|  |  |
| --- | --- |
| Министерство образования и науки Республики Казахстан Северо-Казахстанский университет им. М. Козыбаева | |
| Семестровый проект №2  На тему: «**Модуль Tkinter**»  ФИЦТ.АПО19.06 группа | |
| Выполнили студенты группы АПО-19 | Битанов А.С. Бейсек Б.Ж .......Отеген К.Р ...Мамлютов В.Р. |
| Научный руководитель доцент, PhD | Астапенко Н.В. |

Петропавловск, 2022 г.

Введение

[**1.** **Аналитическая часть** 3](#_Toc97114819)

[**1.1** **Характеристика объекта исследования** 3](#_Toc97114820)

[**2.** **Проектная часть** 5](#_Toc97114821)

[**2.1** **Проектирование DFD** 6](#_Toc97114822)

[2.2 Проектирование средствами UML 6](#_Toc97114823)

[Заключение 8](#_Toc97114824)

[Список используемой литературы 9](#_Toc97114825)

**Введение**

Модуль Tkinter является графической библиотекой, позволяющей создавать программы с оконным интерфейсом. Это является интерфейсом к популярному языку программирования и инструменту создания графических приложений, является кроссплатформенной библиотекой и может быть использована в большинстве распространённых операционных систем.

Tkinter является событийно-ориентированной библиотекой. В приложениях такого типа имеется главный цикл обработки событий.

Целью данной проектной работы является изучение Tkinter и получение готового результата в виде готовой работы и её дальнейшего оценивания по предложенной теме.

В соответствие с поставленной целью необходимо решить следующие задачи:

1. изучить учебную, научную, справочную литературу и видеоролики по теме проекта.
2. Осуществляется ввод и хранение данных в БД (например, в качестве БД можно
3. использовать файл .csv).
4. Наличие минимум 5 видов виджетов.
5. Весь функционал представлен в главном меню.
6. Создать исполняемый файл.

В процессе выполнения проектной работы были использованы: учебная, справочная литература, видеоматериалы.

Работу распределили следующим образом:

Разработку и отладку кода проводил Битанов А.С.

Редактирование и отладку репозитория Отеген К.Р.

Составление отчёта и его редактирование Бейсек Б.Ж и Мамлютов В.Р.

1. **Аналитическая часть**

В результате планирования распределения работы над проектом, мы изучили библиотеку Tkinter. В ходе работы разработали архитектуру программного средства.

По сравнению с предыдущей работой, в этот раз было всё довольно просто, так-как мы по ровно распределили перечень работ по участникам и спокойно выполнили работу. Изучением библиотеки мы занимались все вместе, но саму разработку делал Асанали.

Провели описание используемых алгоритмов, приложили скриншоты и описали их. После провели тщательное тестирование кода, и получили готовый полноценный результат.

Tkinter является графическим интерфейсом. Появляется в приложении путём его импортирования и настройке под нужный формат экран.

Особенностями создания графического интерфейса пользователя средствами пакета Tkinter языка программирования Python. Требует знания языка Python на уровне структурного программирования, желательно также владение азами ООП.

Проблем с созданием приложений не много, они возможны только если в коде были допущены ошибки.

Очень много действий придется выполнять в рамках одной функции. И хотя может показаться довольно логичным и простым написание методов, которые выполняли бы целый спектр действий от А до Я, это может привести к значительному ухудшению читаемости кода, а также невозможности грамотно его поддерживать.

Так что все зависит от перспективы, в которой человек это оценивает. Если перспектива краткосрочная, то такой подход действительно проще.

Но при разработке сложных приложений это может привести к большим проблемам в дальнейшем. А учитывая то, что с базами данных работают именно сложные приложения, то проблемы при таком подходе возникнут гарантированно.

* 1. **Характеристика объекта исследования**

Tkinter – это пакет для Python, предназначенный для работы с библиотекой Tk. Библиотека Tk содержит компоненты графического интерфейса пользователя (graphical user interface – GUI). Эта библиотека написана на языке программирования Tcl.

Под графическим интерфейсом пользователя (GUI) подразумеваются все те окна, кнопки, текстовые поля для ввода, скроллеры, списки, радиокнопки, флажки и др., которые вы видите на экране, открывая то или иное приложение. Через них вы взаимодействуете с программой и управляете ею. Все эти элементы интерфейса будем называть виджетами (widgets).

В настоящее время почти все приложения, которые создаются для конечного пользователя, имеют GUI. Редкие программы, подразумевающие взаимодействие с человеком, остаются консольными. В предыдущих двух курсах мы писали только консольные программы.

Существует множество библиотек GUI, среди которых Tk не самый популярный инструмент, хотя с его помощью написано немало проектов. Он был выбран для Python по умолчанию. Установочный файл интерпретатора Питона обычно уже включает пакет Tkinter в составе стандартной библиотеки.

Tkinter можно представить как переводчик с языка Python на язык Tcl. Вы пишете программу на Python, а код модуля Tkinter переводит ваши инструкции на язык Tcl, который понимает библиотека Tk.

Программы с графическим интерфейсом пользователя событийно-ориентированные. Вы уже должны иметь представление о структурном и желательно объектно-ориентированном программировании. Событийно-ориентированное ориентировано на события. То есть та или иная часть программного кода начинает выполняться лишь тогда, когда случается то или иное событие.

Событийно-ориентированное программирование базируется на объектно-ориентированном и структурном. Даже если мы не создаем собственных классов и объектов, то все-равно ими пользуемся. Все виджеты – объекты, порожденные встроенными классами.

События бывают разными. Сработал временной фактор, кто-то кликнул мышкой или нажал Enter, начал вводить текст, переключил радиокнопки, прокрутил страницу вниз и т. д. Когда случается что-то подобное, то, если был создан соответствующий обработчик, происходит срабатывание определенной части программы, что приводит к какому-либо результату.

1. **Проектная часть**

Вопросы и ответы – это понимание конечного результата Ваших действий. И чем подробнее Мы представляем этот результат, тем лучше и легче сможете достичь цель. Как раз в этом нам поможет SMART технология.

Вопрос: что именно мы хотим достичь/улучшить/изменить? Наша ли это цель?

Ответ: добавить покупку в наше приложение, изменить интерфейс

Вопрос: Какие результаты покажут завершение цели? Сколько и чего мы хотим приобрести? Какие факты подтвердят, что цель по SMART оказалась достигнута?

Ответ: “Больше времени для реализации”, “Изучение дополнительных материалов, для большего познания”.

Вопрос: сможем ли мы достичь эту цель по SMART? Не слишком ли она сложная или лёгкая? Что может помешать, а что помочь?

Ответ: “Наши конкуренты имеют почти весь рынок, но Мы постараемся улучшить наше приложение и пустить её в реализацию, и попробовать забрать себе хоть малую часть рынка”.

Совокупность исходных данных определяется конкретным видом проектируемой технологии, методики или др. Однако в любом случае на первом этапе проектирования дипломник проводит анализ данных, значимых для его частного исследования. Проектирование ведется в несколько этапов, которые связаны между собой следующим образом: глобальное – общее – конкретное. В соответствии с этим на втором этапе ведется непосредственное проектирование на глобальном или общем уровне: разрабатываются пакеты учебно-программной документации или создаются общая структура и содержание конкретной технологии (информационной, игровой, модульной, укрупнения дидактических единиц и др.), прописывается структура системы контроля и т. п.

На третьем этапе ведется проектирование на конкретном уровне, прорабатываются специальные вопросы технологий, методик, элементы систем. Например, создаются пакеты учебных элементов (обучающих модулей) по модульным технологиям, сценарии деловых игр в рамках игровых технологий – в рамках проблемных, текстовые задания и системы оценки результатов в рейтинговых технологиях и т. д.

* 1. **Проектирование DFD**

Функции, хранилища и внешние сущности на DFD-диаграмме связываются дугами, представляющими потоки данных. При интерпретации DFD-диаграммы используются следующие правила:

* Функции преобразуют входящие потоки данных в выходящие
* Хранилища данных не изменяют потоки данных, а служат только для хранения поступающих объектов
* Преобразования потоков данных во внешних сущностях игнорируется

Помимо этого, для каждого информационного потока и хранилища определяются связанные с ними элементы данных. Каждому элементу данных присваивается имя, также для него может быть указан тип данных и формат. Именно эта информация является исходной на следующем этапе проектирования - построении модели "сущность-связь".

Построим DFD-диаграмму для работы, строящего свою деятельность по принципу "Игровой терминал о играх". На основании полученных данных, пользователь может посмотреть другую игру. Подробная схема представлена на Рис.1

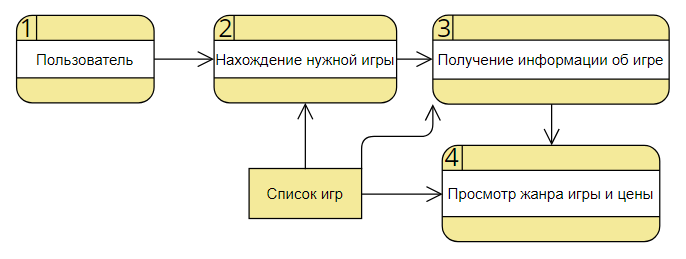


Рис.1 «DFD диаграмма»

# 2.2 Проектирование средствами UML

Вся работа была распределена поровну и почти также мы составляли отчёт. В ходе выполнения работы, проблем никаких не было. Подробная таблица распределения работ представлена на Рис.2

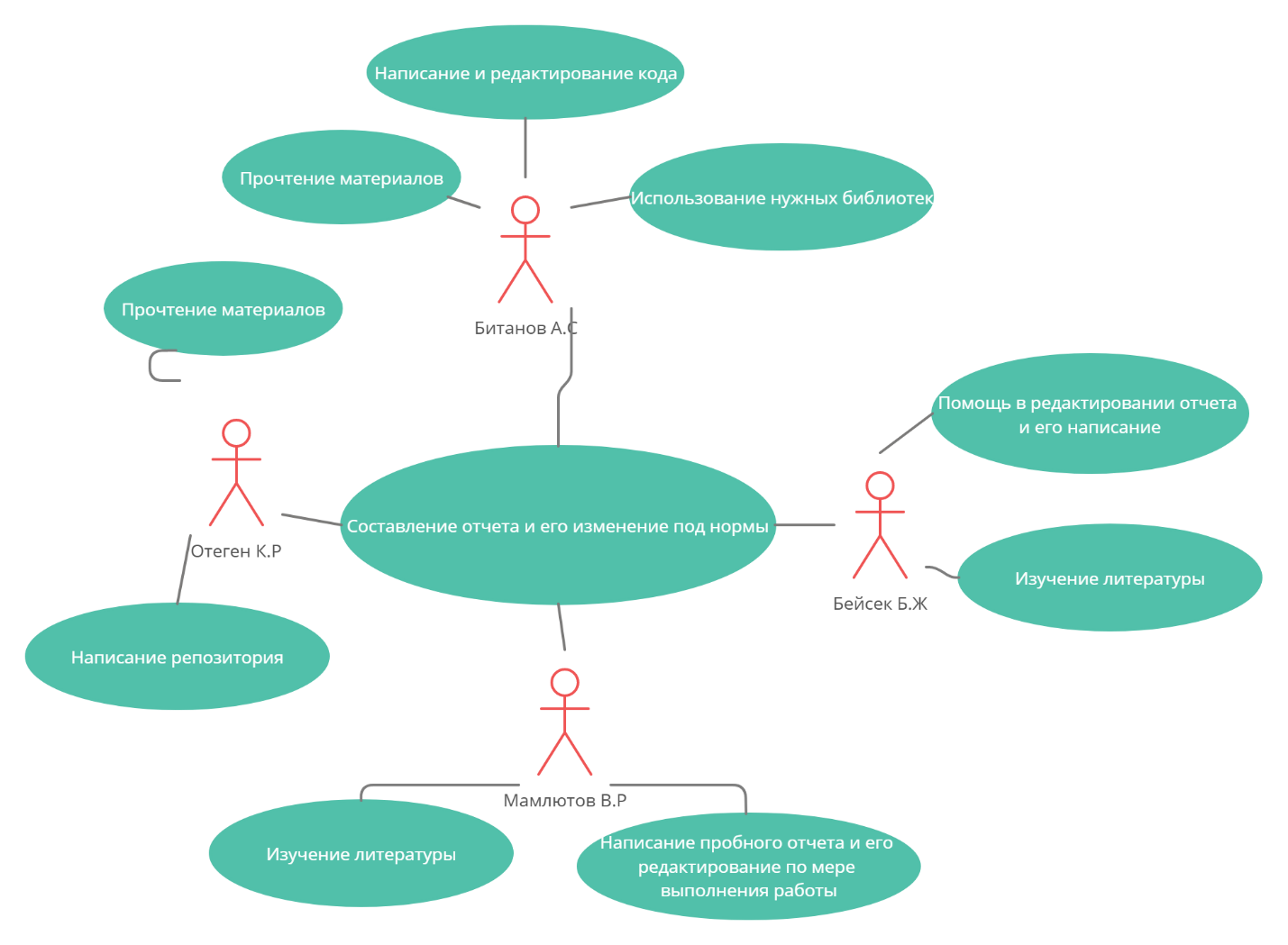


Рис.2 «UML схема создания проекта»

# Заключение

В нашей проектной работе мы спроектировали интерфейс и протестировали его.

По итогу выполнения проектной работы мы выполнили все поставленные перед нами задачи, было выполнено 6 заданий которые были указаны во введении.

1. Изучить учебную, научную, справочную литературу и видеоролики по теме проекта.
2. Осуществляется ввод и хранение данных в БД (например, в качестве БД можно
3. использовать файл .csv).
4. Наличие минимум 5 видов виджетов.
5. Весь функционал представлен в главном меню.
6. Создать исполняемый файл.

Во время выполнения поставленных задач проблем с их реализацией не было.

Итогом работы можно считать закрепление знаний в области модуле Tkinter, полученных в ходе выполнения работы.

В заключение можно отметить, что модуль Tkinter является не таким и сложным, если объяснять обычным языком, это создание приложения с графическим интерфейсом.

Таким образом, можно сделать вывод, что первоначальные задачи, поставленные были полностью решены. Практическая часть работы успешно реализована в среде Python.

# Список используемой литературы

1. ADV-IT (2019) GitHub Actions - Основы Автоматизации - DevOps – GitOps [Любительское видео]//YouTube. 16 декабря (<https://www.youtube.com/watch?v=Yg5rpke79X4>) Просмотрено: 16.02.2022.
2. Хитрый питон (2021) GitHub Actions для автоматической проверки кода [Любительское видео]//YouTube. 3 июня (<https://www.youtube.com/watch?v=NijFSs03Pd4>) Просмотрено: 16.02.2022.
3. Рядченко, В.П. Программирование на языке высокого уровня Python: учебно-методическое пособие / В.П. Рядченко, Л.М. Эльканова, Л.М. Шавтикова. Черкесск: БИЦ СевКавГГТА, 2018. –144с.
4. Федоров Д.Ю. Основы программирования на примере языка Python: учебное пособие. – СПб.: Питер, 2019. – 152 с.
5. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»)
6. Федоров Д.Ю. Основы программирования на примере языка Python: учебное пособие. – СПб.: Питер, 2019. – 152 с.
7. Юдин, Д.Б. Задачи и методы линейного программирования: Математические основы и практические задачи / Д.Б. Юдин, Е.Г. Гольштейн. - М.: КД Либроком, 2020.
8. Бадд, Т. Объектно-ориентированное программирование в действии / Т. Бадд. - СПб.: Питер, 2017.
9. Бадд, Т. Объектно-ориентированное программирование в действии / Т. Бадд. - СПб.: Питер, 2018.
10. Хорев, П.Б. Объектно-ориентированное программирование / П.Б. Хорев. - М.: Academia, 2018.