Оглавление

Цели и задачи лабораторной работы

Целью работы является приобретение навыков моделирования поведения компонентов программных средств с использованием канонических диаграмм состояний и деятельности языка UML.

Решаемые в ходе выполнения работы задачи:

– построение диаграммы состояний выбранных объектов в рамках определенного программного продукта;

– построение диаграммы деятельности для выбранных алгоритмов или методик, реализуемых в рамках выбранного программного продукта.

Описание предметной области

Вариант №7. Графический текстовый редактор.

В данном случае в качестве искомого программного продукта будет рассматриваться простейший вариант графического текстового редактора, представляющий собой оконное приложение, которое позволяет загружать, просматривать и сохранять текстовые файлы с возможностью поиска и замены слов и выражений.

Графический интерфейс программы подразумевает интуитивно понятное управление с командами меню и дублированием основных кнопок на панели инструментов основного окна.

Также в случае невозможности выполнения той или иной команды меню в текущий момент времени предусмотрен вывод информационного окна с соответствующим сообщением.

1. РАЗРАБОТКА ДИАГРАММ СОСТОЯНИЙ

В рамках выбранной предметной области (в контексте диаграммы состояний UML) выбраны два объекта, для которых характерны наличие нескольких (не менее пяти) состояний на разных стадиях работы разрабатываемого программного продукта и переходов между ними.

1.1 Объект «Контроллер замены текста»

В данном случае моделируется метод search класса GeneralController, представляющий собой алгоритм обработки запроса на замену текста.

В ходе штатной работы программного продукта могут наблюдаться следующие состояния (рис. 1.1).

1) Ожидание – простое состояние, в рамках которого редактор ждет, когда пользователь инициирует процедуру замены текста. Нахождение объекта в данном состоянии фактически запускается при старте приложения.

2) Замена текста – составное сложное состояние, представляющее собой собственно процедуру замены текста. Нахождение объекта в данном состоянии запускается при вызове единого метода change класса GeneralController опосредованно либо через метод change класса FxmlController, либо с помощью нажатия горячих клавиш через метод start класса Main и конструктор класса KeyCode. Данное состояние включает в себя шесть подсостояний:

a) Проверка наличия открытой вкладки – с момента запуска контроллера до момента получения результатов проверки. Проверка проводится в теле метода change класса GeneralController. В зависимости от результатов проверки возможны два варианта – либо открытие окна предупреждения (при отсутствии вкладок), либо открытие окна замены (при наличии хотя бы одной вкладки).
b) Предупреждение пользователя – с момента открытия окна предупреждения до момента закрытия окна предупреждения.

Нахождение объекта в данном состоянии осуществляется в теле метода change класса GeneralController.

c) Ввод текста для поиска и замены – с момента открытия окна замены либо до момента завершения ввода данных, либо до момента отмены ввода (закрытия окна). Нахождение объекта в данном состоянии осуществляется в теле метода change класса GeneralController.
d) Ожидание подтверждения поиска и замены – с момента завершения ввода данных либо до момента подтверждения процедуры замены (кнопка «Заменить»), либо до момента отмены ввода (закрытия окна). Нахождение объекта в данном состоянии осуществляется в теле метода change класса GeneralController.
e) Замена текста – с момента подтверждения процедуры замены (кнопка «Заменить») до момента завершения всех замен и открытия окна результатов замены. Данное состояние тождественно основному алгоритму замены текста. Нахождение объекта в данном состоянии осуществляется в теле метода change класса GeneralController. При этом в экземпляре класса TextTabModel вызываются методы setTemplateSearch, setTemplateChange, setCount и setNewMainText.
f) Вывод результатов замены – с момента завершения алгоритма замены и открытия окна результатов до закрытия окна результатов. Нахождение объекта в данном состоянии осуществляется в теле метода change класса GeneralController.

При этом каждое состояние может иметь те или иные метрики entry и exit.

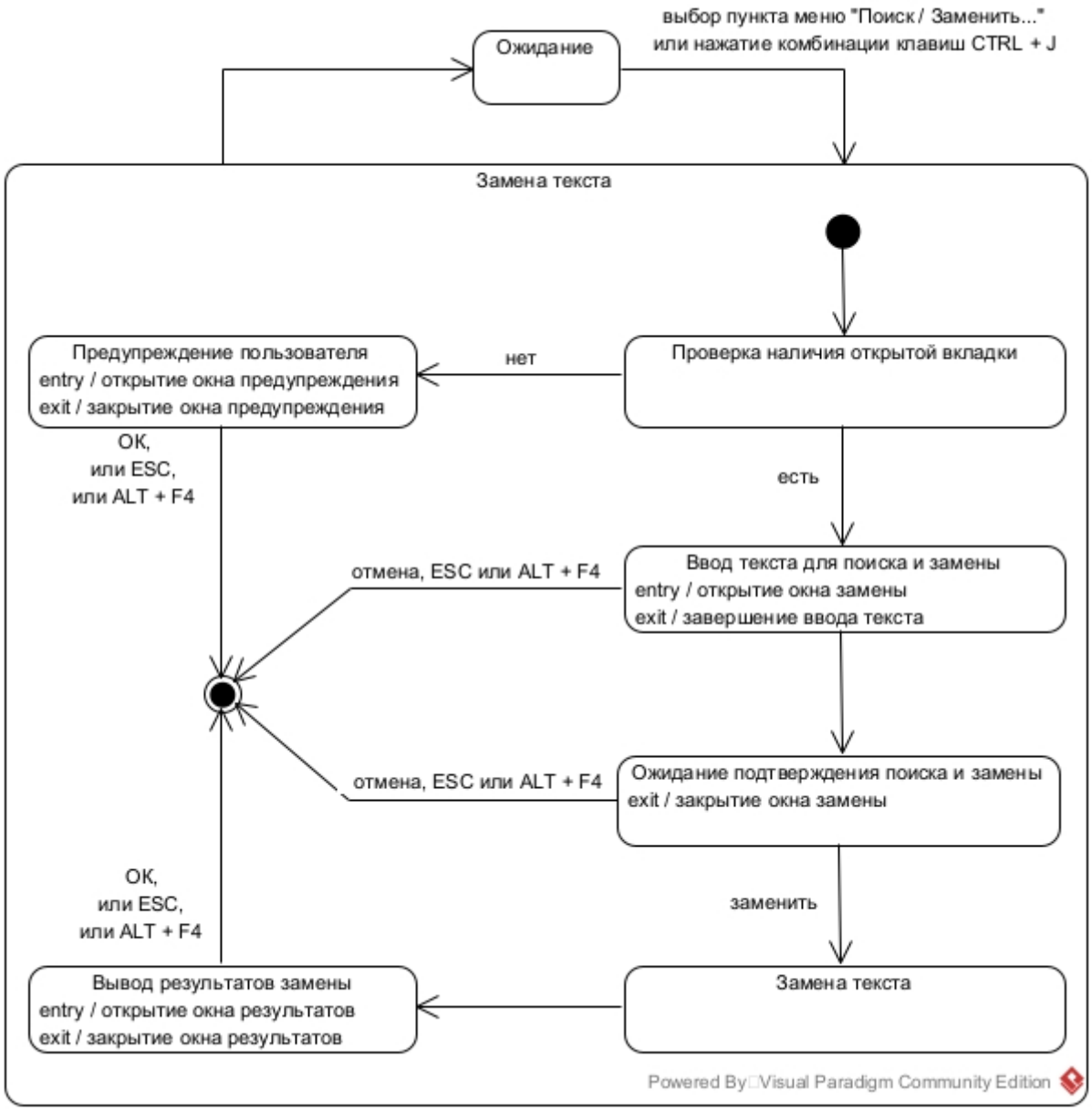

Рисунок 1.1 – Диаграмма состояний объекта «Контроллер замены текста»

1.2 Объект «Контроллер сохранения текста»

В данном случае моделируется метод save класса GeneralController, представляющий собой алгоритм обработки запроса на сохранение текста.

В ходе штатной работы программного продукта могут наблюдаться следующие состояния (рис. 1.2).

1) Ожидание – простое состояние, в рамках которого редактор ждет, когда пользователь инициирует процедуру сохранения текста. Нахождение объекта в данном состоянии фактически запускается при старте приложения.

2) Сохранение текста – составное сложное состояние, представляющее собой собственно процедуру сохранения текста. Нахождение объекта в данном состоянии запускается при вызове единого метода save класса GeneralController опосредованно либо через метод save класса FxmlController, либо с помощью нажатия горячих клавиш через метод start класса Main и конструктор класса KeyCode. Данное состояние включает в себя пять подсостояний:

a) Проверка наличия открытой вкладки – с момента запуска контроллера до момента получения результатов проверки. Проверка проводится в теле метода save класса GeneralController. В зависимости от результатов проверки возможны два варианта – либо открытие окна предупреждения (при отсутствии вкладок), либо открытие системного окна сохранения (при наличии хотя бы одной вкладки).
b) Предупреждение пользователя – с момента открытия окна предупреждения до момента закрытия окна предупреждения. Нахождение объекта в данном состоянии осуществляется в теле метода save класса GeneralController.
c) Выбор имени директории и файла – с момента открытия системного окна сохранения либо до момента завершения выбора, либо до момента отмены процедуры сохранения (закрытия окна). Нахождение объекта в данном состоянии осуществляется в теле метода save класса GeneralController.
d) Ожидание подтверждения сохранения – с момента завершения выбора имени директории и файла либо до момента подтверждения процедуры сохранения (кнопка «Сохранить»), либо до момента отмены процедуры сохранения (закрытия окна). Нахождение объекта в данном состоянии осуществляется в теле метода save класса GeneralController.
e) Сохранение текста – с момента подтверждения процедуры сохранения (кнопка «Сохранить») до момента завершения записи имеющихся данных в файл формата *.txt и закрытия системного окна сохранения. Данное состояние тождественно основному алгоритму сохранения текста.

Нахождение объекта в данном состоянии осуществляется в теле метода save класса GeneralController. Также в экземпляре класса TextTabModel вызываются методы getTextArea и setText (штатный метод родительского класса Tab библиотеки JavaFX).

При этом каждое состояние может иметь те или иные метрики entry и exit.

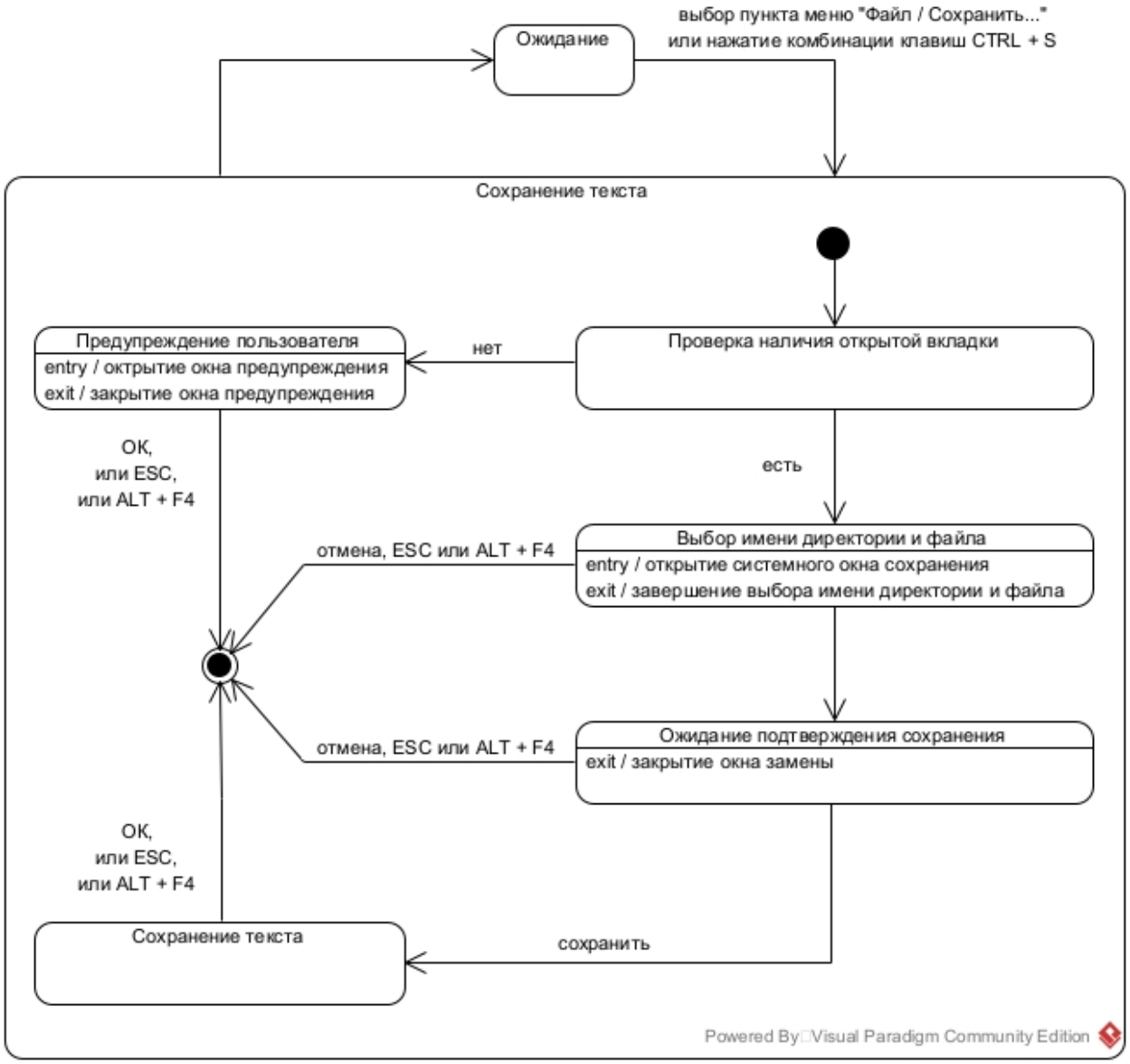

Рисунок 1.2 – Диаграмма состояний объекта «Контроллер сохранения текста»

2. РАЗРАБОТКА ДИАГРАММЫ ДЕЯТЕЛЬНОСТИ

Основными алгоритмами приложения являются более простой алгоритм поиска и алгоритм замены символов, слов и выражений с подсчетом количества найденных совпадений или произведенных замен.

Алгоритм поиска текста состоит из следующих этапов.

1) Проверить наличие открытой вкладки. В зависимости от результатов проверки открывается либо окно предупреждения (при отсутствии открытых вкладок), либо окно поиска (при наличии открытой вкладки).

2) Открыть окно предупреждения – действие при отсутствии открытых вкладок и невозможности поиска.

3) Закрыть окно предупреждения – воздействие на соответствующие элементы интерфейса с помощью мыши или клавиатуры.

4) Открыть окно поиска – действие при наличии хотя бы одной открытой вкладки.

5) Ввести данные для поиска – процесс заполнения строки поиска. Переход к следующим этапам в зависимости от выбора необходимого действия с помощью кнопок «Найти» и «Отмена».

6) Произвести поиск – выполнение собственно алгоритма поиска.

7) Открыть окно с результатами поиска – открывается окно с указанием количества найденных совпадений.

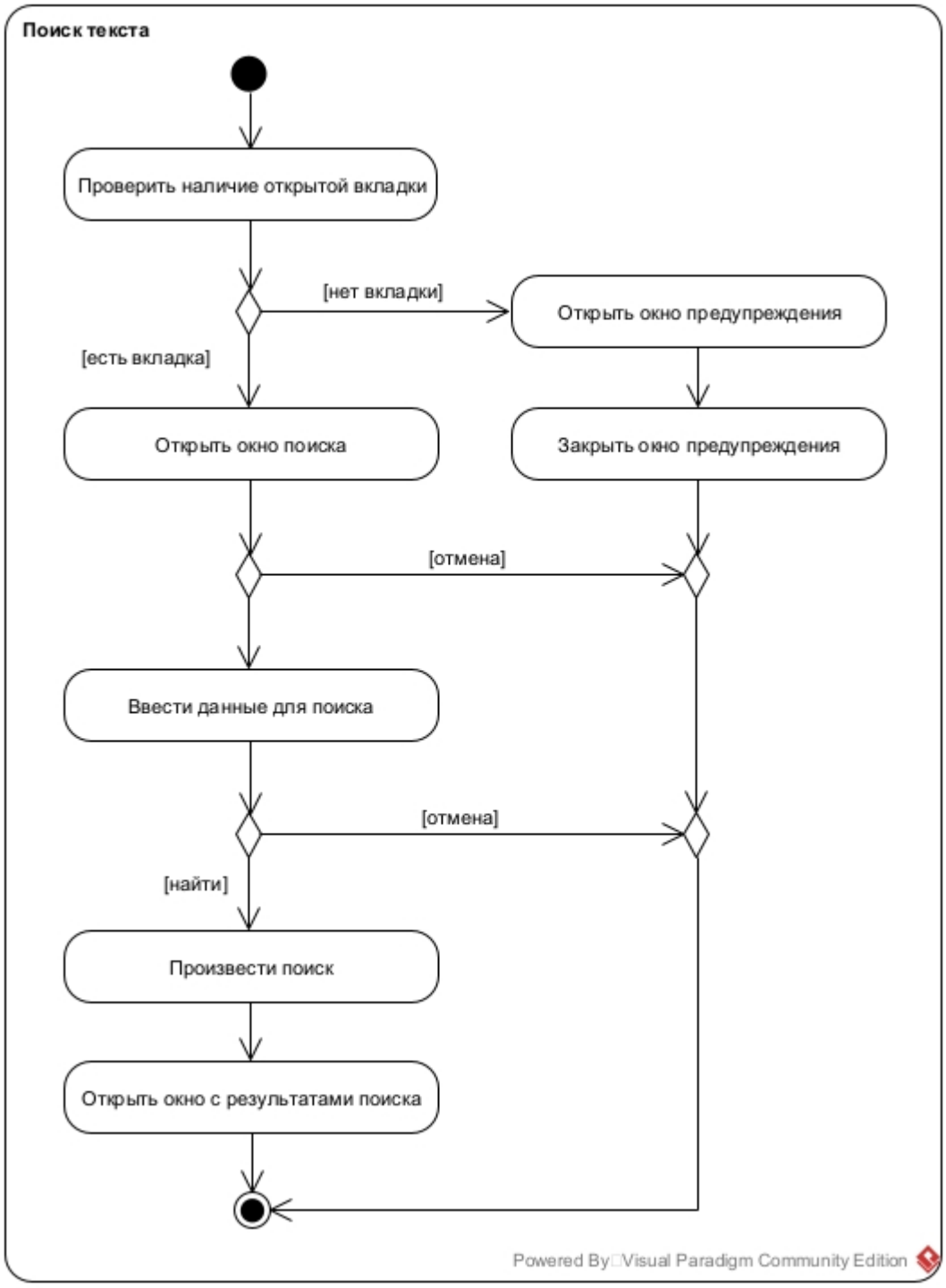

Рисунок 2.1 – Диаграмма деятельности алгоритма «Поиск текста»

Список использованных источников

1. Милихин М.М. Проектирование и архитектура программных средств: методические указания по выполнению лабораторных работ для студентов ФДО направления подготовки 09.03.04 «Программная инженерия». Томск: ФДО, ТУСУР, 2017. 14 с.

2. Милихин М.М. Проектирование и архитектура программных средств: учебное пособие. Томск: факультет дистанционного обучения ТУСУРа, 2015. 138 с.