|  |  |
| --- | --- |
|  | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждениевысшего образования"МИРЭА - Российский технологический университет"РТУ МИРЭА | |
| Институт информационных технологий (ИТ) | |
| Кафедра вычислительной техники | |

|  |  |
| --- | --- |
| **Отчет по практической работе. Тема:**  **«Подготовка и сдача ЕГЭ»** | |
| **по дисциплине** | |
| **«Системы обработки и хранения данных»** | |
|  | |
| Выполнил студент группы ИКБО-20-19 | Московка Артём Александрович |
| Принял | Володина Анна Михайловна  *Старший преподаватель* |

|  |  |  |
| --- | --- | --- |
| Практические работы выполнены | «\_\_» \_\_\_\_\_\_\_\_\_\_2021 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_\_\_\_2021 г. |  |

2021 г.

1. Проектирование информационной модели в нотации IDEF0

*Модель IDEF0 по предметной области «Подготовка и сдача ЕГЭ».*

**Предметная область «Подготовка и сдача ЕГЭ»** — обобщенная информационная структура, демонстрирующая взаимодействия учеников в процессе подготовки к написанию и самой сдаче экзамена.

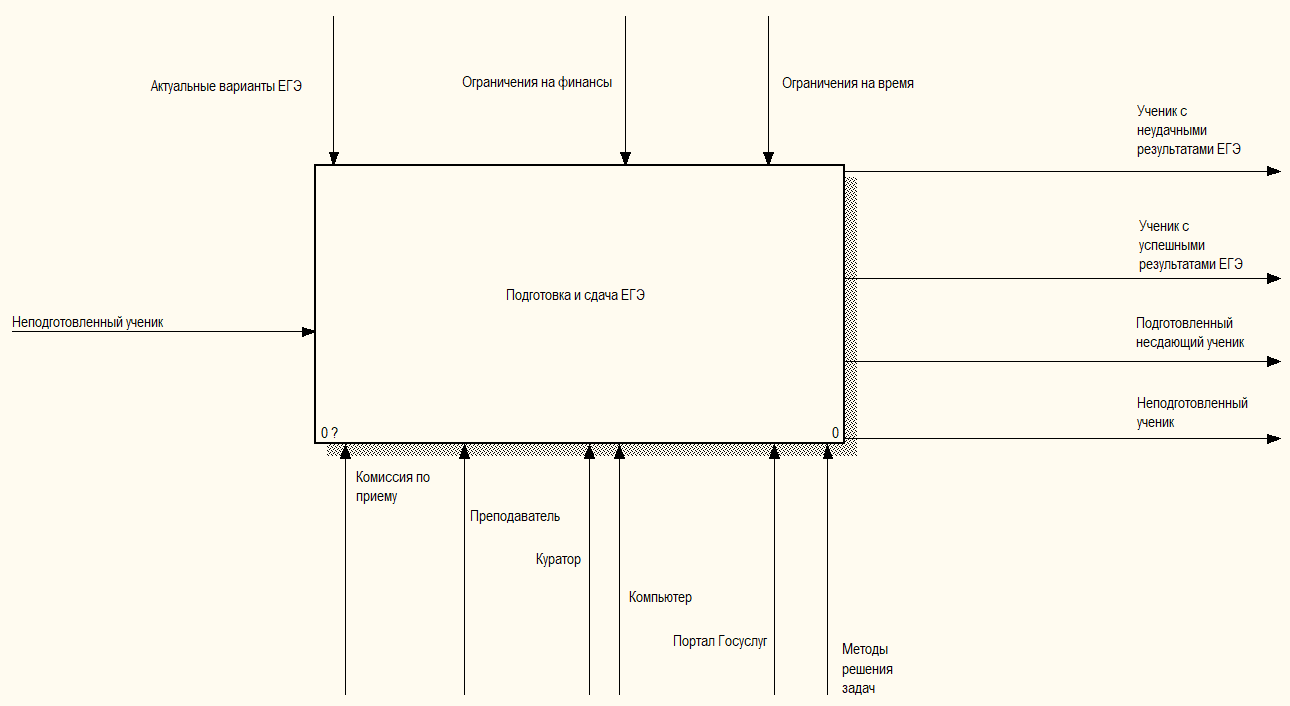


Рис. 1 – Скриншот контекстной диаграммы в нотации IDEF0*.*

Основной блок – Подготовка и сдача ЕГЭ.

Входной информацией системы является:

* **Неподготовленный ученик.**

Выходной информацией системы являются:

* **Ученик с неудачными результатами ЕГЭ;**
* **Ученик с успешными результатами ЕГЭ;**
* **Подготовленный несдающий ученик;**
* **Неподготовленный ученик.**

Механизмами информации системы являются:

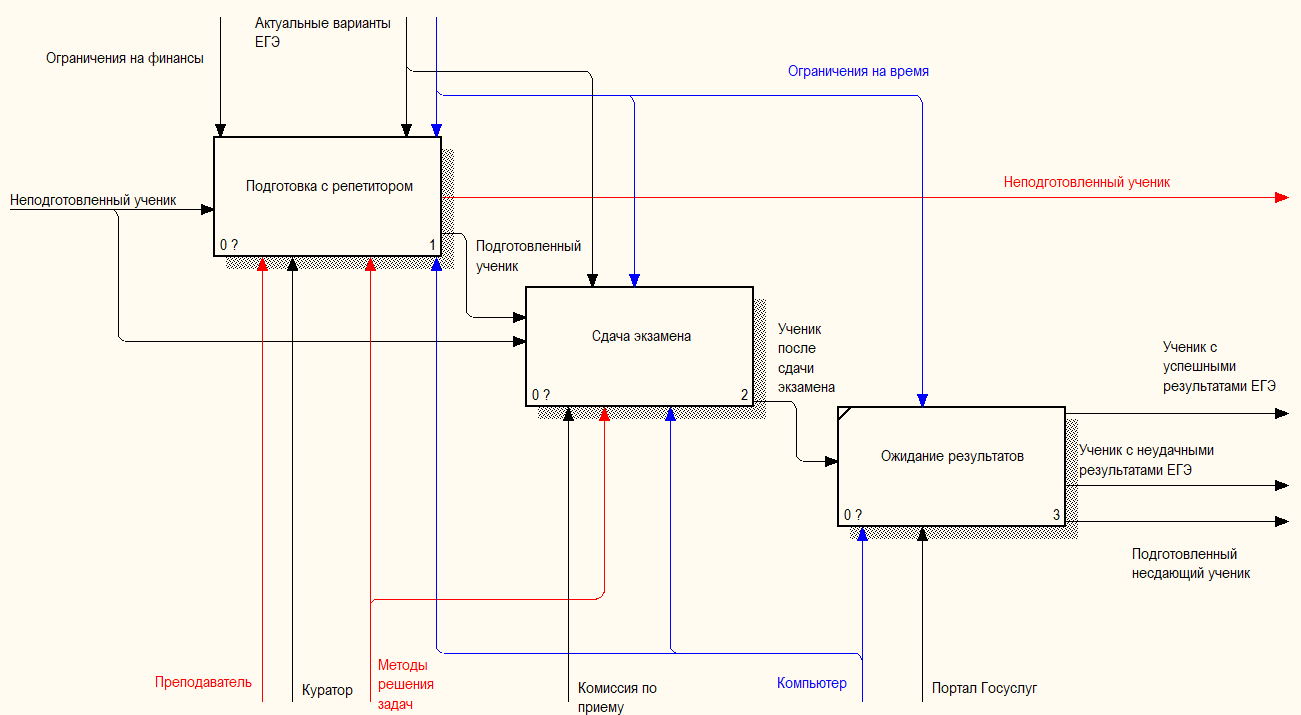
* **Комиссия по приему;**
* **Преподаватель;**
* **Куратор;**
* **Компьютер;**
* **Портал Госуслуг;**
* **Методы решения задач.**

Управляющей информацией системы являются:

* **Актуальные варианты ЕГЭ;**
* **Ограничения на время;**
* **Ограничения на финансы.**

Мы декомпозировали общий блок «Подготовка и сдача ЕГЭ» на связанные между собой элементы. В нашем случае делится на 3 основных этапа:

* **Подготовка с репетитором;**
* **Сдача экзамена;**
* **Ожидание результатов.**

****Рис. 2 – Скриншот декомпозиции контекстной диаграммы

Блок «Подготовка с репетитором» мы декомпозируем еще на три этапа:

* **Поиск репетитора;**
* **Заключение договора;**
* **Учебный процесс.**

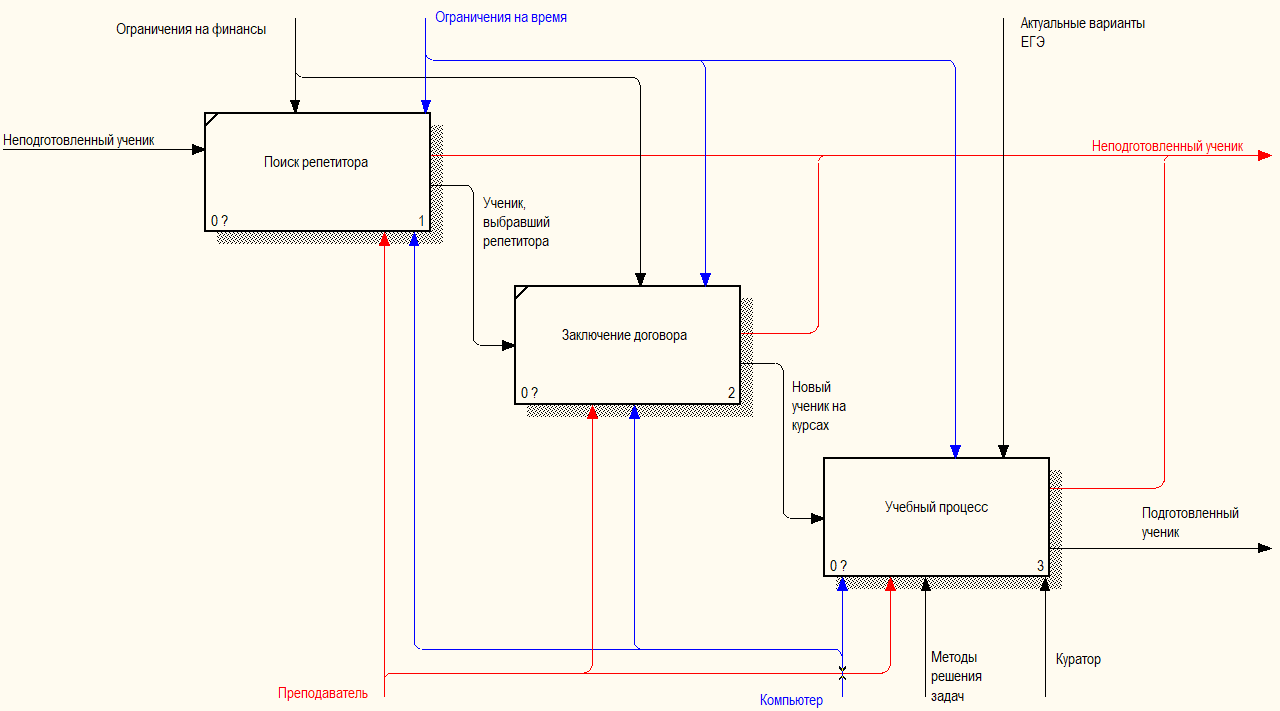


Рис. 3 – Скриншот декомпозиции процесса «Подготовка с репетитором»

Процесс «Учебный процесс» мы декомпозируем еще на 3 этапа:

* **Изучение теории;**
* **Выполнение практических заданий и контрольных работ;**
* **Сдача пробных экзаменов.**

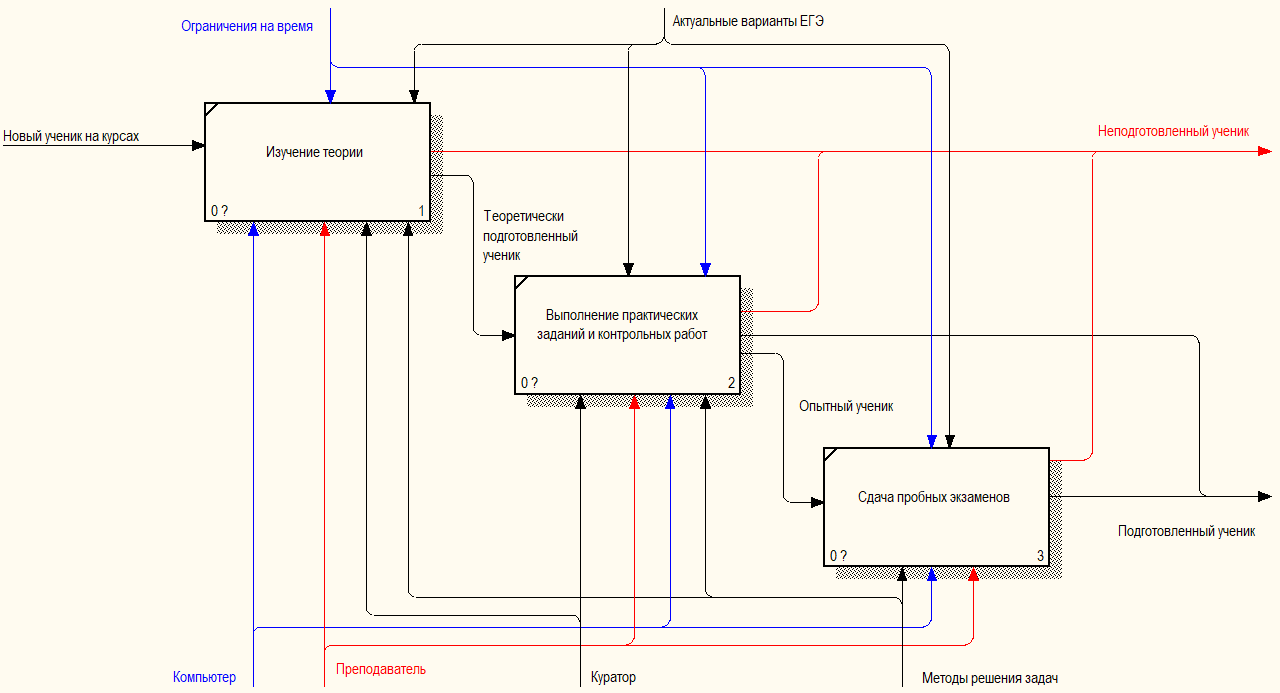
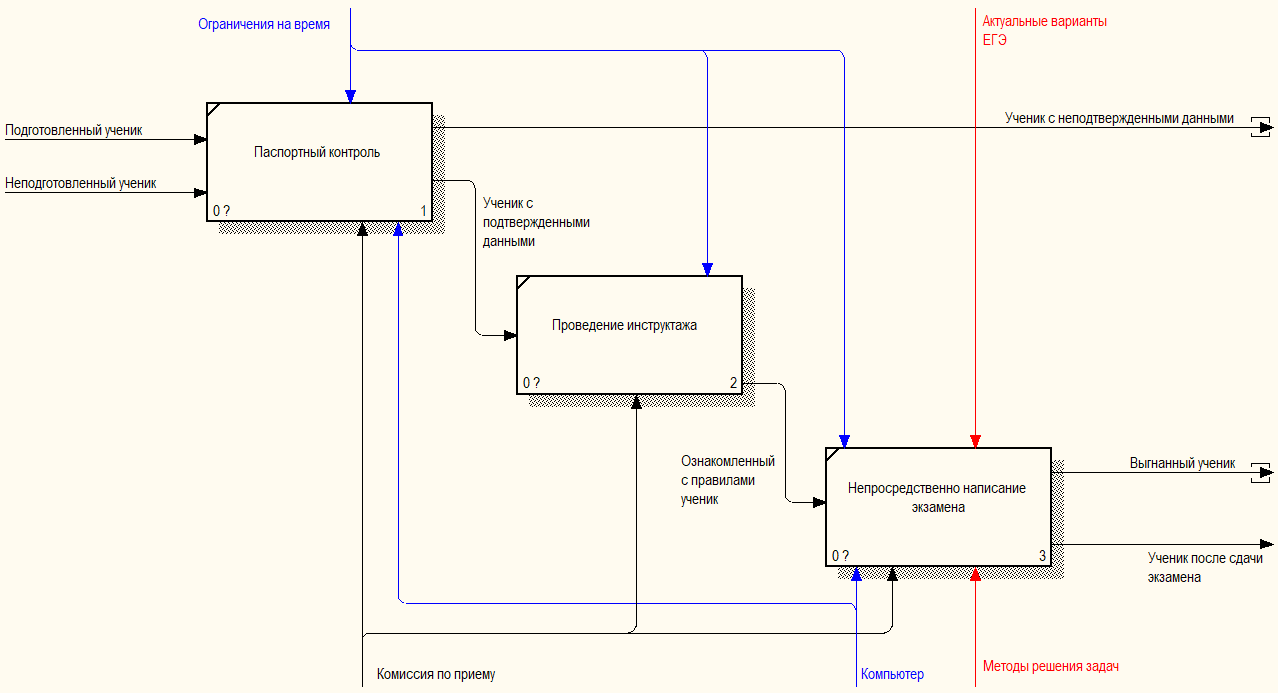
****

Рис. 4 – Скриншот декомпозиции процесса «Учебный процесс»

Процесс «Сдача экзамена» мы декомпозируем еще на 3 этапа:

* **Паспортный контроль;**
* **Проведение инструктажа;**
* **Непосредственно написание экзамена.**

Рис. 5 – Скриншот декомпозиции процесса «Сдача экзамена»

1. Проектирование информационной модели в нотации DFD

*Модель DFD по предметной области «Подготовка и сдача ЕГЭ».*

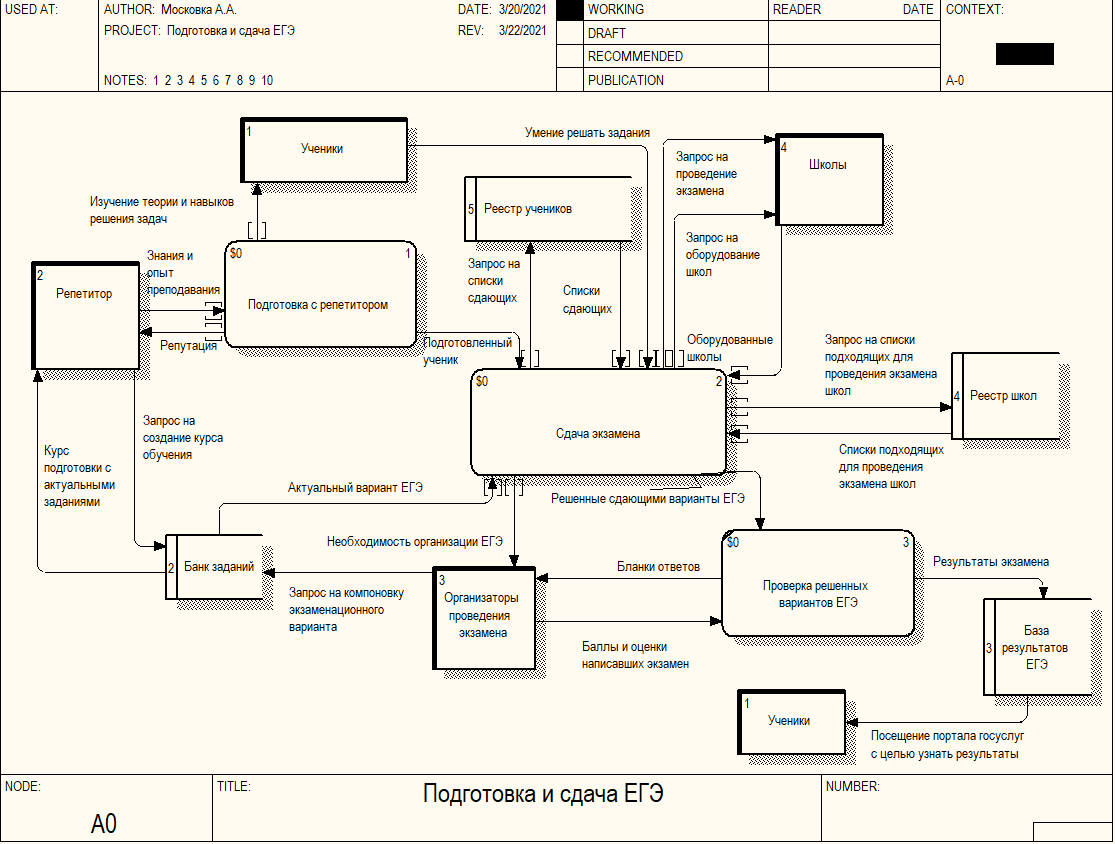
**

Рис. 6 – Скриншот основного процесса «Подготовка и сдача ЕГЭ»

На данной модели отображается основной процесс (сама система в целом) и ее связи с внешней средой (внешними сущностями). Это взаимодействие показывается через потоки данных.

**Внешние сущности** изображают входы в систему и/или выходы из нее. У нас внешние сущности это: Ученики, Репетитор, Школы, Организаторы проведения экзамена.

**Стрелки (потоки данных).** Стрелки описывают движение объектов из одной части системы в другую.

**Хранилище данных.** В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в покое. У нас хранилища данных это: Банк заданий, База результатов ЕГЭ, Реестр школ, Реестр учеников.

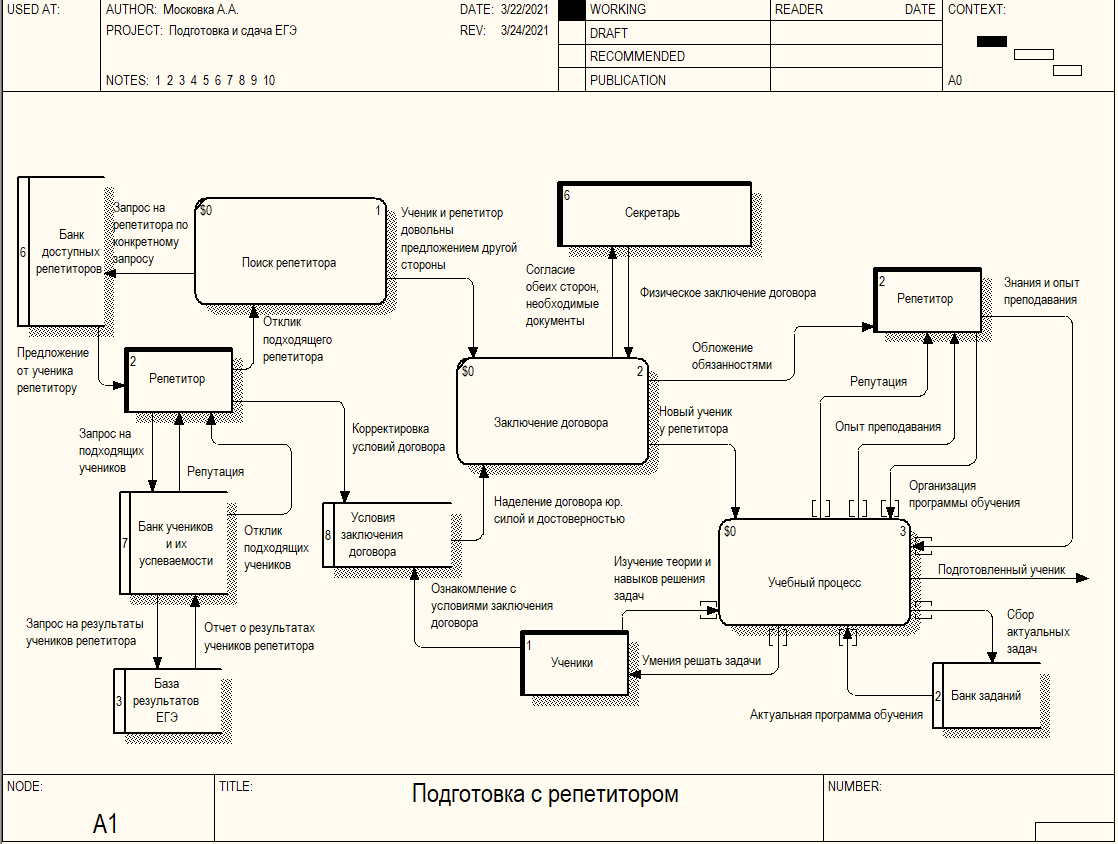


Рис. 7 – Скриншот декомпозиции процесса «Подготовка с репетитором»

Мы декомпозировали общий блок «Подготовка и сдача ЕГЭ» на связанные между собой элементы. В нашем случае это 3 этапа:

* **Поиск репетитора;**
* **Заключение договора;**
* **Учебный процесс.**

На данной модели отображается дочерний процесс (Подготовка с репетитором) и его связи с внешней средой (внешними сущностями). Это взаимодействие показывается через потоки данных.

**Внешние сущности** на данной модели это: Ученики, Репетитор, Секретарь.

**Хранилищами данных** в контексте данной диаграммы являются: База результатов ЕГЭ, Банк учеников и их успеваемости, Условия заключения договора, Банк заданий, Банк доступных репетиторов.

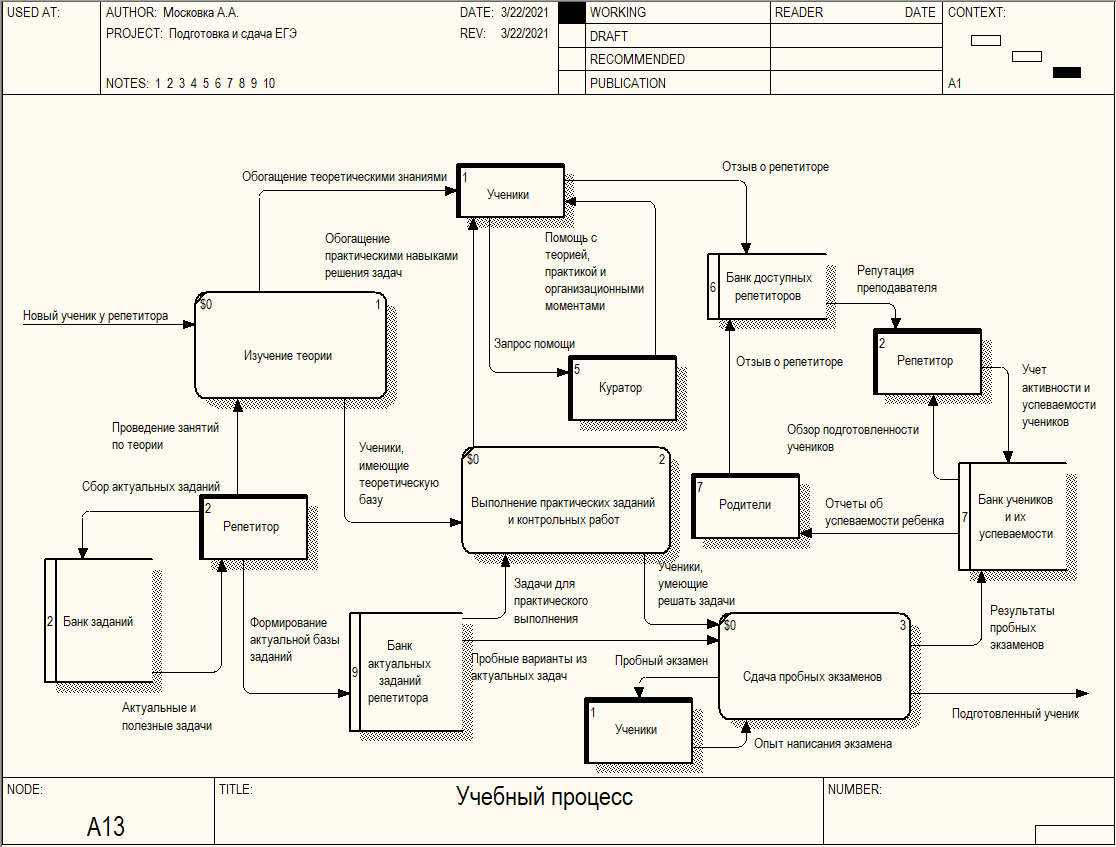


Рис. 8 – Скриншот декомпозиции процесса «Учебный процесс»

Мы декомпозировали общий блок «Подготовка с репетитором» на связанные между собой элементы. В нашем случае это 3 этапа:

* **Изучение теории;**
* **Выполнение практических заданий и контрольных работ;**
* **Сдача пробных экзаменов.**

На данной модели отображается дочерний процесс (Учебный процесс) и его связи с внешней средой (внешними сущностями). Это взаимодействие показывается через потоки данных.

**Внешние сущности** на данной модели это: Ученики, Репетитор, Куратор, Родители.

**Хранилищами данных** в контексте данной диаграммы являются: Банк заданий, Банк актуальных заданий репетитора, Банк учеников и их успеваемости, Банк доступных репетиторов.

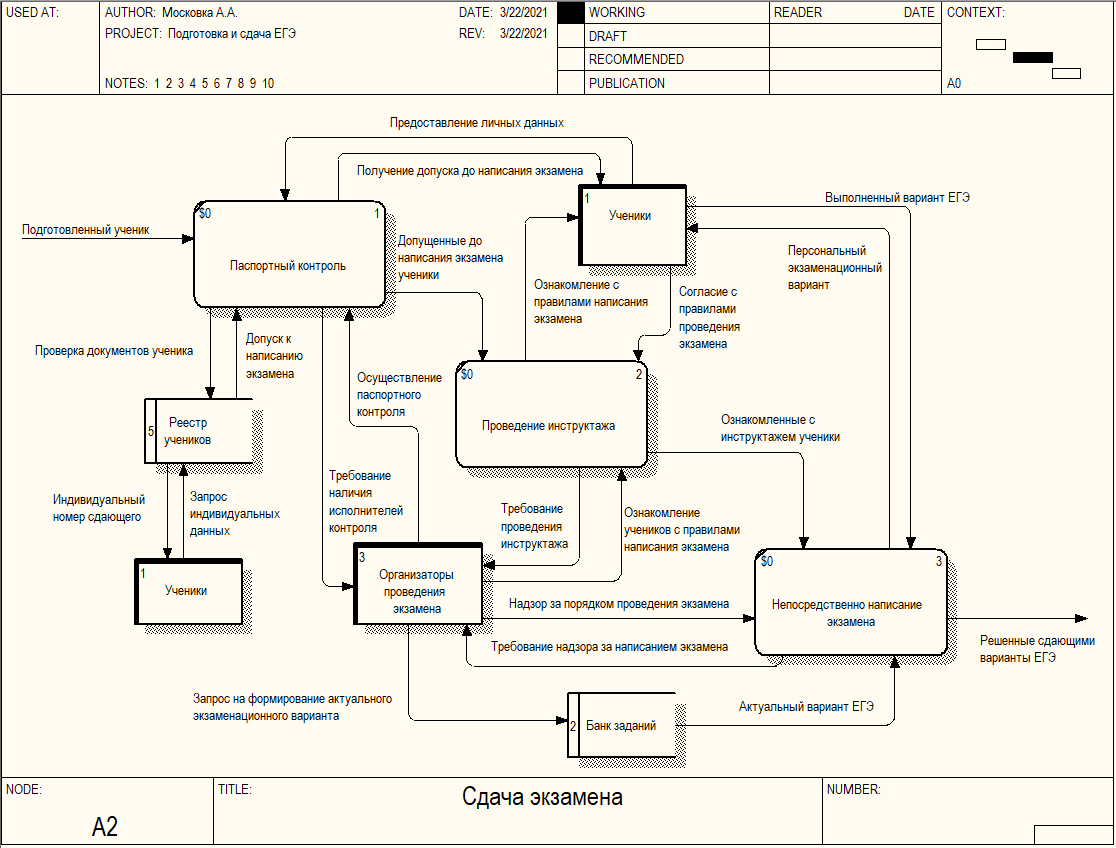


Рис. 9 – Скриншот декомпозиции процесса «Сдача экзамена»

Мы декомпозировали общий блок «Подготовка и сдача ЕГЭ» на связанные между собой элементы. В нашем случае это 3 этапа:

* **Паспортный контроль;**
* **Проведение инструктажа;**
* **Непосредственно написание экзамена.**

На данной модели отображается дочерний процесс (Сдача экзамена) и его связи с внешней средой (внешними сущностями). Это взаимодействие показывается через потоки данных.

**Внешние сущности** на данной модели это: Ученики, Организаторы проведения экзамена.

**Хранилищами данных** в контексте данной диаграммы являются: Банк заданий, Реестр учеников.

1. Проектирование информационной модели в нотации IDEF3

*Модель IDEF3 по предметной области «Подготовка и сдача ЕГЭ».*

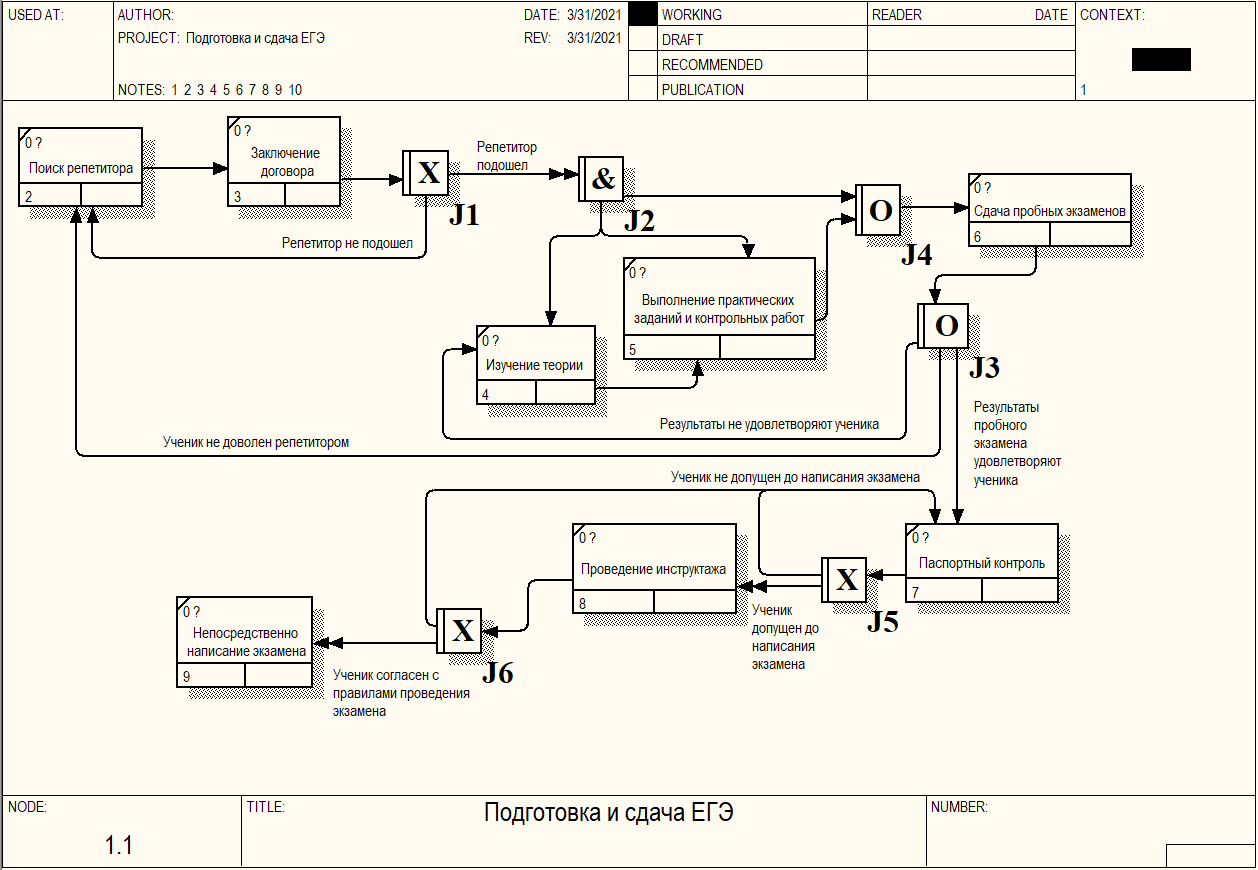
**

Рис. 10 – Скриншот модели IDEF3 процесса «Подготовка и сдача ЕГЭ»

Методология IDEF3 позволяет декомпозировать работу многократно, т. е. работа может иметь множество дочерних работ. Возможность множественной декомпозиции отражается в нумерации работ: номер работы состоит из номера родительской работы, номера декомпозиции и номера работы на текущей диаграмме.

**Слабые связи переходов** изображаются сплошными одинарными стрелками.

**Сильные связи переходов** изображаются двойными однонаправленными стрелками.

1. Проектирование информационной модели на языке UML

Целью данной работы является освоение технологии проектирования информационных систем с позиции объектно-ориентированного проектирования на основе языка UML.

Последовательность работ, следующая: строятся диаграммы логического проектирования, не имеющие прямого отношения к языку программирования. Это диаграммы концептуального моделирования.

* 1. **Диаграмма прецедентов**

Use case diagram (диаграмма прецедентов) – этот вид диаграмм позволяет создать список операций, которые выполняет система. Каждая такая диаграмма – это описание сценария поведения, которому следуют действующие лица (Actors).

Данный тип диаграмм используется при описании бизнес-процессов предметной области, определении требований к будущей программной системе, отражает объекты как системы, так и предметной области, и задачи, ими выполняемые. Окончательный вид диаграммы показан на рисунке 11. На диаграмме находятся: варианты использования и актеры. Также были добавлены ассоциации и описание всех элементов диаграммы.

На данной диаграмме мы отобразили организационную структуру подготовки и сдачи экзамена. Ученик с родителями ищет подходящего репетитора, затем родители с репетитором заключают договор на подготовку. Ученик может готовиться к экзамену и самостоятельно. После подготовки ученик идет сдавать экзамен, проходя КПП, прослушивая инструктаж, а также непосредственно пишет экзамен, после чего ожидает результатов.

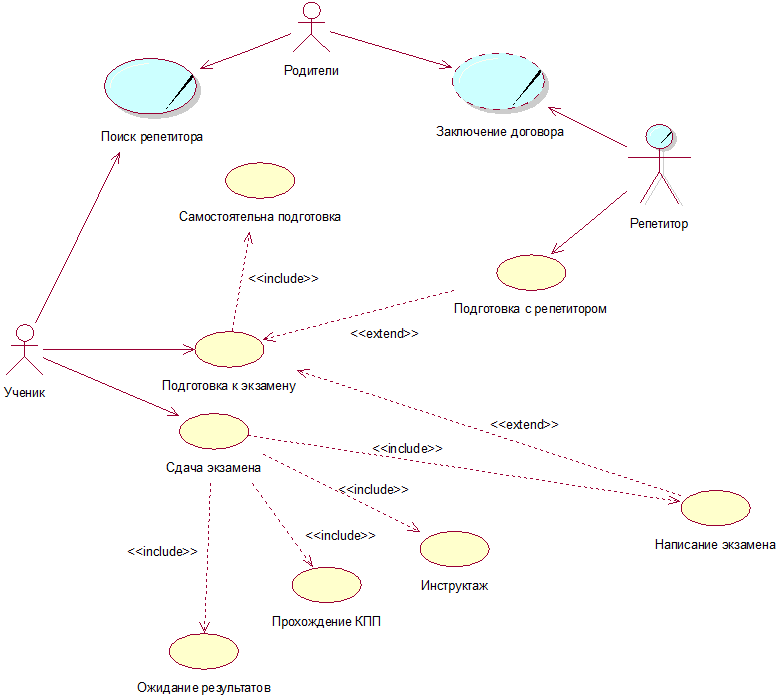


Рис. 11 – Скриншот диаграммы прецедентов

* 1. Диаграмма последовательности

Sequence diagram (диаграмма последовательностей действий). Данный тип диаграмм позволяет отразить последовательность передачи сообщений между объектами. Этот тип диаграммы не акцентирует внимание на конкретном взаимодействии, главный акцент уделяется последовательности приема/передачи сообщений. Окончательный вид диаграммы показан на рисунке 12-14. Диаграмма содержит действующие лица и объекты, так же добавлены такие объекты, как Object Message (Новое сообщение между объектами) и Message to Self (Рефлексивное сообщение самому себе).

На диаграмме показана последовательность подготовки к экзамену с репетитором.

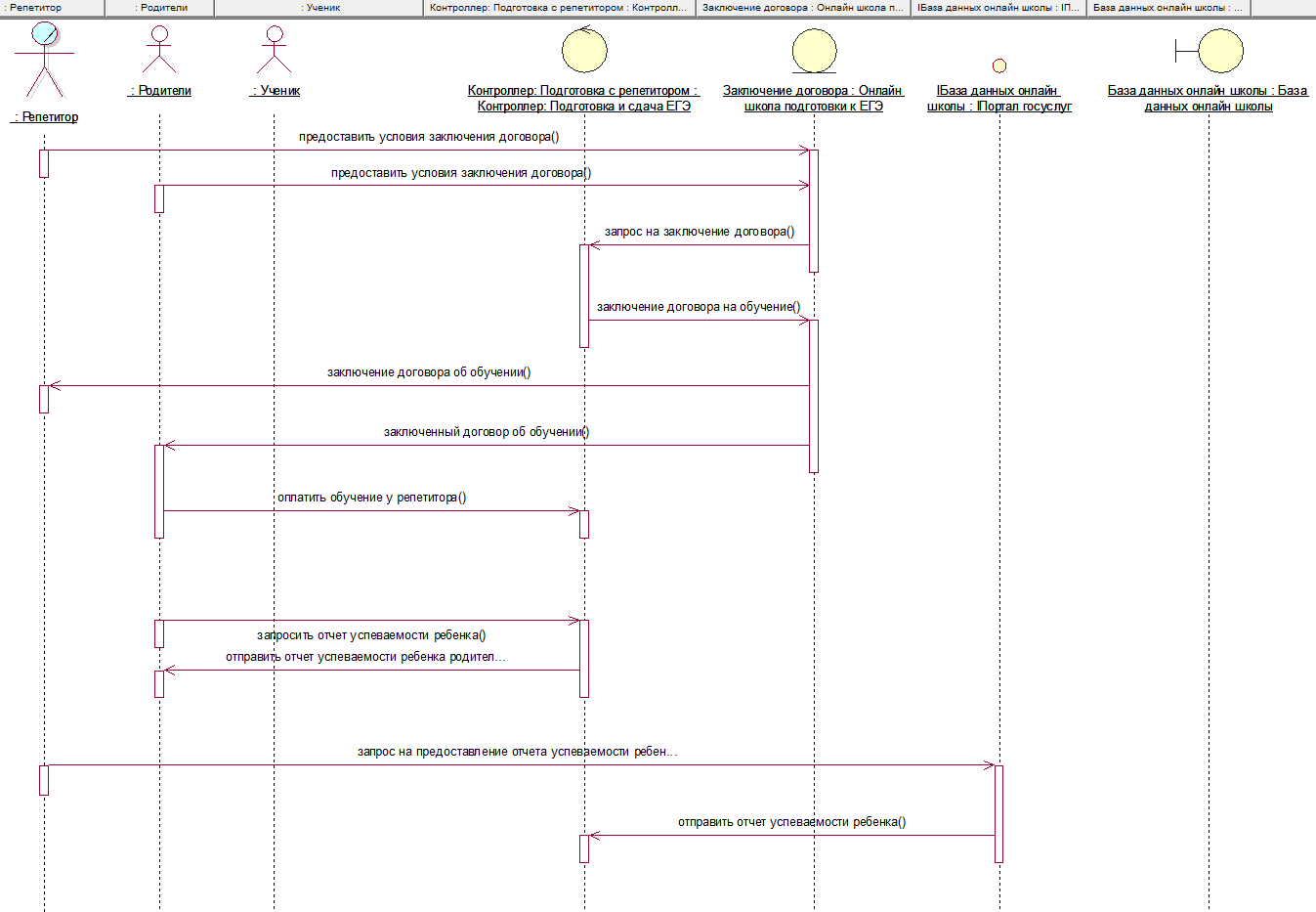


Рис. 12 – Скриншот диаграммы последовательности (ч. 1)

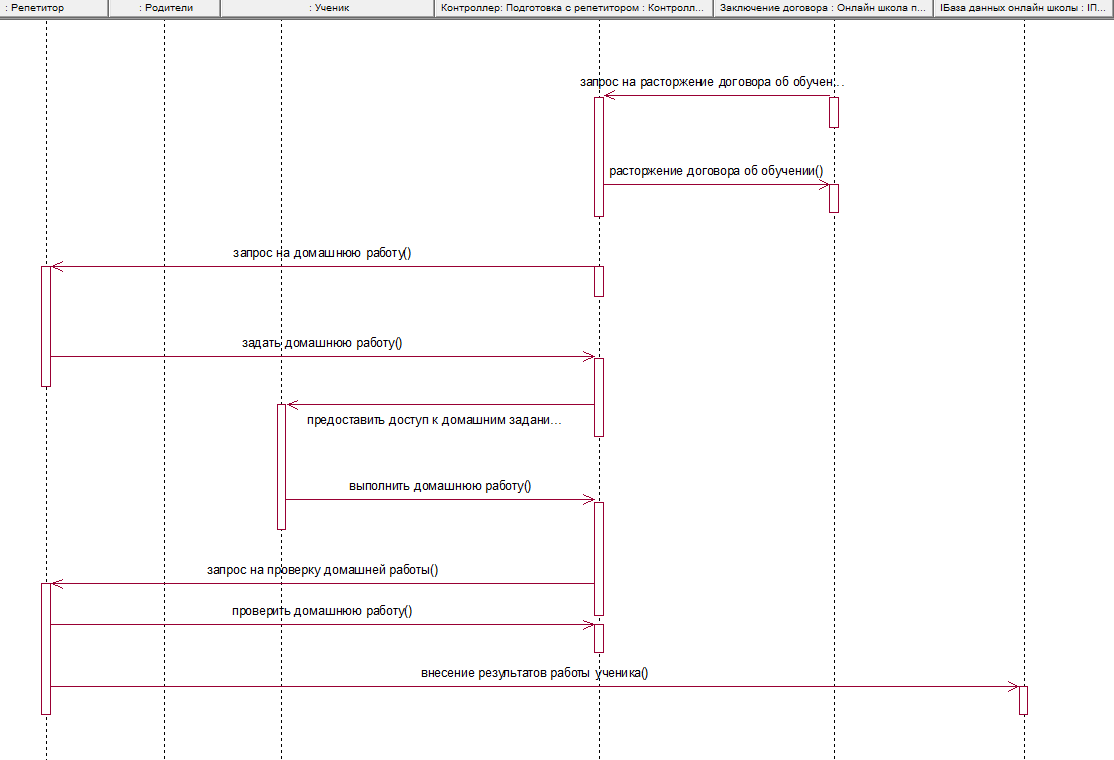


Рис. 13 – Скриншот диаграммы последовательности (ч. 2)

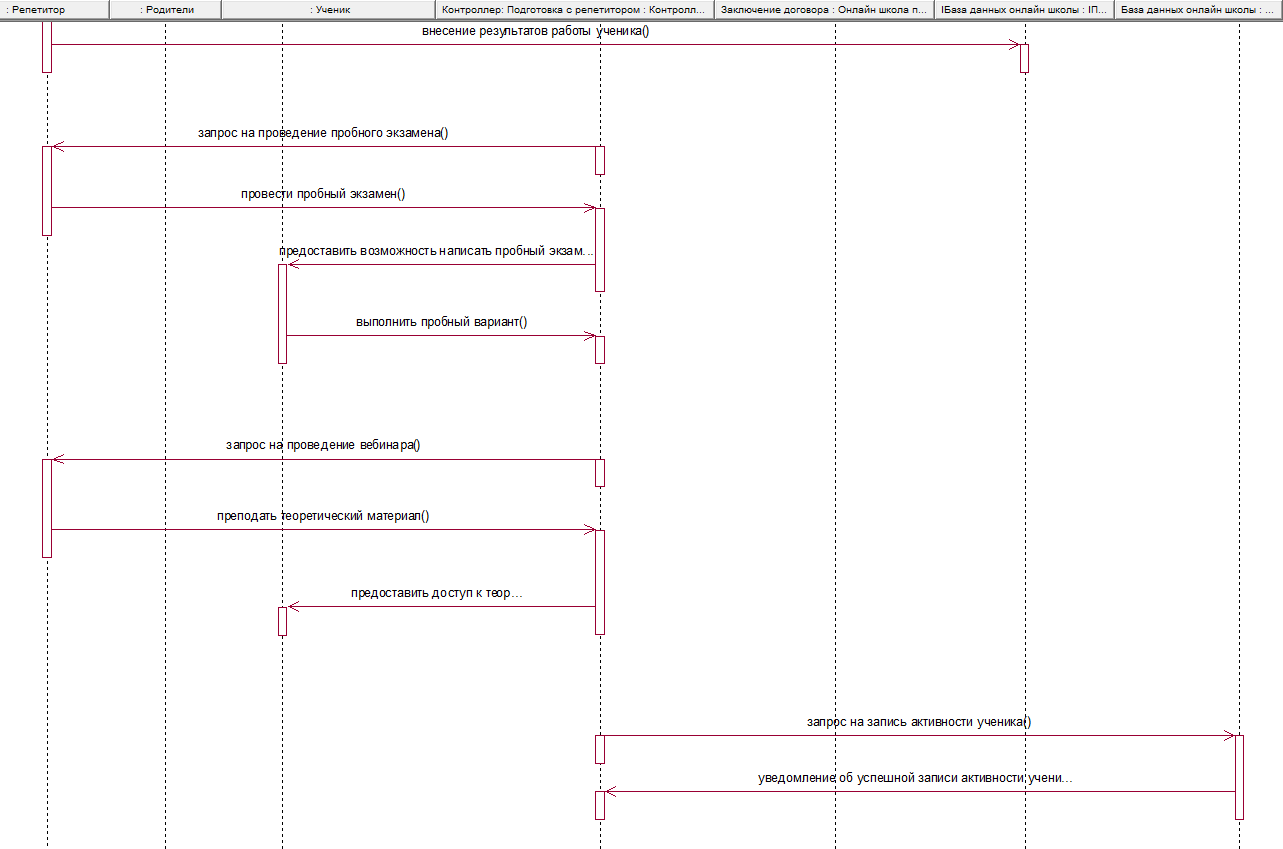


Рис. 14 – Скриншот диаграммы последовательности (ч. 3)

* 1. Кооперативная диаграмма

Collaboration diagram (диаграмма сотрудничества). Этот тип диаграмм позволяет описать взаимодействия объектов, абстрагируясь от последовательности передачи сообщений. На этом типе диаграмм в компактном виде отражаются все принимаемые и передаваемые сообщения конкретного объекта и типы этих сообщений. По причине того, что диаграммыSequenceиCollaborationявляются разными взглядами на одни и те же процессы,RationalRoseпозволяет создавать изSequenceдиаграммы диаграммуCollaborationи наоборот, а также производит автоматическую синхронизацию этих диаграмм. Окончательный вид диаграммы показан на рисунке 15. Диаграмма содержит Link To Self (Связь объекта самого с собой), Object Link (Связь между объектами), объекты и Link Message (сообщение, передаваеме между двумя объектами или объектом самому себе).

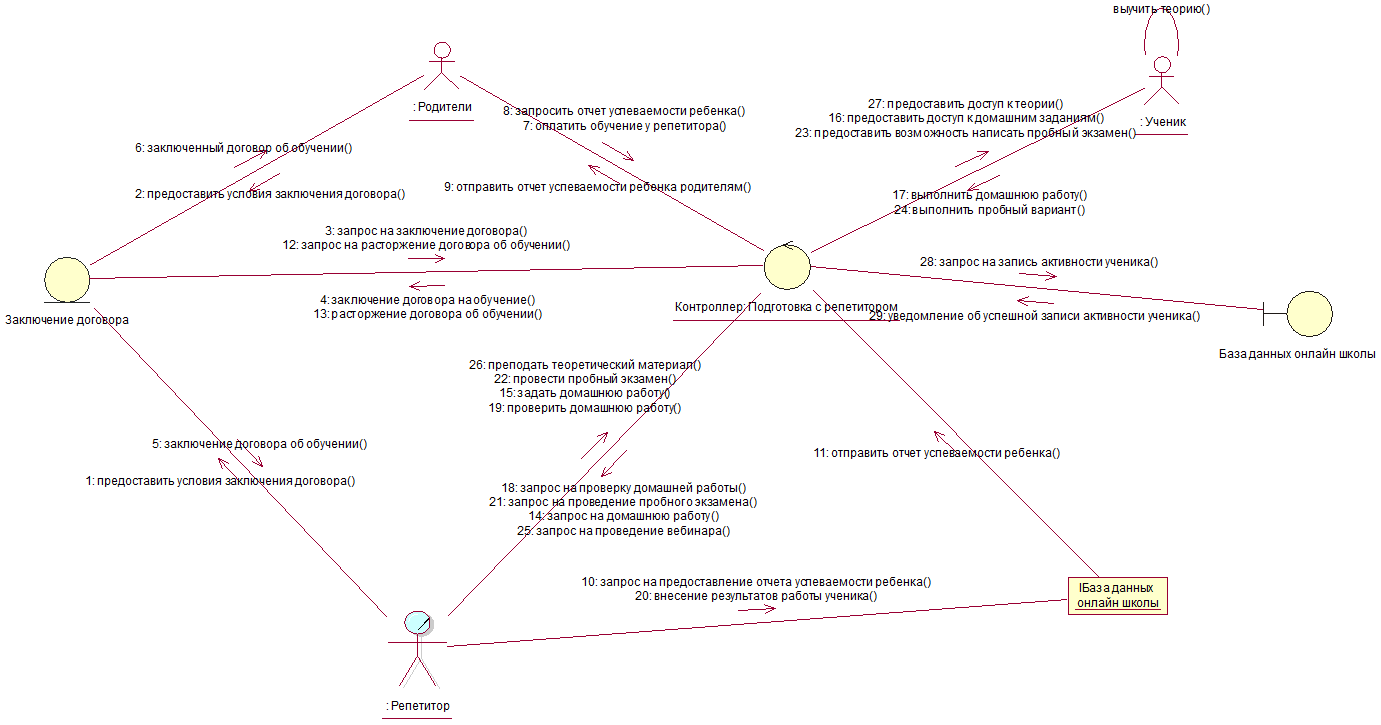


Рис. 15 – Скриншот диаграммы кооперации

* 1. Диаграмма состояний

Каждый объект системы, обладающий определенным поведением, может находится в определенных состояниях, переходить из состояния в состояние, совершая определенные действия в процессе реализации сценария поведения объекта. Поведение большинства объектов реальных систем можно представить с точки зрения теории конечных автоматов, то есть поведение объекта отражается в его состояниях, и данный тип диаграмм позволяет отразить это графически. Для этого используется два вида диаграмм: Statechart diagram (диаграмма состояний) и Activity diagram (диаграмма активности).

Диаграмма состояний предназначена для отображения состояний объектов системы, имеющих сложную модель поведения. Конечный вид диаграммы представлен на рисунке 16. Диаграмма содержит в себе state (суперсостояние), состояния, state transition (переходы), начальное (Start state) и конечное состояние (End State).

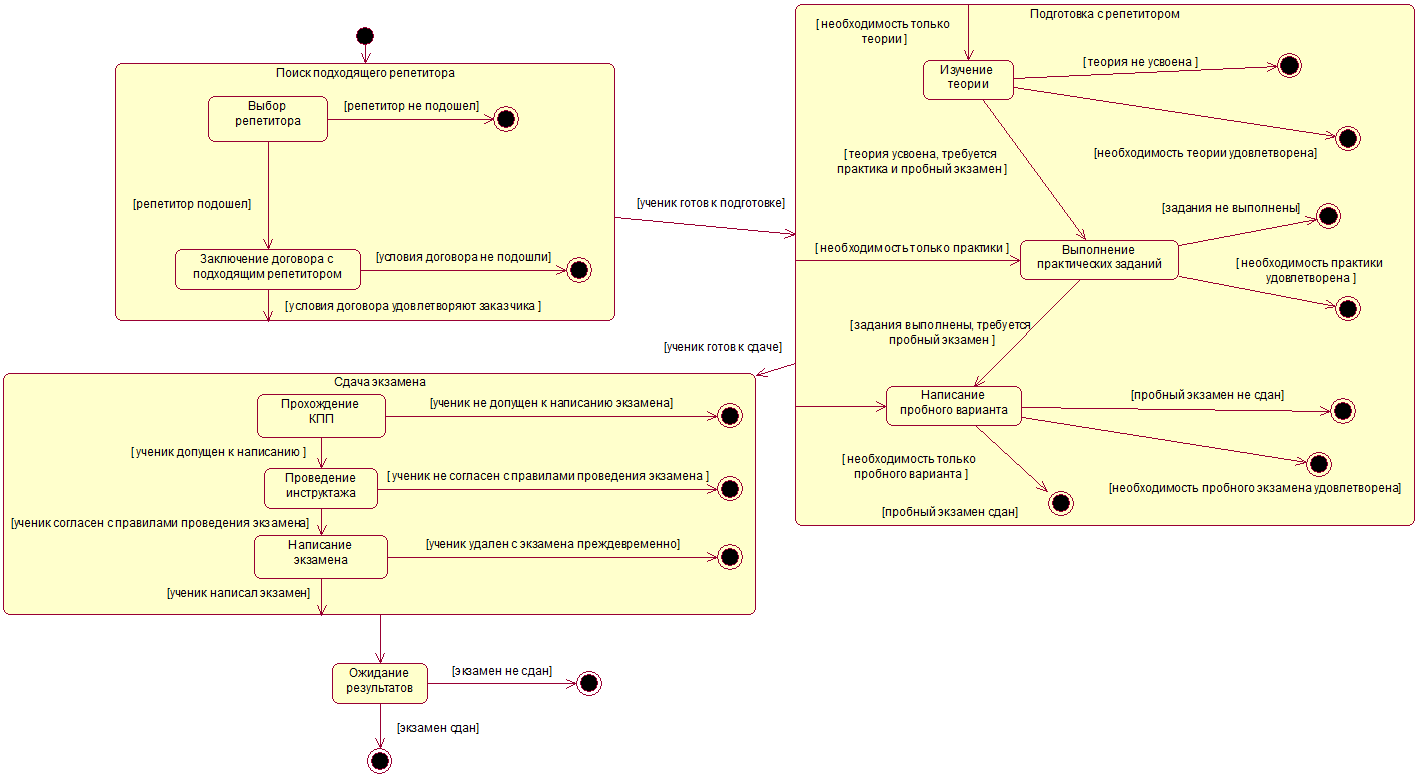


Рис. 16 – Скриншот диаграммы состояний

* 1. Диаграмма активности

Activity diagram (диаграмма активности). Это дальнейшее развитие диаграммы состояний. Фактически данный тип диаграмм может использоваться и для отражения состояний моделируемого объекта, однако, основное назначение Activity diagram в том, чтобы отражать бизнес-процессы объекта. Этот тип диаграмм позволяет показать не только последовательность процессов, но и ветвление и даже синхронизацию процессов. Окончательный вид диаграммы показан на рисунке 17. На диаграмме присутствуют дорожки (Swimlane), объекты состояния (Activity), начальное (Start state) и конечное состояние (End State).

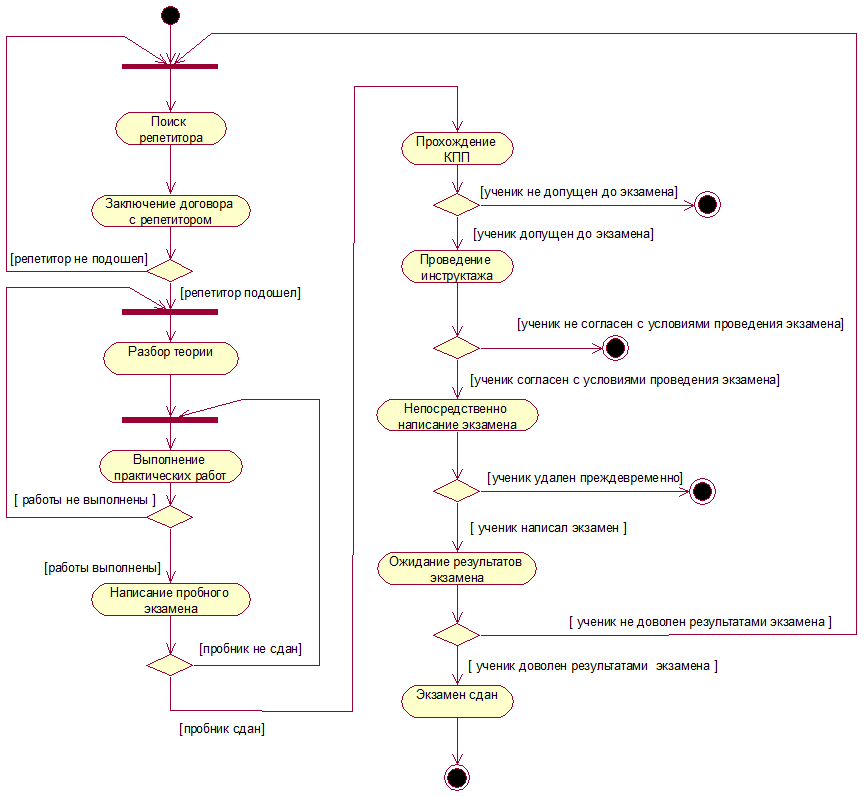


Рис. 17 – Скриншот диаграммы активности

* 1. Диаграмма классов

На данном этапе создается уже готовая диаграмма классов, а именно добавляются новые атрибуты и операции к классу. Происходит подробное описание операций и атрибутов, а именно указываются их типы.

В лабораторной работе 8 происходит описание связей между классами и добавляются ассоциации, далее выбирается множественность связей. В лабораторной работе 9 происходит изменение стереотипов классов. Стереотип позволяет указывать дополнительные особенности для разрабатываемо модели. Понятие стереотипа для этих элементов из уже заданных и представляет собой дополнительную классификацию элементов. В данной работе используется 4 вида стереотипов: Исполнитель (Actor), граница (boundary), сущность (entity) и управление (control). Окончательный вид диаграммы представлен на рисунке 18.

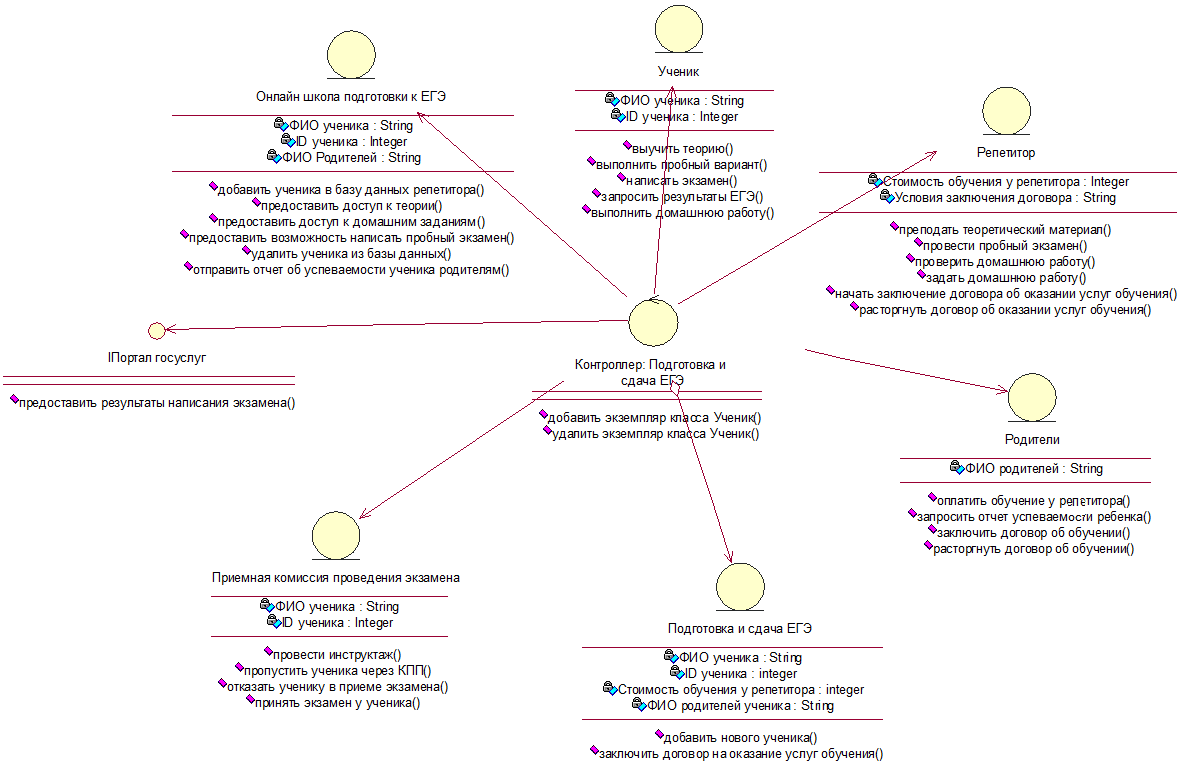


Рис. 18 – Скриншот диаграммы классов

* 1. Диаграмма бизнес-деятельности

Продолжая рассмотрение особенностей разработки диаграмм деятельности, следует отметить, что программа может быть успешно использована для выполнения проектов по моделированию бизнес-процессов. Наиболее подходящим типом диаграмм для визуального представления схем выполнения бизнес-процессов являются диаграммы деятельности, на которых дополнительно размещаются так называемые дорожки (Swimlane). Назначение дорожек состоит в том, чтобы указать зоны ответственности за выполнения отдельных деятельностей в рамках моделируемого бизнес-процесса. В качестве имен дорожек используются либо названия подразделений (департаментов) рассматриваемой компании, либо названия отдельных должностей сотрудников тех или иных подразделений.

Проекты по моделированию бизнес-процессов могут выполняться либо с целью реорганизации или реинжиниринга компании, либо с целью документирования бизнес-процессов. Особенности данных проектов заключаются в том, что в обоих случаях необходимо построить модели бизнес-процессов некоторой существующей компании. Три основные дорожки можно увидеть на рисунках 19-21.

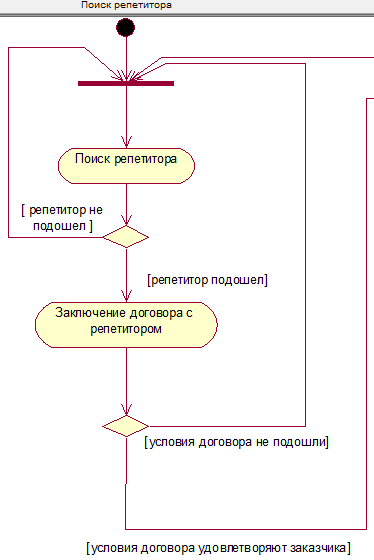


Рис. 19 – Скриншот дорожки «Поиск репетитора» в диаграмме бизнес-деятельности

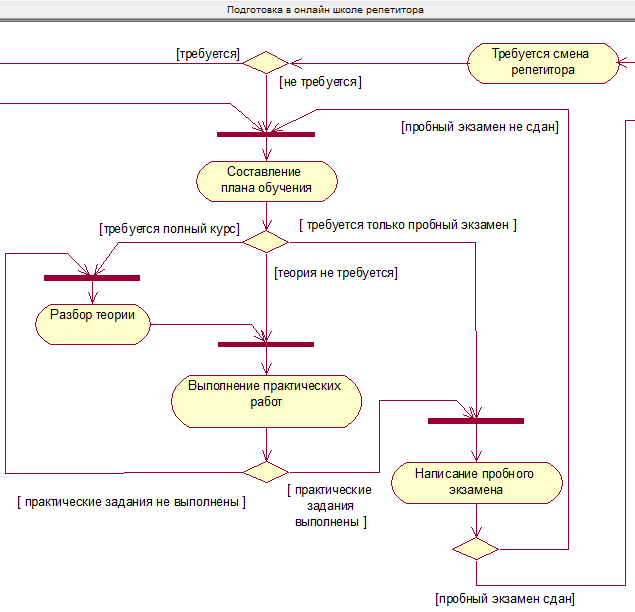


Рис. 20 – Скриншот дорожки «Подготовка в онлайн школе репетитора» в диаграмме бизнес-деятельности

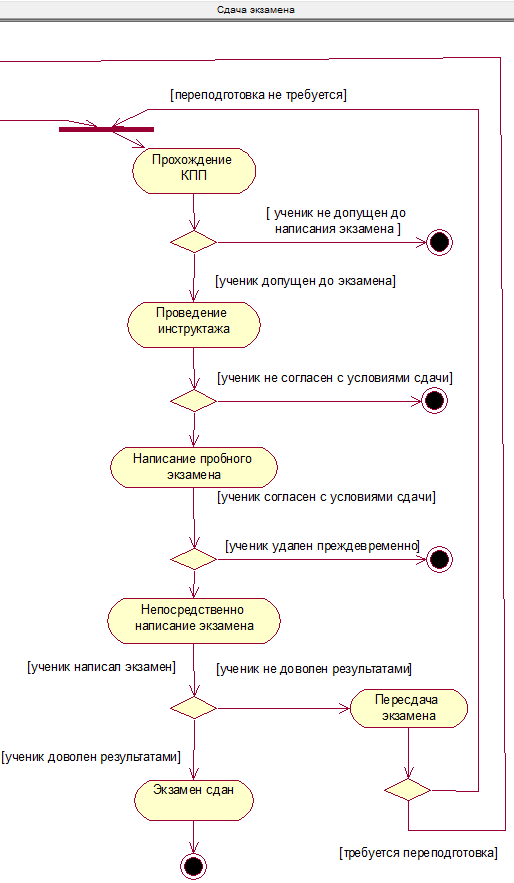


Рис. 21 – Скриншот дорожки «Сдача экзамена» в диаграмме бизнес-деятельности

* 1. Диаграмма компонентов

Component diagram (диаграммы компонентов). Этот тип диаграмм предназначен для распределения классов и объектов по компонентам при физическом проектировании системы. Часто данный тип диаграмм называют диаграммами модулей. Диаграмма компонентов представлена на рисунке 22.

Диаграмма компонентов, в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

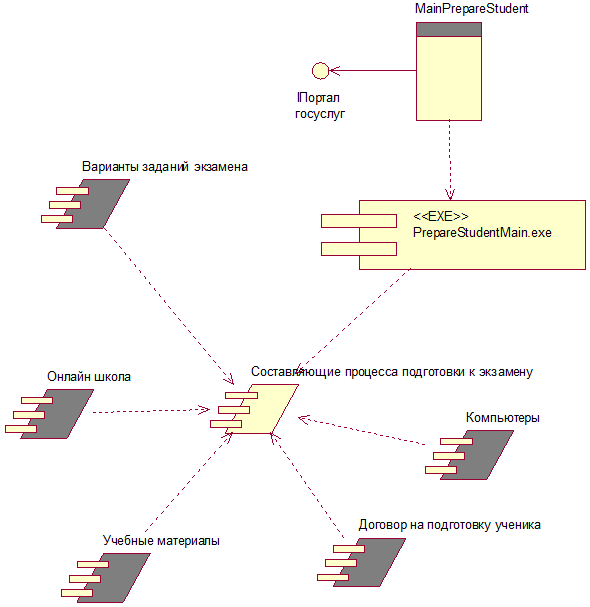


Рис. 22 – Скриншот диаграммы компонентов системы

* 1. Диаграмма развертывания

Диаграмма развертывания является второй составной частью физического представления модели и разрабатывается, как правило, для территориально распределенных систем. Для разработки диаграмм компонентов в браузере проекта предназначено отдельное представление развертывания (Deployment View), в котором уже содержится диаграмма развертывания с пустым содержанием и без собственного имени.

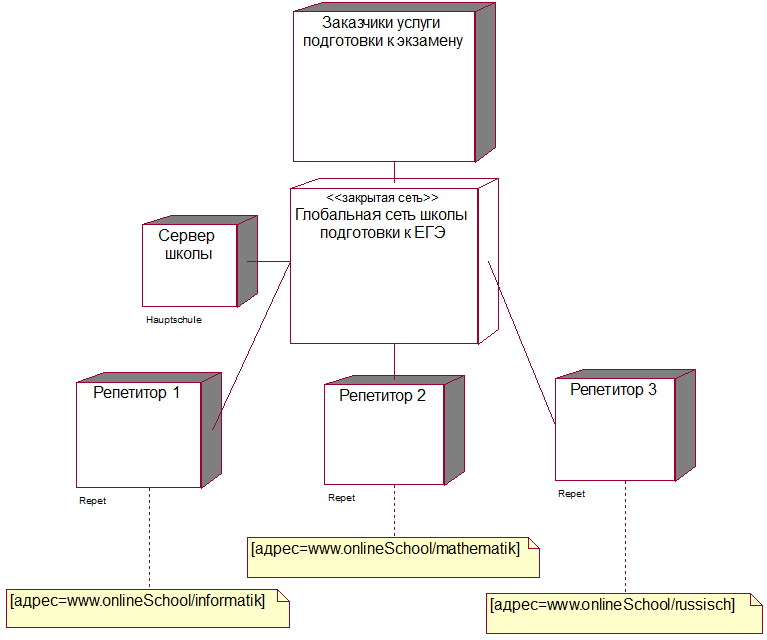


Рис. 23 – Скриншот диаграммы развертывания системы

Заключение

В результате выполнения практических работ данного курса были получены практические и теоретические навыки проектирования и построения информационных систем, развиты умения системно мыслить и анализировать представленную сферу с использованием нотаций IDEF0, DFD, IDEF3, а также языка UML.

Была выполнена основная цель – разработать и внедрить информационную систему подготовки и сдачи ЕГЭ.

Список использованной литературы

1. Боггс У. Боггс М. UML и Rational Rose [Текст]: учеб. Пособие/У.Боггс, М. Боггс:Лори, 2004, - 510с.
2. Кватрани, Т. Rational Rose 2000 и UML [Текст]: учеб. Пособие/ Т. Кватрани: ДМК Пресс, 2001, - 176с.
3. Ларман, К. Применение UML и шаблонов проектирования [Текст]: учеб. Пособие/ К. Ларман: Вильямс, 2002. – 624с.
4. Рамбо Дж., Блаха М.. UML 2.0. Объектно-ориентированное моделирование и разработка. [Текст]: учеб. Пособие/ Дж. Рамбо, М. Блаха: Питер, 2004. – 544с.
5. Трофимов, С. А. - CASE-технологии. Практическая работа в Rational Rose [Текст]/ С. А. Трофимов: Бином-Пресс, 2002. – 288с.
6. Статья С. А. Трофимов UML диаграммы в Rational Rose [Электронный ресурс]. - http://www.caseclub.ru/articles/rose2.html