕТОДОВ ПРОГРАММЫ 12](#_Toc41387624)

[7.ПРОГРАММНАЯ ЧАСТЬ 13](#_Toc41387625)

[ЗАКЛЮЧЕНИЕ 14](#_Toc41387626)

[ПРИЛОЖЕНИЕ 15](#_Toc41387627)

# **ВВЕДЕНИЕ**

Была поставлена задача разработать программный модуль «Учета клиентов стоматологии», предназначенный для использования работниками стоматологии. В базе содержатся сведения о клиентах стоматологической клиники (Имя, фамилия, отчество, возраст, диагноз, дата приёма). При приходе клиента в клиники происходит ввод данных товара. И последующее занесение товара в базу.

Программа создана для учёта и систематизации клиентов стоматологической клиники.

Цель разработки обеспечить работу быстро и максимально эффективно, удовлетворять запросы пользователя, а также должен быть возможность пополнять бузу клиентов, для быстрого использования пользователями. И неотъемлемой частью данной программы должна высокая оптимизация, дабы пользователь мог запустить даже на самом слабом устройстве. Так же клинику посещает множество клиентов, а значит потребуется обрабатывать большое количество данных.

Задачи данного программного продукта: быстро и гибко редактироваться со стороны владельцев клиники. С точки зрения использования пользователя программа должна иметь красивый, понятный и интуитивный интерфейс чтобы было владельцу приятно пользоваться программой. Так же требуется предусмотреть возможность добавления новых функций в программе(масштабируемость). А также возможность обрабатывать большее потоки данных.

Для разработки программы был использован текстовый файл и язык программирования Python созданный Гвидо ван Россум. И среда программирования PyCharm от компании JetBrains.

**PyCharm** — интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA.

**PyCharm** — это кросс-платформенная среда разработки, которая совместима с Windows, macOS, Linux. PyCharm Community Edition (бесплатная версия) находится под лицензией Apache License, а PyCharm Professional Edition (платная версия) является проприетарным ПО.

**Python**— высокоуровневый язык программирования общего назначения с динамической типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным — всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации, сам же язык известен как интерпретируемый и используется в том числе для написания скриптов. Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на Си или C++.

# **1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

Была поставлена задача разработать программный модуль «Учета клиентов стоматологии», предназначенный для использования работниками стоматологии. В базе содержатся сведения о клиентах стоматологической клиники (Имя, фамилия, отчество, возраст, диагноз, дата приёма). При приходе клиента в клиники происходит ввод данных товара. И последующее занесение товара в базу.

Медицина в общем и стоматология, в частности, является обширной сферой бизнеса. Из-за подобного развитие системы здравоохранения, то требуется учёт данных и их автоматизация. Данные условия возможно выполнить только при использовании информационных технологий.

Функции программного продукта: ввод характеристик клиента (Имя, фамилия, отчество, возраст, диагноз, дата приёма) должны иметь понятные данные и возможность обрабатывать их.

Данный программный продукт выгодно отличается от других подобных тем что, пользуясь им, кроме того, что пользователь может заносить данные в приложение, просматривать и редактировать. А выбранная архитектура позволяет быстро просматривать и оптимизировать данные.

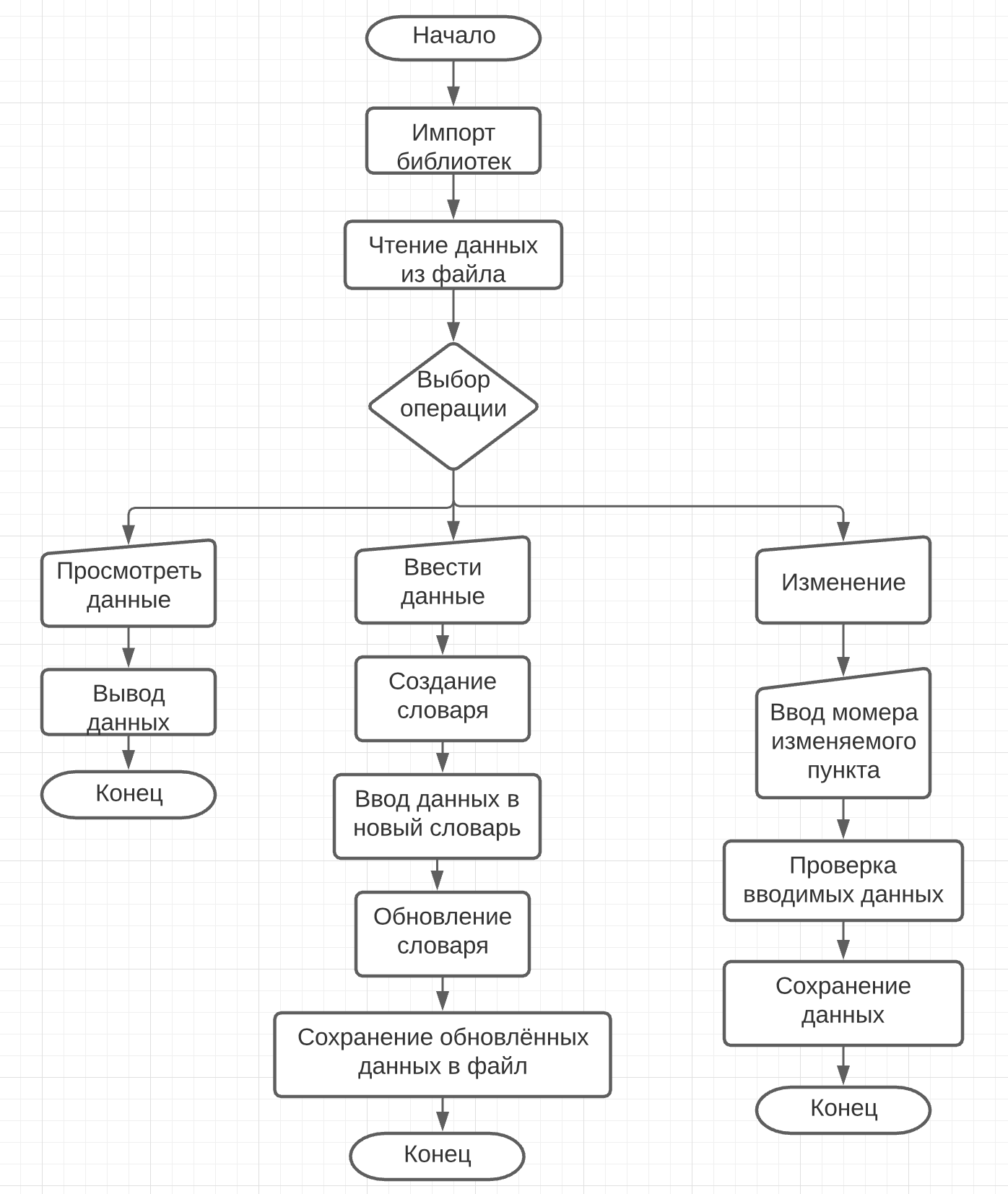
# **2.ТРЕБОВАНИЕ К ПРОГРАММЕ**

Исходя из задач, описанных выше главными требованиями к языку, на котором будет написана программы — это возможность обрабатывать большие потоки данных, а также возможность масштабировать проект.

Для выполнения данных задач очень подходит такой язык программирования как python. Так как его из-за того, что он является высокоуровневым языком программирования, а значит у него простой синтаксис и отсутствует излишние функции. Так же из-за простоты языка его можно легко и не дорого поддерживать, и развивать.

Среда разработки PyCharm. Вид приложения – консольное. Программа построена на базе функционального программирования. Способ организации переменные и структуры данных. Способ хранения данных – файл с разрешением .dat. Каждая логическая завершенная данных подзадача программы реализована в виде функций. К защите курсовой предоставляется: консольное приложение и пояснительная записка

# **3.КОНСТРУИРОВАНИЕ ПРОГРАММЫ**



В текущей главе представлено описание алгоритма, который имеет начало алгоритма и конец, внутри алгоритма имеются различные операций.

Вначале происходит импорт библиотек, затем происходит чтение данных из файла, где храниться данные. После этого происходит выбор из трёх вариантов (Просмотр данных, ввести данные, изменение)

1. **Просмотреть данные:** Циклично выводиться данные всех клиентов на консоль.
2. **Ввести данные:** Создается словарь для последующего изменения. Так как словарь пуст то его необходимо заполнить. Заполняется словарь вручную с клавиатуры. Потом происходит объединение словаря. А затем обновлённый словарь сохраняется, а файл.
3. **Изменение:** Вначале требуется ввести с клавиатуры номер изменяемого пункта. После ввода происходит проверка данных. Проверка состоит из двух пунктов. Первое: проверка введено ли число, а не другой символ. Поле этой проверки происходит проверка на корректность вводимого числа. Есть ли вводимый элемент в словаре.

# **4.ОПИСАНИЕ ИНТЕРФЕЙСА**

В начале происходит выбор желаемой операции. Необходимо ввести в консоли желаемый выбор (рис. 4.1.).

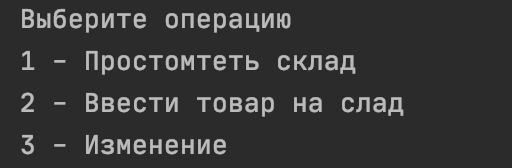


Рис.4.1.

Если же введены некорректное значение, то приложение попросит повторить ввод.

При вводе первого пункта происходит вывод всех данных из файла (рис. 4.2)

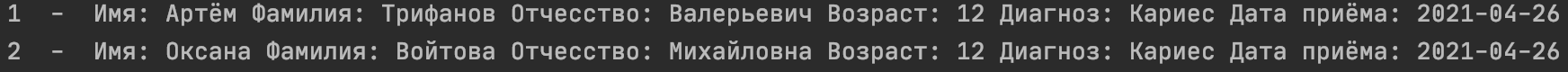


Рис.4.2.

Вывод происходит в одну строчку с поясняющей надписью, следующей последовательностью: Имя, фамилия, отчество, возраст, диагноз, дата приёма.

При выборе второго пункта происходит ввод данных (рис. 4.4.). Данные водятся последовательно, ввод следующего пункта возможно только после ввода преведушего.

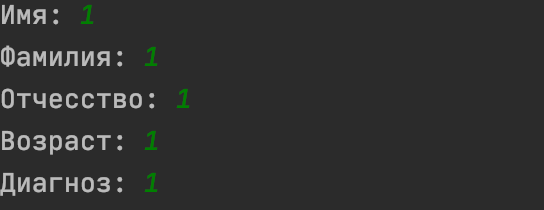


Рис. 4.3.

## Выбор третьего пункта сопровождается следующим сообщением. В нём содержится количество пунктов возможных для изменения.



Рис. 4.4.

Если данные введены некорректно, например введен символ то обработка данной ошибки сопровождается следующей надписью (рис. 4.5.)

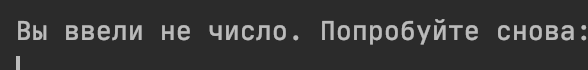


Рис.4.5.

Если же введено число выходящие из диапазона, то выводится надпись (рис 4.6.)

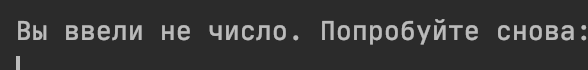
****

Рис 4.6.

Если же данные ведены корректно то поочерёдно выедаться данные для изменения (изменение времени не предусмотрено) рис.4.7.

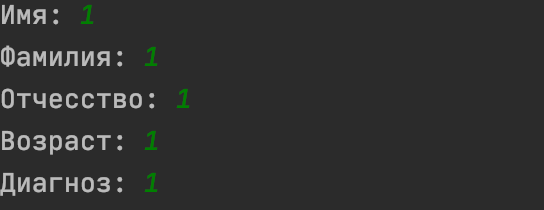


Рис 4.7.

# **5.ВЫБОР СПОСОБА ОРГАНИЗАЦИИ ДАННЫХ**

В Python есть возможность создавать комплексные словари. У них есть множество преимуществ. Одно из них состоит в том, что они могут выполнять функцию классов. Так как в программном продукте необходимо было такая сущность, которая бы хранила данные товара, но не нужны были методы по их обработки. То использование словаря является самым оптимальным вариантом.

В данном программном продукте используются 3 словаря.

1. Klient – основной словарь в котором храниться и обрабатываются данные. Это основная с структура в данном программном продукте.
2. Add – предназначен для изменения данных определённого пункта
3. Change – используется для добавления данных в основной словарь (Klient)

Необходимость использовать два последних словаря обуславливается невозможностью в словарях Python добавлять и изменять словарь. Но эту проблему можно обойти при помощи обновления словаря.

Сохранение же данных происходит в текстовом файле. А для непосредственного сохранения используется бинарная социализация.

# **6.РАЗРАБОТКА ПЕРЕЧНЯ ПОЛЬЗОВАТЕЛЬСКИХ МЕТОДОВ ПРОГРАММЫ**

**def Save():** - функция предназначенная для сохранения данных в файл. Файл имеет разрешение .dat и находиться в той же папке что и непосредственно сам проект.

В данном продукте реализована возможность выбора операции пользователем. Так как каждый выбор операции реализует уникальный алгоритм соответствующей данному проекту, то реляция возможно считать функцией.

**1 - Просмотреть данные** – выводить данные со клиентов клиники построчно.

**2 - Ввести данные -** требует ввести построчно данные клиента с одновременной вариацией данных.

**3 – Изменение** – происходит изменение данных. Пользователь вводит номер изменяемого элемента, а затем происходит непосредственно изменение данных и последующие их сохранение в соответствующий файл

# **7.ПРОГРАММНАЯ ЧАСТЬ**

Вначале происходит импортирование библиотек. **Pickle** – библиотека для возможности, сохранять и считать данные, то есть выполнять социализацию. **DateTime** – библиотека реализующая возможность различных операции с датой и временем.

Затем происходит инициализация глобальных переменных и структур данных для дальнейшей работы.

**Counter** – целочисленная переменная которая хранит в себе количество клиентов в базе.

**FILENAME** – переменная строкового типа хранящая имя файла в которой будет выполнять функции базы данных.

**Klient –** словарь который выполняет функции класса, в нём храниться данные о истории клиентов.

После этого происходит реализация функции **Save ()**. Данная функция выполняет сохранение словаря Klientв текстовый файл.

Следом происходит чтение данных из файла и занесение данных в словарь, а потом подсчёт элементов с помощью цикла for.

После этих действий происходит требуется провести выбор операции. Путём ввода данных с клавиатуры 1 – просмотреть данные, 2 – Ввести данные, 3 – Изменение.

1. При помощи цикла происходит вывод данных всех клиентов.
2. Происходит поочередное внесение данных, а затем последующее их занесение в словарь. Следом один словарь обновляется вторым. И данные из обновлённого словаря сохраняются в файл.
3. Вначале на консоль выводится количество клиентов в текстовом файле, существующих на данной клинике. Затем пользователя просят ввести номер клиента, который он желает изменить. После этого происходит проверка введенных данных. Если введенные данные являются положительным числом и не превышают количество клиентов, то пользователь вводит данные элемента словаря схожим алгоритмом что и во втором пункте. Далее происходит обновление одного словаря другим и обновлённый словарь сохраняется в файл.

# **ЗАКЛЮЧЕНИЕ**

Исходя из поставленных целей и задач было сконструировано, разработана и протестирован программный модуль «Учета клиентов стоматологии», предназначенный для использования работниками стоматологии.

Были реализованы задачи:

1. В базе содержатся сведения о клиенте (Имя, фамилия, отчество, возраст, диагноз, дата приёма).
2. При поступлении клиента происходит ввод данных товара.
3. И последующее занесение товара в базу.

Цель была выполнены, приложение имеет следующие преимущества:

1. Работа продукта происходит быстро и максимально эффективно. Программа удовлетворяет запросам пользователя.
2. Имеет возможность пополнять базу клиентов.
3. Следить за динамикой посещения стоматологии.
4. Так же программы имеет высокую оптимизацию. Из-за этого программа работает весьма быстра и запускается даже на самом слабом устройстве.

Так же программный продукт выгодно отличается от других представленных на рынке, тем что имеет понятный и простой интерфейс.

Таким образом программа работает без сбоев и все задачи были выполнены.

# **ПРИЛОЖЕНИЕ**