**Министерство образования И МОЛОДЕЖНОЙ ПОЛИТИКИ**

**Свердловской области**

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

СВЕРДЛОВСКОЙ ОБЛАСТИ

«ВЕРХНЕСАЛДИНСКИЙ АВИАМЕТАЛЛУРГИЧЕСКИЙ КОЛЛЕДЖ им.А.А.ЕВСТИГНЕЕВА»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КУРСОВОЙ ПРОЕКТ**

***На тему****:*  **Теория разработки программного обеспечения автоматизированной информационной системы "Абитуриент"**

**(ТРПО)**

***Специальность 09.02.07***

Информационные технологии и программирование

Лунев Леонид Михайлович

**Студент**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись Ф.И.О.

ИСП-301

**Курс \_\_\_\_**\_III\_\_\_**\_\_\_\_\_\_\_ Группа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Руководитель:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бардина Т.И.

подпись Ф.И.О.

г.Верхняя Салда

2021г.

[ВВЕДЕНИЕ 3](#_Toc68123972)

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc68123973)

[1.1 Анализ предметной области 4](#_Toc68123974)

[1.2 Сравнительная характеристика информационных систем 5](#_Toc68123975)

[1.3 Требования к информационной системе 6](#_Toc68123976)

[1.4 Принципы построения информационных систем 6](#_Toc68123977)

[2 РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИCТЕМЫ 7](#_Toc68123978)

[2.1 Языки программирования 7](#_Toc68123979)

[2.1.1 Сравнительная характеристика языков программирования 7](#_Toc68123980)

[2.1.2 Обоснование выбора языка программирования 8](#_Toc68123981)

[2.1.3 Преимущества и недостатки 9](#_Toc68123982)

[2.2 Среды программирования 9](#_Toc68123983)

[2.3 Среда программирования Visual Studio 9](#_Toc68123984)

[2.3.1 Преимущества и недостатки 9](#_Toc68123985)

[2.3.2 Основные компоненты 9](#_Toc68123986)

[2.4 Жизненный цикл 10](#_Toc68123987)

[2.5 Процессы разработки программного обеспечения 11](#_Toc68123988)

[2.6 Модели процессов разработки программного обеспечения 12](#_Toc68123989)

[2.7 Язык моделирования UML 13](#_Toc68123990)

[ЗАКЛЮЧЕНИЕ 14](#_Toc68123991)

# ВВЕДЕНИЕ

Поскольку задачи цифровизации всего народного хозяйства, и в том числе образования, не теряют с годами своей актуальности, а, наоборот, становятся всё более насущными, в том числе в связи с пандемией, переходом обучения в режим онлайн, любые попытки компьютеризации учебного процесса должны приветствоваться.

Театр начинается с вешалки, по словам великого русского режиссёра К.С. Станиславского, а учёба абитуриента в нашем колледже начинается со сдачи документов в приёмную комиссию. Качество учёбы, удовлетворённость учёбой, последующие успехи студента во взрослой жизни определяются информационной средой в колледже, одно из первых соприкосновений с которой произойдёт, когда будущий студент передаст свои данные в приёмную комиссию.

Данный курсовой проект посвящён тому, какое программное обеспечение должно использоваться приемной комиссией колледжа, и как можно разработать такое программное обеспечение.

## 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### Анализ предметной области

Как было сказано во Введении, темой нашей работы является теория разработки программного обеспечения, предназначенного для обслуживания работы приёмной комиссией колледжа, ввода персональных данных по поступающим абитуриентам и занесения результатов приёмных экзаменов.

Ввод персональных данных производится во множестве программных систем, при заказах билетов на все виды транспорта, при получении почтовых отправлений, при обслуживании в лечебных учреждениях. Персональные данные содержатся также в общегосударственной системе «Госуслуги», с очень широкой сферой применения. Ввод персональных данных во все эти системы производится с настольных компьютеров, с мобильных устройств, с помощью браузеров или специально установленных программ. Основными средствами идентификации личности практически во всех системах служат паспортные данные, а именно имя, фамилия, отчество, год рождения, а также данные самого документа, в котором они содержаться, то есть, как правило, номер и серия паспорта, место выдачи документа и каким органом власти выдан. После того, как персональные данные введены, обращение к ним в каждой данной конкретной системе производится по некому внутреннему идентификатору. В системе «Госуслуги» таким идентификатором может служит одно из трёх значений: номер пенсионного свидетельства СНИЛС, адрес электронной почты, номер телефона. На заводе наиболее часто используем идентификатором, служащим для установления личности работника, является табельный номер. В «Корпорации ВСМПО-АВИСМА»» это шестизначный номер, первые две цифры которого означают номер цеха, а следующие четыре уникально определяют работающего внутри данного цеха.

Такая короткая комбинация цифр или букв, позволяющего быстро и точно найти персональные данные человека, называется индексом. Индекс можно рассматривать как разновидность хеш-кода, то есть небольшого куска информации, позволяющей найти большого размера информацию, представителем(идентификатором) которого этот кусок, строка, номер, комбинация цифр и букв является.

Правила составления индексов или хеш- кодов зависят от объёма информации, которую необходимо проиндексировать. Для студентом техникума уникальным идентификатором может служить номер студенческого билета. В нашем случае, где мы ведём учёт поступающих в техникум будущих студентов, которым ещё не присвоен номер студенческого билета, достаточным условием поиска может служить просто фамилия абитуриента, редкие случаи однофамильцев ввиду небольшого количества поступающих могут легко разрешаться обращением к имени абитуриента и точной дате рождения.

### 1.2 Сравнительная характеристика информационных систем

Информационные системы служат для решения оперативных, тактических и стратегических задач предприятия. Оперативная информация поступает в оперативную информационную систему постоянно в течение рабочего времени, где она обрабатывается и становится доступной для информационной подсистемы тактического уровня. Поступающая оперативная информация формируется на основе информации тактического уровня и сверяется с ней. Информационные системы самого высшего, стратегического уровня используют информацию, поступающую с тактического уровня, для сверки её со стратегическими целями, её оценки и формирования заданий тактического уровня.  
Для колледжа примером информационной системы стратегического уровня может служит составление отчётности по работе колледжа в соответствии с Уставом и нормативными документами органов государственной власти, определение задач на следующий учебный год и большие строки.

Системой тактического уровня может служит информационная система, осведомляющая об общем числе подавших заявления на каждую специальность, количество и фамилии зачисленных, количество и фамилии успешно сдавших экзамены и количество и фамилии отчисленных студентов, не справившихся с учебными заданиями.  
Автоматизированная информационная система «Абитуриент», теорию разработки которой мы описываем, принадлежит к классу информационных систем оперативного уровня и предназначена для решения одной, строго определённой, узкой задачи: учёта лиц, подавших документы для поступления в колледж и определения среднего проходного балла для зачисления. Накопленные в ней сведения могут быть переданы в информационные системы других , более высоких уровней, для работы с котингентом студентов в течении учебного года (тактический уровень), для обработки результатов обучения студентов в течение всего курса обучения и составления отчётов (стратегический уровень)

### 1.3 Требования к информационной системе

Информационная система должна выполнять задачи, для которых была создана, кроме того, существенными требованиями являются надёжность работы и расширяемость, то есть способность адаптироваться к более широкому кругу задач, чем те, для которых она изначально предназначалась. Важным свойством является свойство адаптируемости под запросы пользователей, которые могут меняться или могут быть не учтены при первоначальном проектировании системы.

Кроме надёжности, расширяемости, адаптируемости важнейшим свойством является эргономичность, то есть удобство использования, дружественность по отношению к пользователю. Современные графические программные оболочки, служащие для взаимодействия с пользователями, хорошо отработаны, однотипны. Стандартный набор элементов управления позволяет пользователю легко использовать новую для него программу. Операционная система Windows так и была названа, потому что в основе её создания была идея многооконного интерфейса с пользователям, снабжённого стандартными элементами управления: кнопками, меню, полями ввода данных, полосами прокрутки для просмотра больших объёмов информации.

Поэтому для нашей автоматизированной информационной системы «Абитуриент» мы используем систему Windows, с которой пользователь попадает сразу в знакомое программное окружение и сможет легко начать использовать нашу программу.

1.4 Принципы построения информационных систем  
Тему, которая стоит в заголовке параграфа, можно раскрыть двумя способами:

1. Как, на основе чего, каким способом можно строить информационные системы и

2. Каким принципам должна удовлетворять информационная система

Основными элементами современных информационных систем являются персональные компьютеры, объединённые в сеть и снабжённые типовым и специализированным программным обеспечением, зависящим от задач предприятия или учреждения, которое обслуживает эта информационная система.

Информационные системы могут быть классифицированы по типу компьютеров, на которых располагается программное обеспечение, которое обеспечивает функционирование информационной системой и по типу взаимодействия между ними. Программное обеспечение может быть разделено на следующие классы:

1) Клиент – серверные

2) Деск-топовые

3) Мобильные

3) Встроенные

Клиент – серверные системы подразумевают наличие главного, центрального компьютера, называемого сервером, на котором находится основная часть программ, составляющих информационную систему. Примером клиент-сервисных систем является программное обеспечение сайтов, с которым пользователь взаимодействует через браузер. Браузер скачивает клиентскую часть программного обеспечения на компьютер пользователя, обеспечивая предоставление информации и ввод необходимых данных.

Деск-топовые программы могут работать на компьютере пользователя без связи с Интернетом или другими компьютерами в локальной сети, полностью используя ресурсы этого компьютера. Преимуществом деск-топовых программ является высокая скорость работы и независимость от сети.

Мобильное программное обеспечение работает на смартфонах и планшетах,

удобно в использовании, может быть использовано большим числом пользователей из-за большой распространённости мобильных устройств, но программирование для них имеет свои особенности из-за используемых на них операционных систем типа Android или iOS, а также небольшого размера экрана. Из-за ограниченности ресурсов и оперативного характера работы программы на мобилных устройствах часто работают по типу клиент-сервер.

Встроенные программные системы работают на компьютерах, которые конструктивно связаны с определёнными типами оборудования, которыми они управляют. Для них характерные ограниченность ресурсов, на которых они работают, и особенные операционные среды, а также повышенные требования к надёжности работы.

Для построения нашей автоматизированной информационной системы «Абитуриент» будем использовать десктопный вариант построения программного обеспечения, то есть все программное обеспечение будет расположено на одном персональном компьютере, работающем под управлением операционной системы Windows, доступ к которому будет осуществляться через стандартное периферийное оборудование этого же компьютера: клавиатуру, мышь, монитор.

## 2 РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИCТЕМЫ

2.1 Языки программирования   
Кроме выбора типа программной системы, для разработки программного обеспечения надо сделать выбор языка программирования, на котором будет вестись эта разработка. В настоящее время для разработки программного обеспечения в его десктопном варианте существует большое число языков. Об их количестве и популярности можно судить по сайту https://www.tiobe.com/tiobe-index/ (индекс TIOBE), которые ежемесячно обновляет данные по языкам программирования, являясь очень авторитетным ресурсом.

2.1.1 Сравнительная характеристика языков программирования   
По области применения выделяется язык JavaScript, это язык WEB, при его поддержке функционирует большинство страниц Всемирной Паутины. Язык Python, который конкурирует за вторую строчку в индексе TIOBE, очень популярен в разработке систем Искусственного Интеллекта, которые находят сейчас очень широкое применение. Языки программирования делятся на языки низкого и высокого уровня. Из первой двадцатки самых популярных языков по индексу TIOBE низкоуровневым языком программирования является только язык Assembler. Занимая 9 место по популярности, он используется только в узких целях системного программирования и для создания встроенных в аппаратное обеспечение систем. Остальные языки являются языками высокого уровня, то есть ориентированы не столько на объём получающегося программного обеспечения и скорость его работы, сколько на удобство разработки систем на их основе, скорость разработки и качество получающихся программ. Язык C, занимающий сейчас первую строчку в индексе TIOBE, ближе всех стоит к языкам низкого уровня, хотя и используется для программирования всех типов систем. Среди остальных языков высокого уровня большинство очень его напоминают, языками другого типа являются языки функционального программирования и язык SQL, который используется при работе с базами данных.

По области применения выделяется язык JavaScript, это язык WEB, при его поддержке функционирует большинство страниц Всемирной Паутины. Язык Python, который конкурирует за вторую строчку в индексе TIOBE, очень популярен в разработке систем Искусственного Интеллекта, которые находят сейчас широчайшее применение.

2.1.2 Обоснование выбора языка программирования  
 Для операционной системы Windows, под управлением которой мы решили писать свою программу, наиболее естественным является выбор языка C#. Для него существует несколько типов графических оболочек, для десктопного программирования наиболее популярны две: WPF(Windows Presentation Foundation) и Windows Forms. Кроме того, язык C# начал свою экспансию в мир WEB-программирования, он может использоваться теперь не только на серверной стороне, но и внутри браузера вместо языка JavaScript. Так что, если мы захотим изменить концепцию построения нашей информационной системы и придать ей клиент-серверный вид, вынести её для работы в Интернет, выбор языка C# будет нам не помехой , а помощником. Огромная библиотека классов .NET Framework, которая поддерживает С#, позволяет выполнить практически любую задачу оперирования с данными, которая может возникнуть по ходу выполнения нашего проекта.

#### 2.1.3 Преимущества и недостатки

Недостатками языка C# является оборотная сторона его достоинств: огромное число возможностей, которое представляет язык, делает необходимым ориентироваться в них, находить нужные для решения поставленных задач, перебирая множество вариантов. Впрочем, большое количество пользователей этого языка, великолепная справочная система Microsoft, множество форумов, блогов для поддержки пользователей C# делает задачу ориентации в его языковых средствах выполнимой. Добрым другом программистам на C# являются сайты docs.microsoft.com, devblogs.microsoft.com содержащие самые свежие, актуальные сведения по последним версиям языка из первых рук создателей C# и его библиотек. К услугам программистов на C# stackoverflow.com, где можно получать помощь от своих коллег в затруднительных ситуациях, конечно, огромное количество видеолекций на youtube.com, и и множество других ресурсов, которые находятся через поисковые системы с набором ключевых слов «язык C#»

### 2.2 Среды программирования

Программное обеспечение создаётся на компьютере, компьютер работает под управлением операционной системы, в нашем случае выбрана Windows 10. Для того чтобы воспользоваться выбранным нами языком программирования C#, необходимы специализированные программы, которые будут переводить написанные нами тексты на языке C# в исполняемый на компьютере код, который может быть запущен.  
Вот как выглядит традиционно простейшая программа на языке C#, которая выводит на консоль системы, специальное чёрное окно для вывода текстовой информации, приветствие-фразу «Привет Верхняя Салда!»:

using System;

class Привет{

public static void Main(){

Console.WriteLine("Привет Верхняя Салда!");

}

Впрочем, в последней, 9-ой версии языка C# пример может выглядеть гораздо проще и состоять всего из двух строчек:  
using System;

Console.WriteLine("Привет Верхняя Салда!");  
Текст это программы можно набрать в стандартном редакторе notepad.exe «блокноте», входящем в состав стандартных программ операционной системы Windows, но для того, чтобы эти текстовые строчки превратились в программу, которая выдаст нам на экран строчку «Привет Верхняя Салда!», нужно что-то ещё, а именно , программу-компилятор csc.exe, которая сможет превратить текстовый файл, в котором находится текст программы, например, с названием Привет.cs в исполняемый файл программы Привет.exe. Расширение .cs у текстового файла указывает на то, что мы имеем дело с программой, написанной на языке C# (C Sharp по-английски), знак # sharp означает в музыке увеличение на полтона высоты звука. C# улучшенный C, так можно интерпретировать идею создателей языка, давших ему такое название.  
Программы вполне можно создавать, обходясь только текстовым редактором и компилятором, но делать это неудобно. Для комфортного и быстрого создания программного обеспечения существуют среды программирования, то есть специальные программы, в которых все подручные средства, и редактор, и компилятор, и проводник для открытия нужных файлов собраны в одном месте, и рутинные действия программиста автоматизированы. Для разных языков программирования существуют разные среды разработки, такие как NetBeans, Eclipse,IntelliJ IDEA для языка Java, PyCharm для Python и множество других IDE. Буквы IDE это сокращение от Integrated Development Environment, что означает Интегрированная Среда разработки.  
В некоторых средах разработки можно использовать несколько разных языков,  
например, среда разработки Visual Studio Code от Microsoft поддерживает практически все языки, достаточно добавить туда соответствующие расширения. Но поскольку мы будем вести разработку исключительно на C#, мы выберем самый удобный, испытанный инструмент от Microsoft, среду программирования Visual Studio. Её название отличается всего на одно слово от упомянутой выше среды Visual Studio Code, но это совсем другой инструмент. Visual Studio Code была создана сравнительно недавно, большей частью чтобы удовлетворить запросы WEB-разработчиков, работающих на разных платформах под Linux и MacOs, и, хотя вести разработку на C# там тоже можно, наиболее рационально применить инструмент, разработанный на нём же и изначально приспособленный для работы с продуктами Microsoft.

### 2.3 Среда программирования Visual Studio

Итак, выбор сделан, в качестве основного инструмента для разработки автоматизированной информационной системы «Абитуриент» мы будем использовать Visual Studio самой последней версии, которая носит название Visual Studio 2019.

#### 2.3.1 Преимущества и недостатки

Чрезвычайно значительным преимуществом сделанного выбора интегрированной системы разработки IDE Visual Studio 2019 является возможность её свободного скачивания и бесплатного использования. Достаточно перейти по ссылке <https://visualstudio.microsoft.com/ru/downloads/> и выбрать кнопку для Community Edition **Скачать бесплатно**. При загрузке и установке должна быть выбрана так называемая загрузка «Разработка классических приложений .NET». Дело в том, что для наших целей вполне достаточно содержимое этого модуля. Если в будущем нам понадобится разрабатывать что-то для WEB, для мобильных устройств, на других языках программирования, отличных от C#, мы сможем ввести в состав Visual Studio дополнительные загрузки. Такая модульность и многосторонность – одно из преимуществ VS 2019. Очень большое количество опций, вариантов и способов работы в Visual Studio создаёт и проблему: нужно время, чтобы ознакомится даже с основными возможностями этой замечательной среды.

#### 2.3.2 Основные компоненты

Основными компонентами интегрированной системы разработки IDE Visual Studio 2019 является стандартный интерфейс пользователя, состоящий из окон, кнопок и меню, и скрытого за ним механизма обработки текстовых файлов и превращения их в готовую программу.

Одним из главных окон, через которые происходит работа над проектом, является окно Обозревателя решений.

### 2.4 Жизненный цикл

Жизненный цикл программного обеспечения состоит из анализа требований, проектирования, когда задаются основные черты будущей программной системы, собственно разработки, когда происходит создание кода программ, и эксплуатации, когда созданное программное обеспечение передаётся конечным потребителям – пользователям ПО.

Во время этапа эксплуатации происходит устранение ошибок, доработка программного обеспечения в связи с вновь открывшимися обстоятельствами, пожеланиями пользователей , изменившихся условий среды, новыми возможностями аппаратного и инструментального программного обеспечения, появлением новых программных сред, языков, методологий разработки.

Во время всех этапов применятся принцип итеративности, последовательных улучшений программного обеспечения, сравнение достигнутых результатов с требованиями, и изменение требований согласно решаемой задачи. Итеративность может происходить вплоть до возврата к анализу требований, если окажется в процессе эксплуатации какие – то важные требования пользователей не были учтены, или успели появиться новые.

### 2.5 Процессы разработки программного обеспечения

Процессы проектирования и разработки программного обеспечения (англ. software development process) — процесс, посредством которого потребности пользователей преобразуются в программный продукт. Его можно разбить на :

1. Анализ требований
2. Спецификация программного обеспечения,
3. Проектирование программного обеспечения
4. Программирование

Каждый из этих подпроцессов тоже может быть детализирован, например, анализ требований может состоять из нескольких этапов извлечения требований,спецификации и проверки.

Спецификация разбивается на выбор программного и аппаратного обеспечения в виде типов компьютеров, их расположения, типа операционных систем, выбора программных сред и языков программирования.

При проектировании выбирается методология пректирования, такая как **Scrum** , **Kanban** ,**RUP** (Rational Unified Process),**DSDM** (Dynamic Systems Development Model),**RAD** (Rapid Application Development),**XP** (Extreme Programming).

Далее по выбранной методологии проетируется программное обеспечение с применением либо объектно – ориентированного подхода, либо приёмов функционального программирования

2.6 Модели процессов разработки программного обеспечения   
Существует несколько моделей процесса разработки программного обеспечения, в которых жизненный цикл организуется в стадии, которые проходятся по -разному, последовательно или итеративно. В зависимости от порядка прохождения этих стадий модели классифицируются на следующие виды:

* Каскадная модель (она же “водопадная” - waterfall)
* Итерационные модели
* Инкрементная модель
* Спиральная модель

Каскадная модель – это модель с минимумом итеративности, все этапы разработки проходятся последовательно, возврат на предыдущие этапы нежелателен и происходит только в виде исключений.

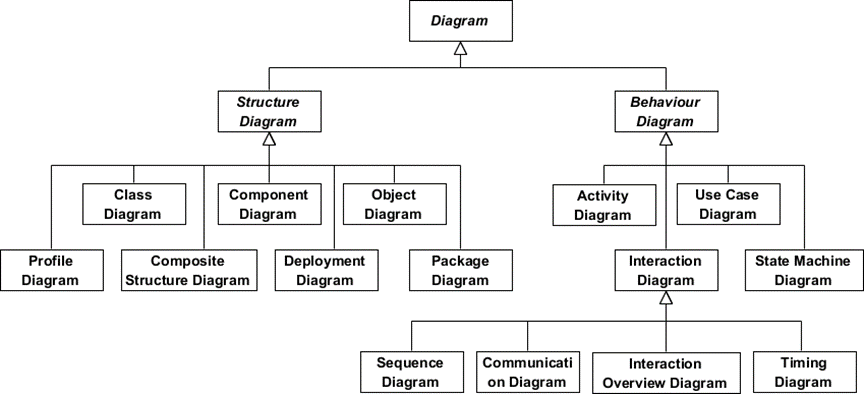
.Итерационные модели разных типов подразумевают возврат к предудущим этапам разработки как законную часть процесса. Например, в инкрементной модели происходит поэтапное улучшение путём добавления новых функций к модели, а в спиральной на каждом этапе происходит улучшение всех функций модели.

В нашей автоматизированной информационной системе «Абитуриент» мы применим инкрементную модель, то есть спроектируем и осуществим первоначально только одну функцию ввода нового абитуриента в базу данных, а затем будеи добавлять функции просмотра, изменеиия, удаления личных данных и увеличение их набора, то есть количества информации, относящейся к абитуриенту : фотокарточка, медицинская справка, любые другие дополнительные сведения, которые могут оказаться необходимыми при работе с поступающими в колледж.

### 2.7 Язык моделирования UML

Язык моделирования UML(Unified Modelling Language) - это унифицированный графический язык для описания прграммных систем и бизнес -процессов. С помощью языка UML можно описывать программные структуры и присходящие в них процессы, а также структуры и процессы реального мира. Ниже перечисляются основные типы диаграмм, которые применяются при использовании в конструкциях этого языка:

1. Диаграммы классов
2. Диаграммы последовательностей
3. Диаграммы объектов
4. Диаграммы пакетов
5. Диаграммы развёртывания
6. Диаграммы вариантов использования
7. Диаграммы состояний
8. Диаграммы деятельности
9. Диаграммы коммуникаций
10. Диаграммы составных структур
11. Диаграммы компонентов
12. Диаграммы профилирования
13. Диаграммы обзора взаимодействий
14. Временные диаграммы

Часть из них относится к структурным диаграммам, часть к диаграммам поведения. 

Наиболле часто используются диаграммы классов, компонентов, диаграммы вариантов использования(взаимодействия с пользователем), диаграммы деятельности, диаграммы состояний.

Чтобы отобразить структуру базы данных,можно применить диаграмму класса. Наша база данных в начале итеративного процесса разработки состоит только из одной таблицы.



Как видно из рисунка, диаграмма UML представляется в виде прямоугольника, сверху подписывается название класса, далее по вертикали располагаются поля класса, которые в данном случае представляют поля базы данных. Отдельные диаграммы могут соединятся стрелками, отображающими связи между ними. (Пример дан в Приложении А).

# ЗАКЛЮЧЕНИЕ

Целью данной работы было исследование вопроса, какие методологии разработки программного обеспечения могут быть применены для разработки программного обеспечения автоматизированной информационной системы «Абитуриент», какие инструменты могут быть использованы для её создания.

В процессе данного исследования была проработана литература, относящаяся к теме исследования, был выполнен поиск информации в сети Интернет, были произведены сравнения различных точек зрения на методологии проектирования и выбор инструментария для создания программного обеспечения.

В процессе исследования автор пришёл к выводу, что язык проектирования UML, наряду с языком BPMN 2.0 могут быть с успехом использованы для описания проектируемой информационной системы, а программное обеспечение для неё может быть создано на инструментах Microsoft, в среде программирования Visual Studio 2019 на языке C#, позволяющих использовать все самые современные достижения в области разработки ПО.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Бахтизин В.В. Технология разработки программного обеспечения: учеб. пособие / В.В.Бахтизин, Л.А.Глухова. – Минск: БГУИР, 2010. – 267 с.

2. Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения: Учебник для вузов. /4-е изд. Стандарт третьего поколения. – СПб.: Питер, 2012. – 608 с.

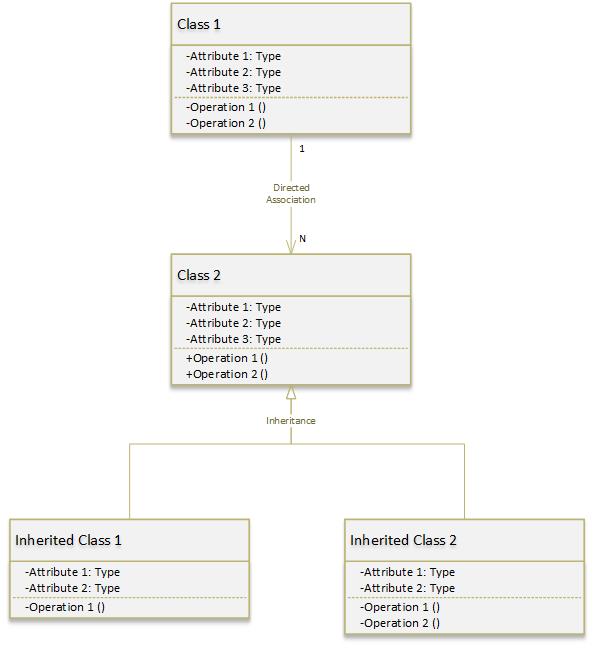
3. Мацяшек Л.А. Практическая программная инженерия на основе учебного примера [Электронный ресурс] / Л.А. Мацяшек, Б.Л. Лионг: пер. с англ. – 2-е изд. (эл.) – М.: БИНОМ. Лаборатория знаний, 2012. – 856с.

4. Ларман, Крег. Применение UML 2.0 шаблонов проектирования. Практическое руководство. /3-е изд.: Пер. с англ. – М.: ООО «И.Д.Вильямс», 2013. – 736 с.

5. И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров Технология разработки программного обеспечения /Учебное пособие для СПО Санкт-Петербур[ский государственный экономическии университет, 2019 г.

# ПРИЛОЖЕНИЕ А

Пример диаграммы классов в нотации UML

****