Министерство образования Республики Беларусь

Учреждение образования

«БелорусскиЙ государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра экономической информатики

Дисциплина «Программирование сетевых приложений»

|  |
| --- |
|  |
|  |
|  |

**Пояснительная записка**

к курсовой работе

на тему

**Разработка** **аВТОМАТИЗИРОВАННОЙ СИСТЕМЫ по расчёту студенческих стипендий**

БГУИР КР 1-40 05 01 №21 ПЗ

Студент

Группы 614302 Михайловская Е.С.

Руководитель Петрович Н.О.

Минск 2018

# **СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение……………………………………………………………………... |  |
| 1. Описание алгоритма расчёта студенческих стипендий………………… |  |
| 2. Постановка задачи и обзор методов ее решения ...................................... |  |
| 3. Описание процесса автоматизации расчета студенческих стипендий.... |  |
| 3.1 |  |
| 4. Информационная модель системы учёта ………………………………... |  |
| 5. Модели представления системы и их описание …………………………. |  |
| 5.1 Диаграмма вариантов использования системы ……………………… |  |
| 5.2 Диаграмма состояния …………………………………………………. |  |
| 5.3 Диаграмма последовательности …………………………………… |  |
| 5.4 Диаграмма классов ……………………………………………………. |  |
| 5.5 Диаграмма компонентов ……………………………………………… |  |
| 5.6 Диаграмма развёртывания ……………………………………………. |  |
| 6. Описание алгоритмов, реализующих бизнес-логику серверной части системы …………………………………………………………………… |  |
| 6.1 |  |
| 6.2 |  |
| 7. Руководство пользователю ………………………………………………. |  |
| 8. Результаты тестирования разработанной системы ……………………. |  |
| Заключение…………………………………………………………………… |  |
| Список использованных источников……………………………….............. |  |
| Приложение А. ……………………………………......................................... |  |
| Приложение Б. ……………………………………………............................. |  |
| Приложение В ………………………………………………………………. |  |
| Приложение Г……………………………………………………………....... |  |
| Приложение Д ………………………………………………………………. |  |
|  |  |
|  |  |

**ВВЕДЕНИЕ**

На сегодняшний день процесс развития информационных технологий очень динамичен, их возможности для общества и экономики начинают масштабно эксплуатироваться. В настоящий момент практически невозможно найти даже малую организацию в нашей стране, которая не использует IT-технологии в своей работе.

От качества и масштабов использования информационных технологий напрямую зависит уровень экономического и социального развития общества. Без внедрения IT-технологий невозможно полноценное функционирование современных учреждений. Они повышают эффективность работы как отдельных сотрудников, так и предприятий в целом. Именно благодаря современным технологиям уменьшается количество ошибок, и при этом увеличивается скорость поиска информации и обработки данных.

В настоящее время невозможно управление предприятием без ПК. Компьютерные технологии давно используются в медицине, науке, менеджменте, бухгалтерском учете.

Всемирная паутина стремительно развивается, использование её технологий даёт безграничное количество возможностей для программистов и пользователей. Интернет объединяет миллиарды компьютеров в сеть и при помощи определённых программ позволяет получить доступ к нужной информации на одном компьютере с другого.

Разработка процесса автоматизации будет гарантировать целостность данных, обеспечит возможность распределённого и одновременного доступа к ним. Также процесс автоматизации приведёт к устойчивой формализации данных и уменьшению бумажного документооборота между отделами.

Целью данного курсового проекта является автоматизация системы расчёта студенческих стипендий.

Сам процесс начисления стипендий довольно трудоёмкий: необходимо определить критерии, по которым будут отбираться студенты для получения стипендии, определить минимальную ставку начисления стипендии, определить таблицу коэффициентов, согласно которым стипендия может индексироваться в зависимости от величины достижений студента и прочих факторов, влияющих на начисление денежной выплаты. Безусловно автоматизация данного процесса является замечательным решением, упрощающим этот трудоёмкий процесс.

Для достижения заданной цели, были поставлены следующие задачи:

* анализировать предметную область;
* разработать объектную модель для выбранной предметной области;
* составить диаграмму классов и БД;
* разработать алгоритм работы приложения;
* создать удобный интерфейс, который будет выполнять действия, необходимые реализации цели курсового проекта;
* протестировать разработанную систему.

1. **ОПИСАНИЕ АЛГОРИТМА РАСЧЁТА СТУДЕНЧЕСКИХ СТИПЕНДИЙ**

Формы бухгалтерского учёта, применяемые технологии обработки учётной информации (используемые компьютерные программы) предусматриваются в учётной политике организации. В Беларуси функционирует развитая система социальной поддержки учащейся молодежи. Система стипендиального обеспечения в республике включает в себя именные, учебные, социальные, персональные стипендии совета учреждения высшего образования, стипендии аспирантам, докторантам, руководящим кадрам, специальные, а также стипендии Президента Республики Беларусь. Стипендии представляют одну из форм государственной материальной помощи студентам. В современных условиях значение стипендий возрастает, усиливается их стимулирующая роль в повышении качества учебы и творческой активности учащейся молодежи.

Условия начисления стипендии зависят от нескольких факторов: успеваемости студента (среднего балла в последней сессии), социального положения студента, участия его в общественной жизни учебного заведения и другое.

Система управления учебным заведением ‒ сравнительно сложная система. Сложность ее объясняется рядом причин, из которых основными являются: большое число элементов системы (объектов и процессов деятельности), высокая степень их связанности и взаимозависимости при реализации процессов управления, неопределенность результатов выполнения ряда процессов, нечеткость связи между элементами, значительная зависимость от внешних факторов.

Определение планового объема затрат – это начальный этап организации любой деятельности, обоснование ее целесообразности, прогнозирование ее результатов. Поэтому и в составе финансовой работы учебного заведения планированию затрат отводится должное внимание.

Для государственного учебного заведения плановый объем затрат предопределяет объем бюджетного финансирования.

*Бюджетное финансирование* – это предоставление средств из государственного бюджета для субъектов хозяйствования (учреждений, организаций и т.д.) на определённые цели, предусмотренные бюджетом. Финансирование из бюджета направлено на выполнение функций государства, оно производится на основании годовой росписи с поквартальным распределением доходов и расходов бюджета.

Выплата стипендий – это элемент технологии подготовки дипломированных специалистов. Как показали исследования, его фактического объема недостаточно для обеспечения жизненного уровня обучающихся.

1. **ПОСТАНОВКА ЗАДАЧИ И ОБЗОР МЕТОДОВ ЕЁ РЕШЕНИЯ**

Прогрессивное развитие компьютерных технологий спровоцировало возникновение совершенно новых знаковых систем для записи алгоритмов – языки программирования.

Задачей данного курсового проекта является создание клиент-серверного программного приложения, позволяющего автоматизировать систему расчета стипендий. В курсовом проекте используется язык Java.

Java – объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems с 1991 года и официально выпущенный 23 мая 1995 года. Изначально новый язык программирования назывался Oak (James Gosling) и разрабатывался для бытовой электроники, но впоследствии был переименован в Java и стал использоваться для написания апплетов, приложений и серверного программного обеспечения.

Язык Java зародился как часть проекта создания передового программного обеспечения для различных бытовых приборов. Реализация проекта была начата на языке С++, но вскоре возник ряд проблем, наилучшим средством борьбы с которыми было изменение самого инструмента – языка программирования. Стало очевидным, что необходим платформо-независимый язык программирования, позволяющий создавать программы, которые не приходилось бы компилировать отдельно для каждой архитектуры и можно было бы использовать на различных процессорах под различными операционными системами.

Одно из основных преимуществ языка Java – независимость от платформы, на которой выполняются программы: один и тот же код можно запускать под управлением операционных систем Windows, Solaris, Linux и др. Это действительно необходимо, когда программы загружаются через Интернет для последующего выполнения под управлением различных операционных систем.

Другое преимущество заключается в том, что синтаксис языка Java похож на синтаксиса языка С++, и программистам, знающим языки С и С++, его изучение не составляет труда. Правда, для программистов, владеющим языком Visual Basic, этот синтаксис, возможно, будет непривычен.

Три ключевых элемента объединились в технологии языка Java:

* Java предоставляет для широкого использования свои апплеты (applets) – небольшие, надежные, динамичные, не зависящие от платформы активные сетевые приложения, встраиваемые в страницы Web. Апплеты Java могут настраиваться и распространяться потребителям с такой же легкостью, как любые документы HTML.
* Java высвобождает мощь объектно-ориентированной разработки приложений, сочетая простой и знакомый синтаксис с надежной и удобной в работе средой разработки. Это позволяет широкому кругу программистов быстро создавать новые программы и новые апплеты.
* Java предоставляет программисту богатый набор классов объектов для ясного абстрагирования многих системных функций, используемых при работе с окнами, сетью и для ввода-вывода. Ключевая черта этих классов заключается в том, что они обеспечивают создание независимых от используемой платформы абстракций для широкого спектра системных интерфейсов.

Автоматизированная информационная система расчёта заработной платы имеет клиент-серверную архитектуру.

Основной принцип клиент-серверной архитектуры заключается в разделении функций приложения на три группы:

* ввод и отображение данных (взаимодействие с пользователем);
* прикладные функции, характерные для данной предметной области;
* функции управления ресурсами (файловой системой, базой даных и т.д.)

Поэтому, в любом приложении выделяются следующие компоненты:

* компонент представления данных;
* прикладной компонент;
* компонент управления ресурсом.

###### **ОПИСАНИЕ ПРОЦЕССА АВТОМАТИЗАЦИИ РАСЧЁТА СТУДЕНЧЕСКИХ СТИПЕНДИЙ**

На основе алгоритма расчета стипендий построена функциональная модель, которая описывает существующую организацию работы. В данной курсовой работе использовалось средство функционального моделирования BPwin.

BPwin – мощный инструмент моделирования, разработанный фирмой Computer Associates Technologies, который используется для анализа, документирования и реорганизации сложных бизнес-процессов. Модель, созданная средствами BPwin, позволяет четко документировать различные аспекты деятельности – действия, которые необходимо предпринять, способы их осуществления, требующиеся для этого ресурсы и др. Таким образом, формируется целостная картина деятельности предприятия – от моделей организации работы в маленьких отделах до сложных иерархических структур. При разработке или закупке программного обеспечения модели бизнес-процессов служат прекрасным средством документирования потребностей, помогая обеспечить высокую эффективность инвестиций в сферу IT. В руках же системных аналитиков и разработчиков BPwin – еще и мощное средство моделирования процессов при создании корпоративных информационных систем (КИС).

Модели BPwin дают основу для осмысления бизнес-процессов и оценки влияния тех или иных событий, а также описывают взаимодействие процессов и потоков информации в организации. Неэффективная, высоко затратная или избыточная деятельность может быть легко выявлена и, следовательно, усовершенствована, изменена или устранена в соответствии с общими целями организации.

С помощью BPwin пользователь может сделать свою работу более продуктивной. Действия и другие объекты создаются буквально несколькими щелчками мыши, а затем легко отбуксированы в нужное место. Интерфейс BPwin, выполненный в стиле "проводника" облегчает навигацию и редактирование сложных процессов с иерархической структурой. Развитые возможности изменения масштаба представления позволяют быстро найти и сосредоточиться на необходимой для работы части модели процесса.

BPwin позволяет:

* Обеспечить эффективность операций, рассматривая текущие бизнес-операции через мощные инструменты моделирования.
* Совершенствовать бизнес-процессы, формулируя и определяя альтернативные реакции на воздействия рынка.
* Быстро исключать непродуктивные операции, легко и интуитивно сопоставляя операционные изменения. Неэффективные, неэкономичные или избыточные операции могут быть легко выявлены и, следовательно, улучшены, изменены или вовсе исключены - в соответствии с целями компании.

BPWin совмещает в одном инструменте средства моделирования функций (IDEF0), потоков данных (DFD) и потоков работ (IDEF3).

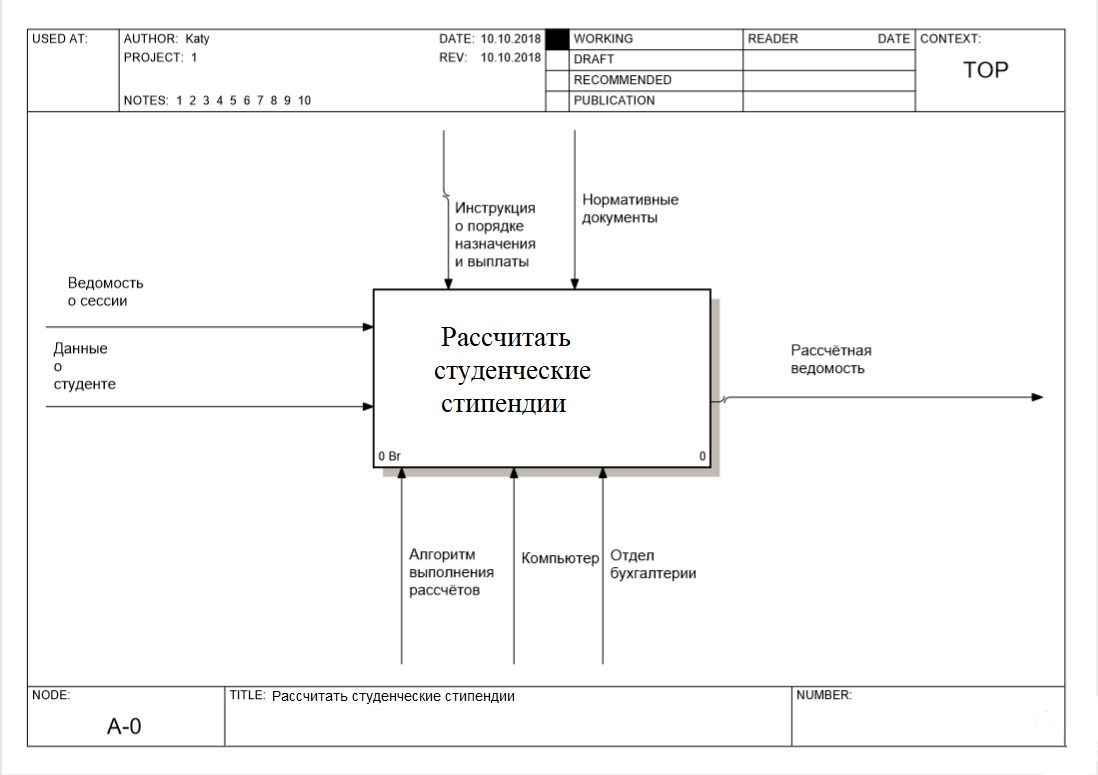
С помощью *функционального моделирования (нотация IDEF0),* можно провести систематический анализ процессов и систем, сосредоточившись на регулярно решаемых задачах, свидетельствующих об их правильном выполнении показателях, необходимых для этого ресурсах, результатах и исходных материалах.

*Моделирование потоков данных (DFD),* сосредоточено вокруг потоков данных, передающихся между различными операциями, включая их хранение, для достижения максимальной доступности и минимального времени ответа.

*Моделирование потоков работ (нотация IDEF3)* позволяет рассмотреть конкретный процесс, проанализировать операции, из которых он состоит, а также точки принятия решений, влияющих на его ход.

Каждая из них может быть выполнена отдельно с помощью BPwin, но их совокупность, заключённая в модель, даёт аналитику полную картину предметной области клиента.

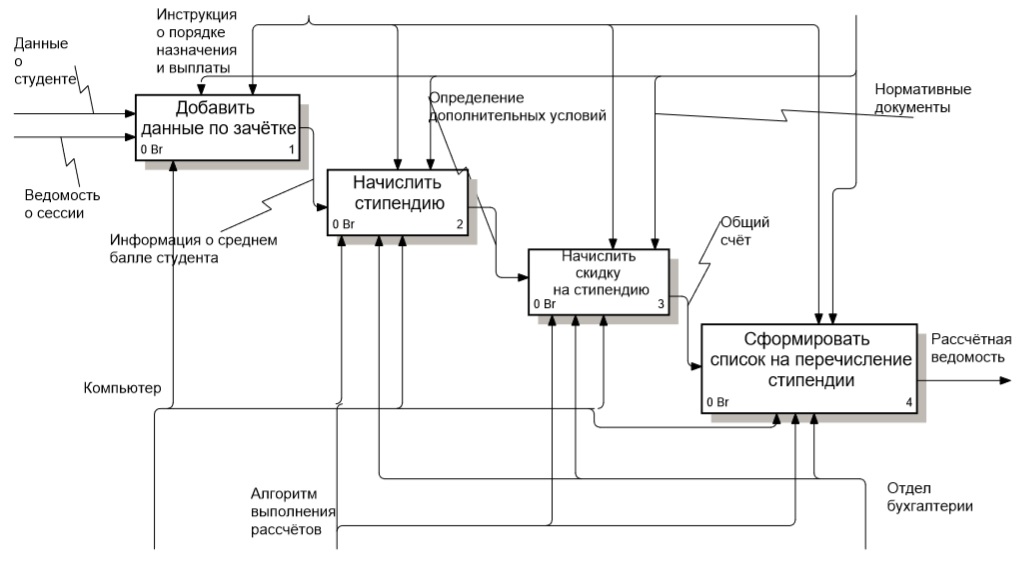
Построим контекстную диаграмму, которая предоставит нам общее описание работы автоматизированной системы расчёта стипендии:



*Рисунок 3.1 Контекстная диаграмма*

Изначально на входе имеется «Данные о студенте», «Ведомость о сессии». Элементами управления выступают «Инструкция о порядке назначения и выплаты стипендий» и «Нормативные документы». А механизмы управления – «Отдел бухгалтерии», «Компьютер» и «Алгоритм выполнения расчётов». На выходе получаем «Расчетную ведомость».

Составляющими блоками диаграммы являются: «Добавить данные по зачётке», «Начислить стипендию», «Начислить скидку на стипендию» и «Сформировать список на перечисление стипендии».

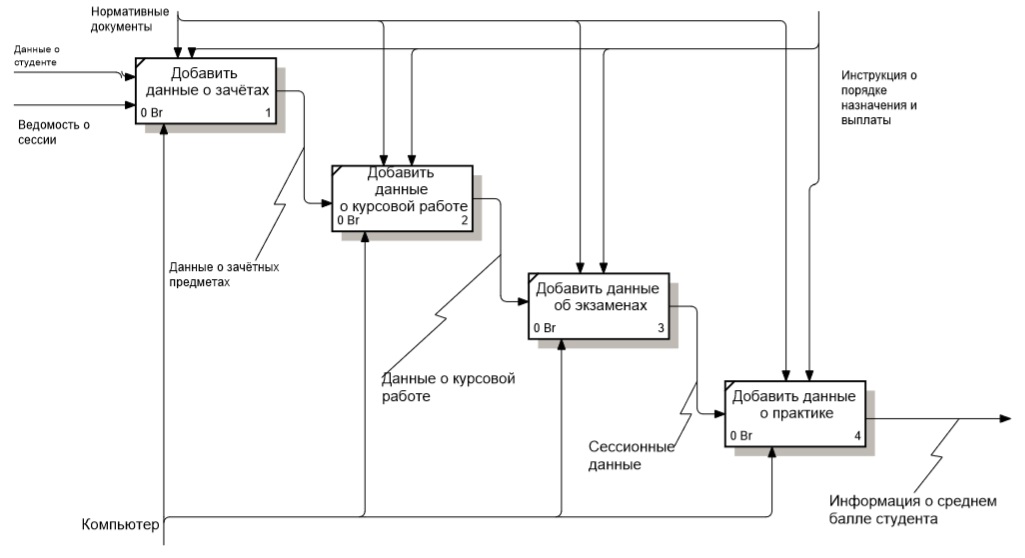


*Рисунок 3.2 Декомпозиция первого уровня*

Теперь более подробно рассмотрим первый элемент представленной выше декомпозиции: «Добавить данные по зачётке».

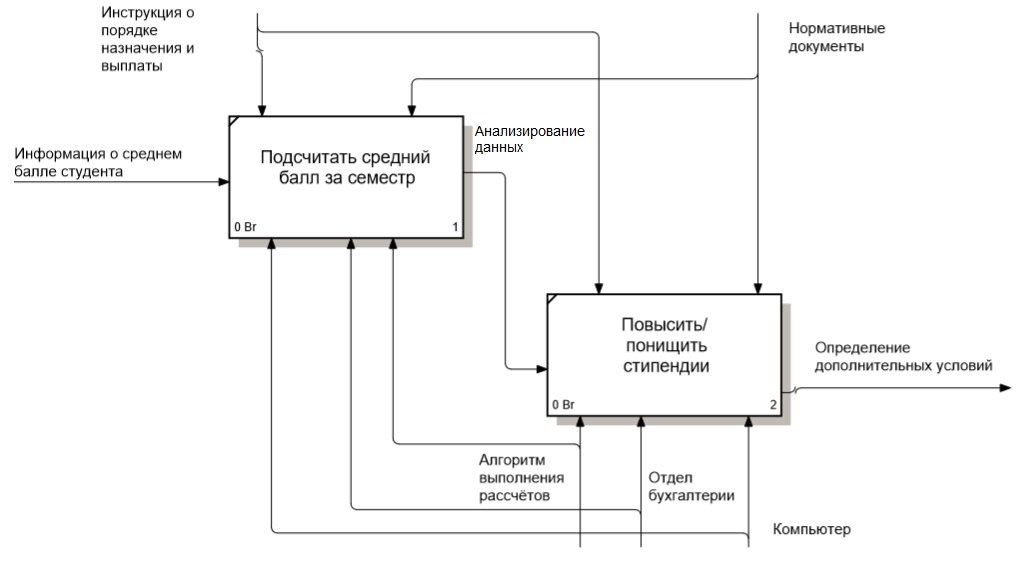
Он состоит из «Добавить данные о зачётке», «Добавить данные о курсовой работе», «Добавить данные об экзаменах» и «Добавить данные о практике».

Зачётка ‒ документ, в котором содержатся записи о сдаче студентом зачётов, экзаменов, защите курсовых, выпускных квалификационных работ (для бакалавров), дипломных работ, и диссертаций (для магистров), а также производственной практики.



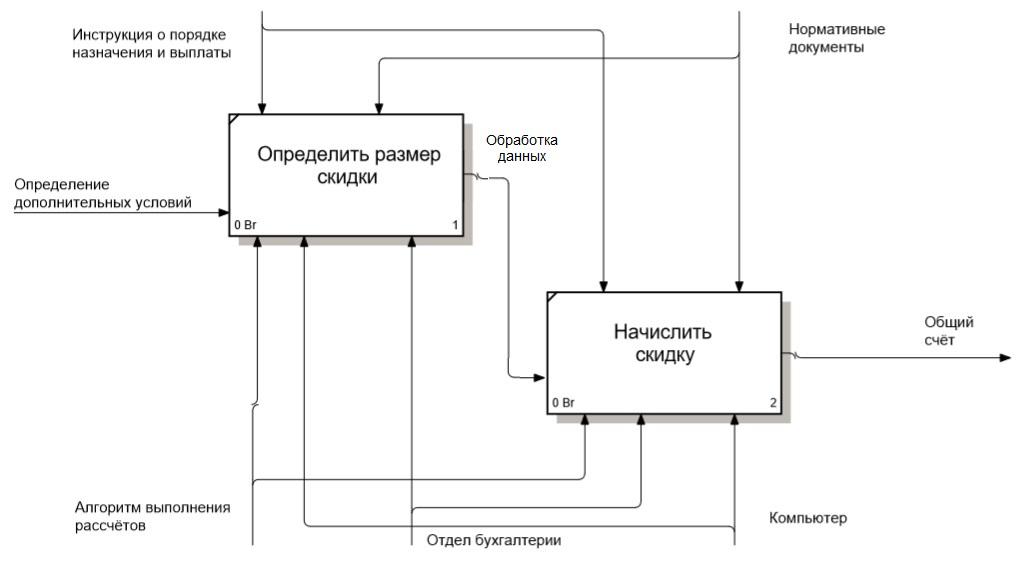
*Рисунок 3.3 Декомпозиция «Добавить данные по зачётке»*

Следующая диаграмма показывает процесс начисления стипендий. Декомпозиция содержит следующие блоки: «Подсчитать средний балл за семестр» и «Повысить/ понизить стипендии».



*Рисунок 3.4 Декомпозиция «Начислить стипендию»*

Пятая диаграмма показывает, как производится расчёт скидки. Декомпозиция содержит следующие блоки: «Определить размер скидки» и «Начислить скидку».

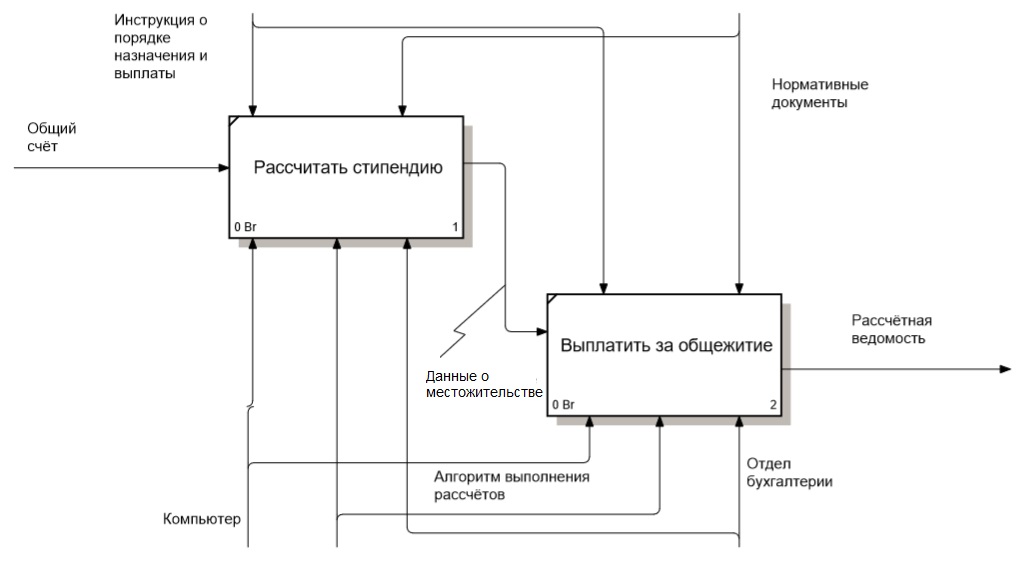


*Рисунок 3.5 Декомпозиция «Начислить скидку на стипендию»*

Последние блок «Сформировать список перечисления стипендии» имеет следующую декомпозицию: «Рассчитать стипендию» и «Выплатить за общежитие».

Наниматели (студенты) жилых помещений в общежитии вносят в соответствии с законодательством плату за пользование жилым помещением, а также плату за жилищно-коммунальные услуги по тарифам для населения, установленным в соответствии с законодательными актами.

Плата за услуги, предоставляемые проживающим в общежитии гражданам (обеспечение постельными принадлежностями, мебелью, стирка белья, прокат предметов культурно-бытового назначения и другое), осуществляется исходя из фактических затрат на оказание этих услуг.



*Рисунок 3.6 Декомпозиция «Сформировать список перечисления стипендии»*

1. **ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ УЧЁТА**

###### **МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ И ИХ ОПИСАНИЕ**

***UML-диаграмма*** – это специализированный язык графического описания, предназначенный для объектного моделирования в сфере разработки различного программного обеспечения. Данный язык имеет широкий профиль и представляет собой открытый стандарт, в котором используются различные графические обозначения, чтобы создать абстрактную модель системы.

Применение UML не заканчивается на моделировании всевозможного ПО. Также данный язык активно сегодня используется для моделирования различных бизнес-процессов, ведения системного проектирования, а также отображения организационных структур. С помощью UML разработчики программного обеспечения могут обеспечить полное соглашение в используемых графических обозначениях, чтобы представить общие понятия, такие как: компонент, обобщение, класс, поведение и агрегация. За счет этого достигается большая степень концентрации на архитектуре и проектировании.

Словарь UML включает три вида строительных блоков:

* Диаграммы.
* Сущности.
* Связи.

***Сущности*** – это абстракции, которые являются основными элементами модели, ***связи*** соединяют их между собой, а *диаграммы* группируют представляющие интерес наборы сущностей.

***Диаграмма*** – это графическое представление набора элементов, чаще всего изображенного в виде связного графа вершин (сущностей) и путей (связей).

UML представляет собой объектно-ориентированный язык моделирования, обладающий следующими основными характеристиками:

* Является языком визуального моделирования, который обеспечивает разработку репрезентативных моделей для организации взаимодействия заказчика и разработчика, различных групп разработчиков ИС;
* Содержит механизмы расширения и специализации базовых концепций языка.
  1. **ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ**

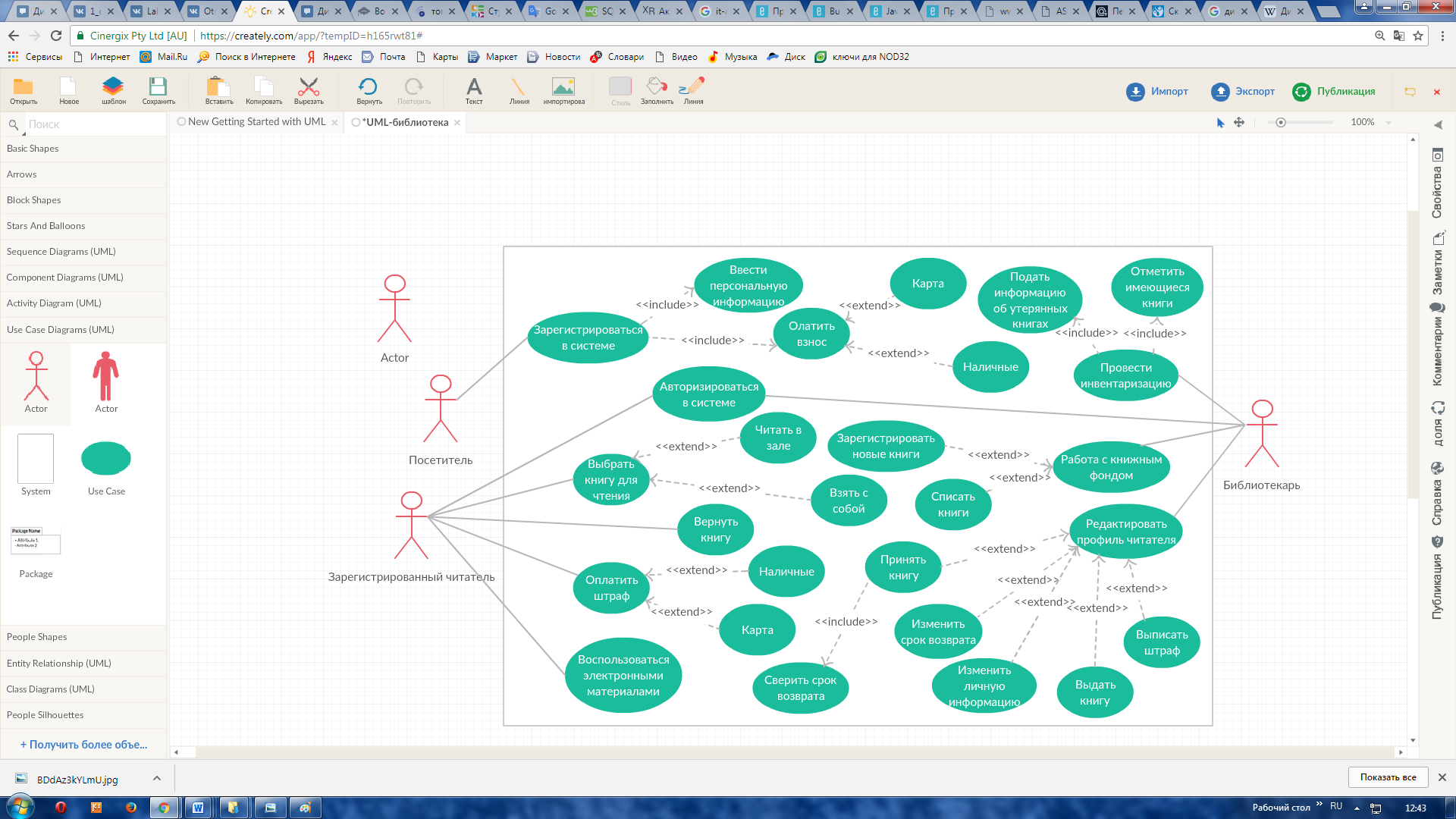
*Диаграмма прецедентов* (диаграмма вариантов использования) в UML – диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Основное назначение диаграммы – описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

Для отражения модели прецедентов на диаграмме используются*:*

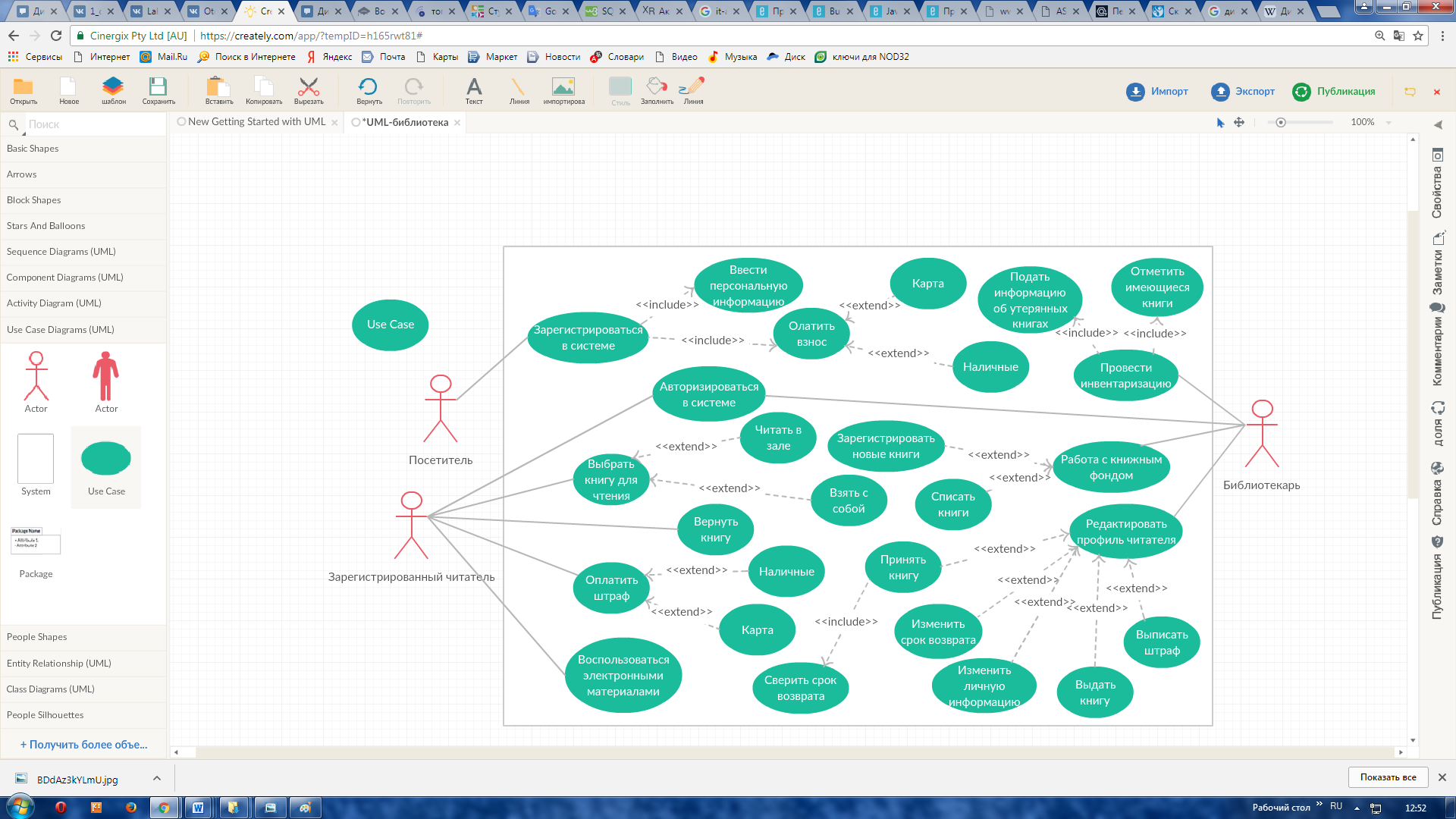
* **рамки системы** – прямоугольник с названием в верхней части и эллипсами (прецедентами) внутри. Часто может быть опущен без потери полезной информации.
* **актёр** – стилизованный человечек, обозначающий набор ролей пользователя (понимается в широком смысле: человек, внешняя сущность, класс, другая система), взаимодействующего с некоторой сущностью. Актёры не могут быть связаны друг с другом (за исключением отношений обобщения/наследования).

Стандартное графическое изображение актера:



* **прецедент** – эллипс с надписью, обозначающий выполняемые системой действия (могут включать возможные варианты), приводящие к наблюдаемым актёрами результатам.

Стандартное графическое изображение прецедента:



Надпись может быть именем или описанием (с точки зрения актёров) того, «что» делает система (а не «как»). Имя прецедента связано с непрерываемым (атомарным) сценарием – конкретной последовательностью действий, иллюстрирующей поведение. В ходе сценария актёры обмениваются с системой сообщениями. Сценарий может быть приведён на диаграмме прецедентов в виде UML-комментария. С одним прецедентом может быть связано несколько различных сценариев.

Виды отношений:

* **ассоциативное отношение.**

Отношение между вариантом использования и актером, отражающее связь между ними. Оно устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования.

* **отношение обобщения** – стрелка с не закрашенным треугольником.

Служит для указания того факта, что некоторый вариант использования А может быть обобщен до варианта использования Б (или актер А может быть обобщен до актера Б).

* **отношение включения** – пунктирная стрелка со стереотипом «include».

Указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования.

* **отношение расширения** – пунктирная стрелка со стереотипом «extend» (стрелка входит в расширяемый прецедент, в дополнительном разделе которого может быть указана точка расширения и, возможно в виде комментария, условие расширения).

Определяет взаимосвязь базового варианта использования с некоторым другим вариантом использования, функциональное поведение которого задействуется базовым не всегда, а только при выполнении некоторых дополнительных условий.

Диаграмма вариантов использования разрабатываемой системы представлена в Приложении А рисунок А.1.

* 1. **ДИАГРАММА СОСТОЯНИЯ**

*Диаграмма состояний* – один из способов детального описания поведения в UML на основе явного выделения состояний и описания переходов между состояниями. Объекты характеризуются поведением и состоянием, в котором находятся. На диаграмме автомата применяют один основной тип сущностей ‒ состояния, и один тип отношений ‒ переходы, но и для тех и для других определено множество разновидностей, специальных случаев и дополнительных обозначений.

На диаграмме состояний используются следующие условные обозначения:

* круг, обозначающий начальное состояние.
* окружность с маленьким кругом внутри, обозначающая конечное состояние (если есть).
* скруглённый прямоугольник, обозначающий состояние. Верхушка прямоугольника содержит название состояния. В середине может быть горизонтальная линия, под которой записываются активности, происходящие в данном состоянии.
* стрелка, обозначающая переход. Название события (если есть), вызывающего переход, отмечается рядом со стрелкой.
* толстая горизонтальная линия с либо множеством входящих линий и одной выходящей, либо одной входящей линией и множеством выходящих.

Диаграмма состояния разрабатываемой системы представлена в Приложении А рисунок А.2.

* 1. **ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ**

*Диаграмма последовательности* (sequence diagram) ‒ способ описания поведения системы на основе указания последовательности передаваемых сообщений. Фактически, диаграмма последовательности – это запись протокола конкретного сеанса работы системы.

Диаграмма последовательностей, так же, как и вариант использования, может быть реализована как в терминах бизнес-объектов, так и в терминах физических сущностей, таких как компоненты или классы.

Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни», отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами.

Диаграмма последовательности разрабатываемой системы представлена в Приложении А рисунок А.3.

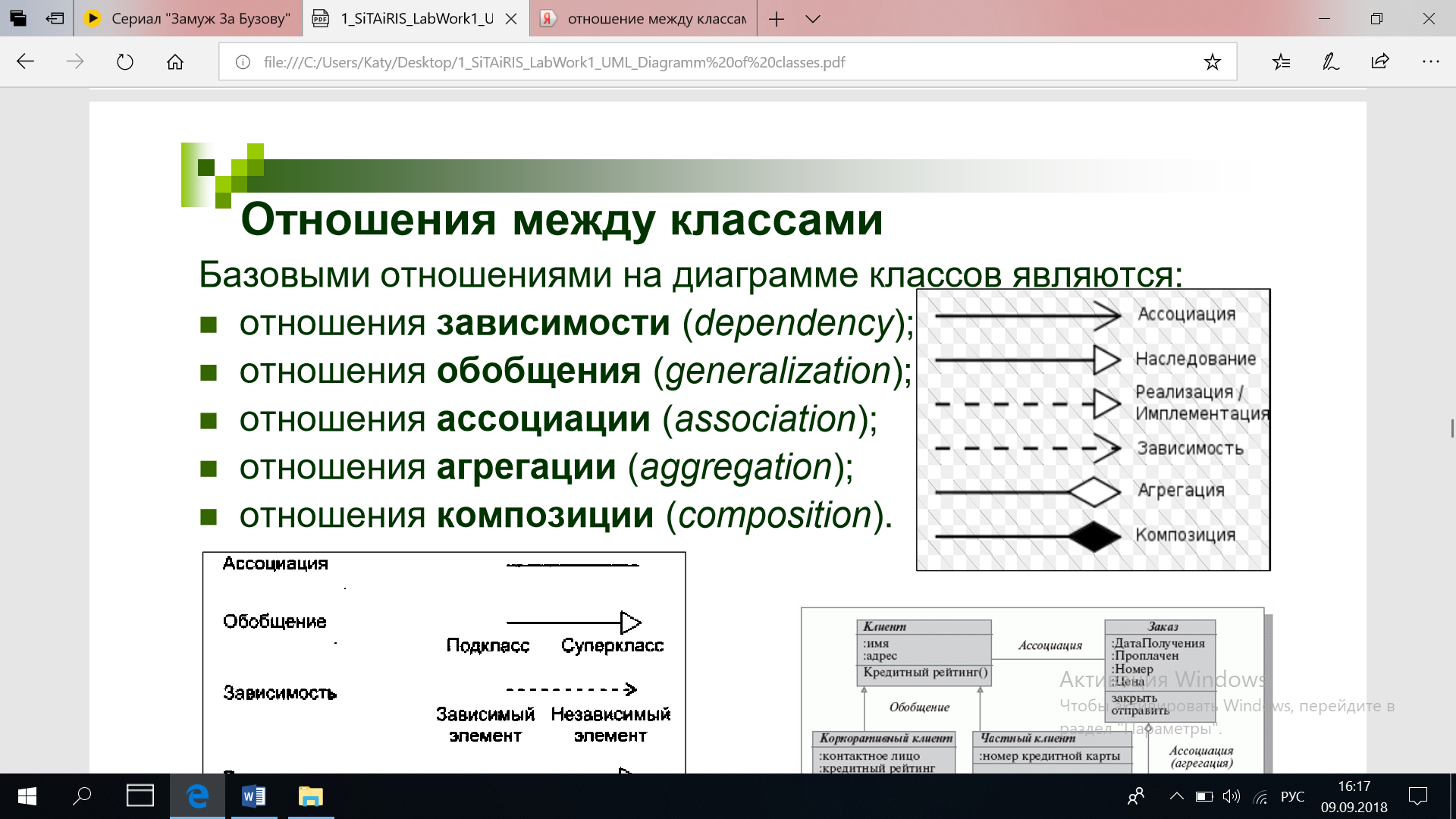
* 1. **ДИАГРАММА КЛАССОВ**

*Диаграмма классов* содержит информацию об объектах системы и статических связях между объектами.

Класс – это множество объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов. Полное описание класса, состоящее из 3 разделов – секции имени, секции атрибутов, секции операций.

Класс может не иметь экземпляров или объектов. В этом случае он называется абстрактным классом, а для обозначения его имени используется курсив.

Базовыми отношениями на диаграмме классов являются:



Существует несколько точек зрения на построение таких диаграмм в зависимости от того, каким образом они будут использоваться:

1. *Концептуальная*. В данном случае диаграмма классов UML осуществляет описание модели определенной предметной области, и в ней предусматриваются только классы прикладных объектов.
2. *Специфическая*. Диаграмма используется в процессе проектирования различных информационных систем.
3. *Реализационная*. Диаграмма классов включает в себя всевозможные классы, которые непосредственно используются в программном коде.

Диаграмма классов разрабатываемой системы представлена в Приложении А рисунок А.4.

* 1. **ДИАГРАММА КОМПОЗИЦИИ**

*Диаграмма компонентов* (component diagram) ‒ показывает взаимосвязи между модулями (логическими, физическими), из которых состоит моделируемая система. Компонент – это физический элемент реализации с четко определенным интерфейсом, предназначенный для использования в качестве заменяемой части системы.

Каждый компонент представляет собой реализацию некоторых классов системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами. Во многих средах разработки модуль или компонент соответствует файлу.

На данной диаграмме применяются следующие отношения:

* реализации между компонентами и интерфейсами (компонент реализует интерфейс);
* зависимости между компонентами и интерфейсами (компонент использует интерфейс).
  1. **ДИАГРАММА РАЗВЁРТЫВАНИЯ**

*Диаграмма развертывания* отражает физические взаимосвязи между программными и аппаратными компонентами разрабатываемой системы. Она относится к статическому виду системы и предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения.

На диаграмме развёртывания, по сравнению с диаграммой компонентов, добавляется два типа сущностей: ***артефакт***, который является реализацией компонента и ***узел*** (может быть как классификатор, описывающий тип узла, так и конкретный экземпляр), а также отношение *ассоциации* между узлами, показывающее, что узлы физически связаны во время выполнения. На диаграммах развертывания может быть показана привязка (в некоторый момент времени или постоянная) компонентов развертывания системы к физическим устройствам.

Диаграмма развёртывания разрабатываемой системы представлена в Приложении А рисунок А.5.

###### **ОПИСАНИЕ АЛГОРИТМОВ, РЕАЛИЗУЮЩИХ БИЗНЕС-ЛОГИКУ СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ**

1. **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЮ**
2. **РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ РАЗРАБОТАННОЙ СИСТЕМЫ**

**ЗАКЛЮЧЕНИЕ**

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Методологии моделирования, поддерживаемые bpWin. [Электронный ресурс]. – Режим доступа: https://studfiles.net/preview/383532/page:2/

[2] UML-диаграмма. Виды диаграмм UML. [Электронный ресурс]. – Режим доступа: https://www.syl.ru/article/206012/new\_uml-diagramma-vidyi-diagramm-uml

[3] Java. Основы и история создания Java. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Java

[4] BPwin. IDEF 0. [Электронный ресурс]. – Режим доступа: https://habr.com/company/trinion/blog/322832/

[5]

[6] Хольцшлаг, М. / Языки HTML и CSS для создания Web-сайтов / М. Хольцшлаг. - М.: Триумф, 2007. – 304 с.

[7]

[8]

**ПРИЛОЖЕНИЕ А**

**(Основные теги html)**

**ПРИЛОЖЕНИЕ В**

**(Листинг кода)**