Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Кубанский государственный технологический университет»
(ФГБОУ ВО «КубГТУ»)

Многоотраслевой институт подготовки и переподготовки специалистов
(МИППС)

Факультет: по работе со студентами ускоренного обучения по индивидуальным учебным планам
Кафедра: информационных систем и программирования
Направление подготовки:09.03.04 Программная инженерия
Профиль: беспрофильный

КУРСОВОЙ ПРОЕКТ

по дисциплине: Проектирование и архитектура программных систем

на тему: «Разработка проекта приложения Конструктор игровых сценариев»

Выполнил студент 3 курса группы 17-ЗКБс-ПР1 Бабич М.М.

Допущен к защите_____

Руководители работы:_____доцент. А.Г. Мурлин

Нормоконтролер _____доцент. А.Г. Мурлин

Защищен _____ Оценка _____

Члены комиссии:_____ст. преп Ю.С. Носова

_____ст. преп К.Е. Тотухов

Краснодар
2020 г.

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Кубанский государственный технологический университет»
(ФГБОУ ВО «КубГТУ»)

Многоотраслевой институт подготовки и переподготовки специалистов
(МИППС)

Факультет: по работе со студентами ускоренного обучения по индивидуальным учебным планам
Кафедра: информационных систем и программирования
Направление подготовки: 09.03.04 Программная инженерия
Профиль: беспрофильный

УТВЕРЖДАЮ
Заведующая кафедрой ИСП
канд. техн. наук, доц. М.В. Янаева
_____

«_____» _____ 20____г.

ЗАДАНИЕ
на курсовой проект

Студенту Бабич М.М.3 курса группы 17-ЗКБс-ПР1
Тема проекта: «Разработка проекта приложения Конструктор игровых сценариев»
(утверждена указанием директора института №10 от 11.02.2020 г.)
План проекта: 1. Постановка задачи
2. Диаграмма вариантов использования
3. Структура классов и методов
Объем проекта:
а) пояснительная записка 26 с.
б) иллюстрированная часть 12 листов
Рекомендуемая литература: 1. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. 2-е изд.

Срок выполнения работы: с «11» 02 2020 г. по «25» 05 2020 г.
Срок защиты: «11» 06 2020 г.
Дата выдачи задания: «11» 02 2020 г.
Дата сдачи работы на кафедру: «25» 05 2020 г.

Руководители работы: _____ доцент. А.Г. Мурлин
Задание принял студент «11» 02 2020 г. _____ Бабич М.М

Реферат

Пояснительная записка курсового проекта 26 с., 12 рис., 5 источников.

ПРОЕКТИРОВАНИЕ И АРХИТЕКТУРА ПРОГРАММНЫХ СИСТЕМ, КОНСТРУКТОР ИГРОВЫХ СЦЕНАРИЕВ, ПОЛЬЗОВАТЕЛЬСКИЕ СЦЕНАРИИ, UML.

В результате выполнения курсового проекта, был разработан проект модели программного приложения «Конструктор игровых сценариев».

Описана общая концепция использования приложения. Приведены примеры пользовательских сценариев. Разработана первичная схема взаимодействия классов приложения.

Разработанных материалов достаточно разработки прототипа приложения и дольнейшего изучения слабых мест в проекти, а также их последующего улучшения.

Содержание

Введение

В ходе данного курсового проекта будет разработан проект модели программного приложения «Конструктор игровых сценариев».

Для решения задачи проектирования программного обеспечения будет использовано UML (унифицированный язык моделирования). Который позволит решить такие кейсы в проектировании как визуализация, структурирование приложения, построить поведенческую модель.

Основным результатом проектирования сиситемы будет визульная и описательная части проекта, которя позволит на ее основе построить прототип приложения для дольнейшего принятия решений в развитии разрабатаемого проекта

1 Постановка задачи

1.1 Общее описание приложения

Требуется разработать проект программного приложения «Конструктор игровых сценариев». Приложение предназначено для упрощения разработки игровых сценариев и взаимодействия сценаристов и разработчиков игр, где есть сюжет и диалоги. В особенности если это касается ветвления сюжета.

Приложение представляет из себя простое оконное приложение, где можно создавать сценарии при помощи графического взаимодействия путем перетаскивания элементов логики или объектов сценария соединяя их между собой. Узлами в этом представления будут логические объекты, диалог, фраза, либо иное действие касаемое изменение/перехода переменных (завязаны на логике).

1.2 Описание функций приложения

В приложении должен быть реализован следующий перечень функций необходимых в приложении:

- Создание проекта сценария
- Создание первого акта в сценарии
- Логические блоки для нелинейного ветвления сюжета
- Узел диалога (ведется последовательное построение фраз персонажей)
- Создание логических переменные для ветвления
- Модуль создание персонажей
- Контрольные точки для перехода между актами

1.3 Прототипа приложения

На рисунке 1 изображен макет первоначальной формы. На первой форме есть список уже существующих, либо по нажатию кнопки добавления нового сценария меняется как на рисунке 2, где заводится проект с именем и местом хранения.

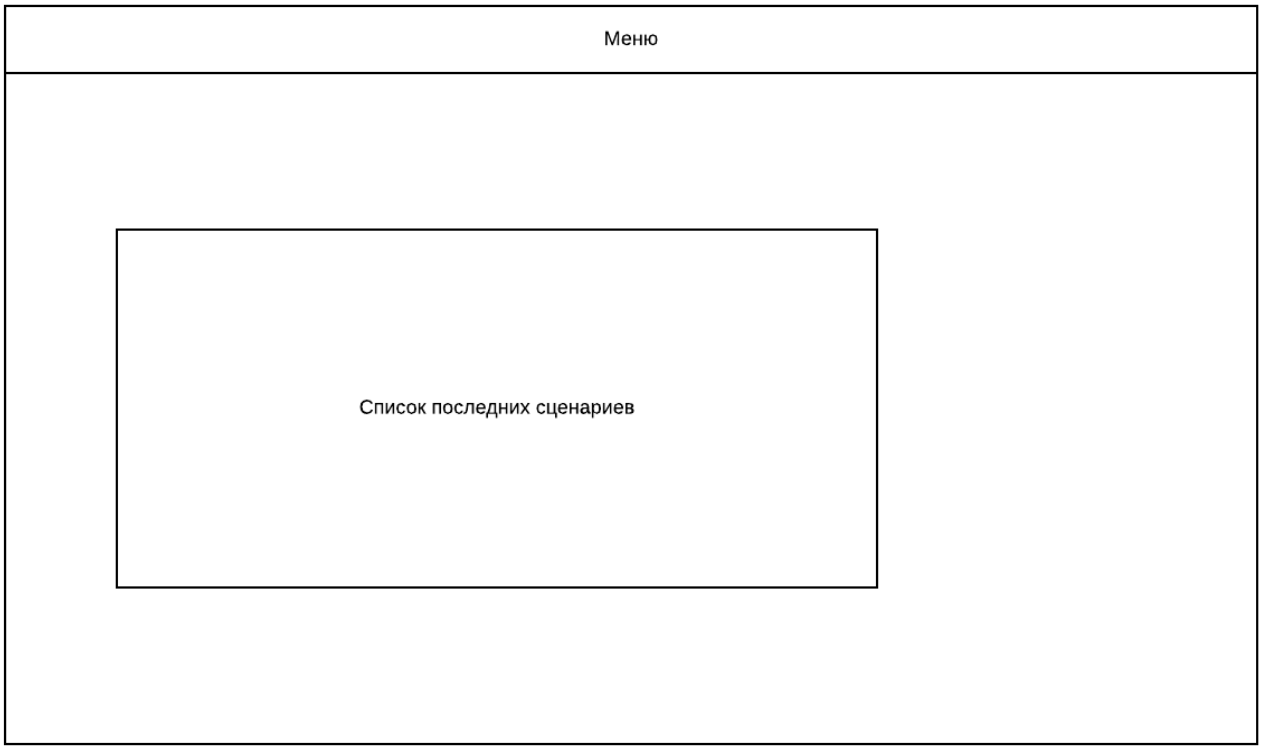

Рисунок 1 – Прототип формы «Меню»

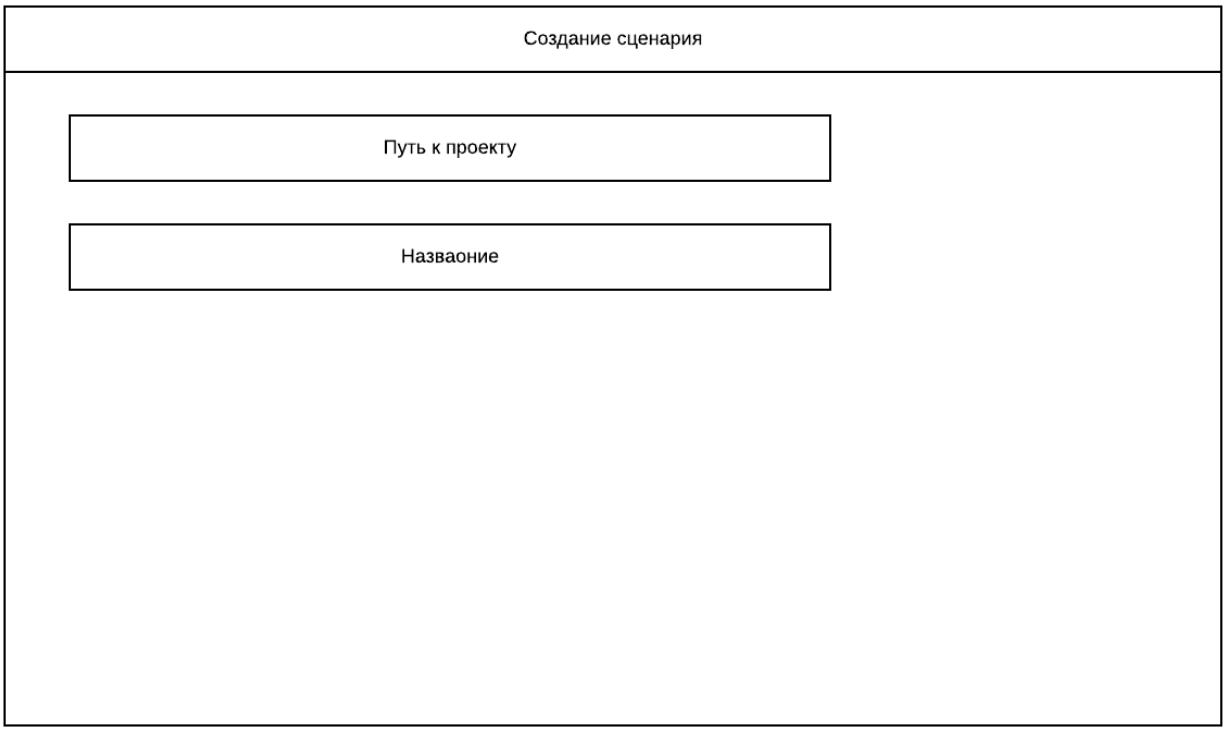

Рисунок 2 – Прототип формы «Меню» при создании сценария

На рисунке 3 отражен макет формы редактирования сценария. Основрыми его элементами является список актов. Актам задается последосвательность перехода в сценарии (сверху в низ), задается путем перетаскивания актов на форме.

Кнопка «Редактирование персонажей» выдет на одноименную форму, где создаются персонажи которые взаимодействуют в сюжетной линии.

«Редактирование переменных» тоже кнопка ведущая на отдельную форму, где создаются переменные.

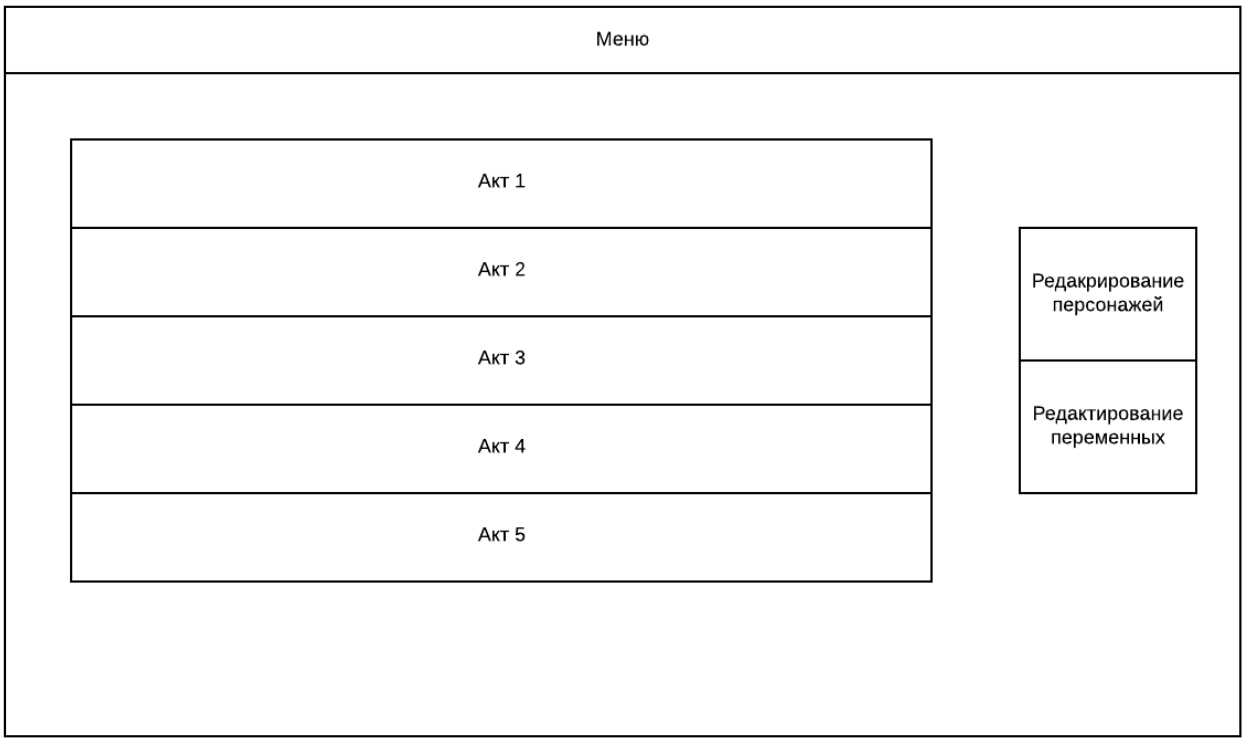

Рисунок 3 – Прототип формы «Сценарий»

На рисунке 4 отражена форма редактрования акта. Форма представляет из себя полатно, на котором раставляются и связываются порядке при помощи графических элементов таких как выбор, условие, переходный узел, узел диалога.

Каждая из нод выполняет собственные функции макеты и описание узлов ниже будет далее

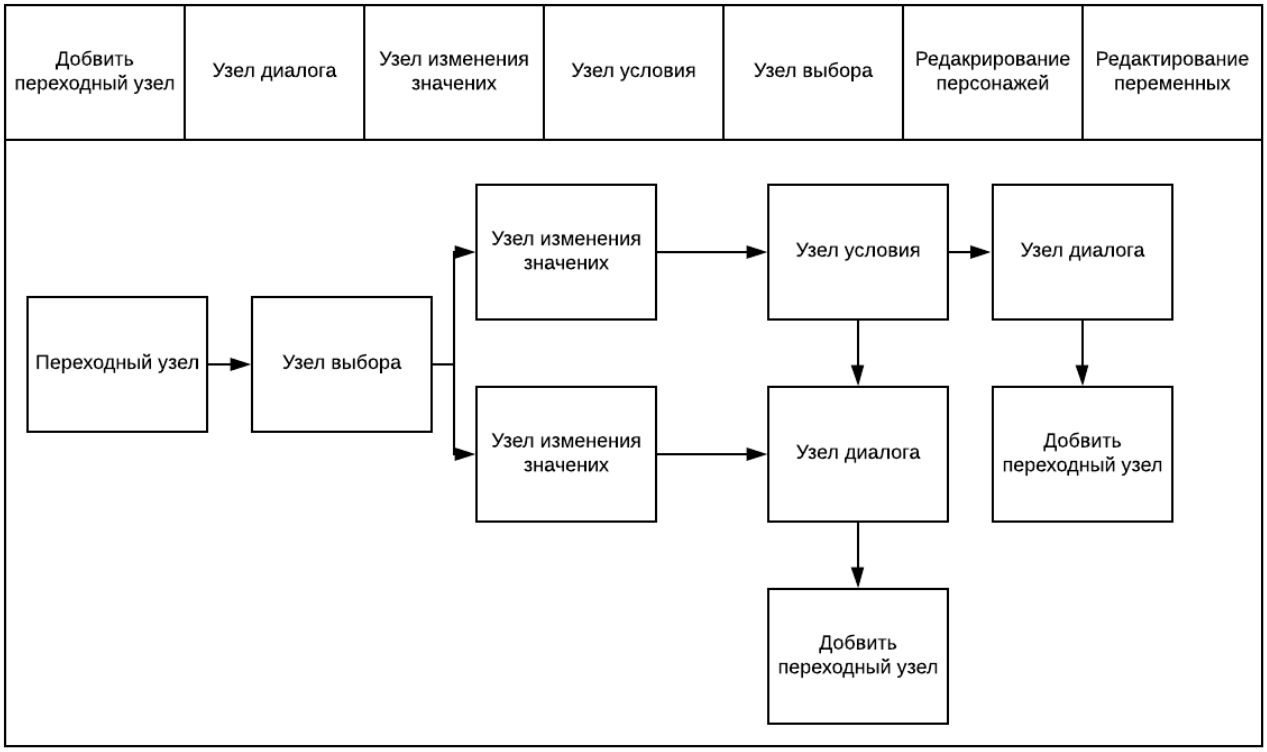

Рисунок 4 – Прототип формы «Акт»

Перходный узел представляет из себя элемет начала в акте. Ставится только в начале акта. Исполнает функцию направления сюжета при переходе из пердыдущих атов сценария. Имеет параметры для прохода по узлу и напраление на следующую ноду в дереве (рисунок 5).

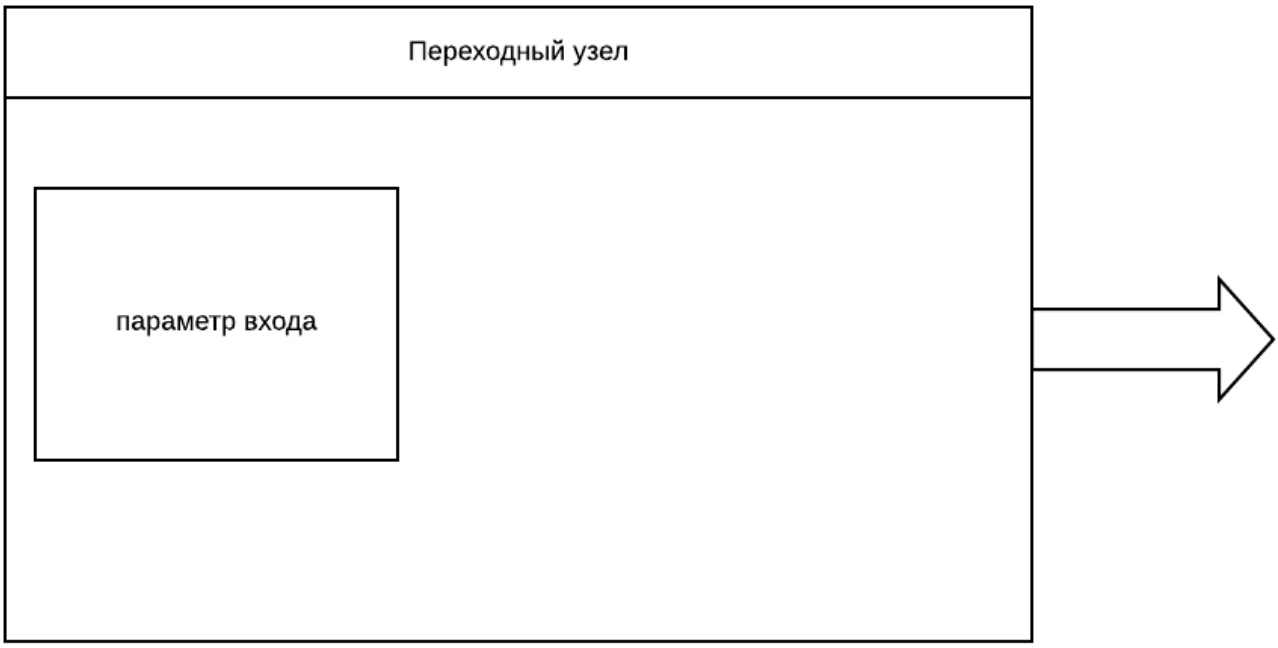

Рисунок 5 – Прототип формы «Переходный узел»

Узел диалога состоит из списка фраз и соответствущие персонажа и эмоции (рисунок 6). Пользователь на данном узле составляет дилог между двумя или группой персонажей. Сами эмоции привязаны к персонажам и выбирается из выподающего списка.

У ноды диалога нет ветвлений и выбора, евляется как факт обработки логики и отражает уже выборанный путь в неленейном сюжете игры, либо повествовании графической новеллы.

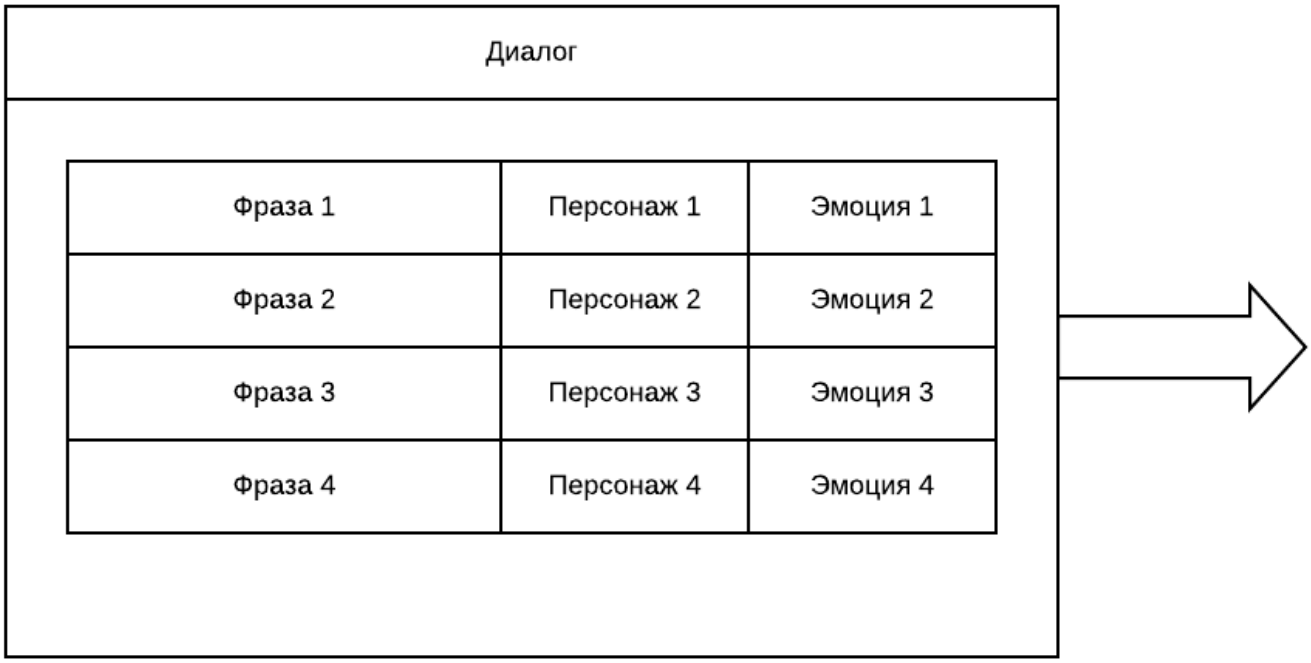

Рисунок 6 – Прототип формы «Диалог»

Узел условие представляет собой графический болок в котором есть список условий которые уходят в связанный маршрут (рисунок 7). Маршрут указывает на следующую ноду, котороя может быть как диальгом, так и выбором.

Условия в ноде указывается как обычное сравнение переменных внесенных в соответствующей форме

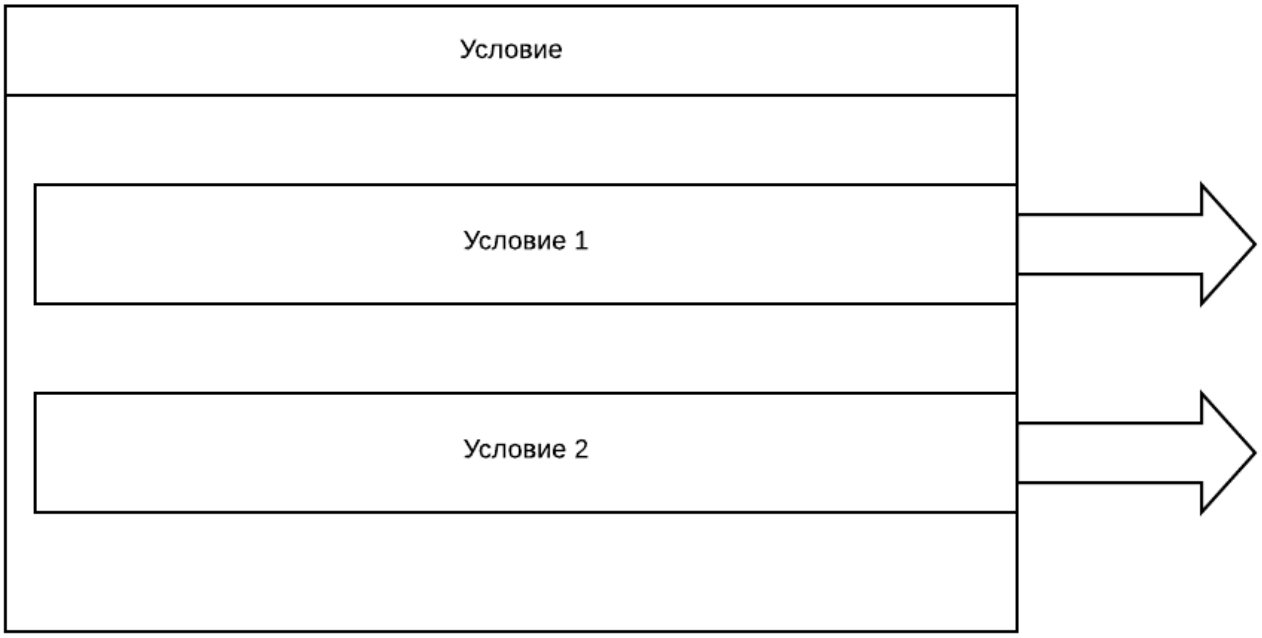

Рисунок 7 – Прототип формы «Условие»

Форма редактора персонажей (рисунок 8). На данной форме заносятся информация о суъектах учавствующих в сценарии. Вход на форму есть из сценария и редактора акта.

У персонажей есть основные характерискики для их отличая это в повествования сюжета:

- Имя
- Цвет
- Список эмоций
- Дополнительные пареметры

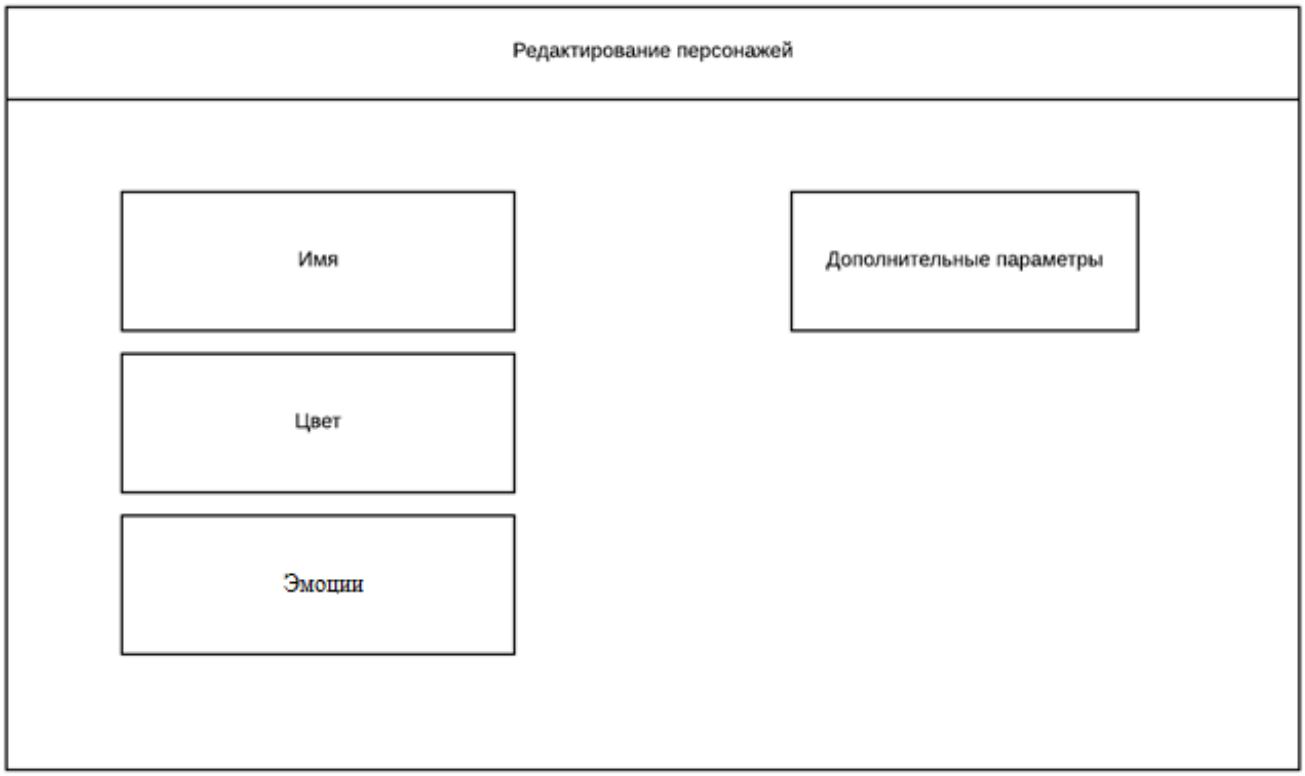

Рисунок 8 – Прототип формы «Персонажи»

Форма заведения переменных. Используются в действиях выборах, то что используется в проверка.

Основные параметры для пере менной это ее название, тип и начальное значение. 3 вида типов это счетчики, значения, флаги.

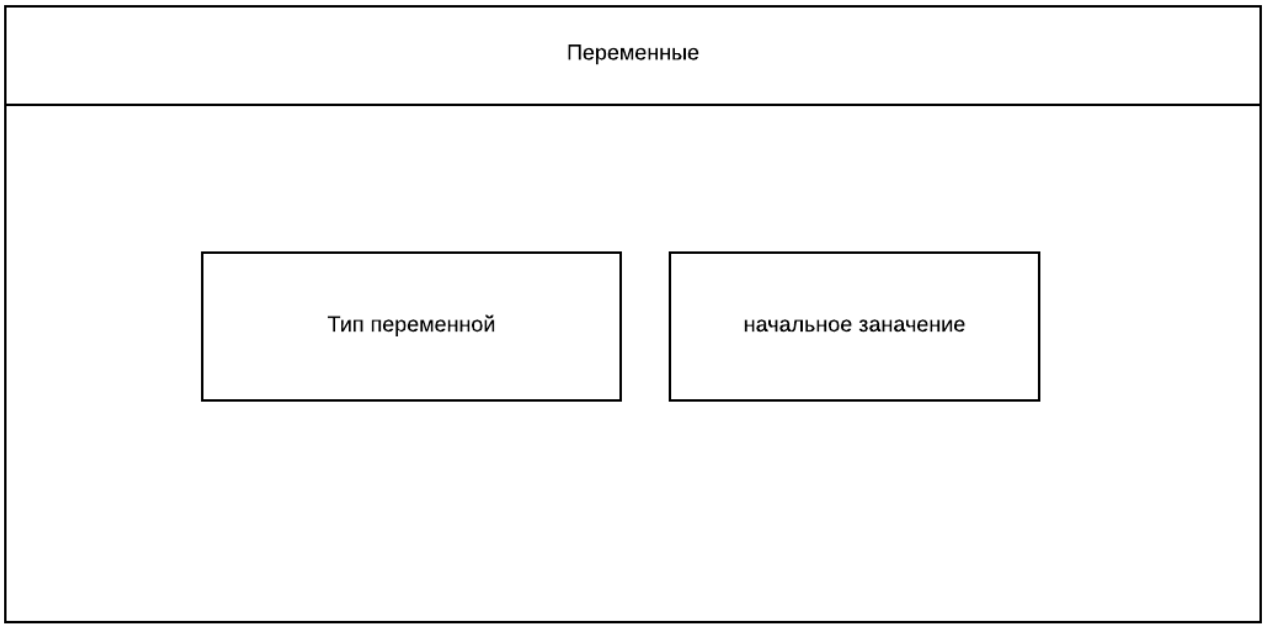

Рисунок 9 – Прототип формы «Переменные»

2 Диаграмма вариантов использования

Конструкция или стандартный элемент языка UML вариант использования применяется для спецификации общих особенностей поведения системы без рассмотрения внутренней структуры этой сущности.

Каждый вариант использования определяет последовательность действий, которые должны быть выполнены проектируемой системой при взаимодействии ее с соответствующим актером.

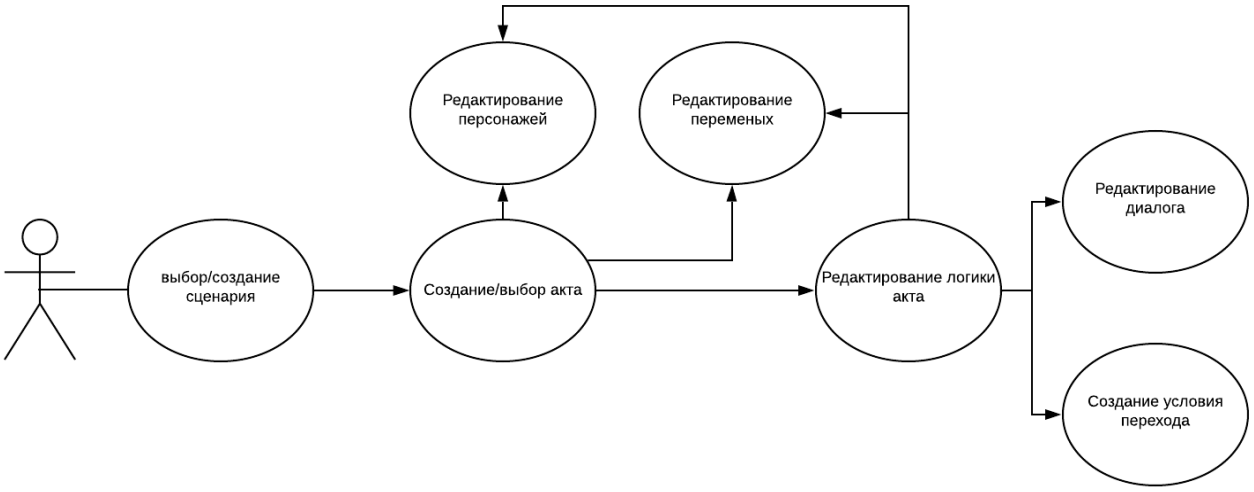

Рисунок 10 - Диаграмма вариантов использования

Поведение и использование форм пользователем практически линейно единственное доступ к редактору персонажей и переменных есть как с формы сценария, так и с редактора акта.

Переходы продемонстрированы на рисунке 11.

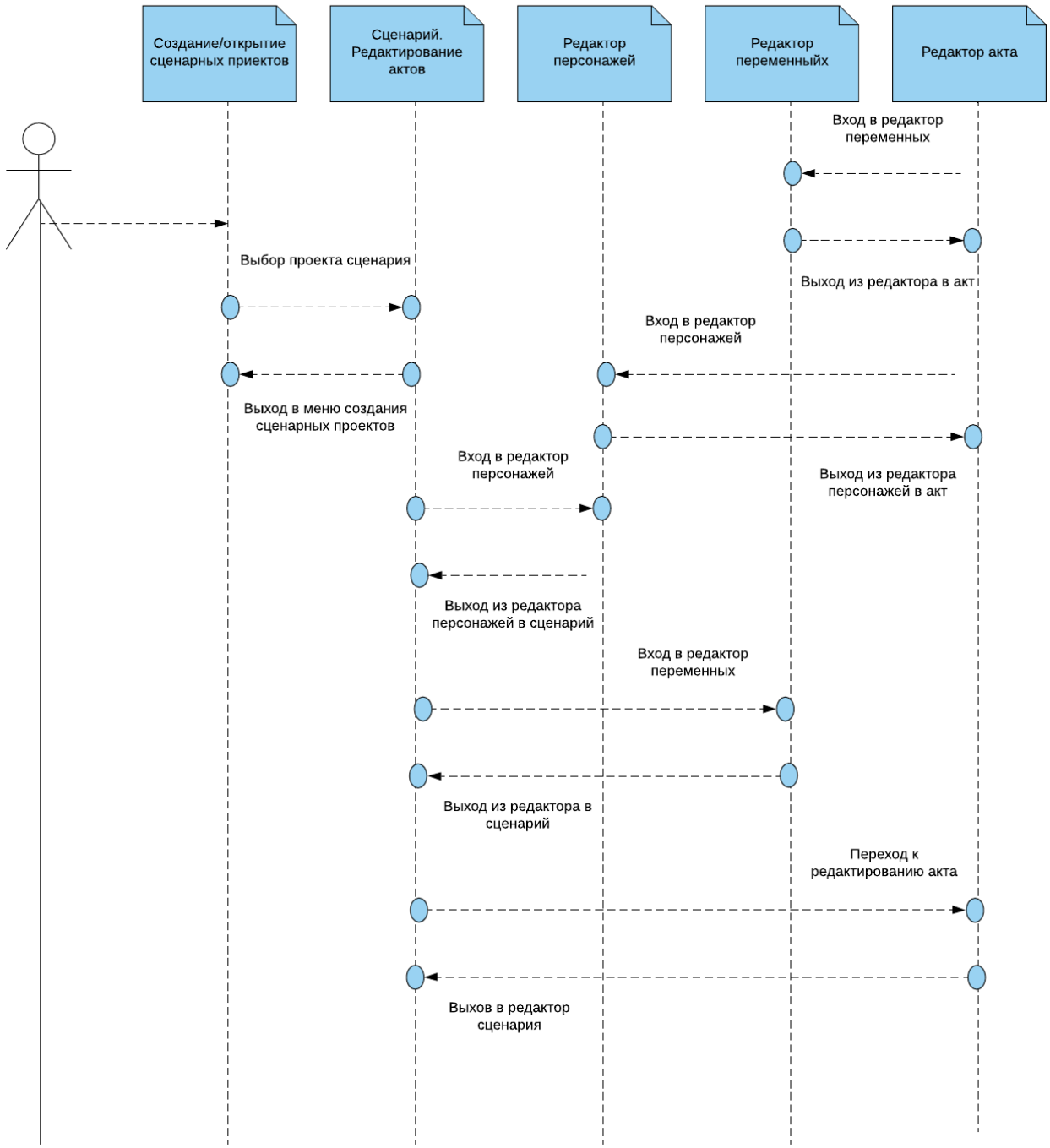

Рисунок 11 – Последовательность действий перехода по основным формам

Большая часть действий пользователя происходит на форме редактора акт. Пользователь выставляет узлы в зависимости от поведения сюжета и условий протекания тех или иных сюжетных поворотов.

Узлы соединяются между собой линиями которая показывает на последовательность действий сценария.

В разных типах узлах есть различное количество входящих выходящих связей. Типы узлов:

- Начало акта
- Условие
- Выбор
- Диалог

3 Структура классов и методов

3.1 Схема классов

На рисунке 5 – «Структура классов и методов», представлены описание классов и методов, которые необходимо реализовать в приложении.

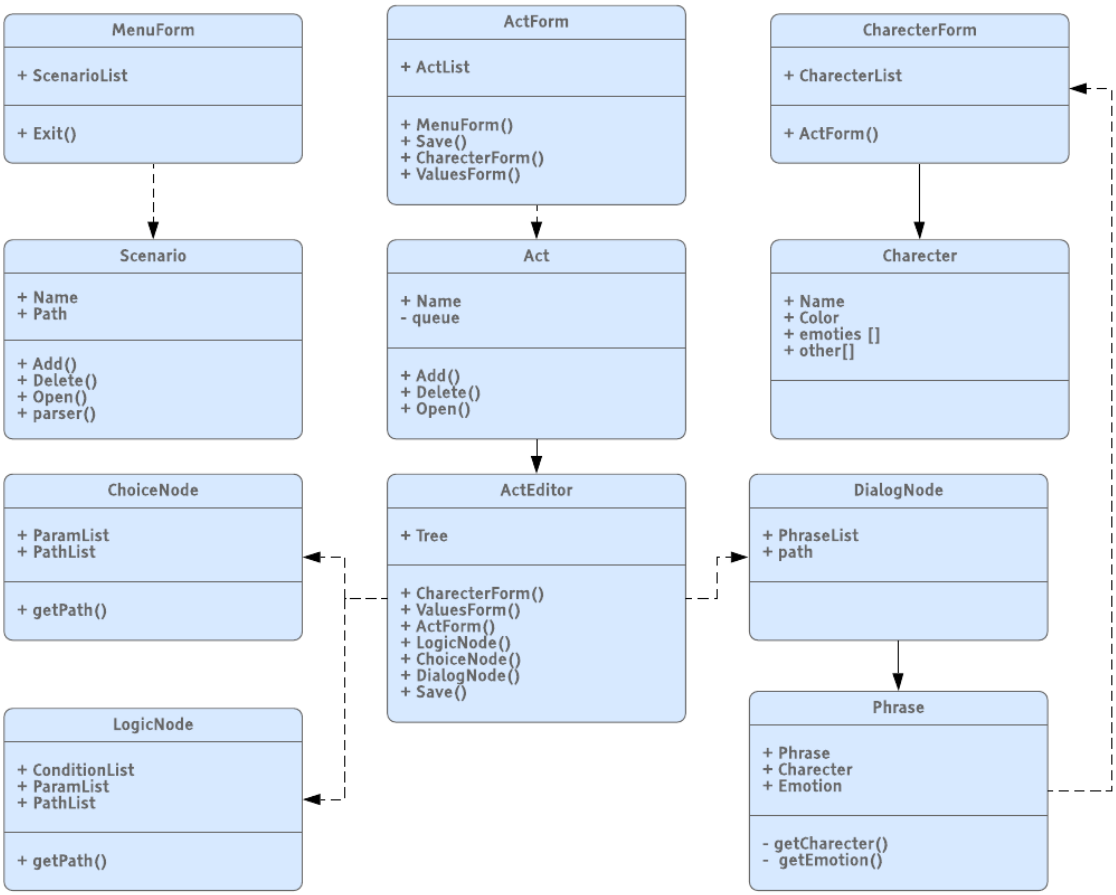

Рисунок 9 – «Структура классов и методов»

3.2 Описание классов методов

MenuForm – Форма меню. Основные действия это создание, удаление, выбор проектов сценария.

ScenarioList – список сценариев.

Exit() – выход из приложения.

Scenario – класс олицитворяющий сущность класса.

Name – название сценария.

Path – путь хранения проекта.

Delete() - удаление объекта.

Open() – открытие сценария в форме сценариев ActForm

parser() – метод сохранения проекта для переноса данных в игровой проет.

ActForm – форма сценария.

ActList – список актов в сценарии.

MenuForm() - переход на форму меню.

Save() - сохранение проекта.

CharecterForm() – переход на форму редактор персонажей.

ValuesForm() – переход на форму редактор перчёных.

Act – Класс описывающий сущность акта.

Name – Название акта.

Queue – номер в очередт.

Delete() - удаление объекта.

Open() – открытие сценария в форме сценариев ActEditor.

ActEditor

CharecterForm() – переход на форму редактор персонажей.

ValuesForm() – переход на форму редактор перчёных.

ActForm() – переход на форму редактор сценария.

LogicNode() – обработка логического блока.

ChoiceNode() – обработка блока выбора.

DialogNode() – обработка ноды диалога.

Save() – Сохранение акта.

CharecterForm – Форма редактирования персонажей.

CharecterList – список персонажей.

ActForm() – переход на форму сценария.

Charecter – класс описывающий объект персонажа.

Name – имя персонажа.

Color – цвет текста персонажа.

emoties [] – эмоции персонажа.

other[] – дополнительные параметры.

ValueForm – форма редактирования переменных.

ValueList – список значений.

ActForm() – переход на форму сценария.

Value класс описывающий объект переменных.

Name – имя переменной.

Value – первоначальное значение переменной.

Type – тип переменной (флаг, значение, счетчик).

ChoiceNode – Элемент усдлвия.

ParamList – список параметров.

PathList – соответствующие параметрам лики до следующих нод.

DialogNode – Элемент диалога.

PhraseList – список фраз составляющих диалог.

Phrase – класс описывающий объект фраза.

Phrase – текст, фраза одного из персонажей.

Charecter – персонаж кому пернадлежит фраза.

Emotion – эмоция персонажа в момент фразы.

getCharecter() – подбирает список персонажей.

getEmotion() - подбирает список эмоций соответствующих выбранному персонажу персонажей.

LogicNode

ConditionList – список условий

PathList – соответствующие условиям лики до следующих нод.

getPath() – метод выбора пути.

3.3 Выбор среды для дальнейшей разработки

В качестве среды для разработки выбран игровой движок Godot Engine. Он подходит не только для создания 2D и 3D игр, но и оконных приложений. У Godot есть очень проработанная система создания элементов управления и написание обработчиков событий.

Основными причинами выбора такой среды разработки в том, что:

Приложение разрабатывается для дальнейшей обработки созданных сценарных проектов в игровых движках.

Удобного создания графических форм.

Из коробки заложенная система связывания графических узлов в граф/дерево

Заключение

В ходе выполнения курсового проекта, был разработан проект модели программного приложения «Планировщик бюджета». Была описана общая концепция использования приложения.

Определены основные пользовательские сценарии. Разработана черновая схема работы классов и их взаимодействуя, которая будет улучшаться и дополняться в процессе разработки прототипа по данному проекту. Разработанных материалов достаточно для начала реализации полноценного приложения.

Данных матерьялов достаточно для разработки прототипа приложения и дольнейшего изучения слабых мест в проекти, а также их последующего улучшения.

Список используемой источников

1 Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. 2-е изд. Пер. с англ. – СПб.: Символ-Плюс, 2007. – 624 с.

2 Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие / Под ред. Л.Г. Гагариной. – М.: ИД «Форум»: Инфра-М, 2013. – 400 с.

3 Мартин Фаулер, «UML. Основы. Краткое руководство по стандартному языку объектного моделирования», 2018г - Символ-Плюс, 192 стр.

4 Крэг Ларман, «Применение UML и шаблонов проектирования. Третье издание», 2018г - Вильямс, 736 стр.

5 Робрет Максимчук, «UML для простых смертных», 2016г - Лори, 400 стр.

ПРИЛОЖЕНИЕ А – Антиплагиат