Шаблон отчета по производствнной практике

ФИО студента: Смыслов Иван

Группа 28 Б

Содержание

[Введение 3](#_Toc169271161)

[1. Общая характеристика МБОУ «СОШ № 6 им. В.А. Горишнего» г. Симферополя 5](#_Toc169271162)

[2. Анализ предметной области «Расчет бюджета школы» 10](#_Toc169271163)

[3. Формирование требований на разработку модуля программного обеспечения 11](#_Toc169271164)

[4. Разработка алгоритма работы окна «Бюджет 1.0» в модуле ПО 20](#_Toc169271165)

[5. Разработка модуля программы в интегрированной среде 22](#_Toc169271166)

[6. Отладка и тестирование на работоспособность модуля 27](#_Toc169271167)

[Заключение 31](#_Toc169271168)

[Список использованных источников 33](#_Toc169271169)

[Приложения 35](#_Toc169271170)

# Введение

Производственная практика по специальности ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем» проходила в [МБОУ «СОШ № 6 им. В.А. Горишнего» г. Симферополя](http://sch6simf.ucoz.com/). Практика была направлена на приобретение практических навыков в профессиональной деятельности и закрепление теоретических знаний, полученных во время учебы в области разработки модулей программного обеспечения. В рамках практики были закреплены теоретические знания и практические навыки по разработке модуля программного обеспечения в интегрированной среде IDLE Python на языке программирования Python. Сроки прохождения практики: 27.05.2024 – 21.06.2024.

Производственная практика проходила на базе  [Муниципальное бюджетное общеобразовательное учреждение «Средняя Общеобразовательная Школа № 6 имени Героя Советского Союза В. А. Горишнего» муниципального образования городской округ Симферополь Республики Крым.](http://sch6simf.ucoz.com/)

Объектом исследования производственной практики была деятельность [МБОУ «СОШ № 6 им. В.А. Горишнего».](http://sch6simf.ucoz.com/)

Предметом исследования производственной практики были процессы проектирования, разработки и тестирования модуля программного обеспечения.

Цель исследования: разработать модуль программного обеспечения «Расчет бюджета школы», позволяющий эффективно распределять и использовать ресурсы школы – финансовые и материальные.

Исходя из поставленной цели были определены следующие задачи практики:

* выполнить анализ видов деятельности и организационной структуры объекта исследования;
* выполнить анализ предметной области «Расчет бюджета школы» и обосновать необходимость разработки модуля программного обеспечения «Расчет бюджета школы»;
* сформировать требования к программному продукту, составить техническое задание на разработку;
* составить алгоритм работы окна «Расчет бюджета школы» в модуле программного обеспечения;
* разработать приложение с использованием современных технологий и инструментов;
* провести отладку и тестирование на работоспособность модуля программного обеспечения.

При прохождении производственной практики можно отметить практическую значимость результатов исследования. Она заключается в том, что разработанный модуль программного обеспечения «Расчет бюджета школы» позволяет автоматизировать процесс управления расходами школы на материальные ресурсы. Данный программный продукт может быть рекомендован образовательным организациям. Результаты исследования могут быть использованы для разработки в дальнейшем клиент-серверной версии приложения.

# 1. Общая характеристика [МБОУ «СОШ № 6 им. В.А. Горишнего» г. Симферополя](http://sch6simf.ucoz.com/)

Муниципальное бюджетное общеобразовательное учреждение «Средняя Общеобразовательная Школа № 6 имени Героя Советского Союза В. А. Горишнего» муниципального образования городской округ Симферополь Республики Крым действует как юридическое лицо на территории полуострова с 2015 г. Сокращенное официальное наименование образовательной организации: [МБОУ «СОШ № 6 им. В.А. Горишнего» г. Симферополя](http://sch6simf.ucoz.com/).

Организационно-правовая форма: бюджетное учреждение. Тип образовательной организации: общеобразовательная организация. Основная цель деятельности организации осуществление образовательной деятельности по образовательным программам начального общего, основного общего и среднего общего образования. ИНН 9102159981. КПП 910201001. ОГРН 1159102028810 (приложение А).

Юридический адрес исследуемой организации: 295022, Республика Крым, г. Симферополь, пр-кт Победы, д. 176. Учредителем образовательной организации является муниципальное образование – Городской округ Симферополя Республики Крым. От имени муниципального образования функции и полномочия учредителя осуществляет администрация Города Симферополя Республики Крым в лице Управления образования, молодежи и спорта администрации города Симферополя Республики Крым.

МБОУ «СОШ № 6 им. В.А. Горишнего» г. Симферополя в своей деятельности подведомственно и подконтрольно Управлению образования.

Виды экономической деятельности осуществляющие школой по Общероссийскому классификатору видов экономической деятельности (ОКВЭД ОК 029-2014 КДЕС. Ред. 2) представлены в таблице 1.1.

Таблица 1.1 – Сведения о видах экономической деятельности МБОУ «СОШ № 6 им. В.А. Горишнего»

|  |  |  |
| --- | --- | --- |
| Категория вида деятельности | Код вида деятельности | Наименование вида деятельности |
| Основной | 85.13 | Образование основное общее |
| Дополнительный | 85.12 | Образование начальное общее |
| Дополнительный | 85.14 | Образование среднее общее |
| Дополнительный | 85.41 | Образование дополнительное детей и взрослых |
| Дополнительный | 68.32.2 | Управление эксплуатацией нежилого фонда за вознаграждение или на договорной основе |
| Дополнительный | 85.11 | Образование дошкольное |

[МБОУ «СОШ № 6 им. В.А.Горишнего» г. Симферополя](http://sch6simf.ucoz.com/) прошла лицензирование и государственную аккредитацию в соответствии с Федеральным законом «Об образовании в Российской Федерации» и другими нормативными правовыми актами Российской Федерации и Республики Крым. Согласно выписки из Единого государственного реестра юридических лиц (приложение А) видом лицензируемой деятельности, на который выдана лицензия № 0694 от 12.11.2022 г. Министерством образования, науки и молодежи Республики Крым является: «Образовательная деятельность, осуществляемая образовательными организациями, организациями, осуществляющими обучение, а также индивидуальными предпринимателями, за исключением индивидуальных предпринимателей, осуществляющих образовательную деятельность непосредственно, лицензирование которой осуществляют органы исполнительной власти субъектов Российской Федерации, осуществляющие переданные полномочия Российской Федерации в сфере образования». В 2022 году школа получила свидетельство о государственной аккредитации № 0694 от 12.11.2022 года серия 82А01 № 0000731.

Образовательное учреждение в соответствии с нормами действующего законодательства принимает локальные нормативные акты по основным вопросам организации и осуществления образовательной деятельности, в том числе регламентирующие правила приема обучающихся, режим занятий обучающихся, формы, периодичность и порядок текущего контроля успеваемости и промежуточной аттестации обучающихся, обучение по индивидуальному учебному плану (в том числе ускоренное обучение) в пределах осваиваемых образовательных программ, порядок и основания перевода, отчисления и восстановления обучающихся, порядок оформления возникновения, приостановления и прекращения отношений между [МБОУ «СОШ № 6 им. В.А. Горишнего»](http://sch6simf.ucoz.com/) и обучающимися и (или) родителями (законными представителями) несовершеннолетних обучающихся.

Управление [МБОУ «СОШ № 6 им. В.А. Горишнего» г. Симферополя](http://sch6simf.ucoz.com/). осуществляется в соответствии с законом № 237-ФЗ «Об образовании в Российской Федерации» и Уставом [МБОУ «СОШ № 6 им. В.А. Горишнего» г. Симферополя](http://sch6simf.ucoz.com/). Республики Крым на принципах демократичности, открытости, приоритета общечеловеческих ценностей, охраны жизни и здоровья человека, свободного развития личности. Управление школой строится на принципах единоначалия и самоуправления. Исходя из целей, принципов построения и стратегии развития школы сложилась структура, в которой выделяется 4 уровня управления (рисунок 1.1).

Первый уровень. Директор Клименко И.А. – главное административное лицо, Контролирует работу и обеспечивает эффективное взаимодействие структурных подразделений организации, утверждает штатное расписание, отчетные документы организации, осуществляет общее руководство Школой.

На этом же уровне модели находятся высшие органы коллегиального и общественного управления, имеющие тот или иной правовой статус: Попечительский совет, Педагогический совет, Общее собрание трудового коллектива.

Общее собрание трудового коллектива является высшим органом самоуправления Школы, реализует право работников участвовать в управлении образовательной организацией, в том числе:

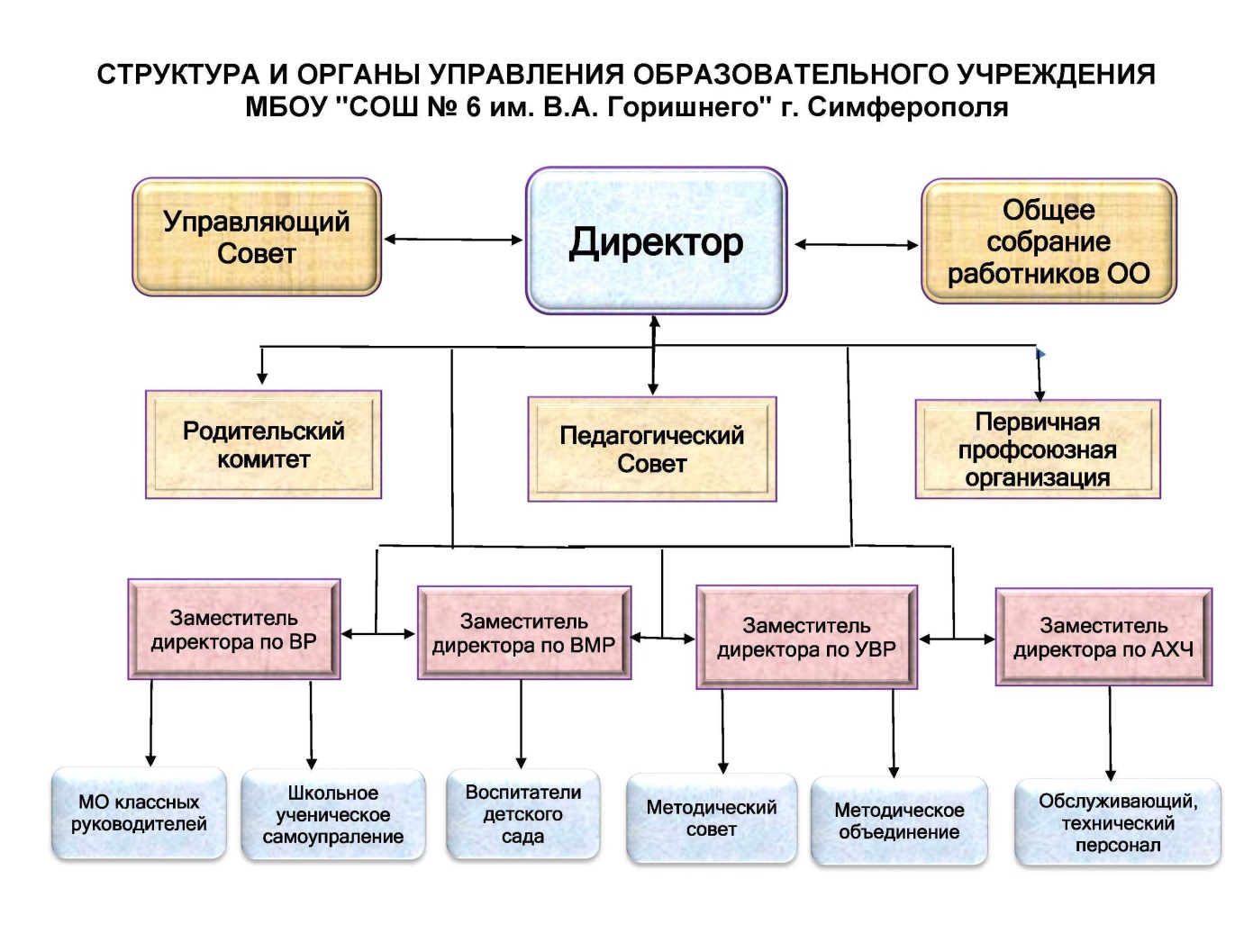


Рисунок 1.1 – Организационная структура [МБОУ «СОШ № 6 им. В.А.Горишнего» г. Симферополя](http://sch6simf.ucoz.com/)

* участвовать в разработке и принятии коллективного договора, Правил трудового распорядка, изменений и дополнений к ним;
* принимать локальные акты, которые регламентируют деятельность образовательной организации и связаны с правами и обязанностями работников;
* разрешать конфликтные ситуации между работниками и администрацией образовательной организации;
* вносить предложения по корректировке плана мероприятий организации, совершенствованию ее работы и развитию материальной базы.

Педагогический совет – коллективный орган управления школой, который решает вопросы, связанные с реализацией программы развития школы, рассматривает проблемы, подготовленные методическим советом школы, администрацией школы, несет коллективную ответственность за принятые решения.

Общее собрание трудового коллектива объединяет всех членов трудового коллектива. Оно решает вопросы, связанные с разработкой Коллективного договора, Правил внутреннего трудового распорядка, проектов локальных актов школы.

Субъекты управления этого уровня обеспечивают единство управляющей системы в целом, определяют стратегическое направление развития образовательного учреждения, всех его подразделений.

Второй уровень – заместители директора образовательного учреждения по учебно-воспитательной работе, заместитель директора по воспитательной работе, заместитель директора по АХЧ. Главная функция – согласование деятельности процесса в соответствии с заданными целями, программой и ожидаемыми результатами.

Третий уровень – методические объединения. К управленцам этого уровня относится руководитель школьного методического объединения. Взаимодействие субъектов управления этого уровня осуществляется через специализацию функций при их одновременной интеграции. Руководитель методического объединения выбирается из преподавательского состава. Школьные предметные объединения ведут методическую работу по предмету, организуют внеклассную деятельность учащихся, проводят анализ результатов образовательного процесса, имеют право выдвигать предложения по улучшению процесса образования, получать методическую помощь научных консультантов, согласуют свою деятельность с администрацией школы и в своей работе подотчётно ей.

Четвёртый уровень – обучающиеся, родители. Здесь органами управления являются общешкольный родительский комитет. В период между заседаниями Попечительского совета школы в роли органа управления выступает Совет родителей, который решает вопросы организации внешкольной и внеклассной работы, развития материальной базы школы, принимает участие в развитии учебного заведения.

Развитие школьного самоуправления обеспечивает реализацию принципов демократизации, общественного характера управления. Участие детей в управляющей системе формирует их организаторские способности и деловые качества. Сложившаяся модель структурных подразделений соответствует функциональным задачам школы, все структурные подразделения выполняют основные задачи, определенные планом работы школы.

Таким образом, целью деятельности Муниципального бюджетного общеобразовательного учреждения [«СОШ № 6 им. В.А.Горишнего» г. Симферополя](http://sch6simf.ucoz.com/). Республики Крым является предоставление качественных образовательных услуг в области начального, общего, среднего и дополнительного образования, обеспечение инновационного характера образования с учетом госзаказа и современных запросов рынка.

# 2. Анализ предметной области «Расчет бюджета школы»

Бюджет – это план доходов и расходов в денежном или натуральном выражении, который составляют на нужный период. Бюджетирование помогает предсказывать результаты деятельности компании. Главная цель бюджетирования – повысить эффективность работы компании на каждом этапе.

Предметная область исследования – расчет бюджета школы.

Основная цель расчета бюджета школы – составление эффективного плана бюджета школы и контроль за его исполнением. Эта предметная область широко используется в образовательных учреждениях. Данная предметная область касается процессов ввода первичных данных о материальных ресурсах образовательной организации, а также выполнение финансовых расчетов по их использованию.

В данной области включены следующие основные составляющие:

1. Внесение первичных данных о материальных ресурсах: процесс, включающий внесение данных о следующих материальных расходах школы учебных пособий, технические средства обучения, расходные материалы, хозяйственные нужды, или иные расходы на приобретение материальных запасов для нужд учебного процесса.
2. Обработка данных: процесс, включающий анализ внесенных данных. В процессе обработки может потребоваться дополнительная информация.
3. Вывод данных: процесс, включающий в себя вывод общей стоимости в отведенную для этого строку в программе, для вывода точной информации нужно проверить правильность введенных данных в начале.

# 3. Формирование требований на разработку модуля программного обеспечения

Для эффективной разработки системы необходимо иметь ясное понимание того, какие функции должны быть созданы и каким должен быть интерфейс. Для того, чтобы гарантировать, что все требования будут учтены и выполнены, используется техническое задание на разработку.

Техническое задание (ТЗ) является одним из ключевых документов при разработке программного продукта. Оно содержит описание требований к разрабатываемому продукту, а также указывает на этапы его создания и сроки исполнения [5, 6, 7, 8].

Основными целями ТЗ являются установление четкого понимания задачи заказчиком и исполнителем, а также определение критериев успеха для проекта. Кроме того, ТЗ служит инструментом контроля качества проекта и позволяет своевременно выявлять и исправлять ошибки в процессе разработки.

В ТЗ должны быть описаны все основные функциональные и нефункциональные требования к продукту, а также требования к его интерфейсу и производительности. Важным элементом ТЗ является также описание предполагаемой архитектуры продукта, используемых технологий и инструментов разработки.

Объем ТЗ может значительно различаться в зависимости от сложности проекта. В некоторых случаях он может быть ограничен несколькими страницами, а в других – составлять сотни страниц.

Важно отметить, что ТЗ должно быть составлено таким образом, чтобы оно было доступно для понимания как заказчику, так и исполнителю. Оно должно быть понятным и конкретным, без двусмысленностей и неоднозначностей. Только в этом случае можно гарантировать успешное выполнение проекта в соответствии с требованиями заказчика.

Техническое задание на разработку программы «Расчет бюджета школы» составлено в соответствии с ГОСТ 34.602-2020. Исходный текстовый файл технического задания на разработку модуля программного обеспечения «Расчет бюджета школы» выложен в репозиторий системы контроля версий по адресу: https://github.com/vanya2007colledj/colledgz.git

Составим техническое задание на разработку мобильного приложения и приведем выдержку из него.

1. Общие сведения.

Полное наименование: модуль программного обеспечения «Расчет бюджета школы».

Заказчиком мобильного приложения является [МБОУ « СОШ № 6 им. В.А. Горишнего».](http://sch6simf.ucoz.com/)

Адрес заказчика: 295022, Республика Крым, г. Симферополь, пр-кт Победы, д. 176.

Данное мобильное приложение разрабатывается на основании задания, выданного в ходе производственной практики.

Плановый срок начала работ – 27 мая 2024 года.

Плановый срок окончания работ – 21 июня 2024 года.

2. Цели и назначение создания программы.

2.1 Цели создания программы.

Цели создания программы «Расчет бюджета школы»:

* Облегчение процесса хранения и управления информацией о ресурсах школы. Приложение должно помочь пользователям сохранять информацию о ресурсах школы.
* Повышение удобства использования. Приложение должно иметь интуитивно понятный и простой интерфейс для удобства использования пользователем.
* Увеличение эффективности работы с ресурсами школы. Пользователи должны иметь возможность быстро и удобно добавлять новую информацию о ресурсах школы, просматривать информацию и рассчитывать общую стоимость внесенных данных которые соответствуют ресурсам школы.

2.2 Назначение программы.

Основным назначением программы «Расчет бюджета школы» является упрощение процесса хранения и управления информацией ресурсах школы , а также повышение эффективности работы. Приложение позволяет пользователям быстро и удобно сохранять информацию о ресурсах школы. Данное программное обеспечение помогает людям удобно и эффективно управлять ресурсами своей школы.

3. Характеристика объекта автоматизации.

3.1 Основные сведения об объекте автоматизации или ссылки на документы, содержащие такие сведения.

Объектом автоматизации в программе «Расчет бюджета школы» является процесс управления информацией о ресурсах школы, включающий в себя сохранение информации о ресурсах. Для автоматизации этого процесса используются функции приложения, такие как добавление информации о ресурсах школы, обновление информации, удаление информации о бюджете школы, расчет общей стоимости всех ресурсов школы.

4. Требования к программе.

4.1 Требования к структуре программе в целом.

Общими для программы «Расчет бюджета школы» являются следующие требования:

* Функционал. Приложение должно обладать необходимым функционалом, включая возможность добавления товаров, редактирования информации о ресурсах школы и удаление информации о ресурсах школы.
* Надежность и отказоустойчивость. Приложение должно быть надежным и отказоустойчивым, чтобы обеспечить доступность сервиса для пользователей в любое время.

4.2 Требования к функциям (задачам), выполняемым программой.

Требования к функциям программы:

* Управление ресурсами школы. Приложение должно обеспечивать возможность добавления различной информации о ресурсах школы в список, редактирования информации о ресурсах а так же удаление информации о ресурсах школы.
* Поиск товаров. Приложение должно обеспечивать возможность поиска ресурсов в списке по различным критериям, таким как название, стоимость, комментарий.

4.3 Общие технические требования к программе.

4.3.1 Технические требования.

Для запуска программы «Расчет бюджета школы» необходимы следующие технические требования:

* Операционная система: Windows 10 и выше.
* Процессор: Intel(R) Core(TM) i5-4670 CPU @ 3.40GHz 3.40 GHz и выше.
* Оперативная память: не менее 4 ГБ
* Свободное место на устройстве: не менее 1 ГБ
* Разрешение экрана: не менее 1366 х 768 пикселей

Для разработки программы «Расчет бюджета школы» необходимо использовать соответствующие инструменты и технологии, включая:

* Язык программирования: Python.
* Интегрированная среда разработки: IDLE Python.
* Библиотеки и фреймворки: sqlite3, tkinter

4.3.2 Требования к показателям назначения.

Показатели назначения программы «Расчет бюджета школы» включают в себя следующие требования:

* Доступность. Программа «Расчет бюджета школы» должна быть доступной для пользователей в любое время, без простоев и сбоев.
* Быстродействие. программа «Расчет бюджета школы» должно обеспечивать высокую скорость работы при добавлении данных о ресурсах школы, редактировании информации о ресурсах.
* Пропускная способность. программы «Расчет бюджета школы» должно иметь достаточную пропускную способность для обработки большого объема данных и пользователей.
* Надежность и отказоустойчивость. программа «Расчет бюджета школы» должна быть надежной и отказоустойчивой, чтобы обеспечивать доступность сервиса для пользователей в любое время.
* Простота использования. программы «Расчет бюджета школы» должна быть простой и удобной в использовании для всех пользователей.

4.3.3 Требования к надежности.

Требования к надежности программы «Расчет бюджета школы» включают в себя следующие пункты:

* Отказоустойчивость. программы «Расчет бюджета школы» должна быть отказоустойчивой, чтобы обеспечивать доступность сервиса для пользователей в любое время.
* Тестирование и анализ. Программа должна проходить регулярное тестирование и анализ, чтобы обеспечивать высокую надежность и производительность.

4.3.4 Требования к эргономике и технической эстетике.

Взаимодействие пользователя с программы «Расчет бюджета школы» происходит посредством визуального графического интерфейса (GUI). Интерфейс программы должен быть удобным и простым в использовании, не должен быть перегружен графическими элементами и должен предоставлять быстрое отображение экранных форм. Элементы навигации должны быть выполнены в удобной для пользователей форме. Ввод и вывод данных, отображение информации о ресурсах в программе должны выполняться в интерактивном режиме. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям приложения.

Интерфейс должен быть разработан под устройства с Windows

Все надписи экранных форм, а также сообщения, выдаваемые пользователю, должны быть на русском языке.

Дизайн приложения должен быть привлекательным и соответствовать современным тенденциям в области компьютерного дизайна. Все элементы интерфейса должны быть четкими и ясно различимыми, а цветовая гамма должна быть гармоничной и не вызывать дискомфорта у пользователей.

Приложение должно обеспечивать удобный поиск данных о ресурсах школы, , а также возможность добавления и редактирования информации о ресурсах школы. Взаимодействие с пользователем должно происходить в режиме реального времени, без задержек и прерываний.

Кроме того, приложение должно обеспечивать безопасность и конфиденциальность сохраненных данных.

4.4 Требования к видам обеспечения.

4.4.1 Требования к программному обеспечению.

программа «Расчет бюджета школы» должно использовать программные методы и модели, необходимые для реализации функций приложения.

Для обеспечения корректной работы приложения, должны быть установлены и настроены все необходимые библиотеки и инструменты, связанные с программным обеспечением.

4.4.2 Требования к информационному обеспечению.

Данные должны храниться в единой базе данных, защищенной от несанкционированного доступа и обеспечивающей быстрый доступ к информации.

Обмен между компонентами приложения должен осуществляться с использованием единой базы данных.

4.4.3 Требования к лингвистическому обеспечению.

Пользовательский интерфейс и вся документация на программу «Расчет бюджета школы» должна быть разработана на русском языке.

4.5 Состав и содержание работ по созданию программы.

Состав и содержание работ по созданию программы включает в себя следующие этапы:

* Анализ требований. Определение требований к программе «Расчет бюджета школы» и их документирование.
* Проектирование программы. Создание архитектуры программы, определение структуры базы данных, выбор языка программирования и технологий.
* Разработка пользовательского интерфейса.
* Тестирование. Проведение тестирования программы для выявления и устранения ошибок и недочетов.
* Оптимизация и улучшение производительности. Оптимизация работы приложения и улучшение его производительности.
* Развитие и сопровождение. Постоянное развитие и сопровождение приложения, включая добавление новых функций, исправление ошибок, обновление программного обеспечения и мониторинг производительности.

Таким образом, разработка технического задания является одним из наиболее ответственных и важных этапов в процессе создания программы «Расчет бюджета школы». Качество разработанного ТЗ может существенно облегчить или затруднить выполнение работ. Поэтому грамотно составленное техническое задание является ключевым фактором успеха реализации проекта.

# 4. Разработка алгоритма работы окна «Бюджет 1.0» в модуле ПО

В рамках автоматизации процесса расчета бюджета школы был разработан алгоритм использования программы с целью расчета бюджета школы в соответствии с ГОСТ 19.701-90.

Алгоритм решения задачи :

1. Пользователь открывает приложение расчет бюджета школы.

2. Пользователь вносит необходимые данные для расчета в отведенные для этого поля а именно: «Название», «Комментарий», «Стоимость».

3. После внесения данных пользователь нажимает кнопку «Добавить» .

4. Данные которые пользователь вносил в пункте 3, заносятся в специальное окно с нумерацией 1, 2, 3... .

5. После внесения всех данных нужных для расчета общей стоимости, пользователь нажимает кнопку «Общая стоимость»

6. После нажатия кнопки «Общая стоимость» открывается отдельно окно в котором рассчитывается общая стоимость всех товаров внесенных в список.

Согласно ГОСТ 19.701-90 [3] разработана блок-схема расчет бюджета школы и отображена на рисунке 4.1.

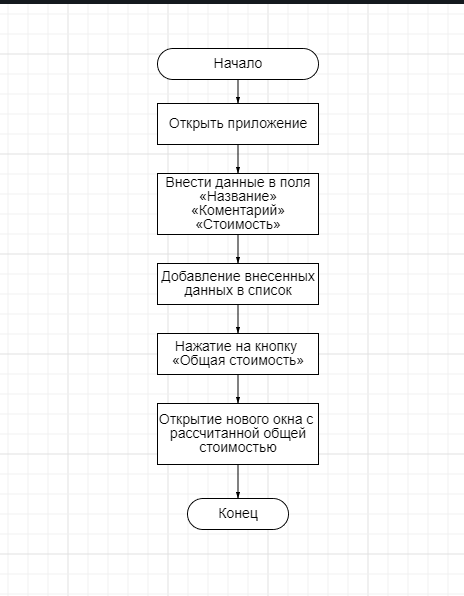


Рисунок 4.1 – Блок-схема расчета бюджета школы

Таким образом, блок-схема представляет алгоритм, который позволяет реализовать процесс расчета бюджета школы. Благодаря блок-схеме можно понять последовательность действий, которые необходимо выполнить для расчета бюджета школы, а также выявить возможные ошибки в процессе расчета бюджета школы.

# 5. Разработка модуля программы в интегрированной среде

Для данного программного продукта были использованы следующие характеристики:

1. IDLE Python

2. Python 3.12

Описание структуры модуля ПО:

1. Присутствует стартовое окно включающая в себя нужные для работы копки и формы выведения информации за это отвечает часть кода представленная в листинге 5.3 с названием «Создание окна».

2. При закрытии окна где вводится нужная пользователю информация, выводиться окно для закрытия окна часть кода отвечающая за данное действие представленная в листинге 5.2 с названием «Закрытие окна».

3. Для ввода информации используется несколько строк под названием «Комментарий» «Стоимость» «Название» листинг кода которых представлен в пункте 5.4 с названием «Создание строк для введения информации».

4. Для работы с программой были разработаны кнопки которые помогают пользоваться программой листинг кода представлен в пункте 5.1 с названием «Обработчики событий кнопок».

5. После ввода информации в отведенные для этого поля, и нажатии кнопки «Добавить» данные вносятся в специальное окно листинг кода которого был предоставлен в пункте 5.5 «Создание списка, где появятся наши покупки»

6. После ввода информации можно рассчитать общую стоимость всех покупок по кнопке под названием «Общая стоимость» листинг кода данной кнопки предоставлен в листинге кода 5.1 с названием «Обработчики событий кнопок»

**Листинг кода окна «Бюджет 1.0» разделяется на два пункта 5.1, 5.2, 5.3, 5.4, 5.5**

**В пункте 5.1 «Обработчики событий кнопок» будет представлен фрагмент кода предназначенный для обработки событий кнопок окна.**

# обработчик нажатия на кнопку «Посмотреть всё»

def view\_command():

list1.delete(0, END)

# проходим все записи в БД

for row in db.view():

# и сразу добавляем их на экран

list1.insert(END, row)

# обработчик нажатия на кнопку «Поиск»

def search\_command():

# очищаем список в приложении

list1.delete(0, END)

# находим все записи по названию покупки

for row in db.search(product\_text.get()):

# и добавляем их в список в приложение

list1.insert(END, row)

# обработчик нажатия на кнопку «Добавить»

def add\_command():

# добавляем запись в БД

db.insert(product\_text.get(), price\_text.get(), comment\_text.get())

# обновляем общий список в приложении

view\_command()

# обработчик нажатия на кнопку «Удалить»

def delete\_command():

# удаляем запись из базы данных по индексу выделенного элемента

db.delete(selected\_tuple[0])

# обновляем общий список расходов в приложении

view\_command()

# обработчик нажатия на кнопку «Обновить»

def update\_command():

# обновляем данные в БД о выделенной записи

db.update(selected\_tuple[0], product\_text.get(), price\_text.get())

# обновляем общий список расходов в приложении

view\_command()

# обрабатываем закрытие окна

def on\_closing():

# показываем диалоговое окно с кнопкой

if messagebox.askokcancel("", "Закрыть программу?"):

# удаляем окно и освобождаем память

window.destroy()

**В пункте 5.2 «Закрытие окна» будет представлен листинг кода обработчика события при нажатии на кнопку «Общая стоимость»**

def calculate\_total():

total = 0

for row in db.view():

total += float(row[2])

messagebox.showinfo("Общая стоимость",f "Общая стоимость: {total:.2f}")

**В пункте 5.3 «Создание окна» ниже будет представлен листинг кода подключения графической библиотеки для создания окна «Бюджет 1.0»**

# подключаем графическую библиотеку

window = Tk()

# заголовок окна

window.title("Бюджет 0.1")

**В пункте 5.4 «Создание строк для введения информации» ниже будет представлен листинг кода для создания определенных полей для ввода информации пользователем**

# создаём поле ввода названия покупки, говорим, что это будут строковые переменные и размещаем их тоже по сетке

product\_text = StringVar()

e1 = Entry(window, textvariable=product\_text)

e1.grid(row=0, column=1)

# то же самое для комментариев и цен

price\_text = StringVar()

e2 = Entry(window, textvariable=price\_text)

e2.grid(row=0, column=3)

comment\_text = StringVar()

e3 = Entry(window, textvariable=comment\_text)

e3.grid(row=1, column=1)

**В пункте 5.5 «Создание списка, где появятся наши покупки» ниже будет представлен листинг кода**

# создаём список, где появятся наши покупки, и сразу определяем его размеры в окне

list1 = Listbox(window, height=30, width=70)

list1.grid(row=2, column=0, rowspan=6, columnspan=2)

**Взаимодействие с компонентами:**

1. Эта часть кода отвечает за просмотр всех данных добавленных пользователем:

# обработчик нажатия на кнопку «Посмотреть всё»

def view\_command():

list1.delete(0, END)

for row in db.view():

list1.insert(END, row)

2. Эта часть кода отвечает за добавление данных внесенных пользователем:

# обработчик нажатия на кнопку «Добавить»

def add\_command():

db.insert(product\_text.get(), price\_text.get(), comment\_text.get())

view\_command()

3. Эта часть кода отвечает за поиск данных внесенных пользователем:

# обработчик нажатия на кнопку «Поиск»

def search\_command():

list1.delete(0, END)

for row in db.search(product\_text.get()):

list1.insert(END, row)

4. Эта часть кода отвечает за удаление записей внесенных пользователем:

def delete\_command():

db.delete(selected\_tuple[0])

view\_command()

5. Эта часть кода отвечает за обновление записей внесенных пользователем:

def update\_command():

db.update(selected\_tuple[0], product\_text.get(), price\_text.get())

view\_command()

6. Эта часть кода отвечает за расчет общей стоимости покупок внесенных пользователем:

def calculate\_total():

total = 0

for row in db.view():

total += float(row[2])

messagebox.showinfo("Общая стоимость", f"Общая стоимость: {total:.2f}")

**Интерфейс приложения представлен на рисунке ниже:**

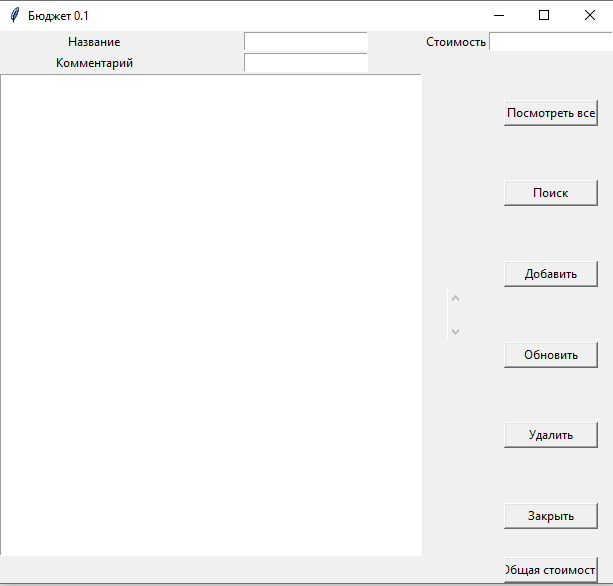


Рисунок 5.1 «Интерфейс окна»

# 6. Отладка и тестирование на работоспособность модуля

Для тестирования работоспособности модуля «Расчет бюджета школы» был использован ручной метод тестирования для проверки пользовательского интерфейса и функциональности приложения. Также был составлен и применен набор тестовых данных, который позволил проверить корректность работы различных функций приложения, включая добавление покупок, расчет общей стоимости и удаление покупок.  
Было произведено тестирование следующих событий: добавление покупок, удаление покупки и расчет общей стоимости покупок.

Результаты ручного тестирования:

1. Проверка правильности добавления покупки:

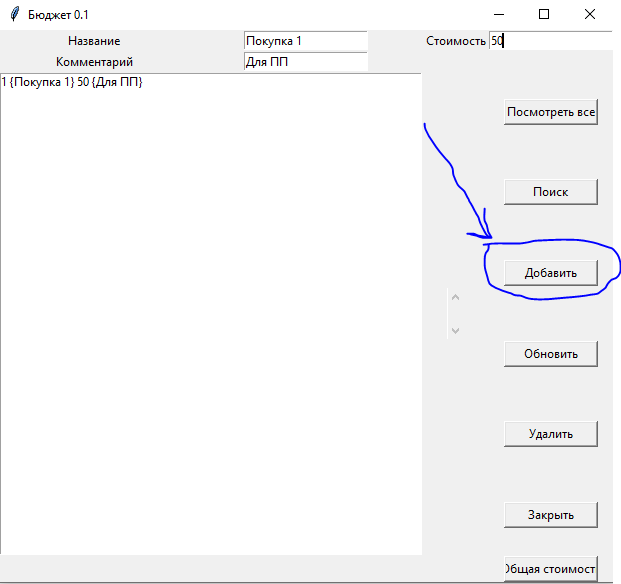


Рисунок 6.1 «Добавление покупки»

Итог: Проверка прошла успешно данные добавились корректно

2. Проверка правильности удаления покупки:

# C:\Новая папка (2)\Снимок3.PNG

# Рисунок 6.2 «Удаление покупки»

# Итог: Проверка прошла корректно, данные о покупке успешно удалились

# безвозвратно.

# 3. Проверка на правильность расчета общей стоимости покупок:

# C:\Новая папка (2)\Снимок4.PNG

# Рисунок 6.3 «Расчет общей стоимости»

# Итог: Проверка прошла корректно, общая стоимость покупок рассчиталась

# верно без ошибок.

# Заключение

В ходе производственной практики, проведенной в школе, была разработана программа на языке программирования Python. Целью практики было изучение основных принципов программирования и применение их на практике для создания полноценной программы.

В процессе работы над проектом были выполнены следующие этапы:

1. Изучение основных концепций языка программирования Python, таких как переменные, операторы, условные выражения, циклы и функции. Были изучены также особенности синтаксиса и структуры программ на Python.

2. Проектирование программы. Были определены требования к программе, ее функциональность и интерфейс. Была разработана структура программы и определены основные модули и классы.

3. Реализация программы. На основе разработанной структуры был написан код программы на языке Python. В процессе разработки использовались различные инструменты и библиотеки, специфичные для Python, для решения задач и упрощения разработки.

4. Тестирование программы. Были проведены тесты, чтобы убедиться в правильности работы программы и соответствии ее функциональности требованиям. В случае обнаружения ошибок или недочетов, в код программы вносились соответствующие исправления.

5. Документация и отчетность. Был подготовлен отчет о производственной практике, в котором были подробно описаны все этапы работы над проектом и достигнутые результаты.

В результате производственной практики была разработана полноценная программа на языке программирования Python, которая соответствует поставленным требованиям и выполняет свою функцию. Работа над проектом позволила углубить знания и навыки в области программирования, а также применить их на практике.

# Список использованных источников

1. ГОСТ 34.602-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы = Information technology. Set of standards for automated systems. Technical assignment for developing of automated system: межгосударственный стандарт утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 19 ноября 2021 г. № 1522-ст: введен взамен ГОСТ 34.602-89: дата введения 2022-01-01 / разработан Акционерным обществом «Всероссийский научно-исследовательский институт сертификации» и Обществом с ограниченной ответственностью «Информационно-аналитический вычислительный центр». – Москва: Российский институт стандартизации, 2022. – 9 с. – Текст непосредственный.
2. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения = Unified system for program documentation. Data, program and system flowcharts, program network charts and system resources charts. Documentation symbols and conventions for flowcharting: межгосударственный стандарт утвержден и введен в действие Постановлением Государственного комитета СССР по управлению качеством продукции и стандартам от 26.12.90 № 3294: введен взамен ГОСТ 19.002-80, ГОСТ 19.003-80: дата введения 1992-01-01 / разработан и внесен Государственным комитетом СССР по вычислительной технике и информатике. – Москва: Стандартинформ, 2010. – 158 с. – Текст непосредственный.
3. Грекул, В. И. Управление внедрением информационных систем : учебное пособие для СПО / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. — Саратов : Профобразование, 2021. — 277 c. — ISBN 978-5-4488-1016-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/102209.html (дата обращения: 13.06.2023). — Режим доступа: для авторизир. пользователей.
4. Иванова, О. Г. Методы и средства проектирования информационных систем и технологий. Основы UML : учебное пособие / О. Г. Иванова, Ю. Ю. Громов. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2020. — 80 c. — ISBN 978-5-8265-2308- 7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/115768.html (дата обращения: 13.06.2023). — Режим доступа: для авторизир. пользователей.
5. Кудинов, Ю. И. Основы алгоритмизации и программирования : учебное пособие для СПО / Ю. И. Кудинов, А. Ю. Келина. — 2-е изд. — Липецк, Саратов : Липецкий государственный технический университет, Профобразование, 2020. — 71 c. — ISBN 978-5-88247-956-4, 978-5-4488-0757-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/92834.html (дата обращения: 15.06.2023). — Режим доступа: для авторизир. пользователей.
6. Чурина, Т. Г. Основы алгоритмизации и программирования : учебное пособие для СПО / Т. Г. Чурина, Т. В. Нестеренко. — Саратов, Москва : Профобразование, Ай Пи Ар Медиа, 2020. — 214 c. — ISBN 978-5-4488-0802-9, 978-5-4497-0465-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/96017.html (дата обращения: 23.06.2023). — Режим доступа: для авторизир. пользователей.
7. Предоставление сведений из ЕГРЮЛ/ЕГРИП. – Текст: электронный // Федеральная налоговая служба: [сайт]. – URL: <https://egrul.nalog.ru/index.html> (дата обращения: 16.06.2023).

# Приложения

# подключаем библиотеку для работы с базой данных

import sqlite3

# подключаем графическую библиотеку для создания интерфейсов

from tkinter import \*

from tkinter import messagebox

# создаём класс для работы с базой данных

class DB:

# конструктор класса

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

# соединяемся с файлом базы данных

self.conn = sqlite3.connect("mybooks.db")

# создаём курсор для виртуального управления базой данных

self.cur = self.conn.cursor()

# если нужной нам таблицы в базе нет — создаём её

self.cur.execute(

"CREATE TABLE IF NOT EXISTS buy (id INTEGER PRIMARY KEY, product TEXT, price TEXT, comment TEXT)")

# сохраняем сделанные изменения в базе

self.conn.commit()

# деструктор класса

def \_\_del\_\_(self):

# отключаемся от базы при завершении работы

self.conn.close()

# просмотр всех записей

def view(self):

# выбираем все записи о покупках

self.cur.execute("SELECT \* FROM buy")

# собираем все найденные записи в колонку со строками

rows = self.cur.fetchall()

# возвращаем сроки с записями расходов

return rows

# добавляем новую запись

def insert(self, product, price, comment):

# формируем запрос с добавлением новой записи в БД

self.cur.execute("INSERT INTO buy VALUES (NULL,?,?,?)", (product, price, comment,))

# сохраняем изменения

self.conn.commit()

# обновляем информацию о покупке

def update(self, id, product, price):

# формируем запрос на обновление записи в БД

self.cur.execute("UPDATE buy SET product=?, price=? WHERE id=?", (product, price, id,))

# сохраняем изменения

self.conn.commit()

# удаляем запись

def delete(self, id):

# формируем запрос на удаление выделенной записи по внутреннему порядковому номеру

self.cur.execute("DELETE FROM buy WHERE id=?", (id,))

# сохраняем изменения

self.conn.commit()

# ищем запись по названию покупки

def search(self, product="", price=""):

# формируем запрос на поиск по точному совпадению

self.cur.execute("SELECT \* FROM buy WHERE product=?", (product,))

# формируем полученные строки и возвращаем их как ответ

rows = self.cur.fetchall()

return rows

# создаём новый экземпляр базы данных на основе класса

db = DB()

# заполняем поля ввода значениями выделенной позиции в общем списке

def get\_selected\_row(event):

# будем обращаться к глобальной переменной

global selected\_tuple

# получаем позицию выделенной записи в списке

index = list1.curselection()[0] #this is the id of the selected tuple

# получаем значение выделенной записи

selected\_tuple = list1.get(index)

# удаляем то, что было раньше в поле ввода

e1.delete(0, END)

# и добавляем туда текущее значение названия покупки

e1.insert(END, selected\_tuple[1])

# делаем то же самое с другими полями

e2.delete(0, END)

e2.insert(END, selected\_tuple[2])

e3.delete(0, END)

e3.insert(END, selected\_tuple[3])

# обработчик нажатия на кнопку «Посмотреть всё»

def view\_command():

# очищаем список в приложении

list1.delete(0, END)

# проходим все записи в БД

for row in db.view():

# и сразу добавляем их на экран

list1.insert(END, row)

# обработчик нажатия на кнопку «Поиск»

def search\_command():

# очищаем список в приложении

list1.delete(0, END)

# находим все записи по названию покупки

for row in db.search(product\_text.get()):

# и добавляем их в список в приложение

list1.insert(END, row)

# обработчик нажатия на кнопку «Добавить»

def add\_command():

# добавляем запись в БД

db.insert(product\_text.get(), price\_text.get(), comment\_text.get())

# обновляем общий список в приложении

view\_command()

# обработчик нажатия на кнопку «Удалить»

def delete\_command():

# удаляем запись из базы данных по индексу выделенного элемента

db.delete(selected\_tuple[0])

# обновляем общий список расходов в приложении

view\_command()

# обработчик нажатия на кнопку «Обновить»

def update\_command():

# обновляем данные в БД о выделенной записи

db.update(selected\_tuple[0], product\_text.get(), price\_text.get())

# обновляем общий список расходов в приложении

view\_command()

def calculate\_total():

total = 0

for row in db.view():

total += float(row[2])

messagebox.showinfo("Общая стоимость", f"Общая стоимость: {total:.2f}")

# подключаем графическую библиотеку

window = Tk()

# заголовок окна

window.title("Бюджет 0.1")

# обрабатываем закрытие окна

def on\_closing():

# показываем диалоговое окно с кнопкой

if messagebox.askokcancel("", "Закрыть программу?"):

# удаляем окно и освобождаем память

window.destroy()

# сообщаем системе о том, что делать, когда окно закрывается

window.protocol("WM\_DELETE\_WINDOW", on\_closing)

# создаём надписи для полей ввода и размещаем их по сетке

l1 = Label(window, text="Название")

l1.grid(row=0, column=0)

l2 = Label(window, text="Стоимость")

l2.grid(row=0, column=2)

l3 = Label(window, text="Комментарий")

l3.grid(row=1, column=0)

# создаём поле ввода названия покупки, говорим, что это будут строковые переменные и размещаем их тоже по сетке

product\_text = StringVar()

e1 = Entry(window, textvariable=product\_text)

e1.grid(row=0, column=1)

# то же самое для комментариев и цен

price\_text = StringVar()

e2 = Entry(window, textvariable=price\_text)

e2.grid(row=0, column=3)

comment\_text = StringVar()

e3 = Entry(window, textvariable=comment\_text)

e3.grid(row=1, column=1)

# создаём список, где появятся наши покупки, и сразу определяем его размеры в окне

list1 = Listbox(window, height=30, width=70)

list1.grid(row=2, column=0, rowspan=6, columnspan=2)

# на всякий случай добавим сбоку скролл, чтобы можно было быстро прокручивать длинные списки

sb1 = Scrollbar(window)

sb1.grid(row=2, column=2, rowspan=6)

# привязываем скролл к списку

list1.configure(yscrollcommand=sb1.set)

sb1.configure(command=list1.yview)

# привязываем выбор любого элемента списка к запуску функции выбора

list1.bind('<<ListboxSelect>>', get\_selected\_row)

# создаём кнопки действий и привязываем их к своим функциям

# кнопки размещаем тоже по сетке

b1 = Button(window, text="Посмотреть все", width=12, command=view\_command)

b1.grid(row=2, column=3) #size of the button

b2 = Button(window, text="Поиск", width=12, command=search\_command)

b2.grid(row=3, column=3)

b3 = Button(window, text="Добавить", width=12, command=add\_command)

b3.grid(row=4, column=3)

b4 = Button(window, text="Обновить", width=12, command=update\_command)

b4.grid(row=5, column=3)

b5 = Button(window, text="Удалить", width=12, command=delete\_command)

b5.grid(row=6, column=3)

b6 = Button(window, text="Закрыть", width=12, command=on\_closing)

b6.grid(row=7, column=3)

b7 = Button(window, text="Общая стоимость", width=12, command=calculate\_total)

b7.grid(row=8, column=3)

# обновляем общий список расходов

view\_command()

# пусть окно работает всё время до закрытия

window.mainloop()