

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Специальность: «Программное обеспечение информационных технологий»

**КОНТРОЛЬНАЯ РАБОТА**

По курсу: «Технология разработки программного обеспечения»  
На тему: «Информационно-справочная система  
социально-педагогического центра»

Студент-заочник 2 курса  
Группы № 581072  
ФИО: Богданова (Попенко)  
Кристина Евгеньевна  
Адрес: Польша, г. Реда,  
ул. Млынська 5Б-5  
Тел. +48 576 335 295

Минск, 2017

# СОДЕРЖАНИЕ

Введение.....	3
1 Анализ предметной области .....	5
1.1 Постановка задачи .....	6
1.2 Анализ требований к автоматизированной системе .....	6
2 Системное проектирование программного средства. ....	8
3 Логическое моделирование.....	11
3.1 Диаграмма вариантов использования .....	11
3.2 Диаграмма классов.....	12
3.3 Диаграмма деятельности .....	13
3.4 Диаграмма коопераций.....	15
3.5 Диаграмма последовательности .....	16
3.6 Диаграмма развертывания .....	17
3.7 Диаграмма компонентов .....	18
3.7 Генерация кода .....	19
Заключение .....	25
Список использованных источников .....	26

## ВВЕДЕНИЕ

Проблемы социально-педагогического характера (социализация, социальное воспитание) находились в поле зрения ученых Беларуси задолго до официального признания социальной педагогики в качестве самостоятельной отрасли научного знания. Практическая социально-педагогическая деятельность в Беларуси в конце XIX века развивалась в форме частной благотворительности и общественного призрения.

В истории складывания системы благотворительной помощи обездоленным детям на Беларуси весьма значительна роль многочисленных конфессиональных организаций, в первую очередь православных братств с их школами и приютами, а также представителей магнатских белорусских родов с их богатыми пожертвованиями на организацию и содержание приютов для детей-сирот и детей из бедных семей.

В 1917 году, в силу коренных политических и социально-экономических изменений, началось создание новой модели защиты обездоленных, которая включала помощь и поддержку нуждающихся государственными структурами и общественными организациями. Реформирование образования в послеоктябрьский период детерминировало становление социальной педагогики как профессионального вида деятельности. Функционировала сеть учреждений, ориентированных на социально-педагогическую деятельность, которая была направлена на социализацию и оказание помощи, защиты, поддержки нуждающимся детям. Выполняли эту деятельность специалисты, ставшие прообразом современного социального педагога. Должность «социальный педагог» была введена в 2000 году. Это нововведение было продиктовано такими фактами, как увеличение количества детей с отклоняющимся поведением, рост детской преступности.

На сегодняшний день в стране создан механизм, позволяющий в должной мере обеспечивать охрану детства, защиту детей от ненадлежащих условий жизни и воспитания.

Проводится активная работа по расширению семейных форм устройства детей на воспитание, в результате которой количество детей, находящихся в интернатных учреждениях, сократилось на 60%. На 30% (с 74 в 2005 г. до 52 в 2010 г.) сократилось количество интернатных учреждений.

Социальную помощь детям и семьям, реабилитацию несовершеннолетних, находящихся в социально опасном положении и признанных нуждающимися в государственной защите, оказывают 142 социально-педагогических учреждения (социально-педагогические центры, детские социальные приюты).

Социально-педагогический центр - это социально-педагогическое учреждение, которое реализует программу воспитания и защиты прав и законных интересов детей, находящихся в социально опасном положении,

образовательную программу дошкольного образования, образовательную программу дополнительного образования детей и молодежи, координирует деятельность иных учреждений образования в пределах административно-территориальной единицы по выявлению и учету детей, находящихся в социально опасном положении, их социально-педагогической реабилитации и оказанию им психологической помощи, осуществляет разработку и координацию выполнения индивидуальных планов защиты прав и законных интересов детей, находящихся в социально опасном положении, оказывает психологическую и социально-педагогическую помощь законным представителям обучающихся и предназначено для временного проживания (не более шести месяцев) и социальной реабилитации детей в возрасте от трех до восемнадцати лет, находящихся в социально опасном положении, детей-сирот и детей, оставшихся без попечения родителей, до их возвращения в семью или определения их дальнейшего жизнеустройства.

Темой данной контрольной работы является разработка базы данных для информационно-справочной системы социально-педагогического центра. Эта тема является весьма актуальной. Достижение целей работы социально-педагогического центра с приютом требует большого количества времени и человеческих ресурсов, причем, зачастую, не только работников центра, но и внешних. Разрабатываемая информационно-справочная система социально-педагогического центра должна значительно упростит его деятельность, предоставив полную информацию о социально-педагогическом центре в свободном доступе в сети Интернет.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Социально-педагогический центр является учреждением образования, которое осуществляет социально-педагогическую, психолого-педагогическую, социально-психологическую, социально-правовую, коррективно-реабилитационную помощь детям и родителям (лицам, которые их замещают), имеющим необходимость в такой помощи.

Детский социальный приют — это воспитательно-образовательное учреждение системы образования для оказания социально-педагогической помощи несовершеннолетним, оказавшимся в экстремальной ситуации;

Социально-педагогический центр с приютом — это социально-педагогическое учреждение, которое реализует программу воспитания и защиты прав и законных интересов детей, находящихся в социально опасном положении, образовательную программу дошкольного образования, образовательную программу дополнительного образования детей и молодежи, обеспечивает необходимые условия для развития личности несовершеннолетних, укрепления физического и психического здоровья, профессионального самоопределения и творческого труда, социальной адаптации несовершеннолетних к жизни в обществе, проводит профилактическую работу по предупреждению социального сиротства, семейного неблагополучия, асоциального поведения, безнадзорности и правонарушений несовершеннолетних, осуществляет разработку и координацию выполнения индивидуальных планов защиты прав и законных интересов детей, находящихся в социально опасном положении, оказывает психологическую и социально-педагогическую помощь законным представителям обучающихся и осуществляет психолого-педагогическое сопровождение семей, принявших на воспитание детей, оставшихся без попечения родителей.

Вопросы реабилитации в приюте призваны решать медицинская, социально-психологическая, социально-педагогическая службы приюта. Это стало возможным благодаря разработанной модели реабилитации, построенной с учетом основной цели учреждения, контингента детей, степени тяжести их социальной дезадаптации. Основной целью работы приюта является оптимальное жизнеустройство ребенка, т.е. предоставление ему не только возможности проживания, но и снятие остроты психического напряжения, защита его прав, законных интересов, помощь в социальной адаптации, представление возможности проживания в семье. Главное в реабилитационной работе — вернуть детям детство и подготовить их к взрослой жизни, то есть привить им ответственность за их жизнь, за их поступки. Достигается это созданием доброжелательного единого реабилитационного пространства, созданием ситуации успеха, возможностью двигаться в развитии в своем темпе, с учетом персональной зоны ближайшего развития, восстановлением ведущих видов деятельности

(общения, игры, познания, труда), восстановлением семейных связей.

## **1.1 ПОСТАНОВКА ЗАДАЧИ**

Тема данной контрольной работы – «Информационно-справочная система социально-педагогического центра». Поэтому ее основной целью являлась разработка базы данных для информационно-справочной системы, которая будет в последствии представлена в виде веб-сайта и позволит быстро и удобно предоставить полную информацию о социально-педагогическом центре, значительно упростив его деятельность.

Данная цель подразумевает следующие задачи проектирования:

- разработать базу данных для информационно-справочной системы социально-педагогического центра с приютом;
- изучить особенности разрабатываемой базы данных для сайта, ознакомиться с требованиями, предъявляемыми к ним;
- систематизировать, закрепить и углубить теоретические знания и практические умения и навыки, полученные при изучении курса «Технология разработки программного обеспечения»;
- применить полученные знания для решения поставленных цели и задач;
- усовершенствовать навыки ведения самостоятельной проектной работы и реализации проекта, приемы разработки базы данных на всех ее этапах;
- развить алгоритмическое и логическое мышления, а также поспособствовать самостоятельному инженерному творчеству.

## **1.2 АНАЛИЗ ТРЕБОВАНИЙ К АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ**

В результате изучения особенностей разрабатываемой базы данных для сайта и ознакомления с требованиями, предъявляемыми к ним, было составлено следующее функциональное назначение.

Информационно-справочная система социально-педагогического центра разрабатывается в виде сайта, который должен предоставлять в сети Интернет информацию социально-педагогическом центр с приютом, проводимых административных процедурах, организации работы с приемными семьями и семьями, в которых ребенок признан находящимся в социально опасном положении, реализуемых проектах, службах социально-реабилитационного отдела СПЦ, организации учебно-воспитательного

процесса, графиках приема граждан и посетителей, администрации, специалистах центра и детях, месте размещения, необходимой помощи воспитанникам приюта. Сайт должен содержать всю необходимую нормативно-правовую базу. Кроме того, в информационно-справочной системе должны присутствовать фото- и видеогалереи.

Информационно-справочная система также должна осуществлять отображение глобального RSS, т.е. ленты новостей с различных внешних ресурсов (например, сайта Министерства образования).

Посредством форм и ссылок сайт должен обеспечивать возможность обратной связи с СПЦ.

Все опубликованные разделы сайта должны открываться для просмотра без регистрации и аутентификации пользователя. Из любого раздела пользователь может перейти на главную страницу, любой пункт меню первого уровня и любой подпункт меню текущего раздела.

## 2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

Описание системы с помощью IDEF0 называется функциональной моделью. Функциональная модель предназначена для описания существующих бизнес-процессов, в котором используются как естественный, так и графический языки. Для передачи информации о конкретной системе источником графического языка является сама методология IDEF0.

Методология IDEF0 предписывает построение иерархической системы диаграмм - единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция - система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности.

Каждая IDEF0-диаграмма содержит блоки и дуги. Блоки изображают функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.

Функциональные блоки (работы) на диаграммах изображаются прямоугольниками, означающими поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Имя работы должно быть выражено отглагольным существительным, обозначающим действие. IDEF0 требует, чтобы в диаграмме было не менее трех и не более шести блоков. Эти ограничения поддерживают сложность диаграмм и модели на уровне, доступном для чтения, понимания и использования.

Каждая сторона блока имеет особое, вполне определенное назначение. Левая сторона блока предназначена для входов, верхняя - для управления, правая - для выходов, нижняя - для механизмов. Такое обозначение отражает определенные системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, что и как выполняет функция.

Блоки в IDEF0 размещаются по степени важности, как ее понимает автор диаграммы. Этот относительный порядок называется доминированием. Доминирование понимается как влияние, которое один блок оказывает на другие блоки диаграммы. Наиболее доминирующий блок обычно размещается в верхнем левом углу диаграммы, а наименее доминирующий - в правом углу.

Расположение блоков на странице отражает авторское определение доминирования. Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные. Чтобы подчеркнуть это, аналитик может перенумеровать блоки в соответствии с порядком их



доминирования. Порядок доминирования может обозначаться цифрой, размещенной в правом нижнем углу каждого прямоугольника: 1 будет указывать на наибольшее доминирование, 2 - на следующее и т. д.

Взаимодействие работ с внешним миром и между собой описывается в виде стрелок, изображаемых одинарными линиями со стрелками на концах. Стрелки представляют собой некую информацию и именуются существительными. Различают пять типов стрелок: вход, управление, выход, механизм, вызов.

В методологии IDEF0 требуется только пять типов взаимодействий между блоками для описания их отношений: управление, вход, обратная связь по управлению, обратная связь по входу, выход-механизм. Связи по управлению и входу являются простейшими, поскольку они отражают прямые воздействия, которые интуитивно понятны и очень просты.

IDEF1 — одна из методологий семейства IDEF. Применяется для построения информационной модели, которая представляет структуру информации, необходимой для поддержки функций производственной системы или среды. В настоящее время на основе совершенствования методологии IDEF1 создана её новая версия — методология IDEF1X. IDEF1X разработана с учетом таких требований, как простота изучения и возможность автоматизации. IDEF1X-диаграммы используются рядом распространённых CASE-средств (в частности, ERwin, Design/IDEF).

Хотя терминология IDEF1X практически совпадает с терминологией IDEF1, существует ряд фундаментальных отличий в теоретических концепциях этих методологий. Сущность в IDEF1X описывает собой совокупность или набор экземпляров похожих по свойствам, но однозначно отличающихся друг от друга по одному или нескольким признакам. Каждый экземпляр является реализацией сущности. Таким образом, сущность в IDEF1X описывает конкретный набор экземпляров реального мира, в отличие от сущности в IDEF1, которая представляет собой абстрактный набор информационных отображений реального мира. Связи в IDEF1X представляют собой ссылки, соединения и ассоциации между сущностями. Связи это суть глаголы, которые показывают, как соотносятся сущности между собой.

Декомпозиция контекстной диаграммы функциональной модели показана на рис. 2.1. Контекстная диаграмма модели показана на рисунке 2.2. Для построения модели использовался продукт BPwin 4.0 фирмы Computer Associates.

Субъектом моделирования является программное средство «Информационно-справочная система социально-педагогического центра», цель — описать функциональность работы программного средства, точка зрения — пользователь, администратор.

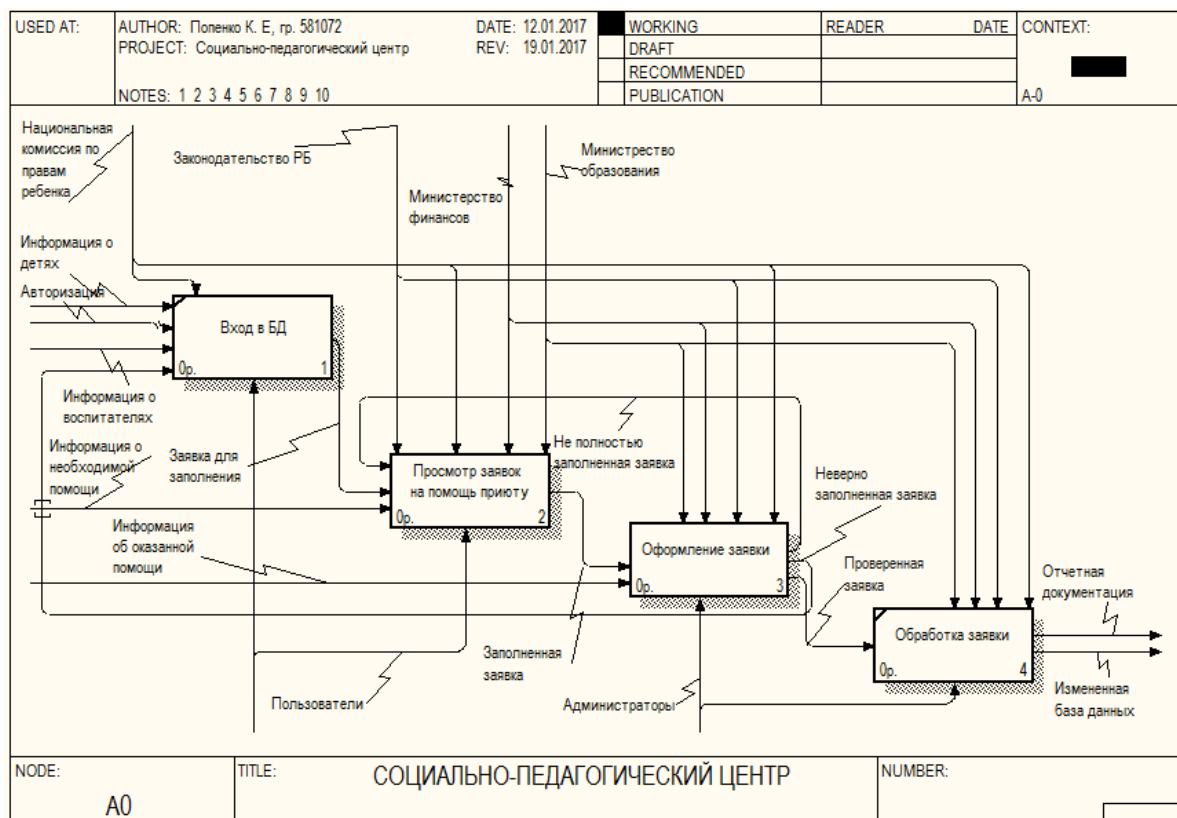


Рисунок 2.1– Декомпозиция контекстной диаграммы функциональной модели

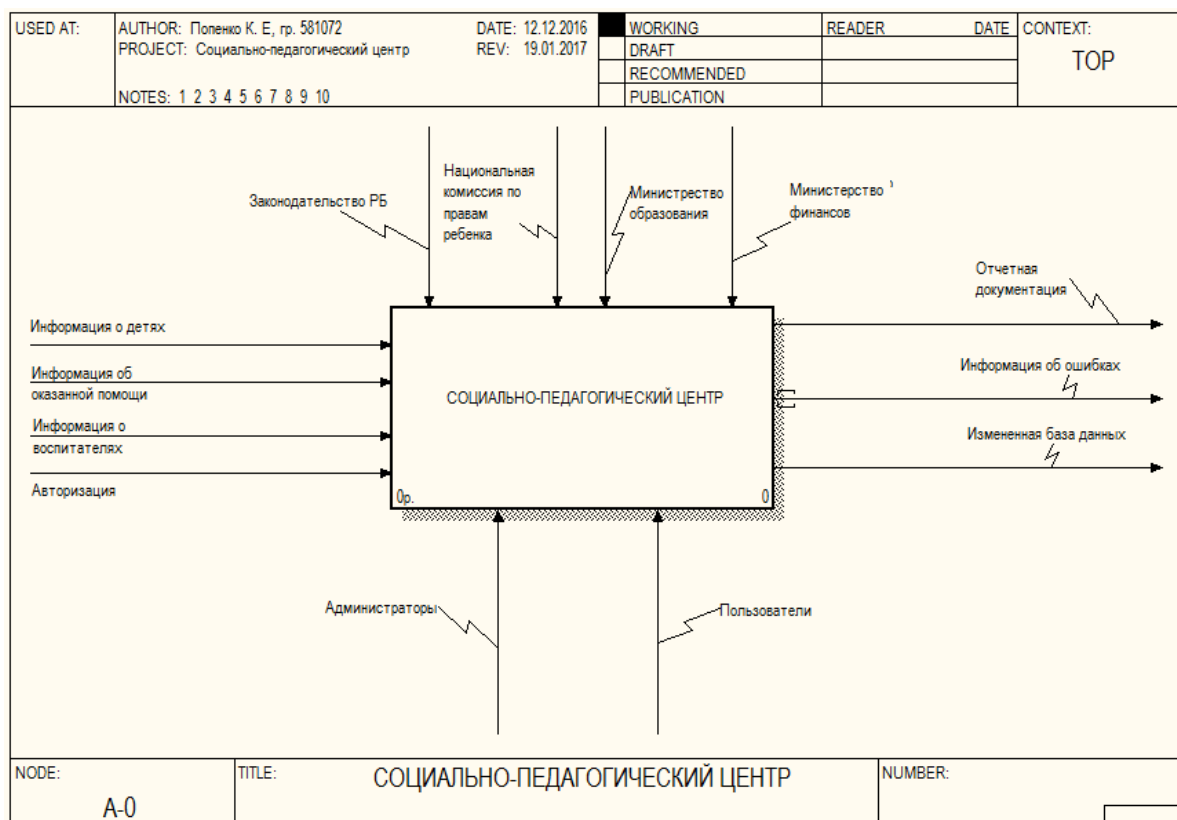


Рисунок 2.2 – Контекстная диаграмма модели

## **3 ЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ**

Основное назначение логического представления состоит в анализе структурных и функциональных отношений между элементами модели системы. Различные элементы логического представления, такие как классы, ассоциации, состояния, сообщения, не существуют материально или физически. Они лишь отражают наше понимание структуры физической системы или аспекты ее поведения.

### **3.1 ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ**

Модель вариантов использования предназначена для определения требований к системе. Она включает в себя актеров, варианты использования и связи между ними. Для отображения этой модели язык UML предлагает использовать диаграммы Use Case (вариант использования) совместно с моделями State Diagram (диаграммы состояний) и Activity Diagram (диаграммы деятельности/активности). Последние используются для конкретизации вариантов использования системы

Разработка диаграммы вариантов использования преследует цели:

- определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
- сформулировать общие требования к функциональному поведению проектируемой системы;
- разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Диаграмма вариантов использования представлена на рисунке 3.1.1.

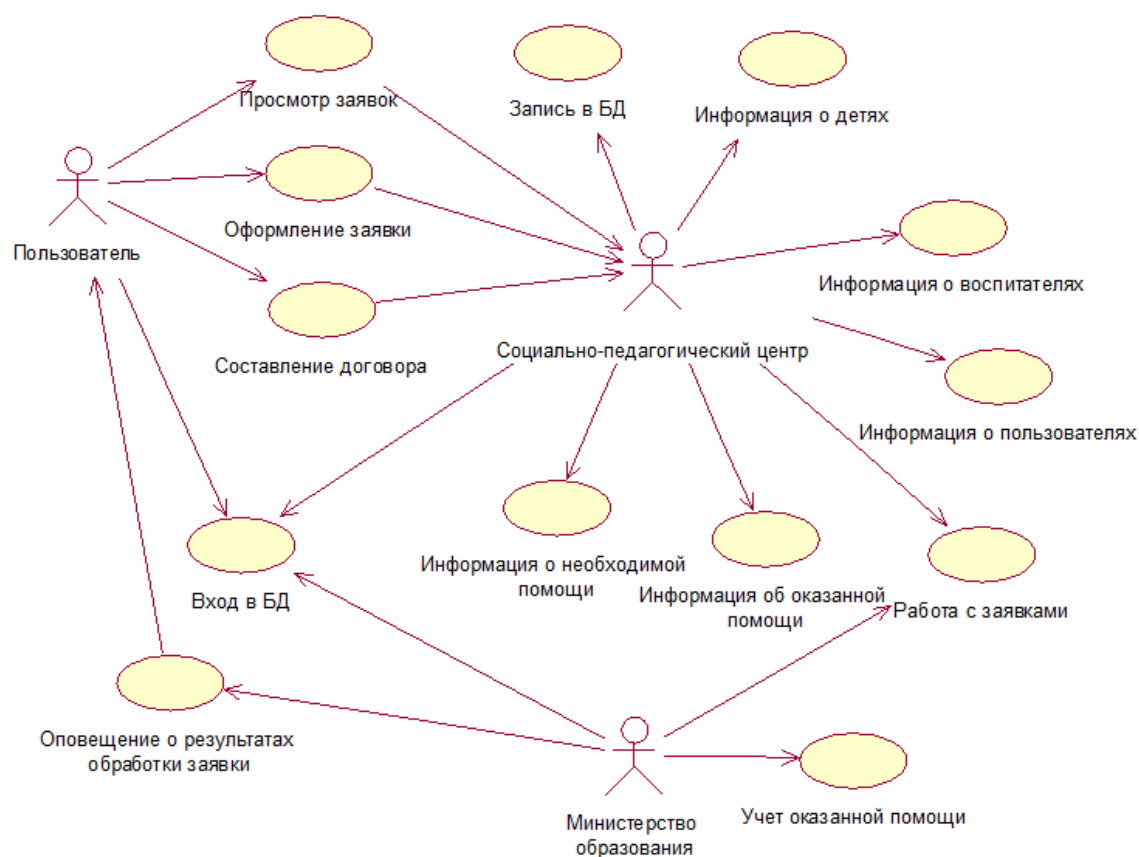


Рисунок 3.1.1 – Диаграмма вариантов использования

## 3.2 ДИАГРАММА КЛАССОВ

Диаграмма классов — диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними. Входит в UML.

Существует два вида:

- статический вид диаграммы рассматривает логические взаимосвязи классов между собой;
- аналитический вид диаграммы рассматривает общий вид и взаимосвязи классов, входящих в систему.

Существуют разные точки зрения на построение диаграмм классов в зависимости от целей их применения:

- концептуальная точка зрения — диаграмма классов описывает модель предметной области, в ней присутствуют только классы прикладных объектов;
- точка зрения спецификации — диаграмма классов применяется при проектировании информационных систем;
- точка зрения реализации — диаграмма классов содержит классы, используемые непосредственно в программном коде (при использовании объектно-ориентированных языков программирования).

Диаграмма классов представлена на рисунке 3.2.1.

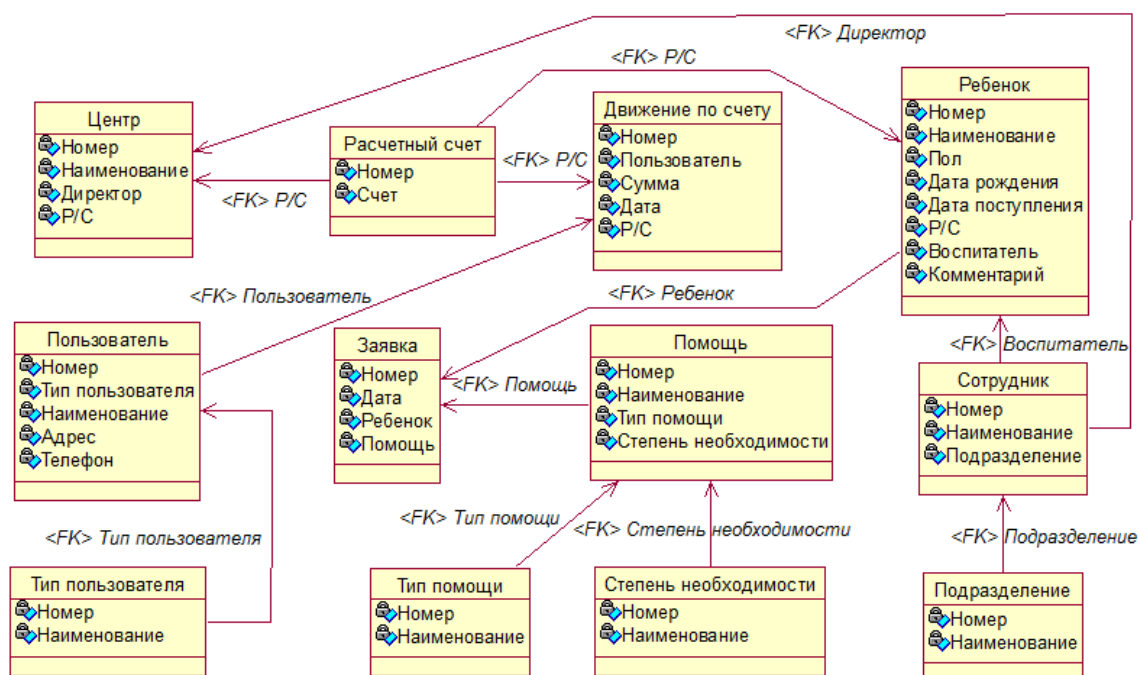


Рисунок 3.2.1 – Диаграмма классов

### 3.3 ДИАГРАММА ДЕЯТЕЛЬНОСТИ

Диаграмма деятельности — UML-диаграмма, на которой показано разложение некоторой деятельности на ее составные части. Под деятельностью (англ. activity) понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчиненных элементов — вложенных видов деятельности и отдельных действий англ. action, соединенных между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

Диаграмма деятельности представлена на рисунке 3.3.1.

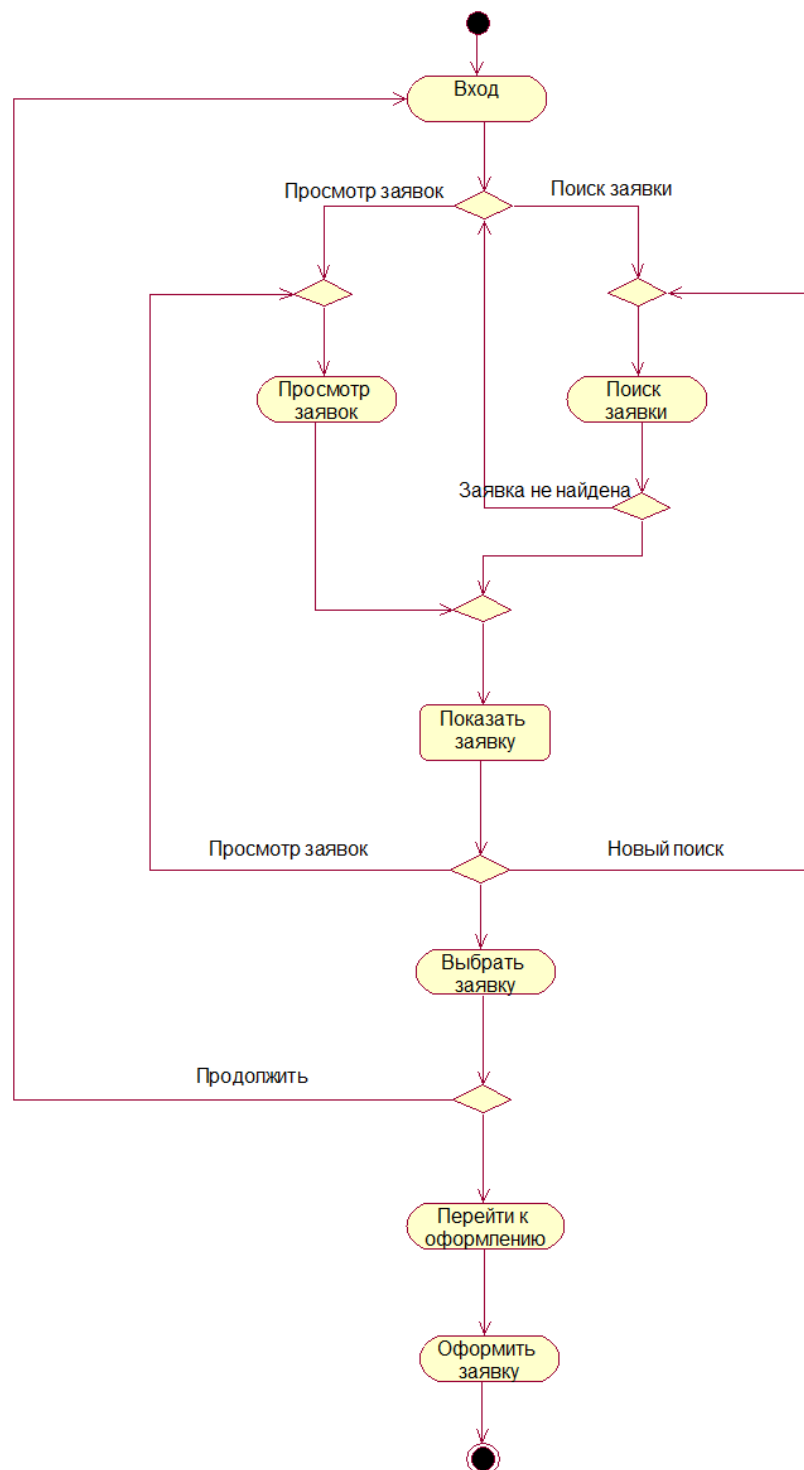


Рисунок 3.3.1 – Диаграмма деятельности

### 3.4 ДИАГРАММА КООПЕРАЦИЙ

Главная особенность диаграммы кооперации заключается в возможности графически представить не только последовательность взаимодействия, но и все структурные отношения между объектами, участвующими в этом взаимодействии.

Прежде всего, на диаграмме кооперации в виде прямоугольников изображаются участвующие во взаимодействии объекты, содержащие имя объекта, его класс и, возможно, значения атрибутов. Далее, как и на диаграмме классов, указываются ассоциации между объектами в виде различных соединительных линий. При этом можно явно указать имена ассоциации и ролей, которые играют объекты в данной ассоциации. Дополнительно могут быть изображены динамические связи - потоки сообщений. Они представляются также в виде соединительных линий между объектами, над которыми располагается стрелка с указанием направления, имени сообщения и порядкового номера в общей последовательности инициализации сообщений.

В отличие от диаграммы последовательности, на диаграмме кооперации изображаются только отношения между объектами, играющими определенные роли во взаимодействии. На этой диаграмме не указывается время в виде отдельного измерения. Поэтому последовательность взаимодействий и параллельных потоков может быть определена с помощью порядковых номеров. Следовательно, если необходимо явно специфицировать взаимосвязи между объектами в реальном времени, лучше это делать на диаграмме последовательности.

Диаграмма коопераций представлена на рисунке 3.4.1.

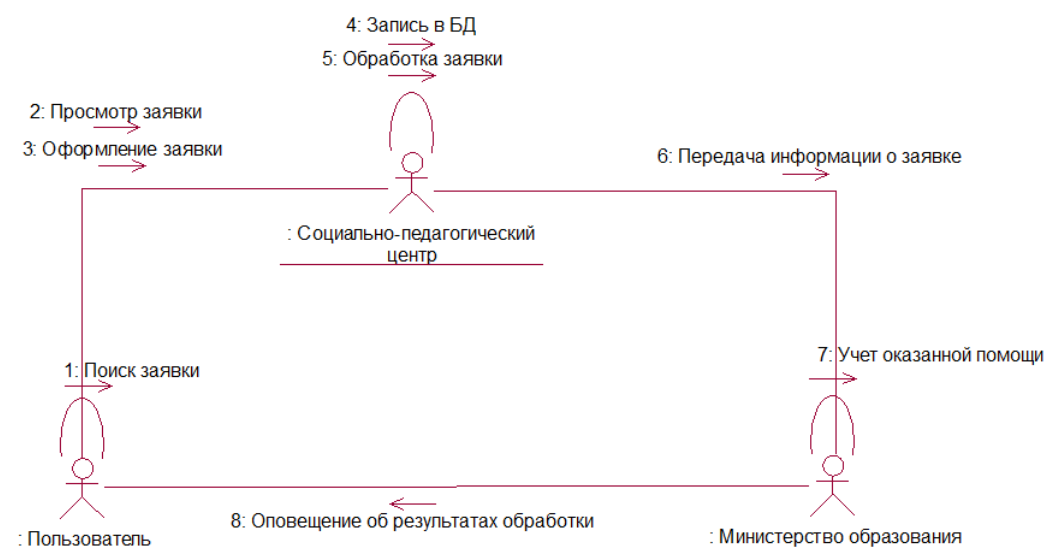


Рисунок 3.4.1 – Диаграмма коопераций

### 3.5 ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ

Диаграмма последовательности — диаграмма, на которой показано взаимодействие объектов, упорядоченное по времени, с отражением продолжительности обработки и последовательности их проявления. Используется в языке UML.

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 представлена на рисунке 3.5.1.

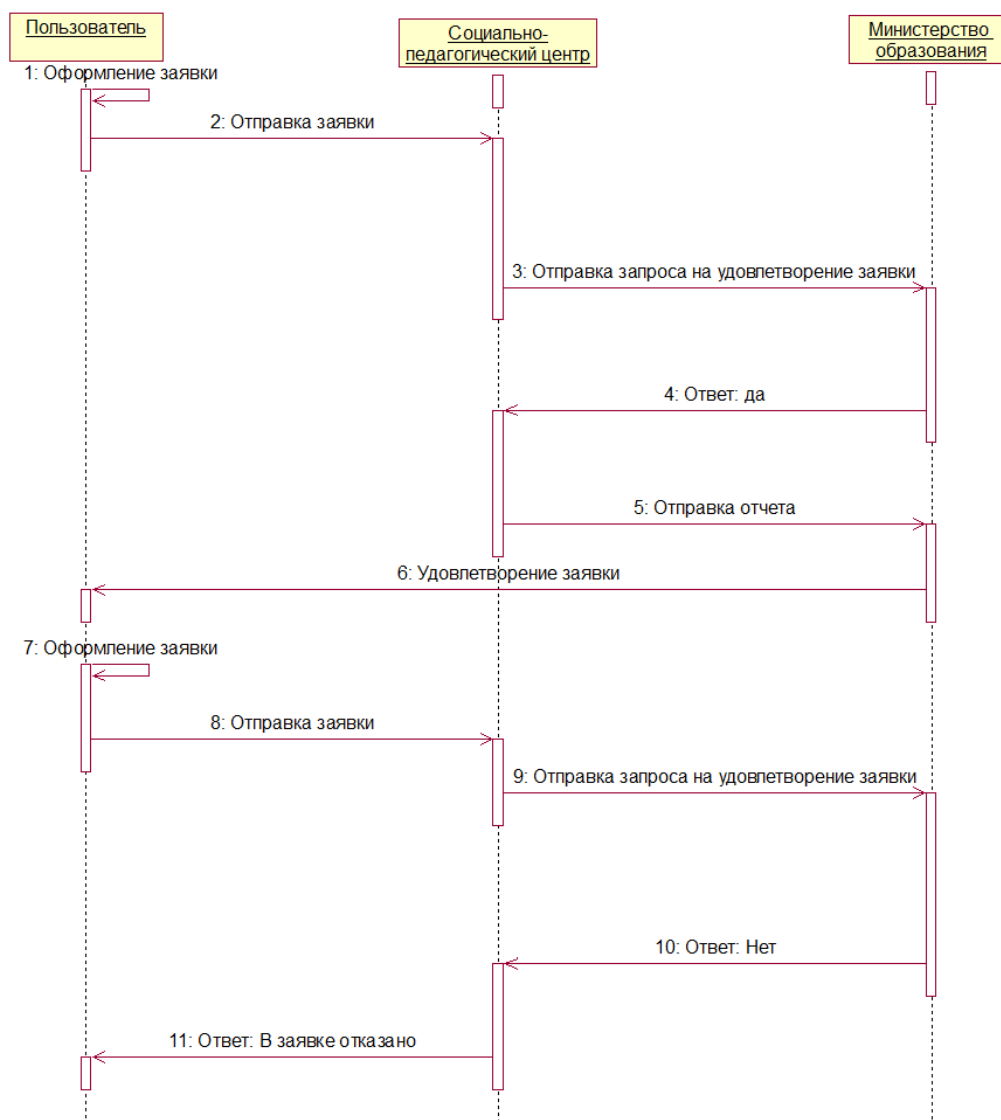


Рисунок 3.5.1 – Диаграмма последовательности



## 3.6 ДИАГРАММА РАЗВЕРТЫВАНИЯ

Диаграмма развертывания в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать веб-сайт диаграмма развертывания должна показывать, какие аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты («артефакты») работают на каждом узле (например, веб-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, REST, RMI).

Узлы представляются как прямоугольные параллелепипеды с артефактами, расположенными в них, изображенными в виде прямоугольников. Узлы могут иметь под узлы, которые представляются как вложенные прямоугольные параллелепипеды. Один узел диаграммы развертывания может концептуально представлять множество физических узлов, таких как кластер серверов баз данных.

Существует два типа узлов:

- узел устройства;
- узел среды выполнения.

Узлы устройств — это физические вычислительные ресурсы со своей памятью и сервисами для выполнения программного обеспечения, такие как обычные ПК, мобильные телефоны. Узел среды выполнения — это программный вычислительный ресурс, который работает внутри внешнего узла и который предоставляет собой сервис, выполняющий другие исполняемые программные элементы.

Диаграмма развертывания представлена на рисунке 3.6.1.

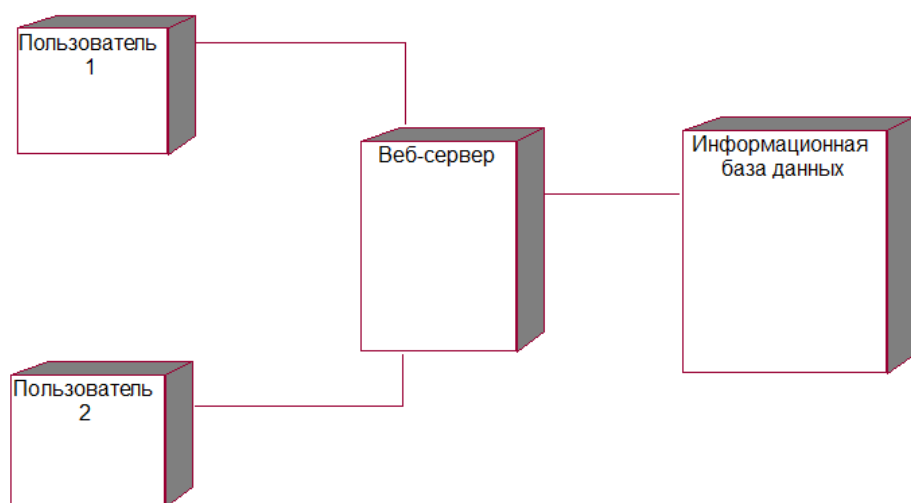


Рисунок 3.6.1 – Диаграмма развертывания

### 3.7 ДИАГРАММА КОМПОНЕНТОВ

Диаграмма компонентов — статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п.

Компоненты связываются через зависимости, когда соединяется требуемый интерфейс одного компонента с имеющимся интерфейсом другого компонента. Таким образом иллюстрируются отношения клиент-источник между двумя компонентами.

Зависимость показывает, что один компонент предоставляет сервис, необходимый другому компоненту. Зависимость изображается стрелкой от интерфейса или порта клиента к импортируемому интерфейсу.

Когда диаграмма компонентов используется, чтобы показать внутреннюю структуру компонентов, предоставляемый и требуемый интерфейсы составного компонента могут делегироваться в соответствующие интерфейсы внутренних компонентов.

Делегация показывается связь внешнего контракта компонента с внутренней реализацией этого поведения внутренними компонентами.

Диаграмма компонентов представлена на рисунке 3.7.1.

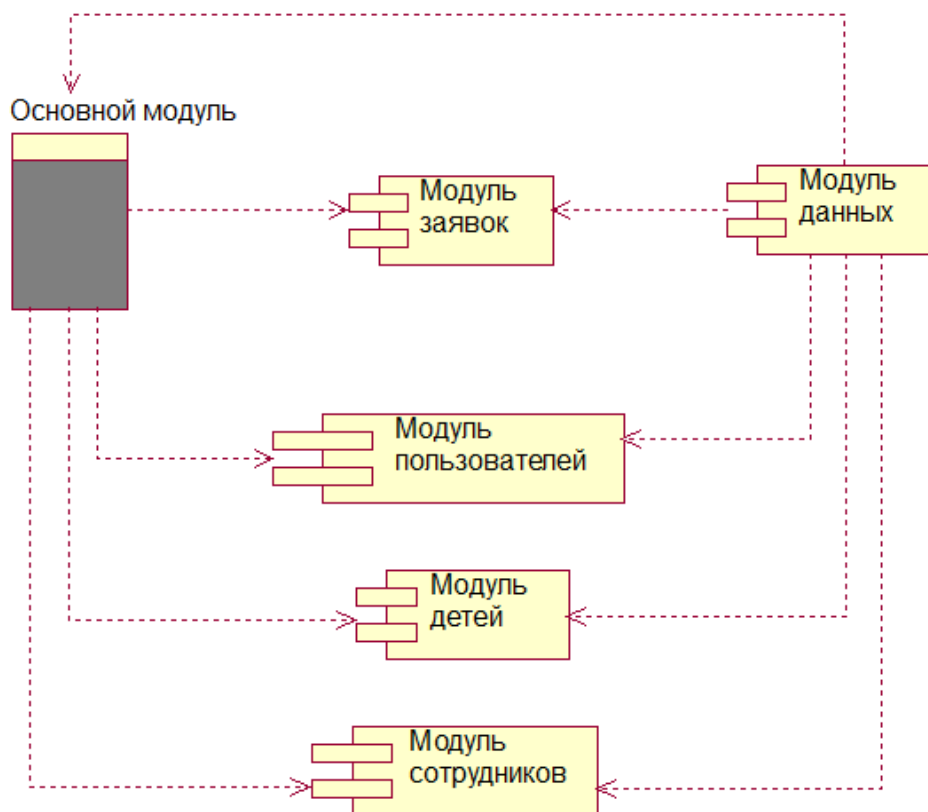


Рисунок 3.7.1 – Диаграмма компонентов

## 4 ГЕНЕРАЦИЯ КОДА

На рисунках 4.1 — 4.5 представлен процесс генерации кода на языке Java, непосредственно сам код расположен ниже.

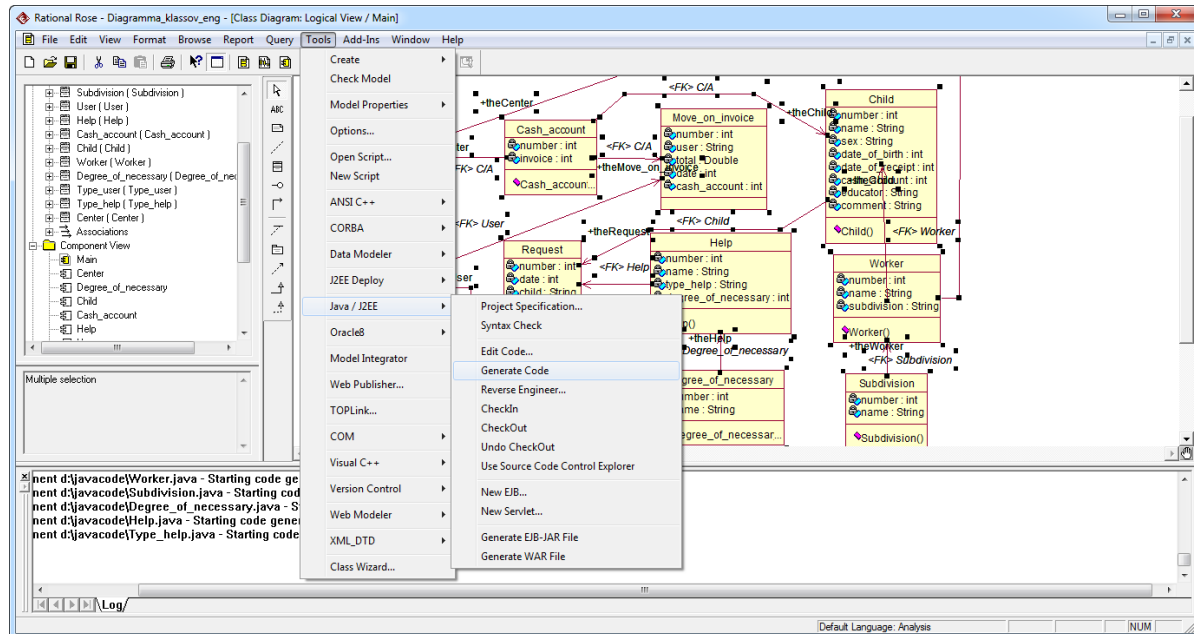


Рисунок 4.1 – Процесс генерации кода

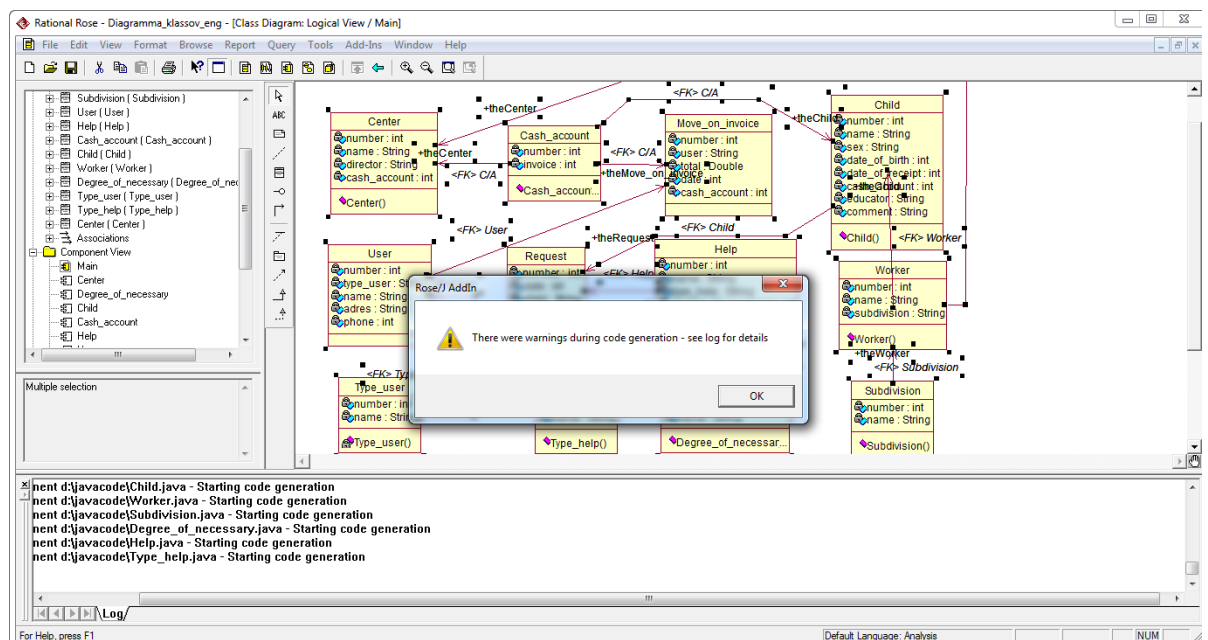


Рисунок 4.2 – Процесс генерации кода



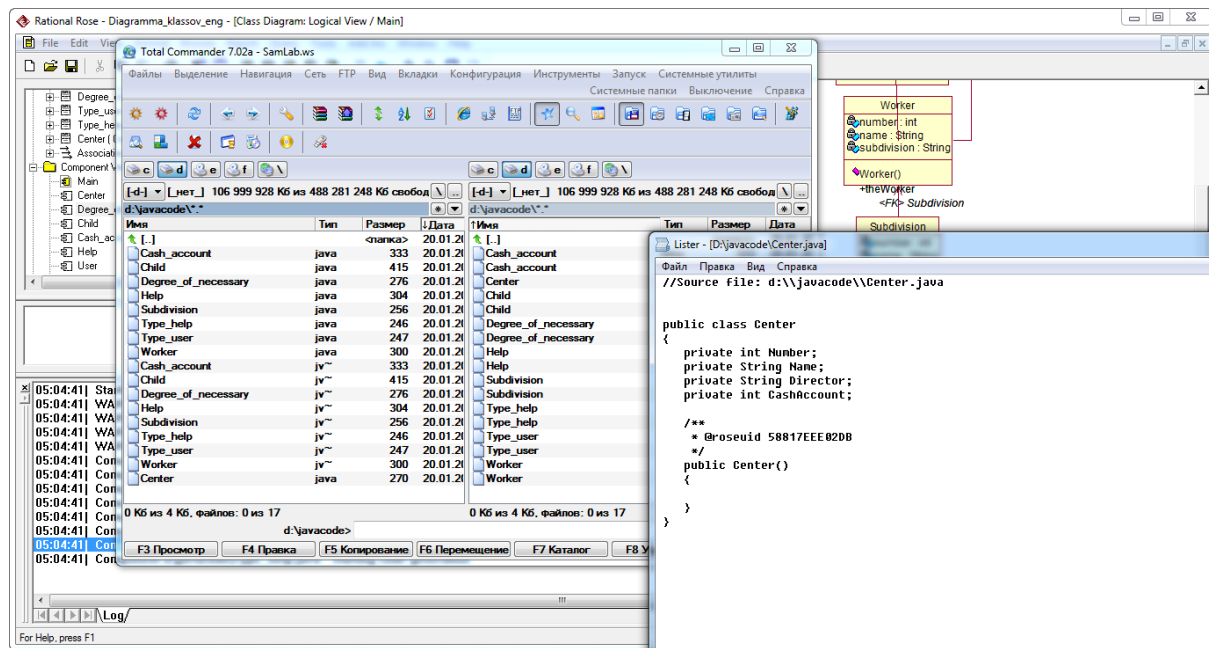


Рисунок 4.5 – Процесс генерации кода

Java — строго типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретенной компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года.

Язык Java активно используется для создания мобильных приложений под операционную систему Android. При этом программы компилируются в нестандартный байт-код, для использования их виртуальной машиной Dalvik (начиная с Android 5.0 Lollipop виртуальная машина заменена на ART). Для такой компиляции используется дополнительный инструмент, а именно Android SDK (Software Development Kit), разработанный компанией Google.

Разработку приложений можно вести в среде Android Studio, NetBeans, в среде Eclipse, используя при этом плагин Android Development Tools (ADT), или в IntelliJ IDEA. Версия JDK при этом должна быть 5.0 или выше.

Сгенерированный код:

```
public class Center
{
    private int Number;
    private String Name;
    private String Director;
    private int CashAccount;
```

```

/**
 * @roseuid 58817EEE02DB
 */
public Center()
{

}
}
public class Cash_account
{
    private int number;
    private int invoice;
    public Center theCenter;
    public Move_on_invoice theMove_on_invoice;
    public Child theChild;

/**
 * @roseuid 58818B5A02B1
 */
    public Cash_account()
    {

    }
}
public class Child
{
    private int number;
    private String name;
    private String sex;
    private int date_of_birth;
    private int date_of_receipt;
    private int cash_account;
    private String educator;
    private String comment;
    public Request theRequest;

/**
 * @roseuid 58818B5A033C
 */
    public Child()
    {

    }
}

```

```

public class Degree_of_necessary
{
    private int number;
    private String name;
    public Help theHelp;

    /**
     * @roseuid 58818B5B0041
     */
    public Degree_of_necessary()
    {

    }
}
public class Help
{
    private int number;
    private String name;
    private String type_help;
    private int degree_of_necessary;
    public Request theRequest;

    /**
     * @roseuid 58818B5B0076
     */
    public Help()
    {

    }
}
public class Subdivision
{
    private int number;
    private String name;
    public Worker theWorker;

    /**
     * @roseuid 58818B5B000E
     */
    public Subdivision()
    {

    }
}

```

```

public class Type_help
{
    private int number;
    private String name;
    public Help theHelp;

    /**
     * @roseuid 58818B5B00B4
     */
    public Type_help()
    {

    }
}

```

```

public class Type_user
{
    private int number;
    private String name;
    public User theUser;

    /**
     * @roseuid 58818B5A024F
     */
    private Type_user()
    {

    }
}

```

```

public class Worker
{
    private int number;
    private String name;
    private String subdivision;
    public Center theCenter;
    public Child theChild;

    /**
     * @roseuid 58818B5A0390
     */
    public Worker()
    {
    }
}

```



## ЗАКЛЮЧЕНИЕ

Предметной областью курсового проектирования является веб-сервис «Информационно-справочная система социально-педагогического центра». Разработанное программное средство позволит быстро и удобно предоставить полную информацию о социально-педагогическом центре, значительно упростив его деятельность.

Пояснительная записка состоит из следующих разделов:

1. Постановка задачи. В данном разделе была поставлена задача по разработке веб-сервиса «Информационно-справочная система социально-педагогического центра».
2. Описание основного процесса предметной области с использованием стандарта IDEF0.
3. Описание информационной модели с использованием стандарта IDEF1.
4. Описание моделей информационной системы с использованием языка UML.
5. Генерация кода на основе моделей Rational Rose.

В ходе написания контрольной работы были закреплены теоретические знания по таким разделам как составление функциональной модели (IDEF0), разработка логической и физической модели IDEF1X, проектирование системы с помощью унифицированного языка моделирования (UML).

Контрольная работа выполнялась в соответствии с методическими требованиями и указаниями.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 2.105-95 ЕСКД. Введ. 1996–07–01. – М.: Изд-во стандартов, 1996.
2. ГОСТ 19.104–78 – Единая система программной документации. Основные надписи.
3. Маклаков С.В. BPWIN и ERWIN: CASE - средства для разработки информационных систем.
4. Федотов Д.Э., Семенов Ю.Д., Чижик К.Н. Практикум для высших учебных заведений. CASE-технологии.-157с.
5. Интернет обучение [Электронный ресурс] / Котляр Д.С. Использование CASE-средства ERwin для автоматизации проектирования и разработки базы данных – Режим доступа: <http://royallib.com/read/bezopasnost> – Дата доступа: 15.12.2016.
6. Интернет обучение [Электронный ресурс] / Информационные системы и технологии – Режим доступа: <http://www.narfu.ru> – Дата доступа: 19.12.2016.
7. Интернет обучение [Электронный ресурс] / Нормализация структурных данных – Режим доступа: <http://infostart.ru/public/269803/> – Дата доступа: 15.01.2016.
8. Интернет обучение [Электронный ресурс] / Программа компьютерного моделирования BpWin – Режим доступа: <http://bourabai.kz/cm/bpwin.htm>– Дата доступа: 11.11.2016.