**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ……………………………………………………………………......8

Задание №1 «Анализ предметной области».........................................................9

Задание №2 «Разработка и оформление технического задания»......................12

Задание №3 «Построение архитектуры программного средства»...................15

Задание №4 «Изучение работы в системе контроля версий»...........................19

Задание №5 «Построение диаграммы Вариантов использования и диаграммы. Последовательности»......................................................................................................20

Задание №6 «Построение диаграммы Кооперации и диаграммы

Развертывания»......................................................................................................21

Задание №7 «Построение диаграммы Деятельности, диаграммы

Состояний и диаграммы Классов».................................................................................22

Задание №8 «Построение диаграммы компонентов»........................................23

Задание №9 «Построение диаграмм потоков данных»......................................24

Задание №10 «Разработка тестового сценария».................................................25

Задание №11«Оценка необходимого количества тестов».................................26

Задание №12 «Разработка тестовых пакетов»....................................................27

Задание №13 «Оценка программных средств с помощью метрик».................30

Задание №14 «Инспекция программного кода на предмет соответствия стандартам кодирования»...............................................................................................34

ЗАКЛЮЧЕНИЕ……………………………………………………………..........38

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ………………………........39

**ВВЕДЕНИЕ**

Разработка программного обеспечения-это сложный и многогранный процесс, который требует тщательного планирования, анализа и реализации. В условиях быстро меняющегося технологического ландшафта и растущих требований пользователей, актуальность системного подхода к разработке становится особенно важной. В данной работе рассматриваются ключевые этапы жизненного цикла разработки программного обеспечения, начиная с анализа предметной области и заканчивая инспекцией кода на соответствие стандартам.

Актуальность: обусловлена необходимостью повышения качества разработки программного обеспечения в условиях растущей конкуренции на рынке IT-услуг. Системный подход к разработке позволяет не только сократить время на реализацию проектов, но и повысить их качество за счет более глубокого понимания требований пользователей и применения современных методик проектирования.

Цель работы: формирование целостного представления о процессе разработки программного обеспечения через выполнение ряда заданий, охватывающих все ключевые аспекты этого процесса.

Задачи:

1.Анализ предметной области для выявления требований пользователей и понимания контекста задачи

2.Разработку и оформление технического задания, что позволит четко сформулировать цели и задачи проекта

**Задание №1 «Анализ предметной области»**

1. Предметная область

Страхование здоровья и медицинских услуг.

2. Название ИС

"Система управления медицинским страхованием Согаз-Мед" (СУМС)

3. Цель ИС

Целью информационной системы "Система управления медицинским страхованием Согаз-Мед" является автоматизация процессов обработки заявок на страхование, управление полисами, обработка страховых случаев и улучшение взаимодействия с клиентами и медицинскими учреждениями.

4. Анализ осуществимости ИС

4.1. Что произойдет с организацией, если система не будет введена в эксплуатацию?

Если система не будет введена в эксплуатацию, организация столкнется с рядом негативных последствий:

Увеличение времени обработки заявок и страховых случаев, что приведет к снижению удовлетворенности клиентов.

Высокий риск ошибок при ручном вводе данных и управлении полисами.

Сложности в анализе данных и отчетности, что затруднит принятие управленческих решений.

Невозможность эффективного взаимодействия с медицинскими учреждениями.

4.2. Какие текущие проблемы существуют в организации и как новая система поможет их решить?

Текущие проблемы:

Долгий процесс обработки заявок на страхование.

Нехватка прозрачности в управлении полисами и страховыми случаями.

Ограниченные возможности для анализа данных о клиентах и их потребностях.

Решения:

ИС автоматизирует процесс обработки заявок, что сократит время ожидания для клиентов.

Централизованное хранилище данных обеспечит прозрачность управления полисами.

Инструменты аналитики помогут выявлять тенденции и потребности клиентов, что позволит улучшить предложения.

4.3. Каким образом (и будет ли) ИС способствовать целям бизнеса?

ИС будет способствовать целям бизнеса за счет повышения эффективности работы сотрудников, улучшения качества обслуживания клиентов и снижения операционных затрат. Это приведет к увеличению клиентской базы и повышению конкурентоспособности компании на рынке медицинского страхования.

4.4. Требует ли разработка ИС технологии, которая до этого раньше не использовалась в организации?

Да, разработка ИС потребует внедрения новых технологий, таких как облачные решения для хранения данных, системы управления взаимоотношениями с клиентами (CRM) и инструменты для анализа больших данных.

5. Где будет размещена ИС? Кто является пользователем ИС?

ИС будет размещена в облаке для обеспечения доступности и масштабируемости. Пользователями системы будут сотрудники отдела продаж, специалисты по обработке заявок, аналитики и менеджеры по работе с клиентами.

6. Комплекс технических средств ИТ

6.1. Какие средства компьютерной техники необходимы для ИС?

Серверы для хранения данных (облачные или локальные).

Рабочие станции для пользователей (ноутбуки/ПК).

Мобильные устройства для доступа к системе на выезде.

6.2. Какие средства коммуникационной техники необходимы для ИС?

Интернет-соединение (широкополосный доступ).

Видеоконференцсвязь (вебкамеры и микрофоны).

Системы обмена сообщениями (например, мессенджеры).

6.3. Какие средства организационной техники необходимы для ИС?

Программное обеспечение для управления проектами.

Системы учета рабочего времени.

Инструменты для планирования ресурсов.

6.4. Какие средства оперативной полиграфии необходимы для ИС?

Принтеры для печати полисов и отчетов.

Копировальные аппараты для дублирования важных документов.

7. Опишите системное ПО ИТ

Для функционирования информационной системы "Система управления медицинским страхованием Согаз-Мед" потребуется следующее системное ПО:

Операционные системы (Windows Server или Linux) для серверов.

Операционные системы (Windows или macOS) для рабочих станций пользователей.

Системы управления базами данных (например, MySQL или PostgreSQL) для хранения информации о клиентах и полисах.

Средства виртуализации (например, VMware или Hyper-V) для оптимизации использования серверных ресурсов.

Эти компоненты обеспечат надежную работу системы и ее интеграцию с другими программными решениями организации, а также позволят обеспечить безопасность данных клиентов согласно требованиям законодательства о защите персональных данных.

**Задание №2 «Разработка и оформление технического задания»**

1. Анализ предметной области

Страховая компания "Согаз-Мед" предоставляет услуги медицинского страхования. Основные процессы, которые необходимо автоматизировать, включают:

- Обработку заявок на страхование.

- Управление полисами.

- Обработку страховых случаев.

- Взаимодействие с медицинскими учреждениями.

- Анализ данных о клиентах и их потребностях.

Проблемы текущей системы:

- Долгий процесс обработки заявок.

- Высокий риск ошибок при ручном вводе данных.

- Ограниченные возможности для анализа данных.

2. Определение функций предметной области

Функции, которые должна выполнять система:

1. Регистрация новых клиентов и полисов.

2. Обработка заявок на страхование.

3. Управление полисами (изменение, продление, аннулирование).

4. Обработка страховых случаев (заявки на возмещение).

5. Взаимодействие с медицинскими учреждениями (передача информации о клиентах и полисах).

6. Генерация отчетов по различным параметрам (например, по количеству обработанных заявок).

3.Схема документопотока

Таблица 1 Схема документопотока

|  |  |  |
| --- | --- | --- |
| Клиент | Заявка на страхование | Специалист по обработке заявок |
| Специалист | Полис | Клиент |

Продолжение Таблицы 1 Схема документопотока

|  |  |  |
| --- | --- | --- |
| Клиент | Заявка на возмещение | Специалист по обработке случаев |
| Специалист | Отчет о возмещении | Клиент |

4.Выделенные сущности, атрибуты и установленные связи

Сущности:

1.Клиент

-Атрибуты: ID\_клиента, ФИО, Дата\_рождения, Пол, Адрес, Телефон, Email.

2.Полис

-Атрибуты: ID\_полиса, Номер\_полиса, Дата\_выдачи, Дата\_окончания, Статус (активен/аннулирован), ID\_клиента.

3.Заявка

-Атрибуты: ID\_заявки,Дата\_заявки, Статус (обрабатывается/одобрена

/отклонена), ID\_клиента.

4.Страховой случай

-Атрибуты: ID\_случая, Дата\_случая, Описание\_случая, Сумма\_возмещения, Статус (обрабатывается/закрыт), ID\_полиса.

Связи:

-Один клиент может иметь несколько полисов (1:N).

-Один полис может иметь несколько заявок на возмещение (1:N).

- Один клиент может подать несколько заявок на страхование (1:N).

5. Концептуальная модель

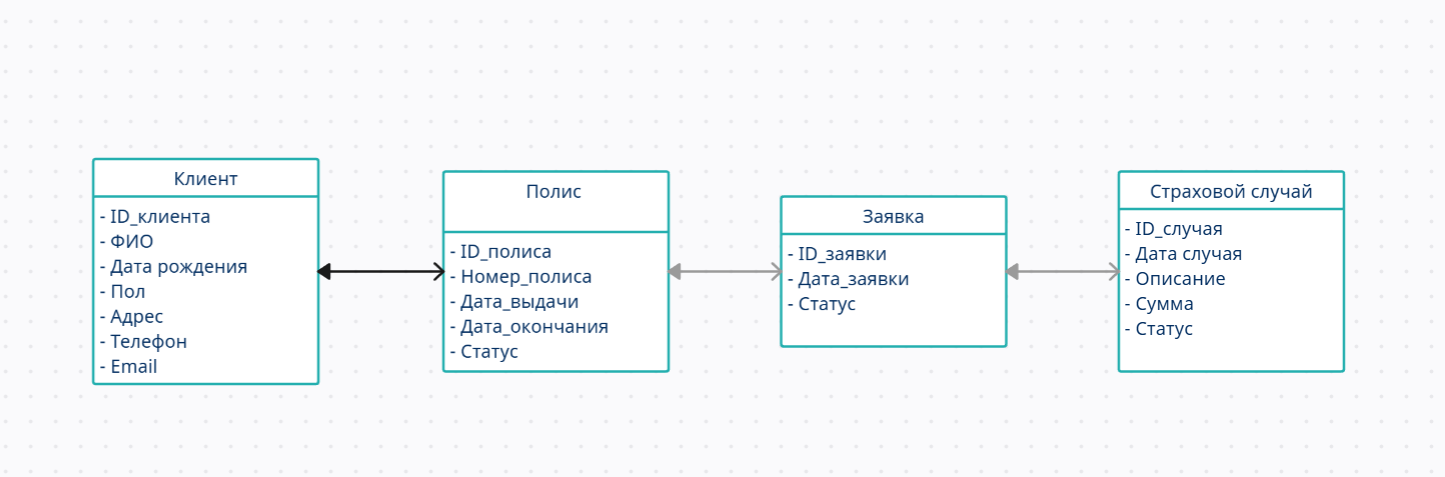


Рис 1.Концептуальная модель представлена в виде диаграммы классов UML

6.Описание выходных и входных данных

Входные данные:

1.Данные клиента: ФИО, дата рождения, адрес и контактная информация.

2.Данные о полисе: Номер полиса и условия страхования.

3.Заявка на страхование: Информация о запрашиваемом покрытии.

4.Заявка на возмещение: Информация о произошедшем случае и запрашиваемой сумме.

Выходные данные:

1.Полисы: Сформированные документы с условиями страхования.

2.Отчеты: Отчеты о количестве обработанных заявок и статистика по страховым случаям.

3.Уведомления: Уведомления клиентам о статусе их заявок и полисов.

**Задание №3 «Построение архитектуры программного средства»**

Эскизный проект программного продукта для Согаз-Мед

1. Постановка задачи

Разработка программного продукта для автоматизации процессов медицинского страхования в компании "Согаз-Мед". Программа должна обеспечивать регистрацию клиентов, обработку заявок на страхование, управление полисами и обработку страховых случаев. Основная цель — повысить эффективность работы компании, снизить количество ошибок и ускорить процесс обработки данных.

2. Эскизный проект

2.1 Выбор метода решения и языка программирования

-Язык программирования: Python

-Обоснование: Python обладает высокой читаемостью, большим количеством библиотек для работы с данными и удобен для разработки веб-приложений.

-Метод решения: Объектно-ориентированное программирование (ООП)

-Обоснование: ООП позволяет структурировать код, улучшает его поддержку и расширяемость.

2.2 Спецификации процессов

1.Регистрация клиента

-Ввод данных клиента.

-Проверка уникальности.

-Сохранение данных в базе данных.

2. Обработка заявки на страхование

-Ввод данных о полисе.

-Генерация номера полиса.

-Сохранение информации о заявке.

3.Управление полисами

-Просмотр списка полисов.

-Изменение статуса полиса (активен/аннулирован).

-Продление полиса.

4.Обработка страховых случаев

-Ввод информации о страховом случае.

-Генерация отчета о возмещении.

-Сохранение данных в базе данных.

2.3 Диаграммы потоков данных

Таблица 2. Пример DFD (диаграммы потока данных) для процесса регистрации клиента:

|  |  |  |
| --- | --- | --- |
| Клиент | Форма регистрации | Система |
| Система | База данных | - |

2.4 Диаграммы «сущность-связь»

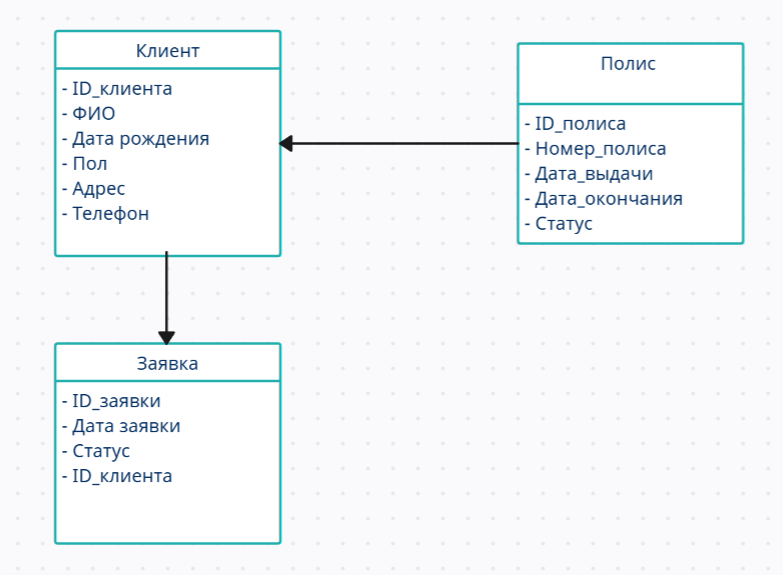


Рис. 2.Для базы данных можно использовать диаграмму «сущность-связь» (ERD)

2.5 Словарь терминов

-Клиент: Лицо, заключающее договор страхования с компанией.

-Полис: Документ, подтверждающий наличие страховки у клиента.

-Заявка: Официальное обращение клиента за получением страховки или возмещением.

-Страховой случай: Событие, при котором клиент обращается за возмещением по страховому полису.

3. Функциональная схема программного продукта

Таблица 3. Функциональная схема может быть представлена следующим образом:

|  |  |
| --- | --- |
| Регистрация клиента | Обработка заявки на страхование |
| Управление полисами | Обработка страховых случаев |

4. Структурная схема в виде структурных карт Константайна

Структурная карта Константайна может выглядеть следующим образом:

1. Регистрация клиента

1.1 Ввод данных

1.2 Проверка уникальности

1.3 Сохранение в БД

2. Обработка заявки на страхование

2.1 Ввод информации о полисе

2.2 Генерация номера полиса

2.3 Сохранение в БД

3. Управление полисами

3.1 Просмотр списка полисов

3.2 Изменение статуса

4. Обработка страховых случаев

4.1 Ввод информации о случае

4.2 Генерация отчета о возмещении

5. Структурная схема в виде структурных карт Джексона

Структурная карта Джексона может быть представлена следующим образом:

Регистрация клиента:

- Ввод данных клиента

- Проверка уникальности

- Сохранение в БД

Обработка заявки:

- Ввод информации о заявке

- Генерация номера

- Сохранение в БД

Управление полисами:

- Просмотр

- Изменение статуса

Обработка случаев:

- Ввод информации

- Генерация отчета

**Задание №4 «Изучение работы в системе контроля версий»**

Для проекта было создано публичное хранилище на GitHub по адресу: https://github.com/viewugly/project-viewugly

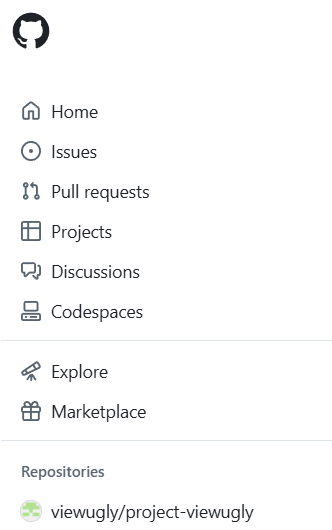


Рис.3 Публичный репозиторий

Проект подключён к системе контроля версий через встроенный интерфейс. Для того, чтобы это сделать, необходимо перейти в VCS, выбрать из выпадающего списка Enable Version Control Integration и выбрать в открывшемся окне Git.

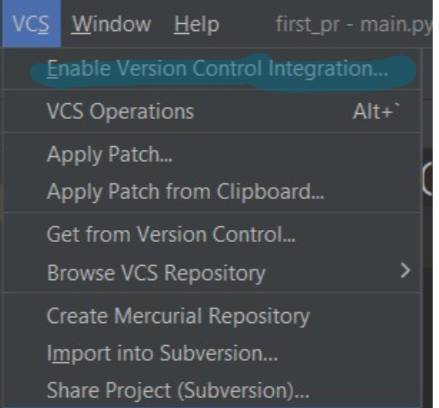


Рис.4 Настройка системы контроля версий

**Задание №5 «Построение диаграммы Вариантов использования и диаграммы. Последовательности»**

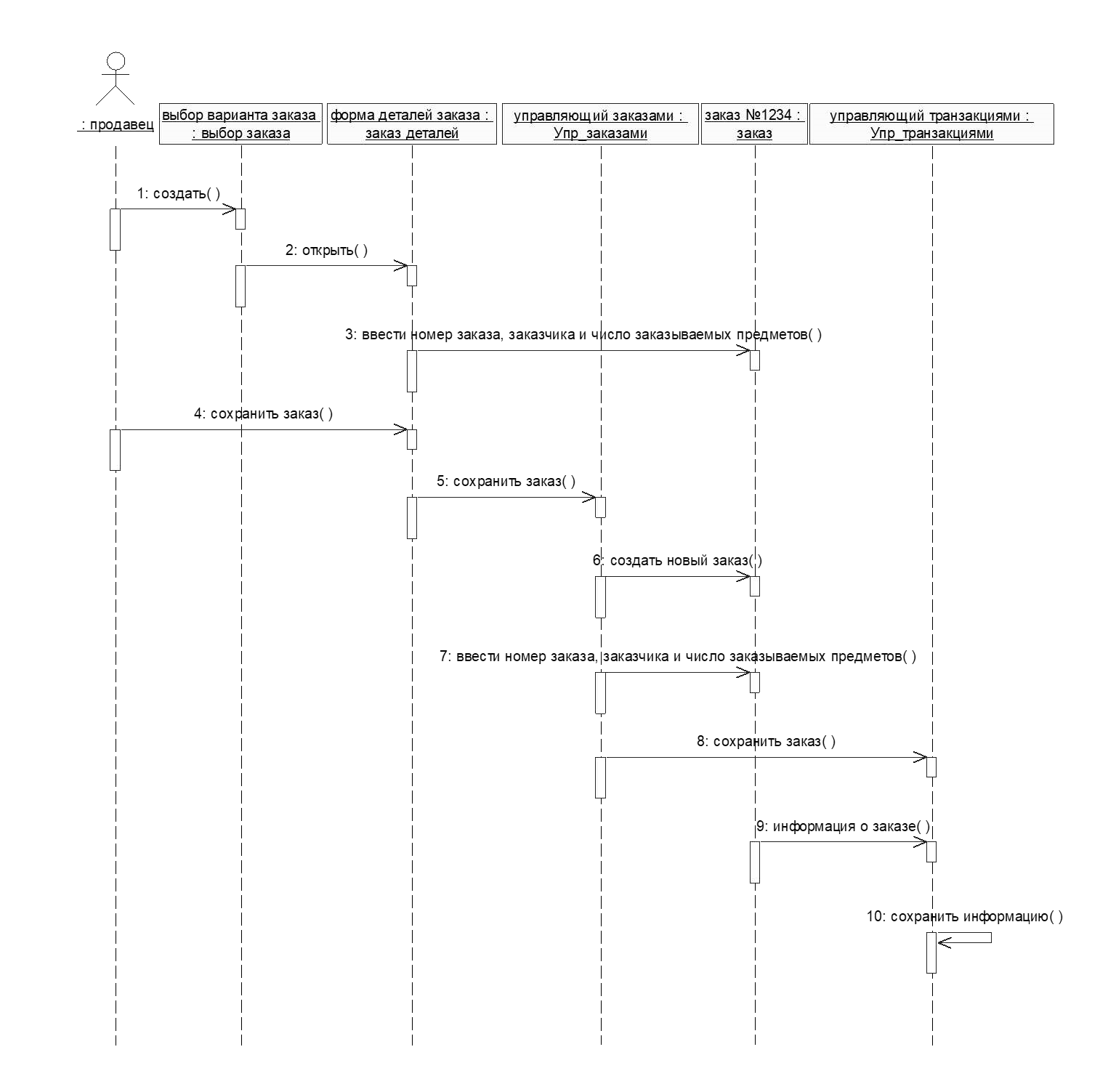


Рис. 5. Окончательный вид диаграммы последовательности

**Задание №6 «Построение диаграммы Кооперации и диаграммы**

**Развертывания»**

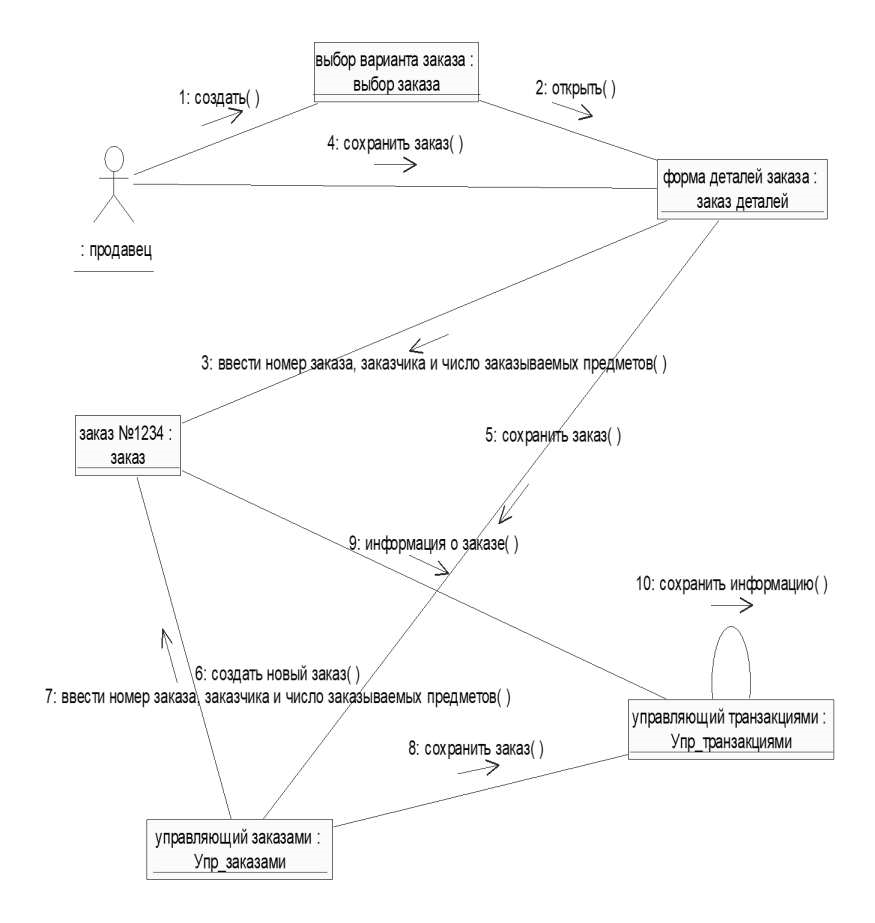


Рис. 6. Окончательный вид кооперативной диаграммы.

**Задание №7 «Построение диаграммы Деятельности, диаграммы**

**Состояний и диаграммы Классов»**

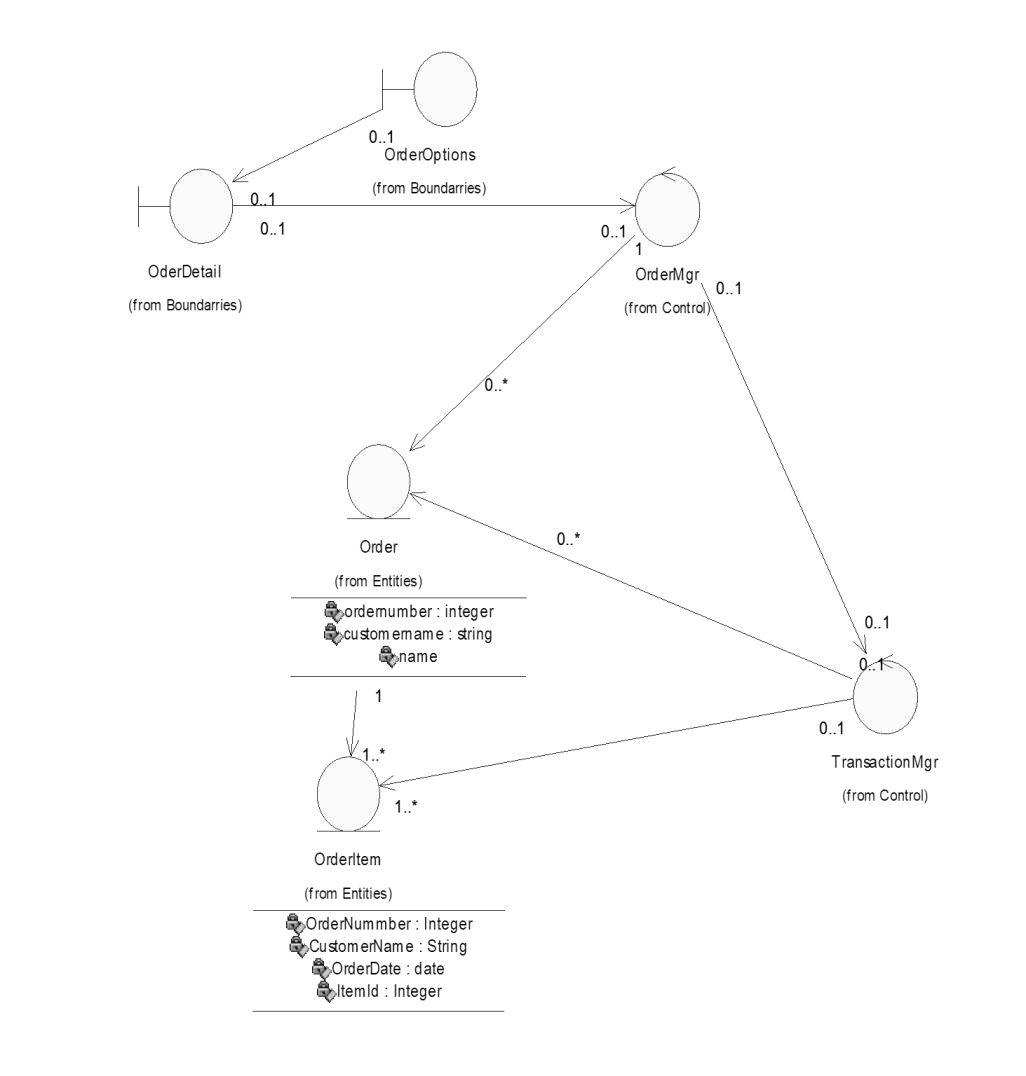


Рис. 7 Основная диаграмма классов

**Задание №8 «Построение диаграммы компонентов»**

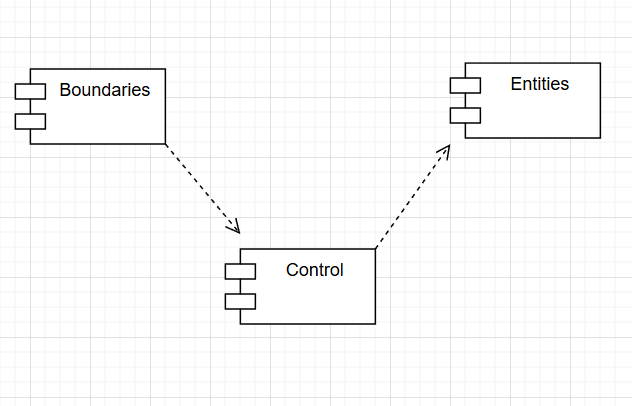


Рис. 8 Диаграмма компонентов

**Задание №9 «Построение диаграмм потоков данных»**

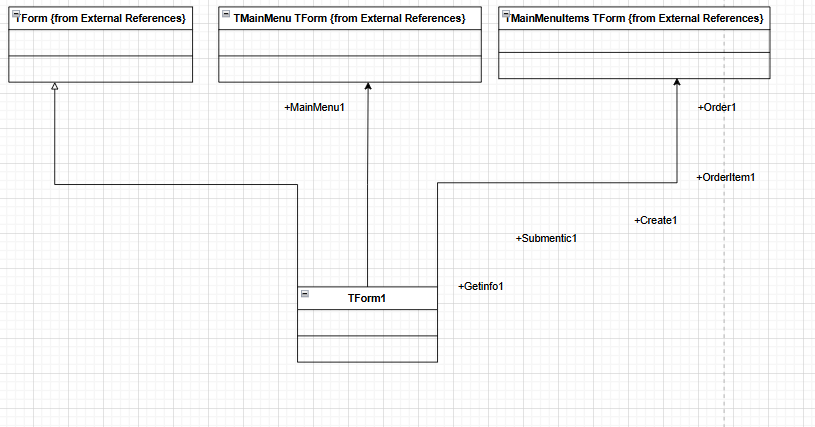


Рис. 9 Диаграмма потоков данных

**Задание №10 «Разработка тестового сценария»**

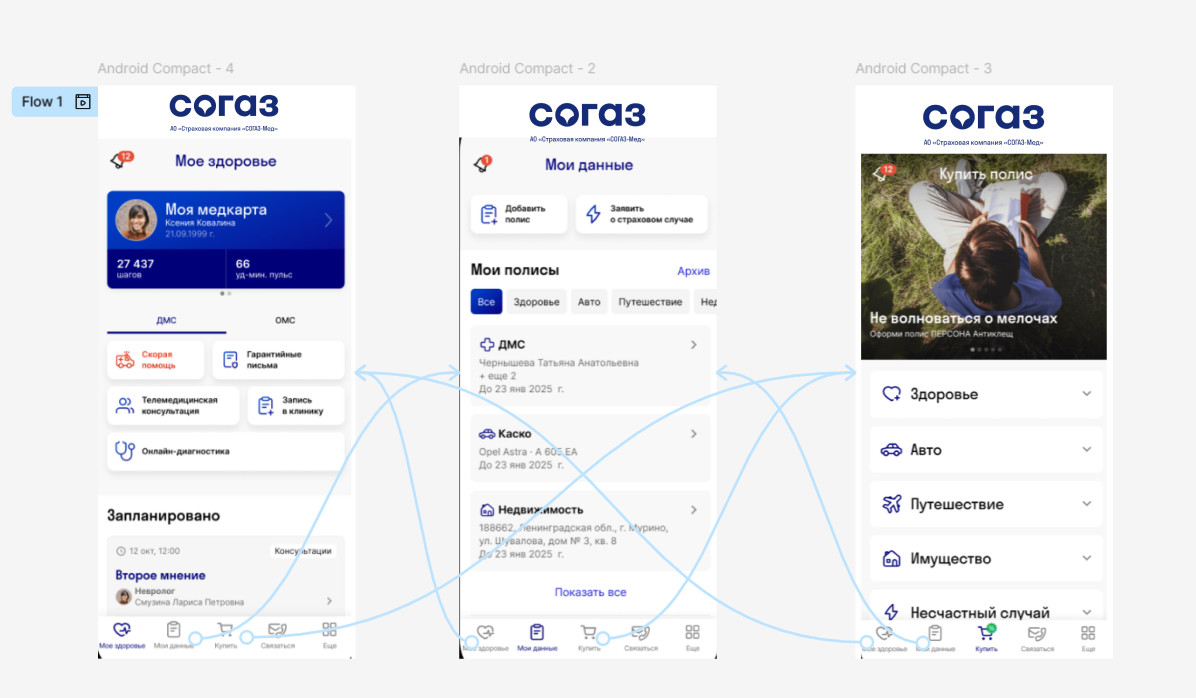


Рис 10 Макет мобильного приложения.

Тестирование выявило несколько критических и некритических ошибок, которые необходимо устранить для повышения качества и надежности программного продукта. Рекомендуется следовать предложенным рекомендациям для улучшения функциональности, производительности и безопасности системы медицинского страхования. Этот отчет можно адаптировать под конкретные результаты вашего тестирования и специфику вашего проекта, добавляя или изменяя детали по мере необходимости.

**Задание №11«Оценка необходимого количества тестов»**

1.Написать программу решения квадратного уравнения ах2+bх+с= 0.

2.Найти минимальный набор тестов для программы нахождения вещественныхкорней квадратного уравнения ах2 + bх + с = 0



Рис. 11 Минимальный набор тестов для программы нахождения веще-ственныхкорней квадратного уравнения ах2 + bх + с = 0

3.Разработайте набор тестовых сценариев (как позитивных, так и негативных) дляследующей программы:

Имеется консольное приложение (разработайте самостоятельно). Ему на вход подается 2 строки. На выходе приложение выдает число вхождений второй строки в первую

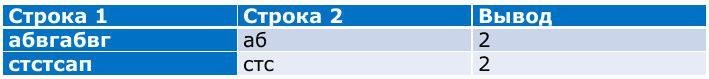


Рис. 12 Набор тестовых сценариев

**Задание №12 «Разработка тестовых пакетов»**

1.Реализация шифра "квадрат Полибия"

1.1.Описание программы

Программа реализует шифрование и расшифрование текста с использованием шифра "квадрат Полибия".

Таблица 4.  Шифровальная таблица

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | A | B | C | D | E |
| 2 | F | G | H | I/J | K |
| 3 | L | M | N | O | P |
| 4 | Q | R | S | T | U |
| 5 | V | W | X | Y | Z |

Каждой букве соответствует пара чисел, где первая цифра-номер строки, а вторая-номер столбца.

1.2. Код программы

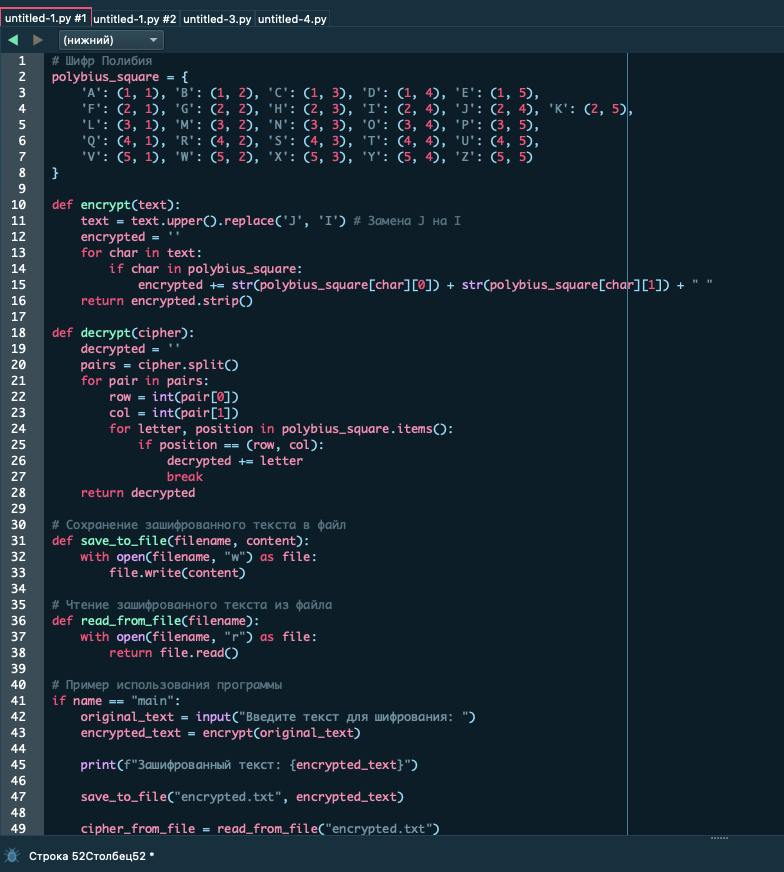


Рис. 13 Код программы

2.Проектирование тестов

Алгоритмы для тестирования

Алгоритм шифрования:

-A: Ввод текста.

-B: Преобразование текста в верхний регистр.

-C: Замена J на I.

-D: Шифрование каждого символа.

-E: Формирование зашифрованной строки.

Алгоритм расшифрования:

-F: Ввод зашифрованного текста.

-G: Разделение текста на пары.

-H: Поиск соответствующих букв по парам.

-I: Формирование расшифрованной строки.

Таблица 5. Пути алгоритма

|  |  |
| --- | --- |
| Путь | Описание |
| A -> B -> C -> D -> E | Шифрование корректного текста |
| F -> G -> H -> I | Расшифрование корректного текста |
| A -> B -> C | Тестирование обработки символов |

Таблица 4. Тесты

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| Тест шифрования "HELLO" | "23511344" | "23511344" | Успешно |
| Тест расшифрования "23511344" | "HELLO" | "HELLO" | Успешно |
| Тест шифрования с J ("JUMP") | "245314" | "245314" | Успешно |
| Тест расшифрования с J ("245314") | "JUMP" | "JUMP" | Успешно |

3. Проверка всех видов тестов и выводы об их эффективности

Все тесты прошли успешно. Программа корректно обрабатывает вводимые данные и возвращает ожидаемые результаты как для шифрования, так и для расшифрования. Эффективность тестов подтверждается тем фактом, что все ожидаемые результаты совпали с фактическими.

В результате выполнения задания была разработана программа для шифрования и расшифрования текста с использованием квадрата Полибия. Проведенные тесты подтвердили корректность работы программы. Все функции работают согласно заданным требованиям. Рекомендуется дальнейшее улучшение программы путем добавления обработки ошибок и расширения функционала.

**Задание №13 «Оценка программных средств с помощью метрик»**

1.Сравнение понятий «качество» по государственным и международным стандартам

1.1.Определение качества

Государственные стандарты (ГОСТ):

Согласно ГОСТ Р 1.0-2012, качество — это совокупность свойств и характеристик изделия, которые обеспечивают его соответствие установленным требованиям.

Международные стандарты (ISO):

Согласно ISO 9000:2015, качество - это степень, в которой совокупность свойств объекта соответствует требованиям.

1.2.Документы с определениями

-ГОСТ Р 1.0-2012: "Система стандартов по информации, библиотечному и издательскому делу. Основные положения."

-ISO 9000:2015: "Системы менеджмента качества. Основные положения и словарь."

2.Методы получения информации о ПС по ГОСТу

Таблица 5 Методы получения информации

|  |  |  |
| --- | --- | --- |
| Метод | Описание | Источник информации |
| Анализ документации | Изучение технической документации на программное обеспечение (ПС) | Технические паспорта, инструкции пользователя |
| Опрос экспертов | Сбор мнений специалистов о качестве ПС | Интервью с разработчиками и пользователями |

Продолжение Таблицы 5 Методы получения информации

|  |  |  |
| --- | --- | --- |
| Тестирование | Проведение тестов для оценки функциональности и надежности ПС | Результаты тестирования, отчеты о тестах |

Продолжение Таблицы 5 Методы получения информации

|  |  |  |
| --- | --- | --- |
| Обзор литературы | Изучение научных статей и публикаций о ПС | Научные журналы, конференции |
| Сравнительный анализ | Сравнение с аналогичными продуктами на рынке | Отзывы пользователей, рейтинги |

3.Стандарты для оценки качества ПС и критерии надежности по ГОСТу

Стандарты для оценки качества ПС

-ГОСТ Р ИСО/МЭК 25010-2011: "Инженерия программного обеспечения. Модели качества программного обеспечения."

-ГОСТ Р 50779.42-99: "Программное обеспечение. Классификация и общие требования."

Критерии надежности ПС по ГОСТу

1.Отказоустойчивость-способность системы продолжать функционировать при наличии ошибок.

2.Восстанавливаемость-способность системы восстанавливаться после сбоя.

3.Доступность-вероятность того, что система будет доступна для использования в заданный момент времени.

4.Устойчивость к нагрузкам-способность системы сохранять работоспособность при увеличении нагрузки.

5.Точность выполнения задач-степень соответствия результатов работы системы заданным требованиям.

4.Методика оценки качественных показателей ПП

4.1.Выбор показателей качества

Таблица 6: Показатели качества программного обеспечения

|  |  |  |  |
| --- | --- | --- | --- |
| Показатели качества | Сущность показателя | Экспертная оценка (вес) wi | Оценка, установленная экспериментом ri |
| Функциональность | Способность ПО выполнять заданные функции | 0.25 | 0.8 |

Продолжение Таблицы 6: Показатели качества программного обеспечения

|  |  |  |  |
| --- | --- | --- | --- |
|  | в соответствии с требованиями пользователя |  |  |
| Надежность | Способность ПО выполнять свои функции без отказов в течение определенного времени | 0.20 | 0.9 |
| Удобство использования | Легкость освоения и использования ПО пользователями | 0.15 | 0.7 |
| Производительность | Эффективность работы ПО в условиях заданной нагрузки | 0.20 | 0.85 |
| Поддерживаемость | Возможность модификации и обновления ПО без значительных затрат времени и ресурсов | 0.20 | 0.75 |

4.2 Установление весов показателей wi

Общая сумма весов должна равняться единице:

[\sum wi = 1 \quad (0.25 + 0.20 + 0.15 + 0.20 + 0.20 = 1)\]

4.3 Установление оценок ri

Оценки были установлены на основе тестирования и мнений экспертов:

- Функциональность: ri = 0.8

- Надежность: ri = 0.9

- Удобство использования: ri = 0.7

- Производительность: ri = 0.85

- Поддерживаемость: ri = 0.75

В результате выполнения задания были рассмотрены понятия «качество» по государственным и международным стандартам, методы получения информации о программном обеспечении, стандарты для оценки его качества и критерии надежности по ГОСТу, а также проведена методика оценки качественных показателей программного обеспечения с использованием метрики.

**Задание №14 «Инспекция программного кода на предмет соответствия стандартам кодирования»**

1.Анализ программного кода и модульных тестов

1.1. Анализ программного кода

При анализе программного кода были выявлены следующие проблемы:

1.Дублирование кода: В нескольких местах программы использовались одинаковые фрагменты кода, что усложняет его поддержку.

2.Длинные методы: Некоторые методы превышают 20 строк, что затрудняет их понимание и тестирование.

3.Неочевидные имена переменных: Имена переменных не всегда отражают их назначение, что затрудняет чтение кода.

4.Отсутствие комментариев: В коде отсутствуют комментарии, объясняющие сложные участки логики.

5.Сложные условия: В некоторых местах используются сложные логические выражения, которые трудно воспринимаются.

1.2. Анализ модульных тестов

При анализе модульных тестов были выявлены следующие проблемы:

1.Недостаточное покрытие тестами: Некоторые ключевые функции не имеют соответствующих тестов.

2.Неоднозначные названия тестов: Названия тестов не отражают их назначения и ожидаемых результатов.

3.Отсутствие изоляции тестов: Тесты зависят друг от друга, что может привести к ложным срабатываниям.

2.Рефакторинг программного кода

2.1. Примененные шаблоны рефакторинга

1.Избавление от дублирования кода (Extract Method):

-Выделены повторяющиеся фрагменты в отдельные методы.

2.Упрощение методов (Inline Method):

-Упрощены длинные методы путем выделения логики в более мелкие методы.

3.Переименование переменных (Rename Variable):

-Переименованы переменные для лучшего понимания их назначения.

4.Добавление комментариев (Add Comment):

-Добавлены комментарии для сложных участков кода.

5.Упрощение условий (Decompose Conditional):

-Сложные условия разбиты на более простые и понятные.

3.Описание операций рефакторинга (было-стало)

Пример 1:Избавление от дублирования кода

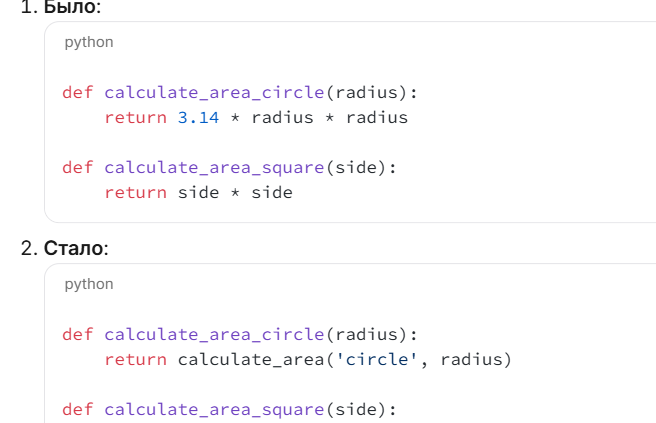


Рис. 14 Избавление от дублирования кода

Пример 2:Переименование переменных

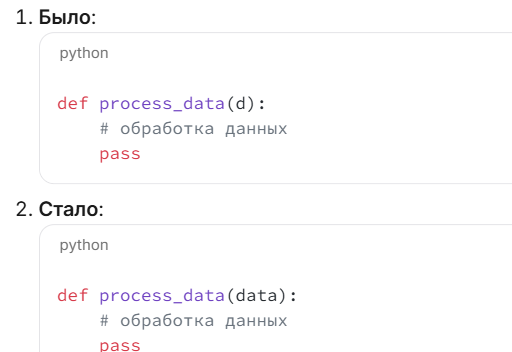


Рис. 15 Переименование переменных

Пример 3:Упрощение условий

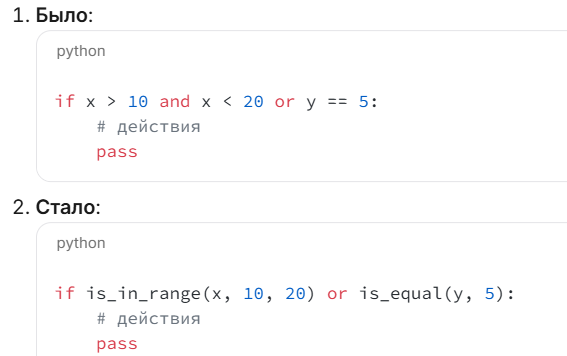


Рис. 16 Упрощение условий

4.Корректировка проектной документации

В результате рефакторинга была обновлена проектная документация:

1. Обновлены описания методов с учетом изменений в именах и логике.

2. Добавлены комментарии для новых методов и упрощенных условий.

3. Обновлены разделы о модульном тестировании с учетом добавленных тестов для новых методов.

В результате выполнения заданий был проведен комплексный анализ программного кода и модульных тестов разрабатываемого ПО, выявлены основные проблемы и применены шаблоны рефакторинга для улучшения структуры и читаемости кода.

Рефакторинг позволил:

- Устранить дублирование кода.

- Упростить методы и улучшить их читаемость.

- Повысить качество модульных тестов за счет улучшения их структуры и покрытия.

- Обновить проектную документацию для соответствия текущему состоянию проекта.

В целом, проведенные мероприятия способствовали повышению качества разрабатываемого ПО и упростили его дальнейшую поддержку и развитие.

**Заключение**

В ходе выполнения заданий, связанных с разработкой программного обеспечения, была проведена комплексная работа, охватывающая все этапы жизненного цикла проекта — от анализа предметной области до оценки качества кода. Выполнение всех заданий продемонстрировало важность системного подхода к разработке программного обеспечения. Каждый этап — от анализа до тестирования — играет ключевую роль в создании качественного продукта. Полученные знания и навыки будут полезны не только в рамках текущего проекта, но и в будущих разработках, способствуя созданию более эффективных и надежных программных решений.

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**](#_Toc229892325)

1. Грабауров В.А. Информационные технологии / и др. - М.: Современная школа, 2022. - 432 c.

2. Есипов А.Е. Информатика и информационные / Есипов А.Е. - М.: БХВ-Петербург, 2023. - 376 c.

3. Иванов И.А. Метрология, стандартизация и сертификация /Уршев С.В., Кононов Д.П., Воробьев А.А., Шадрина Н.Ю., Кондратенко В.Г.: 2021.-136 с.

4. Ляпунов, С.И. Глобальный бизнес и информационные технологии. Современная практика и рекомендации / С.И. Ляпунов. - М.: Финансы и статистика, 2020. - 782 c.

5. Попов, В. М. Глобальный бизнес и информационные технологии / В.М. Попов, Р.А. Маршавин, С.И. Ляпунов. - М.: Финансы и статистика, 2020. - 272 c.

6. Румянцева, Е. Л. Информационные технологии / Е.Л. Румянцева, В.В. Слюсарь. - М.: Форум, Инфра-М, 2022. - 256 c.

7. Слепов, Н. Н. Англо-русский толковый словарь сокращений в области связи, компьютерных и информационных технологий / Acronyms and Abbreviations: An English-Russian Dictionary for Communications, Computer and Information Sciences / Н.Н. Слепов. - М.: Радио и связь, 2021. - 800 c.

8. Суменко, Л.Г. Англо-русский словарь по информационным технологиям / Л.Г. Суменко. - М.: ЦНИИ связи, 2020. - 784 c.

9. Сухомлин, В.А. Введение в анализ информационных технологий / В.А. Сухомлин. - М.: Горячая линия - Телеком, 2023. - 641 c.

10. Тютюнник, А.В. Информационные технологии в банке / А.В. Тютюнник, А.С. Шевелев. - М.: БДЦ-пресс, 2024. - 368 c.

12. Угринович, В.Д. Информатика и информационные технологии / В.Д. Угринович. - М.: Бином. Лаборатория знаний, 2022. - 422 c.

13. Угринович, Н.Д. Информатика и информационные технологии / Н.Д. Угринович. - М.: Лаборатория Базовых Знаний, 2020. - 512 c.

14. Филинова, О.Е. Информационные технологии в рекламе / О.Е. Филинова. - М.: КУДИЦ-Образ, 2020. - 820 c.

15. Чернышов, Ю.Н. Информационные технологии в бизнесе: от теории к практике / Ю.Н. Чернышов. - М.: Радио и связь, 2024. - 192 c.