МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

КУРСОВОЙ ПРОЕКТ  
ЗАЩИЩЕН С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Е.О. Пятлина |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ | | | | |
| ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ «КНИЖНЫЙ МАГАЗИН» СРЕДСТВАМИ ЯЗЫКА UML | | | | |
| по дисциплине: ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ | | | | |

РАБОТУ ВЫПОЛНИЛА

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТКА ГР.№ |  |  |  |  |  |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2017

**CОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc484137852)

[1. Обоснование аппарата и средств проектирования 5](#_Toc484137853)

[1.1 Язык UML, особенности применения, достоинства и недостатки 5](#_Toc484137854)

[1.2 Описание используемых CASE-средств 6](#_Toc484137855)

[2. Проектирование программного обеспечения информационной системы «Книжный магазин» 7](#_Toc484137856)

[2.1 Диаграмма вариантов использования 7](#_Toc484137857)

[*2.1.1* *Описание вариантов использования* 7](#_Toc484137858)

[*2.1.2* *Расчет количественной оценки информационной наполненности диаграммы вариантов использования* 9](#_Toc484137859)

[2.2 Диаграмма классов 11](#_Toc484137860)

[*2.2.1* *Описание диаграммы классов* 11](#_Toc484137861)

[*2.2.2* *Расчет количественной оценки информационной наполненности диаграммы классов* 13](#_Toc484137862)

[2.3 Диаграммы взаимодействия 14](#_Toc484137863)

[*2.3.1* *Описания диаграмм последовательности* 14](#_Toc484137864)

[*2.3.2* *Расчет количественной оценки информационной наполненности диаграмм последовательности* 18](#_Toc484137865)

[*2.3.3* *Описания диаграмм кооперации* 18](#_Toc484137866)

[*2.3.4* *Расчет количественной оценки информационной наполненности диаграмм кооперации* 21](#_Toc484137867)

[2.4 Диаграммы состояний 21](#_Toc484137868)

[*2.4.1* *Описание диаграмм состояний* 22](#_Toc484137869)

[*2.4.2* *Расчет количественной оценки информационной наполненности диаграмм состояний* 26](#_Toc484137870)

[2.5 Диаграммы видов деятельности 27](#_Toc484137871)

[*2.5.1* *Описания диаграмм видов деятельности* 27](#_Toc484137872)

[*2.5.2* *Расчет количественной оценки информационной наполненности диаграмм видов деятельности* 32](#_Toc484137873)

[2.6 Диаграмма пакетов 32](#_Toc484137874)

[*2.6.1* *Описание диаграммы пакетов* 32](#_Toc484137875)

[*2.6.2* *Расчет количественной оценки информационной наполненности диаграммы пакетов* 33](#_Toc484137876)

[2.7 Диаграмма размещения 33](#_Toc484137877)

[*2.7.1* *Описание диаграммы размещения* 34](#_Toc484137878)

[*2.7.2* *Расчет количественной оценки информационной наполненности диаграммы размещения* 34](#_Toc484137879)

[ЗАКЛЮЧЕНИЕ 35](#_Toc484137880)

[СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ 36](#_Toc484137881)

[ПРИЛОЖЕНИE 37](#_Toc484137882)

# ВВЕДЕНИЕ

В процессе создания крупных информационных систем перед разработчиками может возникнуть множество разнообразных проблем, которые возникают из-за ресурсных и других ограничений. Эти барьеры не всегда можно преодолеть классическими средствами структурного подхода, гораздо удобнее будет использование обектно-ориентированных методов. Такие программые средства моделирования, как Rational Rose позволяют уменьшить временные затраты на разработку информационной системы, в следствие чего сокращаются и другие издержки.

Целью выполнения данной курсовой работы является разработка информационной системы «Книжный магазин», которая отражала бы работу реального предприятия и позволила бы упростить управление им.

## Обоснование аппарата и средств проектирования

### Язык UML, особенности применения, достоинства и недостатки

Унифицированный язык моделирования UML – это визуальный язык для определения, конструирования систем, но не принцип или метод объектно-ориентированного анализа и проектирования.

Язык UML можно использовать для следующих задач:

*Для черновиков* – неполные и неформальные диаграммы, создание для прояснения сложные проектных решений.

*Для создания проектной документации* – относительно детализированные диаграммы проектирования.

*В качестве языка программирования* – полные выполняемые спецификации программных средств на языке UML. Выполняемый код можно автоматически сгенерировать.

К достоинствам языка относят следующие особенности:

1. UML объектно-ориентированный.
2. UML позволяет описать систему практически со всех возможных точек зрения и разные аспекты поведения системы.
3. Диаграммы UML сравнительно просты для чтения после достаточно быстрого ознакомления с его синтаксисом.
4. UML расширяет и позволяет вводить собственные текстовые и графические стереотипы.
5. UML получил широкое распространение и динамично развивается.

В то же время, UML обладает рядом недостатков:

1. Избыточность языка. UML часто критикуется, как неоправданно большой и сложный.
2. Неточная семантика. Неточность описания самого UML одинаково отражается на пользователях и поставщиках инструментов.
3. Пытается быть всем для всех. UML — это язык моделирования общего назначения, который пытается достигнуть совместимости со всеми возможными языками разработки.
4. Сложен во внедрении и обучении. Вышеописанные проблемы делают проблематичным изучение и внедрение UML.

### Описание используемых CASE-средств

В настоящее время на рынке CASE-средств представлено более десятка программных инструментов, поддерживающих нотацию языка UML и обеспечивающих интеграцию, включая прямую и обратную генерацию кода программ, с наиболее распространенными языками и средами программирования, такими как MS Visual C++, C#, Java, Object Pascal/Delphi.

Компания Borland реализовала поддержку UML в проекте Together Designer, который интегрирован с Borland Developer Studio. Другим известным примером CASE-системы, ориентированной на поддержку UML, является Rational Rose от компании Rational Software.

Rational Rose популярное средство визуального моделирования объектно-ориентированных информационных систем. Работа продукта основана на язык UML. Благодаря этому Rational Rose способен решать большой спектр задач в проектировании информационных систем.

В распоряжение проектировщика системы Rational Rose предоставляет следующие типы диаграмм:

1. Use case diagram (диаграммы прецедентов);
2. deployment diagram (диаграммы топологии);
3. statechart diagram (диаграммы состояний);
4. activity diagram (диаграммы активности);
5. interaction diagram (диаграммы взаимодействия);
6. sequence diagram (диаграммы последовательностей действий);
7. collaboration diagram (диаграммы сотрудничества);
8. class diagram (диаграммы классов);
9. component diagram (диаграммы компонент).

## 

## Проектирование программного обеспечения информационной системы «Книжный магазин»

### Диаграмма вариантов использования

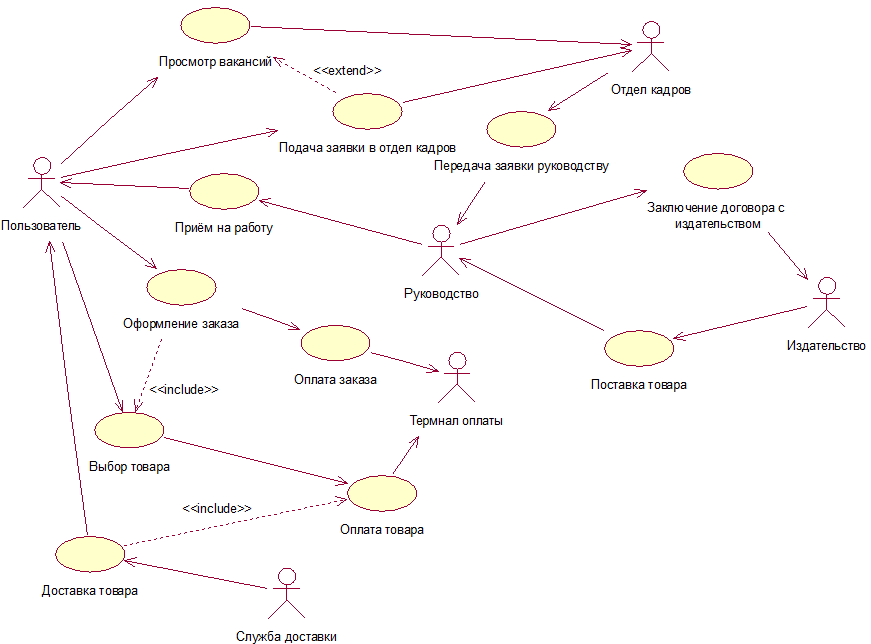
Разрабатываемая информационная система предназначена для осуществления работы основных процессов внутри магазина и его взаимодействия с внешними системами.

Система выполняет такие функции, как:

1. выбор товара и оплата клиентом;
2. оформление заказа и его оплата;
3. просмотр вакансий и приём на работу;
4. заключение договора между магазином и издательством;
5. формирование поставок товаров;
6. обеспечение доставки товаров.

#### *Описание вариантов использования*

В диаграмме вариантов использования (рисунок 1) отражено функционирование информационной системы «Книжный магазин».



1. Диаграмма вариантов использования

Сценарий «Выбор товара» означает, что перед покупкой товара актёр «Пользователь» должен выбрать необходимые товары.

Сценарий «Оплата товара» показывает, что актёр «Пользователь ИС» должен оплатить выбранные товары. За проверку оплаты ответственен актёр «Терминал оплаты».

Сценарий «Оформление заказа» требуется для того, чтобы актёр «Пользователь ИС» мог оформить требуемый заказ перед его оплатой.

Сценарий «Оплата заказа» показывает, чтобы актёр «Пользователь ИС» должен оплатить оформленный заказ. За проверку оплаты ответственен актёр «Терминал оплаты».

Сценарий «Доставка товара» нужен для демонстрации того, что актёр «Служба доставки» осуществляет доставку оплаченных товаров клиенту (актёр «Пользователь ИС»).

Сценарий «Просмотр вакансий» позволяет соискателям просматривать возможные вакансии.

Сценарий «Подача заявки в отдел кадров» описывает возможность подачи соискателем (актёр «Пользователь ИС) заявки на трудоустройство актёру «Отдел кадров».

Сценарий «Передача заявки руководству» означает, что проверенные в отделе кадров заявки перенаправляются к руководству магазина (актёр «Руководство») для принятия решения о трудоустройстве.

Сценарий «Приём на работу» подразумевает одобрение кандидатуры соискателя для работы в магазине.

Сценарий «Заключение договора с издательством» нужен для того, чтобы актёр «Руководство» мог заключить с актёром «Издательство» договор о поставке товара.

Сценарий «Поставка товара» изображает то, что после заключения договора актёр «Издательство» осуществит непосредственно поставку товара.

#### *Расчет количественной оценки информационной наполненности диаграммы вариантов использования*

Для оценки диаграмм UML используется методика, позволяющая рассчитать количественный коэффициент, описывающий такие параметры диаграммы как наглядность, удобство восприятия, степень информативности (или загруженности информацией). Для расчета данного коэффициента используется формула:

*где:*

*Sobj* – оценка элементов

*Slink* – оценка связей

*Sclass* – дополнительное слагаемое, используемое, если для конкурентного класса указаны атрибуты и методы.

*Obj* – количество объектов на диаграмме

*Tobj* – число типов объектов на диаграмме

*Tlink* – число типов *с*вязей на диаграмме

*где*

Op – число методов

Attr – число атрибутов

Оценки классов и элементов берутся из таблиц 1-2.

1. Оценки основных элементов UML

|  |  |
| --- | --- |
| Тип элемента | Оценка элемента |
| Класс class | 5 |
| Интерфейс interface | 4 |
| Сценарий use-case | 2 |
| Компонент component | 4 |
| Узел node | 3 |
| Взаимодействие interaction | 6 |
| Пакет package | 4 |
| Состояние state | 4 |
| Примечание note | 2 |

1. Оценки основных типов связей

|  |  |
| --- | --- |
| Тип связи | Оценка связи |
| Зависимость dependency | 2 |
| Ассоциация association | 1 |
| Агрегирование aggregation | 2 |
| Композиция composition | 3 |
| Обобщение generalization | 3 |
| Реализация realization | 2 |

Остальные типы связей рассматриваются как ассоциации.

Оценка диаграммы должна попадать в оптимальный диапазон, иначе диаграмма или слишком краткая, или перегружена информацией.

Данные по диапазонам для различных диаграмм представлены в таблице 3.

1. Диапазоны оптимальных оценок диаграмм

|  |  |
| --- | --- |
| Class – диаграмма классов с атрибутами и операциями | 5-5,5 |
| Class - диаграмма классов без атрибутов и операций | 3-3,5 |
| Component – диаграмма компонентов | 3,5-4 |
| Use case - диаграмма вариантов использования | 2,5-3 |
| Deployment - диаграмма развёртывания | 2-2,5 |
| Sequences - диаграмма последовательности | 3-3,5 |
| Cooperative - диаграмма кооперации | 3,5-4 |
| Package - диаграмма пакетов | 3,5-4 |
| State – диаграмма состояния | 2,5-3 |

Для данной диаграммы оценка вычислена по формуле 1 и равна:

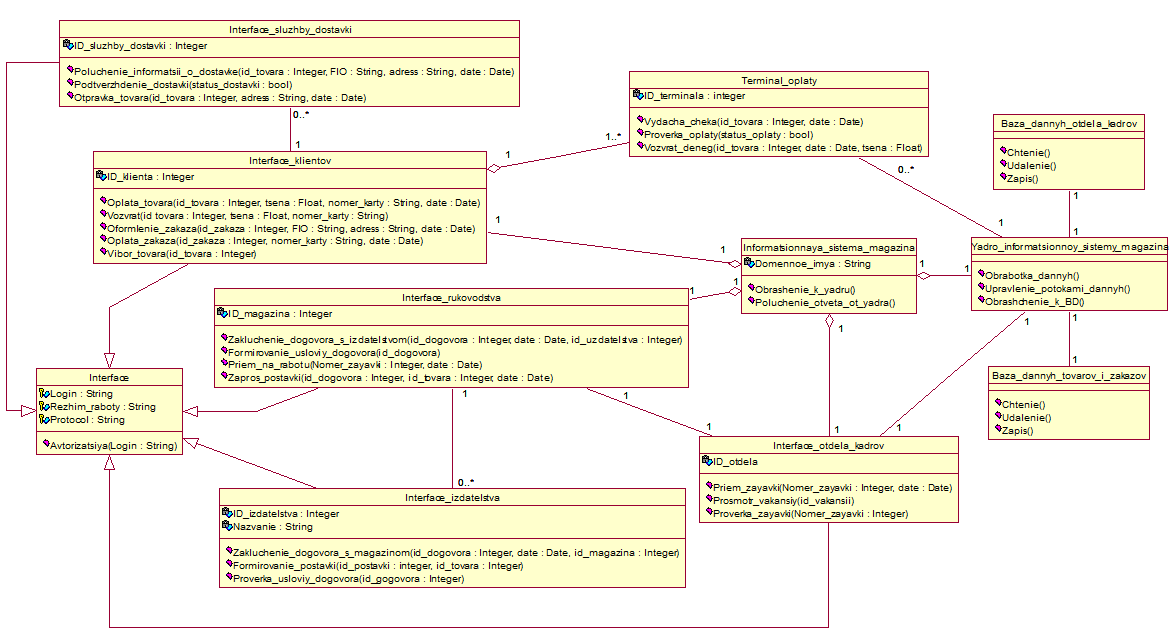
Оценка входит в заданный диапазон для диаграмм данного вида (2,5-3).

### Диаграмма классов

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений.

#### *Описание диаграммы классов*

Диаграмма классов (рисунок 2) служит для наглядного взаимодействия классов системы.



1. Диаграмма классов

#### *Расчет количественной оценки информационной наполненности диаграммы классов*

Расчёт количественной оценки информационной наполненности диаграммы классов оценивается по формулам 1-2.

Оценки классов:

Оценка диаграммы:

Эта оценка незначительно превышает оптимальный диапазон для этого вида диаграмм (5-5,5), но за счёт этого мы получаем более полное представление о взаимодействии классов в системе.

### Диаграммы взаимодействия

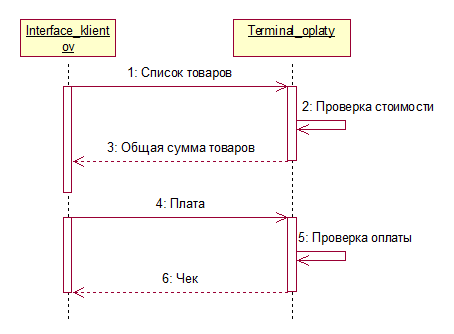
*Диаграмма последовательности* – это диаграмма, отображающая взаимодействия между объектами в рамках текущего сценария, упорядоченные во времени.

Д*иаграмма кооперации* – это диаграмма взаимодействия, которая выделяет структурную организацию объектов, отправляющих и принимающих сообщения. Графически представляет собой набор дуг и вершин.

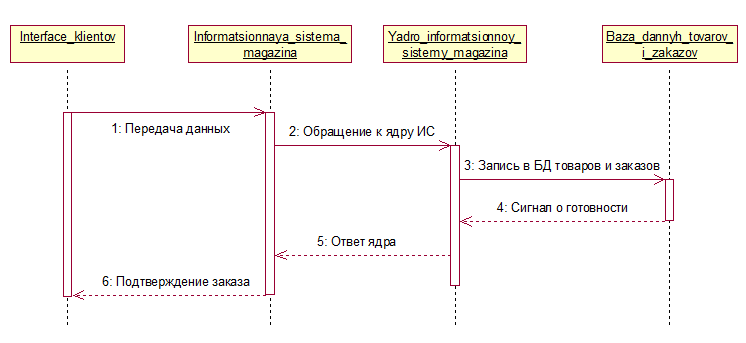
#### *Описания диаграмм последовательности*



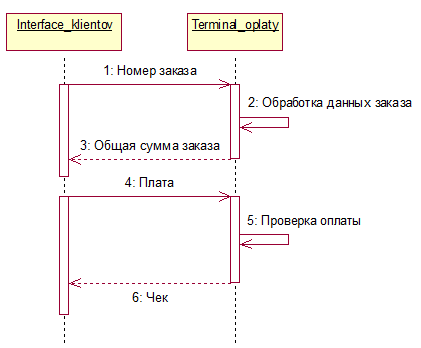
1. Диаграмма последовательности «Выбор товара»



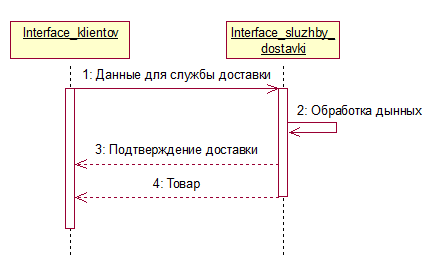
1. Диаграмма последовательности «Оплата товара»



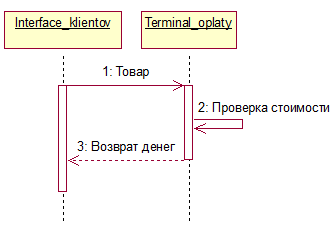
1. Диаграмма последовательности «Оформление заказа»



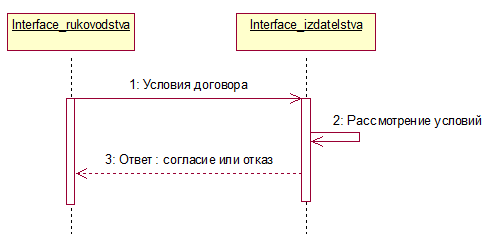
1. Диаграмма последовательности «Оплата заказа»



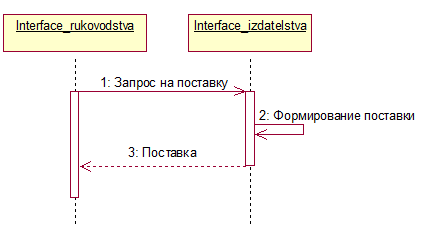
1. Диаграмма последовательности «Доставка товара»



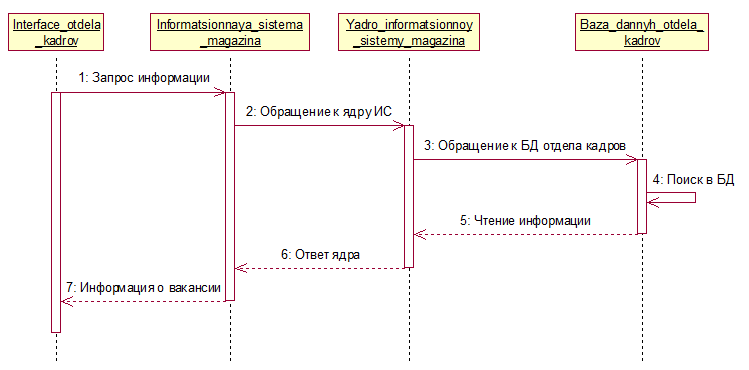
1. Диаграмма последовательности «Возврат товара»



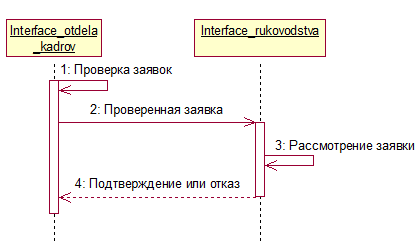
1. Диаграмма последовательности «Заключение договора с издательством»



1. Диаграмма последовательности «Поставка товара»



1. Диаграмма последовательности «Просмотр вакансий»



1. Диаграмма последовательности «Приём на работу»

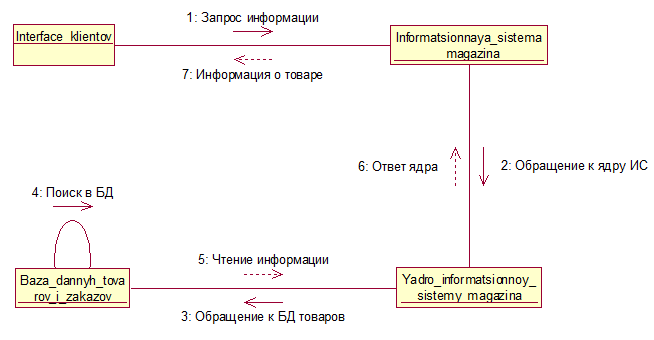
#### *Расчет количественной оценки информационной наполненности диаграмм последовательности*

Расчёт количественной оценки информационной наполненности диаграмм последовательности оценивается по формуле 1 и представлен в таблице 4.

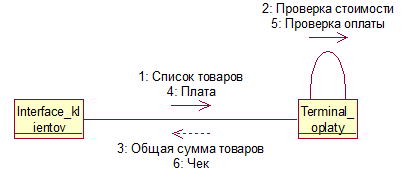
1. Оценка диаграмм последовательности

|  |  |
| --- | --- |
| Диаграмма последовательности | Оценка |
| Выбор товара | 3.85 |
| Оплата товара | 3.38 |
| Оформление заказа | 3.86 |
| Оплата заказа | 3.38 |
| Доставка товара | 2.8 |
| Возврат товара | 2.6 |
| Заключение договора с издательством | 2.6 |
| Поставка товара | 2.6 |
| Просмотр вакансий | 3.85 |
| Приём на работу | 2.8 |

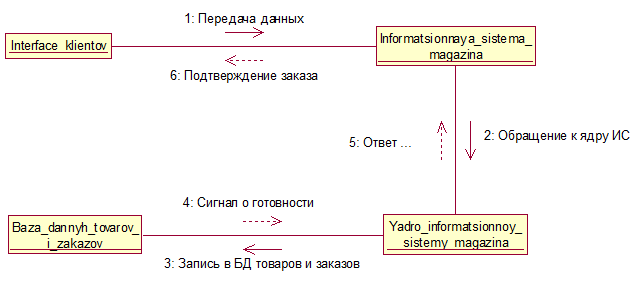
#### *Описания диаграмм кооперации*



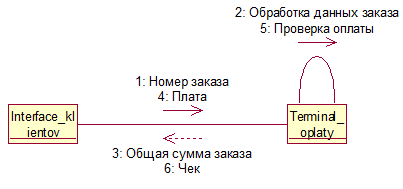
1. Диаграмма кооперации «Выбор товара»



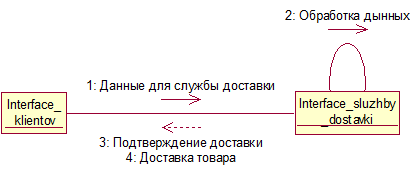
1. Диаграмма кооперации «Оплата товара»



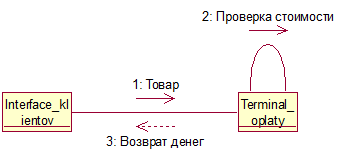
1. Диаграмма кооперации «Оформление заказа»



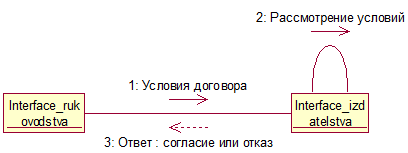
1. Диаграмма кооперации «Оплата заказа»



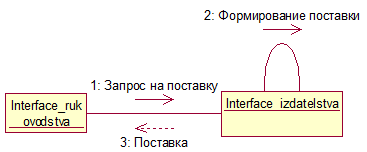
1. Диаграмма кооперации «Доставка товара»



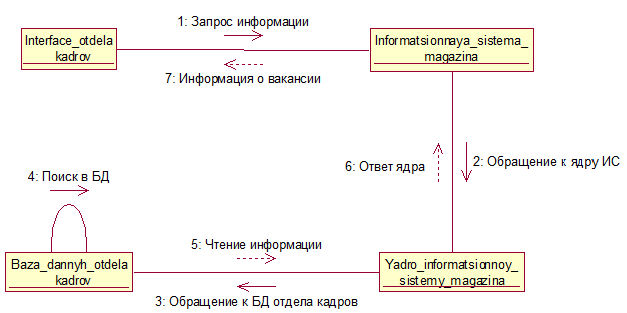
1. Диаграмма кооперации «Возврат товара»



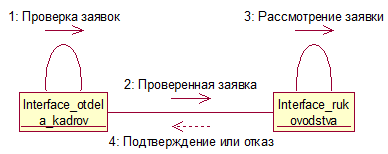
1. Диаграмма кооперации «Заключение договора с издательством»



1. Диаграмма кооперации «Поставка товара»



1. Диаграмма кооперации «Просмотр вакансий»



1. Диаграмма кооперации «Приём на работу»

#### *Расчет количественной оценки информационной наполненности диаграмм кооперации*

Расчёт количественной оценки информационной наполненности диаграмм последовательности оценивается по формуле 1 и представлен в таблице 5.

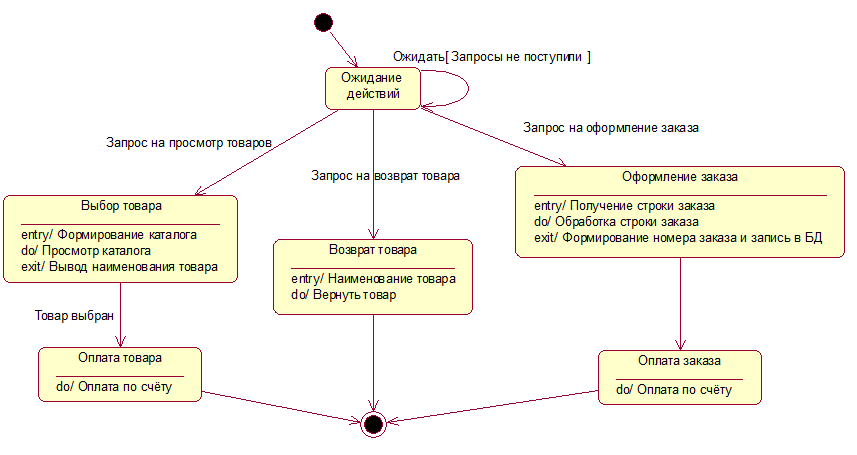
1. Оценка диаграмм кооперации

|  |  |
| --- | --- |
| Диаграмма последовательности | Оценка |
| Выбор товара | 3.85 |
| Оплата товара | 3.38 |
| Оформление заказа | 3.86 |
| Оплата заказа | 3.38 |
| Доставка товара | 2.8 |
| Возврат товара | 2.6 |
| Заключение договора с издательством | 2.6 |
| Поставка товара | 2.6 |
| Просмотр вакансий | 3.85 |
| Приём на работу | 2.8 |

### Диаграммы состояний

*Диаграммы состояний* – это один из пяти видов диаграмм UML, предназначенных для моделирования динамических аспектов поведения систем. Диаграмма состояний показывает конечный автомат. И диаграммы деятельности, и диаграммы состояний подходят для моделирования жизненного цикла объекта. Однако в то время, как диаграмма видов деятельности демонстрирует поток управления от одной деятельности к другой через множество объектов, диаграмма состояний отображает поток управления от состояния к состоянию внутри отдельного объекта.

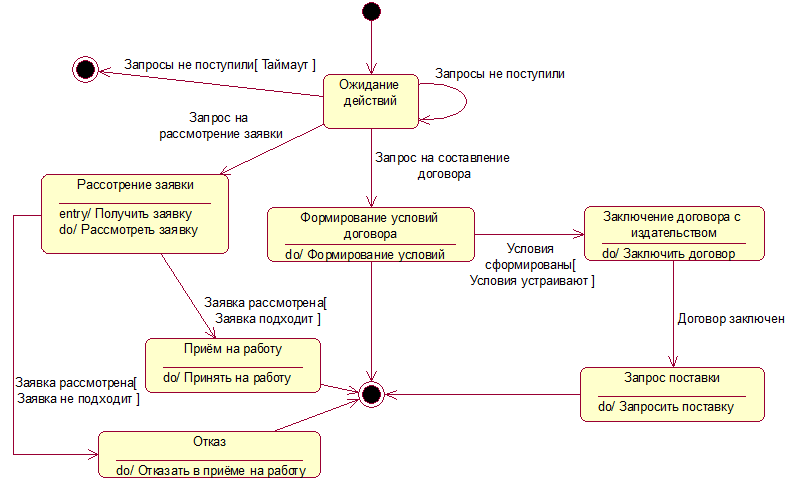
#### *Описание диаграмм состояний*



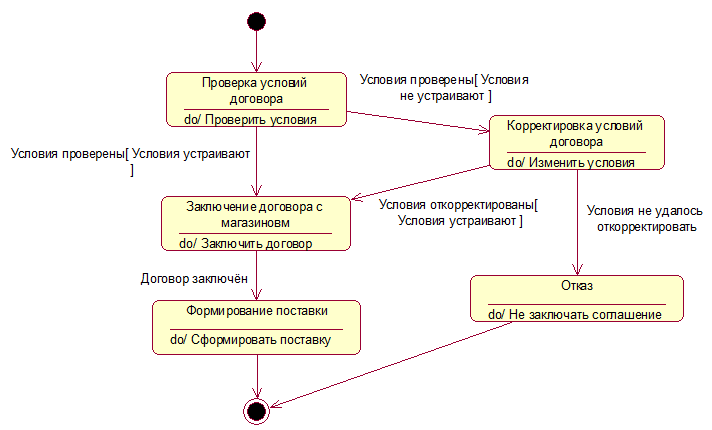
1. Диаграмма состояния «Interface\_klientov»



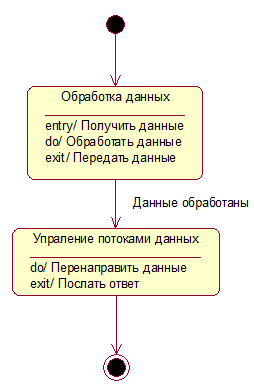
1. Диаграмма состояния «Terminal\_oplaty»



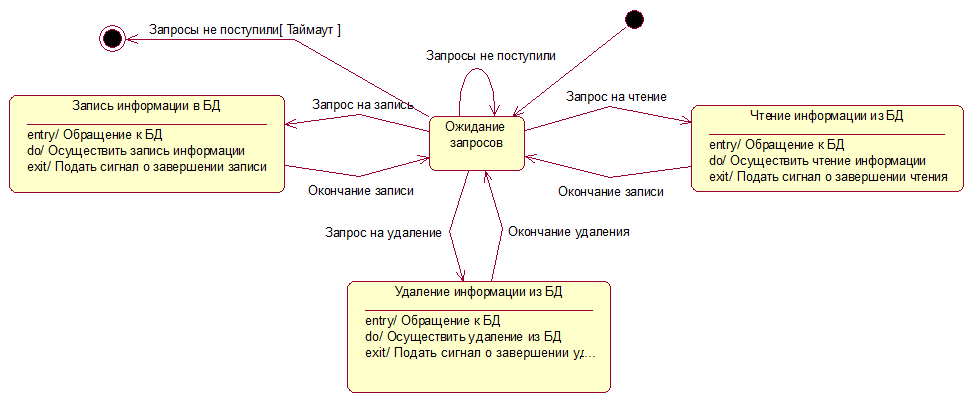
1. Диаграмма состояния «Interface\_rukovodstva»



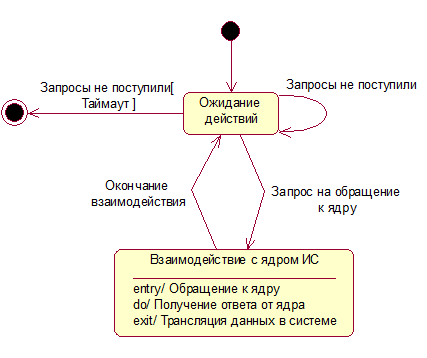
1. Диаграмма состояния «Interface\_izdatelstva»



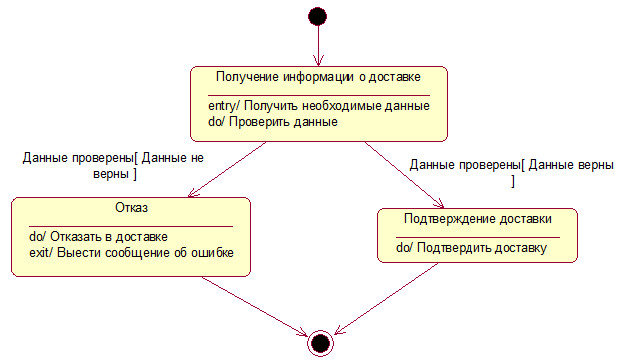
1. Диаграмма состояния «Yadro\_informatsionnoy\_sistemy\_magazina»



1. Диаграмма состояния «Baza\_dannyh\_otdela\_kadrov»



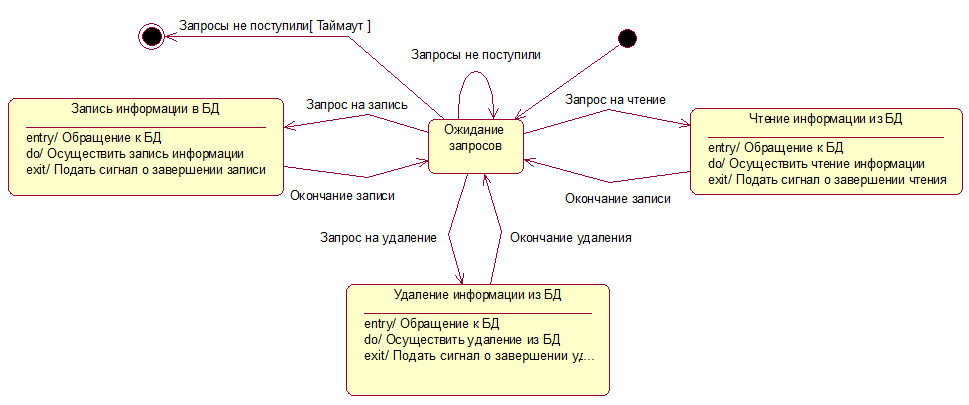
1. Диаграмма состояния «Informatsionnaya\_sistema\_magazina»



1. Диаграмма состояния «Interface\_sluzhby\_dostavki»



1. Диаграмма состояния «Interface\_otdela\_kadrov»



1. Диаграмма состояния «Baza\_dannyh\_tovarov\_i\_zakazov»

#### *Расчет количественной оценки информационной наполненности диаграмм состояний*

Расчёт количественной оценки информационной наполненности диаграмм состояний оценивается по формуле 1 и представлен в таблице 6.

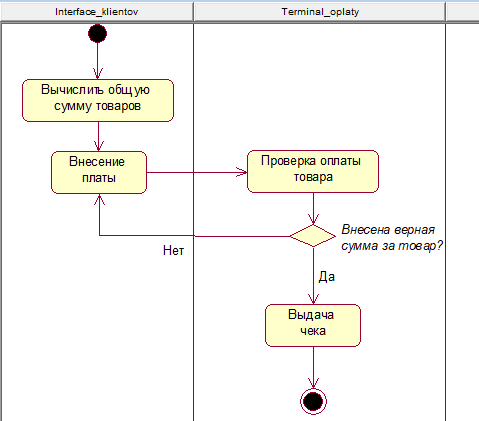
1. Оценка диаграмм состояний

|  |  |
| --- | --- |
| Диаграмма | Оценка |
| Interface\_klientov | 4.15 |
| Terminal\_oplaty | 3.74 |
| Interface\_rukovodstva | 4.35 |
| Interface\_izdatelstva | 3.64 |
| Yadro\_informatsionnoy\_sistemy\_magazina | 2.49 |
| Baza\_dannyh\_otdela\_kadrov | 3.89 |
| Informatsionnaya\_sistema\_magazina | 2.95 |
| Interface\_sluzhby\_dostavki | 3.14 |
| Interface\_otdela\_kadrov | 4.05 |
| Baza\_dannyh\_tovarov\_i\_zakazov | 3.89 |

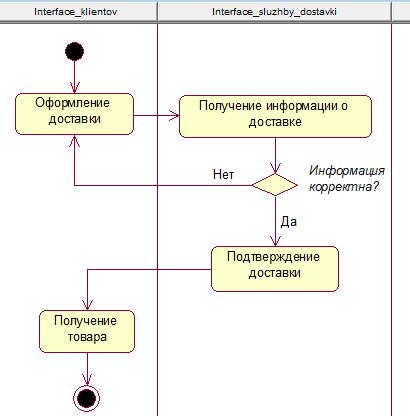
### Диаграммы видов деятельности

*Диаграммы видов деятельности* – это один из пяти видов диаграмм, применяемых в UML для моделирования динамических аспектов систем. По сути, диаграмма деятельности представляет собой блок-схему, которая показывает, как поток управления переходит от одной деятельности к другой. В отличие от традиционной блок-схемы диаграмма деятельности показывает параллелизм так же хорошо, как и ветвление потока управления.

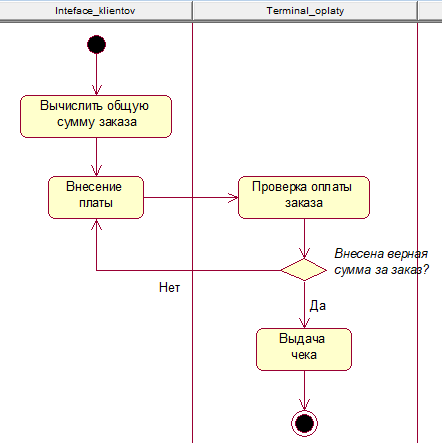
#### *Описания диаграмм видов деятельности*



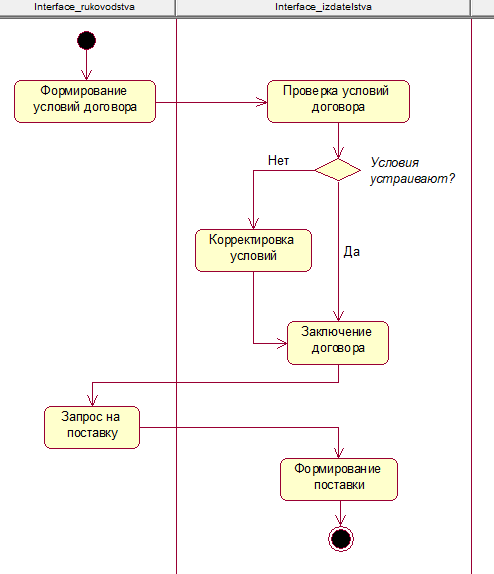
1. Диаграмма видов деятельности «Оплата товара»



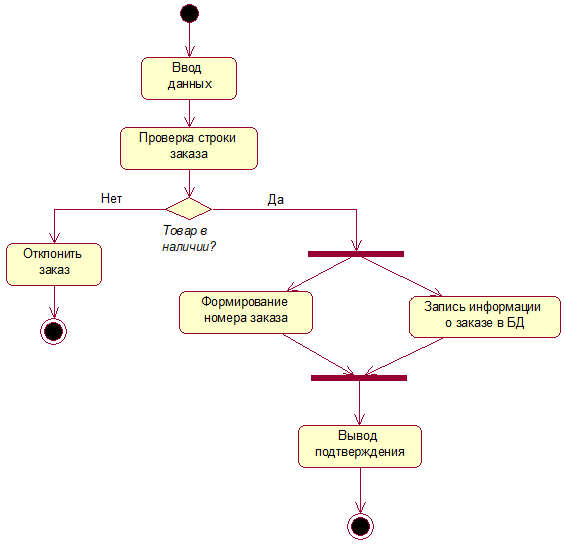
1. Диаграмма видов деятельности «Доставка товара»



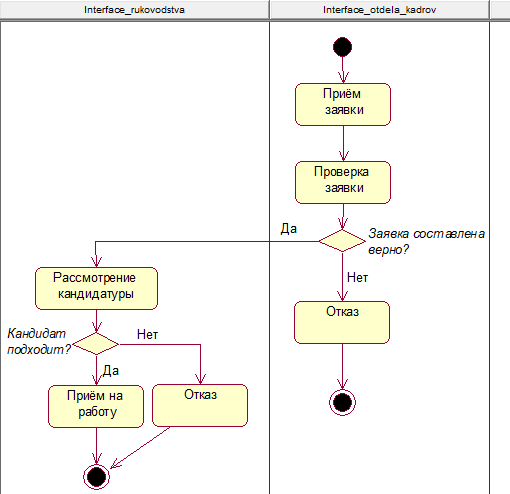
1. Диаграмма видов деятельности «Оплата заказа»



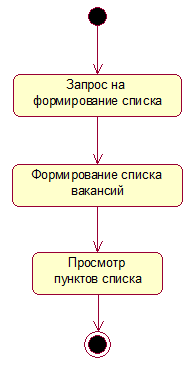
1. Диаграмма видов деятельности «Заключение договора на поставку»



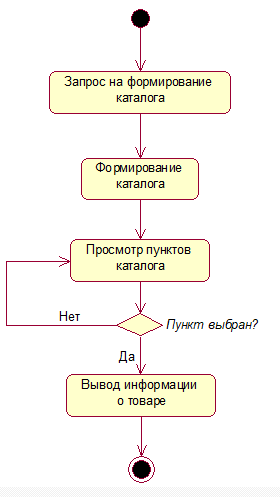
1. Диаграмма видов деятельности «Оформление заказа»



1. Диаграмма видов деятельности «Приём на работу»



1. Диаграмма видов деятельности «Просмотр вакансий»



1. Диаграмма видов деятельности «Выбор товара»

#### *Расчет количественной оценки информационной наполненности диаграмм видов деятельности*

Расчёт количественной оценки информационной наполненности диаграмм видов деятельности оценивается по формуле 1 и представлен в таблице 7.

1. Оценка диаграмм видов деятельности

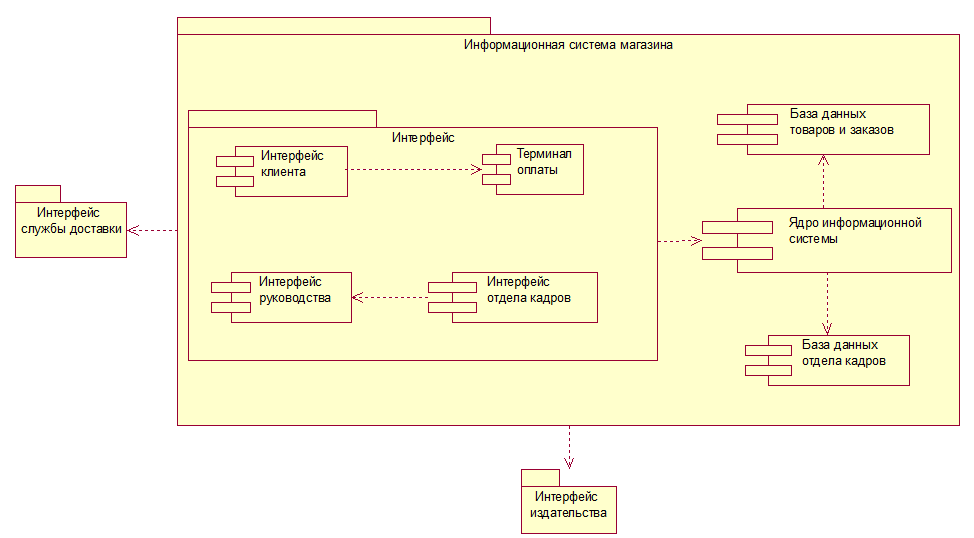
|  |  |
| --- | --- |
| Диаграмма последовательности | Оценка |
| Оплата товара | 2.97 |
| Доставка товара | 2.97 |
| Оплата заказа | 2.97 |
| Заключение договора на поставку | 3.39 |
| Оформление заказа | 3 |
| Приём на работу | 3.26 |
| Просмотр вакансий | 2.96 |
| Выбор товара | 4.01 |

### Диаграмма пакетов

*Диаграмма пакетов* – это диаграмма, представляющая логическую группировку составляющих информационную систему процессов и объектов. Данная диаграмма предназначена для обеспечения удобства разработки, так как позволяет группировать объекты любым способов. При этом один объект или пакет может принадлежать нескольким родительским пакетам.

#### *Описание диаграммы пакетов*

Диаграмма пакетов (рисунок 41) показывает, как хранятся, располагаются и взаимодействую между собой объекты информационной системы.



1. Диаграмма пакетов

#### *Расчет количественной оценки информационной наполненности диаграммы пакетов*

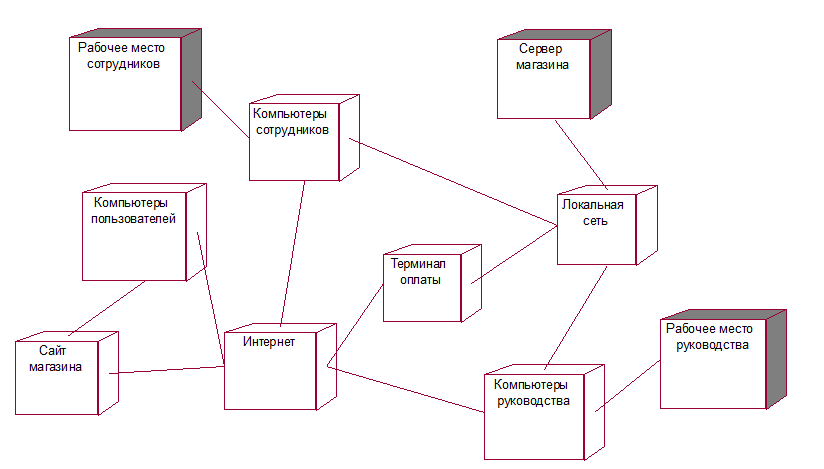
Расчёт количественной оценки информационной наполненности диаграмм пакетов оценивается по формуле 1.

### Диаграмма размещения

*Диаграмма размещения* – это диаграмма, предназначенная для моделирования работающих узлов (аппаратных средств), и компонентов информационной системы, развернутых на них.

#### *Описание диаграммы размещения*

Диаграмма развёртывания представлена на рисунке 42.



1. Диаграмма размещения

#### *Расчет количественной оценки информационной наполненности диаграммы размещения*

Расчёт количественной оценки информационной наполненности диаграмм размещения оценивается по формуле 1.

# ЗАКЛЮЧЕНИЕ

В данной курсовой работе была спроектирована информационная система «Книжный магазин» с помощью языка UML. Информационная система включает в себя: диаграмму вариантов использования (Use Case Diagram), диаграмму классов (Class Diagram), диаграммы последовательности (Sequence Diagram), диаграммы кооперации (Collaboration Diagram), диаграммы состояний (Statechart Diagram), диаграммы видов деятельности (Activity Diagram), диаграмму компонентов (Component Diagram) и диаграмму развертывания (Deployment Diagram).

Спроектированная система является наглядной основой для дальнейших модификаций, результатом которых станет возможность перехода к разработке полноценной информационной системы. На данном же этапе система упрощает работу с имеющимися данными и позволяет эффективно решать необходимые задачи. Rational Rose является универсальным средством моделирования на языке UML, которое позволяет быстро создать необходимую информационную систему и продуктивно ей пользоваться.

В результате генерации кода для каждого класс получены файлы с расширением .cpp и .h. Файлы с расширением .h содержат объявления функций, а файл с расширением .cpp содержат исходные коды классов.

# СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ

1. Максимчук Р.А. Нейбург Э.Дж. UML для простых смертных: Руководство / Максимчук Р.А. Нейбург Э.Дж. – Лори, 2014. – 268 c.
2. Буч Г., Якобсон А., Рамбо Дж. Введение в UML от создателей языка / Буч Г., Якобсон А., Рамбо Дж. – ДМК Пресс, 2015. – 496 с.
3. Лешек А. Мацяшек, Анализ и проектирование информационных систем с помощью UML 2.0 / Лешек А. Мацяшек. – Вильямс, 2016. – 816 c.
4. Мартин Фаулер. UML. Основы. Краткое руководство по стандартному языку объектного моделирования / Мартин Фаулер. – Символ–Плюс, 2016. – 184 c.
5. Леоненков А.В. Нотация и семантика языка UML / Леоненков А.В. – М.: Интуит, 2016. — 205 c.
6. Бабич А.В. Введение в UML / Бабич А.В. – М.: НОУ ИНТУИТ, 2016. – 209 c.
7. Алистер Коберн. Современные методы описания функциональных требований к системам / Алистер Коберн. – Лори, 2017. – 288 с.
8. Эрик Эванс. Предметно–ориентированное проектирование (DDD). Структуризация сложных программных систем / Эрик Эванс. – Вильямс, 2017. – 448 c.

# ПРИЛОЖЕНИE

Приложение А. Листинг сгенерированного кода

Файл «Baza\_dannyh\_magazin.h»

#ifndef BAZA\_DANNYH\_MAGAZIN\_H\_HEADER\_INCLUDED\_A6CF6E20

#define BAZA\_DANNYH\_MAGAZIN\_H\_HEADER\_INCLUDED\_A6CF6E20

//##ModelId=5930A8A1048A

class Baza\_dannyh\_magazin

{

public:

//##ModelId=5930A8A1048B

Chtenie();

//##ModelId=5930A8A1048C

Udalenie();

//##ModelId=5930A8A1048D

Zapis();

};

#endif /\* BAZA\_DANNYH\_MAGAZIN\_H\_HEADER\_INCLUDED\_A6CF6E20 \*/

Файл «Baza\_dannyh\_magazin.cpp»

#include "Baza\_dannyh\_magazin.h"

//##ModelId=5930A8A1048B

Baza\_dannyh\_ magazin::Chtenie()

{

}

//##ModelId=5930A8A1048C

Baza\_dannyh\_ magazin::Udalenie()

{

}

//##ModelId=5930A8A1048D

Baza\_dannyh\_ magazin::Zapis()

{

}

Файл «Interface\_izdatelstva.h»

#ifndef INTERFACE\_IZDATELSTVA\_H\_HEADER\_INCLUDED\_A6CF7C16

#define INTERFACE\_IZDATELSTVA\_H\_HEADER\_INCLUDED\_A6CF7C16

//##ModelId=5930A8A1024C

class Interface\_izdatelstva : public Interface

{

public:

//##ModelId=5930A8A1024D

Zakluchenie\_dogovora\_s\_magazinom(Integer id\_dogovora, Date date, Integer id\_magazina);

//##ModelId=5930A8A10251

Formirovanie\_postavki(integer id\_postavki, Integer id\_tovara);

//##ModelId=5930A8A10254

Proverka\_usloviy\_dogovora(Integer id\_gogovora);

private:

//##ModelId=5930A8A10256

Integer ID\_izdatelstva;

//##ModelId=5930A8A10257

String Nazvanie;

};

#endif /\* INTERFACE\_IZDATELSTVA\_H\_HEADER\_INCLUDED\_A6CF7C16 \*/

Файл «Interface\_izdatelstva.cpp»

#include "Interface\_izdatelstva.h"

//##ModelId=5930A8A1024D

Interface\_izdatelstva::Zakluchenie\_dogovora\_s\_magazinom(Integer id\_dogovora, Date date, Integer id\_magazina)

{

}

//##ModelId=5930A8A10251

Interface\_izdatelstva::Formirovanie\_postavki(integer id\_postavki, Integer id\_tovara)

{

}

//##ModelId=5930A8A10254

Interface\_izdatelstva::Proverka\_usloviy\_dogovora(Integer id\_gogovora)

{

}

Файл «Interface\_klientov.h»

#ifndef INTERFACE\_KLIENTOV\_H\_HEADER\_INCLUDED\_A6CF43D9

#define INTERFACE\_KLIENTOV\_H\_HEADER\_INCLUDED\_A6CF43D9

//##ModelId=5930A8A103CB

class Interface\_klientov

{

public:

//##ModelId=5930A8A103CD

Oplata\_tovara(Integer id\_tovara, Float tsena, String nomer\_karty, Date date);

//##ModelId=5930A8A103D2

Vozvrat(Integer id tovara, Float tsena, String nomer\_karty);

//##ModelId=5930A8A103D6

Oformlenie\_zakaza(Integer id\_zakaza, String FIO, String adress, Date date);

//##ModelId=5930A8A103DB

Oplata\_zakaza(Integer id\_zakaza, String nomer\_karty, Date date);

//##ModelId=5930A8A103DF

Vibor\_tovara(Integer id\_tovara);

private:

//##ModelId=5930A8A103E1

Integer ID\_klienta;

};

#endif /\* INTERFACE\_KLIENTOV\_H\_HEADER\_INCLUDED\_A6CF43D9 \*/

Файл «Interface\_klientov.cpp»

#include "Interface\_klientov.h"

//##ModelId=5930A8A103CD

Interface\_klientov::Oplata\_tovara(Integer id\_tovara, Float tsena, String nomer\_karty, Date date)

{

}

//##ModelId=5930A8A103D2

Interface\_klientov::Vozvrat(Integer id tovara, Float tsena, String nomer\_karty)

{

}

//##ModelId=5930A8A103D6

Interface\_klientov::Oformlenie\_zakaza(Integer id\_zakaza, String FIO, String adress, Date date)

{

}

//##ModelId=5930A8A103DB

Interface\_klientov::Oplata\_zakaza(Integer id\_zakaza, String nomer\_karty, Date date)

{

}

//##ModelId=5930A8A103DF

Interface\_klientov::Vibor\_tovara(Integer id\_tovara)

{

}

Файл «Interface.h»

#ifndef INTERFACE\_H\_HEADER\_INCLUDED\_A6CF7EA9

#define INTERFACE\_H\_HEADER\_INCLUDED\_A6CF7EA9

//##ModelId=5930A8A10244

class Interface

{

public:

//##ModelId=5930A8A10245

Avtorizatsiya(String Login);

protected:

//##ModelId=5930A8A10247

String Login;

//##ModelId=5930A8A10248

String Rezhim\_raboty;

//##ModelId=5930A8A10249

String Protocol;

};

#endif /\* INTERFACE\_H\_HEADER\_INCLUDED\_A6CF7EA9 \*/

Файл «Interface.cpp»

#include "Interface.h"

//##ModelId=5930A8A10245

Interface::Avtorizatsiya(String Login)

{

}

Файл «Interface\_otdela\_kadrov.h»

#ifndef INTERFACE\_ MAGAZIN \_H\_HEADER\_INCLUDED\_A6CF114A

#define INTERFACE\_ MAGAZIN \_H\_HEADER\_INCLUDED\_A6CF114A

#include "Interface.h"

//##ModelId=5930A8A10495

class Interface\_ magazin: public Interface

{

public:

//##ModelId=5930A8A10497

Priem\_zayavki(Integer Nomer\_zayavki, Date date);

//##ModelId=5930A8A1049A

Prosmotr\_vakansiy(void id\_vakansii);

//##ModelId=5930A8A1049C

Proverka\_zayavki(Integer Nomer\_zayavki);

private:

//##ModelId=5930A8A1049E

ID\_otdela;

};

#endif /\* INTERFACE\_OTDELA\_KADROV\_H\_HEADER\_INCLUDED\_A6CF114A \*/

Файл «Interface\_otdela\_kadrov.cpp»

#include "Interface\_otdela\_kadrov.h"

//##ModelId=5930A8A10497

Interface\_otdela\_kadrov::Priem\_zayavki(Integer Nomer\_zayavki, Date date)

{

}

//##ModelId=5930A8A1049A

Interface\_ magazin::Prosmotr\_vakansiy(void id\_vakansii)

{

}

//##ModelId=5930A8A1049C

Interface\_otdela\_kadrov::Proverka\_zayavki(Integer Nomer\_zayavki)

{

}

Файл «Interface\_rukovodstva.h»

#ifndef INTERFACE\_RUKOVODSTVA\_H\_HEADER\_INCLUDED\_A6CF5463

#define INTERFACE\_RUKOVODSTVA\_H\_HEADER\_INCLUDED\_A6CF5463

//##ModelId=5930A8A1029C

class Interface\_rukovodstva

{

public:

//##ModelId=5930A8A1029E

Zakluchenie\_dogovora\_s\_izdatelstvom(Integer id\_dogovora, Date date, Integer id\_uzdatelstva);

//##ModelId=5930A8A102A2

Formirovanie\_usloviy\_dogovora(void id\_dogovora);

//##ModelId=5930A8A102A4

Priem\_na\_rabotu(Integer Nomer\_zayavki, Date date);

//##ModelId=5930A8A102A7

Zapros\_postavki(Integer id\_dogovora, Integer id\_tovara, Date date);

private:

//##ModelId=5930A8A102AB

Integer ID\_magazina;

};

#endif /\* INTERFACE\_RUKOVODSTVA\_H\_HEADER\_INCLUDED\_A6CF5463 \*/

Файл «Interface\_rukovodstva.cpp»

#include "Interface\_rukovodstva.h"

//##ModelId=5930A8A1029E

Interface\_rukovodstva::Zakluchenie\_dogovora\_s\_izdatelstvom(Integer id\_dogovora, Date date, Integer id\_uzdatelstva)

{

}

//##ModelId=5930A8A102A2

Interface\_rukovodstva::Formirovanie\_usloviy\_dogovora(void id\_dogovora)

{

}

//##ModelId=5930A8A102A4

Interface\_rukovodstva::Priem\_na\_rabotu(Integer Nomer\_zayavki, Date date)

{

}

//##ModelId=5930A8A102A7

Interface\_rukovodstva::Zapros\_postavki(Integer id\_dogovora, Integer id\_tovara, Date date)

{

}

Файл «Interface\_sluzhby\_dostavki.h»

#ifndef INTERFACE\_SLUZHBY\_DOSTAVKI\_H\_HEADER\_INCLUDED\_A6CF0B75

#define INTERFACE\_SLUZHBY\_DOSTAVKI\_H\_HEADER\_INCLUDED\_A6CF0B75

//##ModelId=5930A8A10441

class Interface\_sluzhby\_dostavki

{

public:

//##ModelId=5930A8A10443

Poluchenie\_informatsii\_o\_dostavke(Integer id\_tovara, String FIO, String adress, Date date);

//##ModelId=5930A8A10448

Podtverzhdenie\_dostavki(bool status\_dostavki);

//##ModelId=5930A8A1044A

Otpravka\_tovara(Integer id\_tovara, String adress, Date date);

private:

//##ModelId=5930A8A1044E

Integer ID\_sluzhby\_dostavki;

};

#endif /\* INTERFACE\_SLUZHBY\_DOSTAVKI\_H\_HEADER\_INCLUDED\_A6CF0B75 \*/

Файл «Interface\_sluzhby\_dostavki.cpp»

#include "Interface\_sluzhby\_dostavki.h"

//##ModelId=5930A8A10443

Interface\_sluzhby\_dostavki::Poluchenie\_informatsii\_o\_dostavke(Integer id\_tovara, String FIO, String adress, Date date)

{

}

//##ModelId=5930A8A10448

Interface\_sluzhby\_dostavki::Podtverzhdenie\_dostavki(bool status\_dostavki)

{

}

//##ModelId=5930A8A1044A

Interface\_sluzhby\_dostavki::Otpravka\_tovara(Integer id\_tovara, String adress, Date date)

{

}

Файл «Terminal\_oplaty.h»

#ifndef TERMINAL\_OPLATY\_H\_HEADER\_INCLUDED\_A6CF051F

#define TERMINAL\_OPLATY\_H\_HEADER\_INCLUDED\_A6CF051F

//##ModelId=5930A8A1037B

class Terminal\_oplaty

{

public:

//##ModelId=5930A8A1037C

Vydacha\_cheka(Integer id\_tovara, Date date);

//##ModelId=5930A8A1037F

Proverka\_oplaty(bool status\_oplaty);

//##ModelId=5930A8A10381

Vozvrat\_deneg(Integer id\_tovara, Date date, Float tsena);

private:

//##ModelId=5930A8A10385

integer ID\_terminala;

};

#endif /\* TERMINAL\_OPLATY\_H\_HEADER\_INCLUDED\_A6CF051F \*/

Файл «Terminal\_oplaty.cpp»

#include "Terminal\_oplaty.h"

//##ModelId=5930A8A1037C

Terminal\_oplaty::Vydacha\_cheka(Integer id\_tovara, Date date)

{

}

//##ModelId=5930A8A1037F

Terminal\_oplaty::Proverka\_oplaty(bool status\_oplaty)

{

}

//##ModelId=5930A8A10381

Terminal\_oplaty::Vozvrat\_deneg(Integer id\_tovara, Date date, Float tsena)

{

}

Файл «Yadro\_informatsionnoy\_sistemy\_magazina.h»

#ifndef YADRO\_INFORMATSIONNOY\_SISTEMY\_MAGAZINA\_H\_HEADER\_INCLUDED\_A6CF7703

#define YADRO\_INFORMATSIONNOY\_SISTEMY\_MAGAZINA\_H\_HEADER\_INCLUDED\_A6CF7703

//##ModelId=5930A8A10348

class Yadro\_informatsionnoy\_sistemy\_magazina

{

public:

//##ModelId=5930A8A10349

Obrabotka\_dannyh();

//##ModelId=5930A8A1034A

Upravlenie\_potokami\_dannyh();

//##ModelId=5930A8A1034B

Obrashchenie\_k\_BD();

};

#endif /\* YADRO\_INFORMATSIONNOY\_SISTEMY\_MAGAZINA\_H\_HEADER\_INCLUDED\_A6CF7703 \*/

Файл «Yadro\_informatsionnoy\_sistemy\_magazina.cpp»

#include "Yadro\_informatsionnoy\_sistemy\_magazina.h"

//##ModelId=5930A8A10349

Yadro\_informatsionnoy\_sistemy\_magazina::Obrabotka\_dannyh()

{

}

//##ModelId=5930A8A1034A

Yadro\_informatsionnoy\_sistemy\_magazina::Upravlenie\_potokami\_dannyh()

{

}

//##ModelId=5930A8A1034B

Yadro\_informatsionnoy\_sistemy\_magazina::Obrashchenie\_k\_BD()

{

}

Файл «Baza\_dannyh\_tovarov\_i\_zakazov.h»

#ifndef БАЗА\_ДАННЫХ\_ТОВАРОВ\_H\_HEADER\_INCLUDED\_A6CF2DEE

#define БАЗА\_ДАННЫХ\_ТОВАРОВ\_H\_HEADER\_INCLUDED\_A6CF2DEE

//##ModelId=5930A8A10370

class Baza\_dannyh\_tovarov\_i\_zakazov

{

public:

//##ModelId=5930A8A10371

Chtenie();

//##ModelId=5930A8A10372

Udalenie();

//##ModelId=5930A8A10373

Zapis();

};

#endif /\* БАЗА\_ДАННЫХ\_ТОВАРОВ\_H\_HEADER\_INCLUDED\_A6CF2DEE \*/

Файл «Baza\_dannyh\_tovarov\_i\_zakazov.cpp»

#include "Baza\_dannyh\_tovarov\_i\_zakazov.h"

//##ModelId=5930A8A10371

Baza\_dannyh\_tovarov\_i\_zakazov::Chtenie()

{

}

//##ModelId=5930A8A10372

Baza\_dannyh\_tovarov\_i\_zakazov::Udalenie()

{

}

//##ModelId=5930A8A10373

Baza\_dannyh\_tovarov\_i\_zakazov::Zapis()

{

}

Файл «Informatsionnaya\_sistema\_magazina.h»

#ifndef САЙТ\_H\_HEADER\_INCLUDED\_A6CF056A

#define САЙТ\_H\_HEADER\_INCLUDED\_A6CF056A

//##ModelId=5930A8A10318

class Informatsionnaya\_sistema\_magazina

{

public:

//##ModelId=5930A8A10319

Obrashenie\_k\_yadru();

//##ModelId=5930A8A1031A

Poluchenie\_otveta\_ot\_yadra();

private:

//##ModelId=5930A8A1031B

String Domennoe\_imya;

};

#endif /\* САЙТ\_H\_HEADER\_INCLUDED\_A6CF056A \*/

Файл «Informatsionnaya\_sistema\_magazina.cpp»

#include "Informatsionnaya\_sistema\_magazina.h"

//##ModelId=5930A8A10319

Informatsionnaya\_sistema\_magazina::Obrashenie\_k\_yadru()

{

}

//##ModelId=5930A8A1031A

Informatsionnaya\_sistema\_magazina::Poluchenie\_otveta\_ot\_yadra()

{

}