

Министерство сельского хозяйства Российской Федерации  
Департамент научно-технологической политики и образования  
*Федеральное государственное бюджетное образовательное учреждение  
высшего образования*  
**«Красноярский государственный аграрный университет»**

УТВЕРЖДАЮ:  
Директор института  
\_\_\_\_\_ Шапорова З.Е.  
«21» \_\_\_\_\_ марта 2023 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
(текущего оценивания /промежуточной аттестации)

Институт экономики и управления АПК  
Кафедра Информационные технологии и математическое обеспечение  
информационных систем  
Наименование и код ОПОП: **09.02.07** «Информационные системы и  
программирование»

Дисциплина:  
МОДЕЛИРОВАНИЕ И АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Красноярск 2023

Составители: Шевцова Л.Н., к.с-х.н., доцент  
(ФИО, ученая степень, ученое звание)

\_\_\_\_\_ «20» марта 2023г.

ФОС разработан в соответствии с рабочей программой дисциплины «Моделирование и анализ программного обеспечения» по направлению 09.02.07 «Информационные системы и программирование»

ФОС обсужден на заседании кафедры «Информационные технологии и математическое обеспечение информационных систем»

\_\_\_\_\_ протокол № 7 «20» марта 2023г.

Зав. кафедрой ИТ и МОИС Бронов С.А., д.т.н., доцент

\_\_\_\_\_ «20» » марта 2023г.

ФОС принят методической комиссией института  
Экономики и управления АПК протокол № 7 «21» марта 2023г.

Председатель методической комиссии

Рожкова А.В., ст. преподаватель \_\_\_\_\_ «21» марта 2023г

## Содержание

1. Цель и задачи фонда оценочных средств.....	4
2. Нормативные документы.....	4
3. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины .....	5
4. Показатели и критерии оценивания компетенций .....	6
5. Фонд оценочных средств.....	8
5.1 Фонд оценочных средств для текущего контроля .....	8
5.1.1 Оценочное средство – Практическая работа. Критерии оценивания. ....	8
5.1.2. Оценочное средство: Защита практических работ. Критерии оценивания.....	16
5.2.Фонд оценочных средств для промежуточного контроля.....	17
6. Учебно-методическое и информационное обеспечение дисциплины .....	19
6.1. Основная литература .....	19
6.2. Дополнительная литература.....	19
6.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет».....	19
(далее – сеть «Интернет») .....	19
6.4. Программное обеспечение .....	19
Приложение А. Варианты тестовых заданий по дисциплине .....	21

## **1. Цель и задачи фонда оценочных средств**

Целью создания ФОС дисциплины является установление соответствия учебных достижений запланированным результатам обучения и требованиям образовательных программ, рабочей программы дисциплины.

ФОС по дисциплине решает задачи:

- контроль и управление процессом приобретения студентами необходимых знаний, умений, навыков и уровня сформированности компетенций, определённых в ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование»;
- контроль (с помощью набора оценочных средств) и управление (с помощью элементов обратной связи) достижением целей реализации ОПОП, определенных в виде набора профессиональных компетенций выпускников;
- обеспечение соответствия результатов обучения задачам будущей профессиональной деятельности через совершенствование традиционных и внедрение инновационных методов обучения в образовательный процесс Университета

### **Назначение фонда оценочных средств:**

используется для оперативного и регулярного управления учебной деятельностью (в том числе самостоятельной) студентов. В условиях рейтинговой системы контроля результаты текущего оценивания студента используются как показатель его текущего рейтинга. А также предназначен для оценки степени достижения запланированных результатов обучения по завершению изучения модулей дисциплины «Моделирование и анализ программного обеспечения» в установленной учебным планом форме: 5 семестр – зачет с оценкой.

## **2. Нормативные документы**

ФОС разработан на основе ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование», рабочей программы дисциплины «Моделирование и анализ программного обеспечения».

### 3. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

Таблица 3.1 -Формы контроля формирования компетенций.

Компетенция	Этап формирования компетенции	Образовательные технологии	Тип контроля	Форма контроля
ПК -3.1. Осуществлять ревьюирование программного кода в соответствии с технической документацией	Теоретический (информационный)	Лекции, практическая работа	текущий	Зачет с оценкой
	Практико-ориентированный	Практические работы	текущий	Выполнение и защита практических работ
	оценочный	аттестация	промежуточный	Зачет с оценкой
ПК -3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма	Теоретический (информационный)	Лекции, практическая работа	текущий	Зачет с оценкой
	Практико-ориентированный	Практические работы	текущий	Выполнение и защита практических работ
	оценочный	аттестация	промежуточный	Зачет с оценкой
ПК-3.4. Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием	Теоретический (информационный)	Лекции	текущий	Зачет с оценкой
	Практико-ориентированный	Практические работы	текущий	Выполнение и защита лабораторных работ
	оценочный	аттестация	промежуточный	Зачет с оценкой

#### 4. Показатели и критерии оценивания компетенций

Таблица 4.1 – Показатели и критерии оценки результатов обучения

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
<b>Выполнение анализа и моделирования программных продуктов</b>		
<p>ПК 3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией.</p>	<p>Оценка «отлично» - в системе контроля версий выбрана верная версия проекта, проанализированы архитектура и алгоритм проекта на соответствие спецификации, предложен альтернативный вариант решения поставленной задачи в виде описания и/или UML диаграмм; результаты ревью сохранены в системе контроля версий.</p> <p>Оценка «хорошо» - в системе контроля версий выбрана верная версия проекта, проанализированы архитектура или алгоритм проекта на соответствие спецификации, предложен альтернативный вариант решения поставленной задачи в виде описания или UML диаграмм; результаты ревью сохранены в системе контроля версий.</p> <p>Оценка «удовлетворительно» - в системе контроля версий выбрана верная версия проекта, проанализированы архитектура или алгоритм проекта на соответствие спецификации; результаты ревью в виде описания сохранены в системе контроля версий.</p>	<p>Зачет с оценкой: практическое задание по ревьюированию предложенного программного кода на соответствие требованиям технического задания на проект. Защита отчетов по практическим и лабораторным работам</p> <p>Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной</p>
<p>ПК 3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.</p>	<p>Оценка «отлично» - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; программный код проанализирован на соответствие алгоритму; проведена оптимизация и подтверждено повышение качества программного кода; результаты сохранены в системе контроля версий.</p> <p>Оценка «хорошо» - определены качественные характеристики программного кода с помощью</p>	<p>Зачет с оценкой: практическое задание по оценке качества предложенного программного кода, поиску некачественного программного кода, его анализу и выявлению ошибок.</p> <p>Защита отчетов по практическим и</p>

	инструментальных средств; выявлены фрагменты некачественного кода; программный код проанализирован на соответствие алгоритму; проведена оптимизация и оценка качества программного кода. Оценка «удовлетворительно» - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; программный код проанализирован на соответствие алгоритму; проведена оценка качества программного кода.	лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной
ПК 3.4 Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.	Оценка «отлично» - указан набор возможных средств выполнения поставленной задачи, выполнен анализ достоинств и недостатков не менее, чем трех программных продуктов и средств разработки, обоснован выбор одного (возможно, двух и более) из них. Оценка «хорошо» - выполнен анализ достоинств и недостатков двух программных продуктов и средств разработки, обоснован выбор одного из них. Оценка «удовлетворительно» - выполнен анализ достоинств и недостатков программных продуктов и средств разработки, обоснован выбор одного (возможно, двух и более) из них.	Зачет с оценкой: практическое задание по обоснованию выбора программных продуктов и средств разработки для решения предложенной задачи. Защита отчетов по практическим и лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной

Таблица 4.2. Шкала оценивания

Показатель оценки результатов обучения	Шкала оценивания
Пороговый уровень	70-75 балла
Продвинутый уровень	76-86 баллов
Высокий уровень	87-100 баллов

## 5. Фонд оценочных средств

### 5.1 Фонд оценочных средств для текущего контроля

Текущий контроль используется для оперативного и регулярного управления учебной деятельностью (в том числе самостоятельной) студентов. В условиях рейтинговой системы контроля результаты текущего оценивания студента используются как показатель его текущего рейтинга. Текущий контроль успеваемости студентов включает в себя: опрос-защита выполненной практической работы.

#### 5.1.1 Оценочное средство – Практическая работа. Критерии оценивания.

##### Пример практической работы

**Тема.** Разработка и создание UML диаграмм.

**Цель.** Создание в среде ArgoUML диаграмм UML.

##### Ход работы

1. Ознакомиться с теоретической частью.
2. Выполнить практическое задание.
3. Ответить на вопросы.

##### Теоретическая часть

После запуска ArgoUML появится окно. Окно ArgoUML состоит из следующих компонентов: меню, панели инструментов и четырех панелей. Вверху слева находится Explorer, затем идет панель редактирования, затем на нижней панель деталей и To-Do панель.

В панели Explorer содержится список всех классов, интерфейсов и типов данных модели в виде дерева. На панели редактирования можно редактировать диаграммы.

В панели To-Do отображаются элементы моделей.

Описание пунктов меню:

**Файл (File)** — позволяет создавать новые проекты, сохранять и открывать проекты, импортировать, импортировать источники из другого расположения, скачать и сохранить модель из/в базу данных, печать модель, сохранить модель, сохранять конфигурацию модели и выход из программы.

**Редактировать (Edit)** -позволяет выбрать один или несколько UML элементов в диаграмме, а также повторить действия, перемещать элементы из диаграммы и так далее

**Вид (View)** -позволяет переключаться между диаграммами, искать артефакты в модели, масштабировать диаграмму, выберите особенное представление диаграмм и т.п.

Создать диаграмму (Create Diagram) -позволяет создавать любую диаграмму из семи UML диаграмм, поддерживаемых ARGOUML (class, use case, state, activity, collaboration, deployment и sequence).

**Расставить (Arrange)** – позволяет выровнять, распределить, упорядочить артефакты в диаграмме.

**Генерация кода (Generation)** -позволяет сгенерировать Java код для отдельных или всех классов.

**Критика (Critique)** –переключать авторецензирование, устанавливать уровень важности проблемы дизайна и дизайн целей и увидеть критических изменения. Инструменты (Tools) Помощь (Help)

##### Практическая часть

**Задание 1.** Для создания диаграммы вариантов использования, выполните следующие действия:

- 1) Дважды щелкните на значке untitledModel.
- 2) Нажмите на значок Диаграмма вариантов использования в проводнике, чтобы открыть диаграмму Вариантов использования.
- 3) С помощью кнопки Use Case (вариант использования) на панели инструментов поместите на диаграмму новый вариант использования.



4) Назовите этот новый вариант «Ввести новый заказ», для этого вам нужно дважды нажать на варианте использования и ввести имя или активировать вкладку Свойства To-Do панели в поле Имя: введите имя.

5) Повторите шаги 3 и 4, чтобы разместить на диаграмме остальные варианты.

6) С помощью кнопки Actor (Действующее лицо) инструментов поместите на диаграмму новое действующее лицо

7) Назовите его «Продавец», действия аналогичные вариантам использования.

8) Повторите шаги 6 и 7, поставив на диаграмме остальных актеров.

**Задание 2.** Для указания абстрактного варианта использования, выполните следующие действия:

1) ЛКМ нажмите на вариант использования «Отклонить заказ» на диаграмме.

2) В панели Свойств в области modifiers выберите контрольный переключатель isAbstract (Абстрактный), чтобы сделать этот вариант использования абстрактным.

**Задание 3.** Для добавления ассоциации выполните следующие действия:

1) Нажав на кнопку Association (Ассоциация), выберите UniAssociation (Однонаправленная Ассоциация), изобразите ассоциацию между актером Продавец и вариантом использования «Ввести новый заказ».

2) Повторите этот шаг, чтобы разместить на диаграмме остальные ассоциации.

**Задание 4.** Для добавления связи расширения выполните следующие действия:

Нажав на кнопку Extend на панели инструментов нарисуйте связь между вариантом использования «Отклонить заказы» и вариантом использования «Оформить заказ». Стрелка должна быть протянута от первого варианта использования ко второму. Связь расширения означает, что вариант использования «Отклонить заказ», при необходимости, дополняет функциональные возможности варианта использования «Оформить заказ». Диаграмма должна иметь вид, как на рисунке 1.

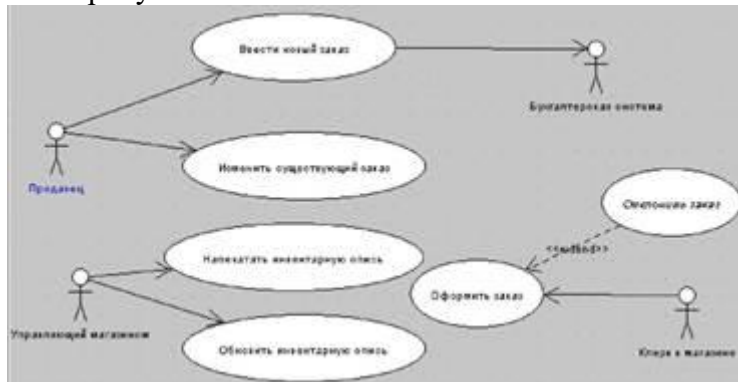


Рисунок 1 – Диаграмма вариантов использования для системы обработки заказов

**Задание 5.** Создать главную диаграмму классов.

1) Дважды щелкните на значке UntitledModel, отобразится содержимое пакета. Выберите Диаграмма классов. С помощью кнопки Новый пакет и панели инструментов разместите на диаграмме новый пакет. Во вкладке Свойства, укажите имя пакета Entities (Сущность).

2) Создайте пакеты Boundaries (Границы) и Control (Управление).



Рисунок 2 – Главная диаграмма классов системы обработки заказов

**Задание 6.** Создание диаграммы классов для сценария «Ввести новый заказ» со всеми классами

- 1) В меню Создать диаграмму, выберите Диаграмма классов.
- 2) Во вкладке Свойства введите имя для новой диаграммы класса Add New Order (Ввод нового заказа).
- 3) Щелкните в браузере на этой диаграмме, чтобы открыть ее.
- 4) Создайте в окне диаграммы классов классы OrderOptions (Выбор заказа), OrderDetail (Детали заказа), Order (Заказ), OrderMgr (Менеджер заказов) и TransactionMgr (Менеджер транзакций). Имя класса вводится во вкладке Свойства.
- 5) Добавьте операции, указанные на диаграмме классов (рис.3). Чтобы сделать это, выберите соответствующий класс, нажмите на кнопку Добавить операцию, здесь вы можете ввести видимость операции. Диаграмма классов должен выглядеть как на рисунке 3.

**Задание 7.** Добавление стереотипов в классы

- 1) Нажмите на класс OrderOptions.
- 2) Введите в поле имя класса стереотип <<Boundary>>.
- 3) В раскрывающемся списке стереотипов теперь будет стереотип Boundary .  
Укажите его.
- 4) Нажмите с помощью мыши на классе OrderDetail.
- 5) Из раскрывающегося списка Стереотип выберите стереотип Boundary.
- 6) Свяжите классы OrderMgr и TransactionMgr со стереотипом Control, а класс Order – со стереотипом Entity. Теперь диаграмма классов должна выглядеть как на рисунке 3.

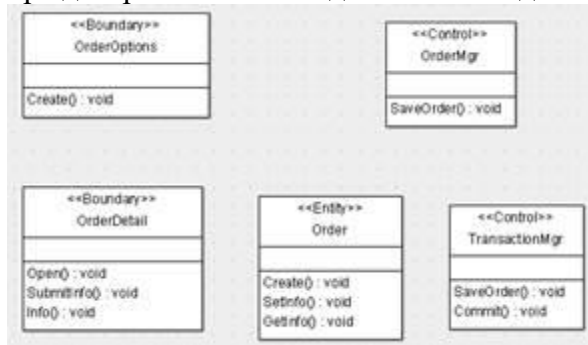


Рисунок 3 – Стереотипы классов для варианта использования Ввести новый заказ

**Задание 8.** Объединение классов в пакеты

- 1) Перетащите в браузере класс OrderDetail на пакет Boundaries.
- 2) Перетащите класс OrderOptions на пакет Boundaries.
- 3) Перетащить классы OrderMgr и TransactionMgr на пакет Control.
- 4) Перетащите класс Order на пакет Entities.

**Задание 9.** Добавить диаграмму классов к каждому пакету

- 1) Дважды щелкните левой кнопкой на пакете Boundaries в окне диаграмм.
- 2) В появившемся окне выберите Да.
- 3) Во вкладка Свойства, введите имя для новой диаграммы – MainB.
- 4) Щелкните на диаграмму, чтобы открыть его.
- 5) В браузере выделите класс OrderOptions (потом OrderDetail), щелкните ПКМ и выберите опцию Добавить в диаграмму, а затем нажмите мышью в поле диаграммы.
- 6) Классы будут отображаться на диаграмме.
- 7) Выполните эти же шаги, чтобы поместить классы OrderMgr и TransactionMgr на главную диаграмму классов пакета Control (MainC).
- 8) Выполните эти же действия, чтобы поместить класс Order на главную диаграмму классов пакета Entities (MainE).

**Задание 10.** Для того чтобы создать диаграммы кооперации выполните следующие действия:

- 1) Нажмите на значок untitledModel в браузере.
- 2) Из меню Создать диаграмму, чтобы выберите Диаграмму коопераций.

3) Назовите эту диаграмму Ввод заказа.

4) Нажмите на нее, чтобы открыть.

**Задание 11.** Для добавления объектов к диаграмме коопераций, выполните следующие действия:

1) Перетащите с диаграммы вариантов использования объект Продавец на рабочую область.

2) На панели инструментов нажмите кнопку ClassifierRole.

3) Щелкните в любом месте диаграммы, чтобы разместить там новый объект.

4) Назовите объект «Order Options Form».

5) Помести на диаграмме остальные объекты: Order Detail Form, Order Manager, Transaction Manager, Order #1234.

**Задание 12.** Добавления сообщений на диаграмму коопераций:


1) На панели инструментов нажмите кнопку Новая ассоциация.

2) Свяжите Продавец с объектом Order Options Form.

3) Соедините остальные объекты.

4) Нажмите на связь между Продавцом и Order Options Form. На панели инструментов нажмите кнопку Добавить сообщение. Нажмите на связь между Продавцом и Order Options Form. Выбрав сообщение, введите его имя Create ().

5) Поместите на диаграмме оставшиеся сообщения: Open (), SubmitInfo (), Save (), SaveOrder(), SetInfo (), GetInfo ().

6) Нажмите на объекте для добавления к нему сообщения рефлексии. По бокам объекта будут показаны значки, среди которых необходимо выбрать .

7) Нажмите на связь рефлексии Transaction Manager, чтобы ввести сообщение. На панели инструментов нажмите кнопку Добавить сообщение. Назовите новое сообщение Commit () (Сохранить информацию о заказе в базе данных) (рисунок 4).

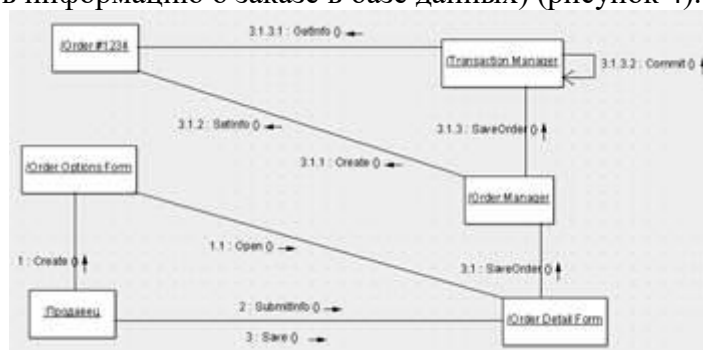


Рисунок 4 – Диаграмма коопераций

**Задание 13.** Для создания схемы последовательностей, выполните следующие действия: щелкните ПКМ логическое представление браузера. В открывшемся меню нажмите Create Diagram → Диаграмма последовательности. Назовите новую диаграмму «Ввод заказ». Дважды щелкните на нем, чтобы открыть

**Задание 14.** Для добавления на диаграмму актера и объектов, выполните следующие действия:

1) Перетащите актера Продавец из браузера на диаграмму. Для панели панель инструментов, нажмите кнопку New Classifier Role. Имя объекта Order Options Form – Выбор варианта заказа.

2) Таким же образом добавьте все элементы, описанные в пункте 11.5 (рисунок 5).



Рисунок 5 – Диаграмма последовательности

**Задание 15.** Для добавления сообщений на диаграмму, выполните следующие действия:

1) На панели инструментов нажмите кнопку Действие отправки, перетащите от линии жизни актера к линии жизни объекта Order Options Form. В ArgoUML в данном виде диаграмм сообщения автоматически не нумеруются, поэтому нумерация переносится из диаграммы кооперации. Выбрав сообщение, введите имя 1: Create ().

2) Добавьте оставшиеся сообщения, описанные в пункте 12.5

**Задание 16.** Сопоставление объектов с классами

1) Нажмите на объекте Order Options Form.

2) В панели Свойств раскройте список База.

3) В открывшемся списке нажмите на кнопку с плюсом..

4) В открывшемся окне выберите класс OrderOptions и нажмите на кнопку со стрелкой для добавления в список.

5) Нажмите кнопку ОК.

6) Сопоставьте остальные объекты с классами: Order Detail Form - OrderDetail, Order Manager – OrderMgr, Transaction Manager - TransactionMgr, Order #1234 - Order.

**Задание 17.** Соотношение сообщений с операциями.

1) Нажмите на сообщении 1: Create () заказ.

2) В окне Свойств дважды щелкните в поле Действие. В открывшемся окне введите Create ().

3) Сопоставьте остальные сообщения.

**Задание 18.** Добавление нового класса

1) Найдите в дереве (браузере) диаграмму классов «Add New Order». Нажмите на нее дважды, чтобы открыть.

2) Поместите на этой диаграмме новый класс OrderItem (Позиция заказа). Назначьте этому классу стереотип Entity.

3) В дереве перетащите класса в пакет Entities.

**Задание 19.** Добавление атрибутов

1) Щелкните ПКМ на классе Order (Заказ).

2) В открывшемся меню выберите Добавить  Новый атрибут.

3) Введите новый атрибут OrderNumber : Integer и нажмите Enter. Тип атрибута можно выбрать из раскрывающегося списка в поле Тип или ввести вручную.

4) Введите следующий атрибут CustomerName : String.

5) Добавьте атрибуты OrderDate : Date и OrderFillDate : Date.

6) Щелкните ПКМ на классе OrderItem.

7) В открывшемся меню выберите Добавить  Новый атрибут.

8) Введите новый атрибут ITEMID : Integer.

9) Введите следующий атрибут ItemDescription : String.

**Задание 20.** Добавления операций к классу OrderItem

1) Щелкните ПКМ на классе OrderItem.

2) В открывшемся меню выберите Добавить  Новая операция.

3) Введите новую операцию Create.

4) Введите следующую операцию SetInfo.

5) Введите следующую операцию GetInfo.

**Задание 21.** Подробное описание операций с помощью диаграммы класса

1) Нажмите на класс Order, выбрав его таким образом.

2) Нажмите на этот класс в разделе операций.

3) Отредактируйте операцию Create() так, чтобы она выглядела таким образом:

Create () : Boolean. Для этого дважды нажмите на операцию Create в разделе операций класса.

4) Отредактируйте операцию SetInfo(), чтобы она выглядела следующим образом:

SetInfo(OrderNum : Integer, Customer : String, OrderDate : Date, FillDate : Date) : Boolean

5) Отредактируйте операцию GetInfo (), чтобы она выглядела следующим образом:

GetInfo () : String

**Задание 22.** Подробное описание операций остальных классов

1) С помощью браузера или диаграмм, введите следующую сигнатуру операций класса OrderDetail: Open () : Boolean, SubmitInfo() : Boolean, Save() : Boolean.

2) С помощью браузера или диаграмм, введите следующую сигнатуру операций класса OrderOptions : Create () : Boolean .

3) С помощью браузера или диаграмм, следующую сигнатуру операций класса OrderMgr: SaveOrder(ORDERID: Integer): Boolean.

4) С помощью браузера или диаграмм, введите следующую сигнатуру операций класса TransactionMgr: SaveOrder (ORDERID: Integer): Boolean, Commit () : Integer.

**Задание 23.** Добавление отношения между классами.

Чтобы найти отношения, были изучены диаграммы последовательности. Все классы, которые там взаимодействуют, требовали определения соответствующих связей на диаграммах классов. После выявления связей их добавили в модель. Добавьте отношения, как показано на рисунке 6.

Множественность указывается выбором из контекстного меню связи опции Multiplicity. После добавления связей диаграмма классов должна выглядеть следующим образом.

**Задание 24.** Создание диаграммы состояний.

1) Найдите в браузере класс Order.

2) Выделите этот класс.

3) В меню Создать диаграмму выберите Диаграмма состояний.

**Задание 25.** Добавление начального и конечного состояний.

1) На панели инструментов нажмите кнопку New Initial (Начальное состояние).

Поместите это состояние на диаграмму.

2) На панели инструментов нажмите Конечное состояние. Поместите на диаграмму.

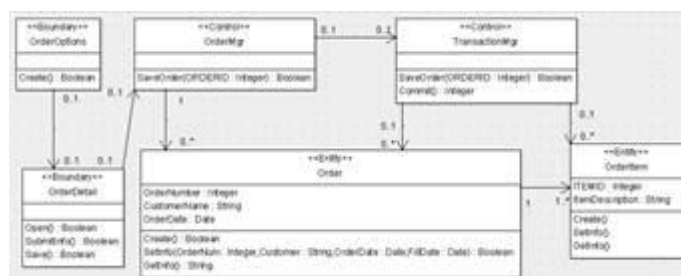


Рисунок 6 – Диаграмма классов Add New Order

**Задание 26.** Добавление суперсостояния.

1) На панели инструментов нажмите кнопку Композитное состояние.

2) Поместите это состояние на диаграмму.

**Задание 27.** Добавление оставшихся состояний.

1) На панели инструментов нажмите кнопку Простое состояние. Поместите его на диаграмму. Назовите состояние Отменен.

2) На панели инструментов нажмите кнопку Простое состояние. Поместите его на диаграмму. Назовите состояние Выполнен.

3) На панели инструментов нажмите кнопку Простое состояние. Поместите это состояние внутрь суперсостояния. Назовите состояние Инициализация.

4) На панели инструментов нажмите кнопку Простое состояние. Поместите это состояние внутрь суперсостояния. Назовите состояние Сборка заказа.

#### **Задание 28.** Подробное описание состояний

Для добавления в состояния действий на входе, выходе и деятельности исполнения на вкладке Свойства состояния выберите соответственно: действие при входе, действие при выходе, деятельность выполнения.

#### **Задание 29.** Добавление переходов

1) На панели инструментов нажмите кнопку New Transition (Переход).

2) Нажмите на начальной точке. Проведите линию перехода к состоянию

Инициализация.

3) Повторить предыдущие шаги для добавления оставшихся переходов.

#### **Задание 30.** Подробное описание переходов

1) Нажмите на переход от состояния Инициализация к состоянию Сборка заказа, открывая окно его свойств.

2) В поле Имя введите Выполнить заказ.

3) Повторите этапы, добавив событие Отменить заказ для переключения между суперсостоянием и состоянием Отменен.

4) Нажмите на переход от состояния Сборка заказа к состоянию Заполнены (Выполнено), открывая окно его свойств.

5) В поле Имя введите фразу Добавить Порядок пункта (Добавить в заказ на новое место).

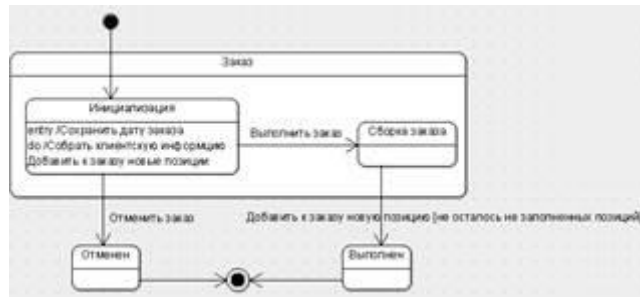


Рисунок 7 – Диаграмма состояний

#### **Задание 31.** Добавление компонентов и изображение зависимостей

1) Выполните команду Создать диаграмму □ Диаграмма развертывания.

2) Нажмите на название диаграммы в браузере. Введите имя Entities.

3) На панели инструментов нажмите кнопку New Component (Компонент).

4) Щелкните в поле диаграммы. Введите имя компонента.

5) Добавить все компоненты.

6) На панели инструментов нажмите кнопку New Dependency (Зависимость).

Проведите необходимые зависимости. Аналогично создайте диаграммы Control и Boundaries.

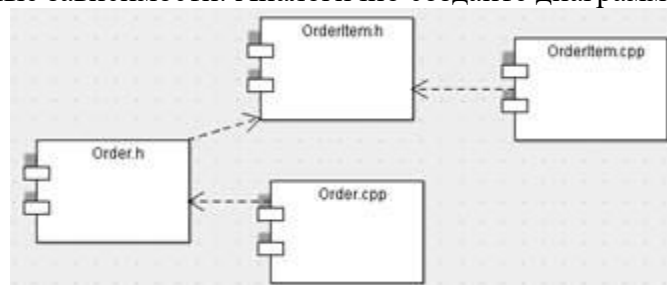


Рисунок 8 – Диаграмма компонентов Entities

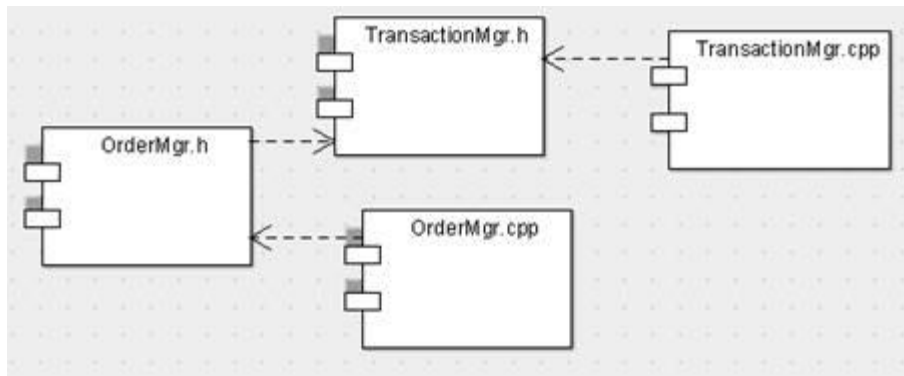


Рисунок 9 – Диаграмма компонентов Control

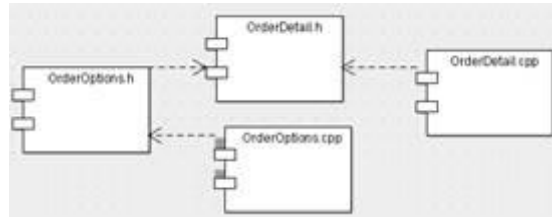


Рисунок 10 – Диаграмма компонентов Boundaries

**Задание 32.** Создание диаграммы компонентов системы

- 1) Выполните команду Создать диаграмму  Диаграмма развертывания.
- 2) Нажмите на название диаграммы Диаграмма компонентов системы.
- 3) Перетащите из дерева на рабочую область диаграммы компоненты OrderDetail.h, OrderOptions.h, OrderMgr.h, TransactionMgr.h, Order.h, OrderItem.h.
- 4) Создайте дополнительные компоненты OrderClient.exe и OrderServer.exe.
- 5) Создайте зависимости между компонентами, которые не были созданы автоматически.

**Задание 33.** Добавление узлов на диаграмму размещения

- 1) Выберите пункт меню Создать диаграмму  Диаграмма развертывания.
- 2) На панели инструментов нажмите кнопку New Node.
- 3) Введите имя узла «Сервер базы данных».
- 4) Добавьте оставшиеся узлы.

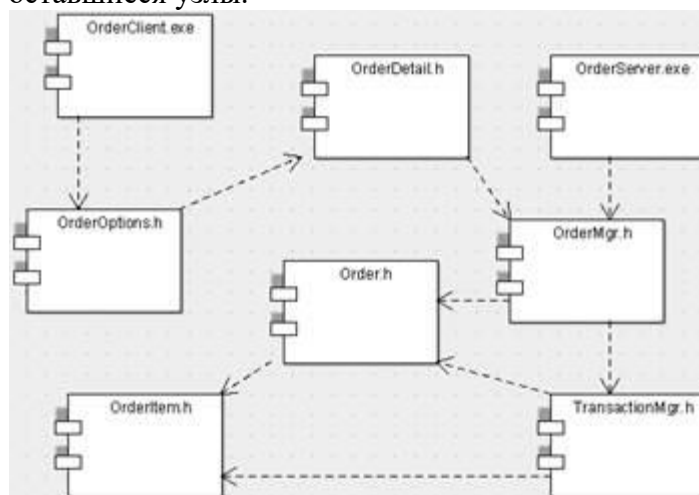


Рисунок 12 – Диаграмма компонентов системы

**Задание 34.** Добавление связей

- 1) На панели инструментов нажмите кнопку New Link.
- 2) Нажмите на «Сервер базы данных».
- 3) Проведите линию до узла «Сервер приложений».

4) Добавьте остальные связи.



Рисунок 13 – Диаграмма размещений

**Задание 35.** Добавление процессов и операций

- 1) Щелкните ПКМ на «Сервер приложений». Из раскрывающегося списка выберите Apply Stereotypes  «process».
- 2) Выполните аналогичные действия для объектов Рабочая станция 1 и Рабочая станция 2.
- 3) Выделите «Сервер приложений». На панели Свойств в поле Операции нажмите ПКМ и выберите Новая операция. В поле Имя введите OrderServerExe.
- 4) Добавьте остальные процессы: Рабочая станция 1 – OrderClientExe, Рабочая станция 2 – ATMClientExe.

### Вопросы

1. Для чего используется язык UML?
2. Опишите основные диаграммы UML.

### Критерии оценивания:

Правильное выполнение всех заданий – 1 балл;

Итого за выполнение практических работ в течение семестра студент может получить 7 баллов.

### 5.1.2. Оценочное средство: Защита практических работ. Критерии оценивания

Обязательными видами текущей аттестации является выполнение всех практических заданий.

*Защита практических работ* проводится следующим образом: преподаватель выбирает любой пункт задания из выполненных практических работ по модулям и просит сделать какие-то конкретные пункты заданий под его непосредственным контролем. Если студент повторил в присутствии преподавателя и выполнил правильно задания (не более 4-х пунктов) – получает баллы согласно рейтинг-плану по дисциплине. Если студент не смог выполнить задания – готовится, вспоминает, может проконсультироваться у однокурсников и преподавателя и повторить попытку сдачи.

**Оценивание студентов** производится в дискретные временные интервалы преподавателем(ями), ведущими лабораторно-практические занятия по дисциплине по следующим позициям: выполнение практических работ – 1 балл за каждое занятие и 1 балл посещение лекций; защита практических работ по модулям – 4 балла за ДМ 2, 14 баллов за ДМ3 и 36 баллов за ДМ4.

Общий рейтинг-план дисциплины приведен ниже:

#### Рейтинг-план

Дисциплинарные модули	Максимально возможный балл по видам работ		Итого баллов
	Текущая работа	Аттестация	



	Выполнение практической работы	Защита практической работы	Лекции	Зачет с оценкой	
ДМ1	-	-	1	-	1
ДМ2	1	4	1	-	6
ДМ3	2	14	2		18
ДМ4	4	36	5		45
Зачет с оценкой				30	30
Итого	7	54	9	30	100

## 5.2. Фонд оценочных средств для промежуточного контроля

Промежуточная аттестация по дисциплине (зачет с оценкой) проводится в виде выполнения одного из проектов, по предложенной ниже тематике. Все задания выполняются с использованием бесплатного ПО.

### Темы проектов

- 1 Составление модели IDEF 0.
- 2 Составление UML- диаграммы прецедентов
- 3.Разработка программы с использованием NetBeans
- 4.Создание приложения с использованием среды разработки Eclipse
- 5.Ревьюирование предложенного программного кода на соответствие требованиям технического задания на проект
- 6.Задание по оценке качества предложенного программного кода, поиску некачественного программного кода, его анализу
- 7.Задание по обоснованию выбора программных продуктов и средств разработки для решения предложенной задачи

### Критерии оценки результатов выполнения проекта

Оценка согласно рейтинг-плана, балл	Критерии оценки
0-12	Ставится, если тема контрольной работы не раскрыта, обнаруживается существенное непонимание проблемы или задание не выполнено.
13-18	Ставится, если имеются существенные отступления от требований к работам. В частности: тема освещена лишь частично; допущены фактические ошибки в при ответе на вопросы; допущены ошибки в стоимости ресурсов и построении графика проекта, во время защиты отсутствует вывод.
19-23	Ставится, если основные требования к работе и его защите выполнены, но при этом допущены недочёты. В частности, имеются неточности в расчетах стоимости ресурсов и бюджета проекта; на дополнительные вопросы при защите даны неполные ответы.
24-30	Ставится, если выполнены все требования к созданию и защите работы: обозначена проблема и обоснована её актуальность, сделан проект и правильно выполнены все расчеты по ресурсам и задачам, а также в целом по бюджету проекта, даны правильные ответы на дополнительные вопросы.

Всего за выполнение заданий обучающийся может набрать 0-30 баллов.

*Перечень вопросов для подготовки к зачету:*

1. Методы организации работы в команде разработчиков. Системы контроля версий.
2. Цели, задачи, этапы и объекты ревьюирования. Планирование ревьюирования.
3. Цели, корректность и направления анализа программных продуктов. Выбор критериев сравнения. Представление результатов сравнения.
4. Примеры сравнительного анализа программных продуктов.
5. Цели, задачи и методы исследования программного кода.
6. Механизмы и контроль внесения изменений в код.
7. Обратное проектирование. Анализ потоков данных. Дизассемблирование.
8. Предпроцессинг кода. Интеграция в IDE.
9. Валидация кода на стороне сервера и разработчика.
10. Совместимость и использование инструментов ревьюирования в различных системах контроля версий.
11. Методы организации работы в команде разработчиков.
12. Системы контроля версий.
13. Цели, задачи, этапы и объекты ревьюирования.
14. Планирование ревьюирования.
15. Цели, корректность и направления анализа программных продуктов. Выбор критериев сравнения. Представление результатов сравнения.
16. Примеры сравнительного анализа программных продуктов.
17. Цели, задачи и методы исследования программного кода.
18. Механизмы и контроль внесения изменений в код.
19. Обратное проектирование.
20. Анализ потоков данных.
21. Дизассемблирование.
22. Предпроцессинг кода.
23. Интеграция в IDE.
24. Валидация кода на стороне сервера и разработчика.
25. Совместимость и использование инструментов ревьюирования в различных системах контроля версий.
26. Особенности ревьюирования в Linux. Настройки доступа.
27. Типовые инструменты и методы анализа программных проектов.
28. Инструментарий различных сред разработки.
29. Инструментарий JavaDevelopmentKit.
30. Инструментарий Eclipse
31. Инструментарий NetBeans и другие.
32. Методологии (подходы) разработки ПО: XP, RUP, MSF

**Итоговая сумма** баллов по дисциплине, складывается из баллов за текущую аттестацию и баллов, полученных за выполнение проекта, и выводится итоговая оценка по следующим критериям:

70 -75 балла – удовлетворительно;

76 - 86 баллов – хорошо;

87- 100 баллов – отлично.

Обучающийся, не сдавший зачет с оценкой, приходит на пересдачу в сроки в соответствии с графиком ликвидации академических задолженностей: [http://www.kgau.ru/new/news/news/2017/grafik\\_lz.pdf](http://www.kgau.ru/new/news/news/2017/grafik_lz.pdf).

Дисциплина входит в состав профессионального модуля ПМ.03 «Ревьюирование программных продуктов». Формой промежуточной аттестации по профессиональному модулю ПМ.03 является экзамен по профессиональному модулю, по результатам сдачи которого выносится решение: «вид профессиональной деятельности освоен/оценка».

## **6. Учебно-методическое и информационное обеспечение дисциплины**

### **6.1. Основная литература**

Е. А. Черткова Программная инженерия. Визуальное моделирование программных систем : учебник для среднего профессионального образования, 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023

### **6.2. Дополнительная литература**

П. Г. Белов Управление рисками, системный анализ и моделирование в 3 ч. Часть 1 : учебник и практикум для среднего профессионального образования, Москва : Издательство Юрайт, 2023

### **6.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»)**

#### *Интернет-ресурсы*

1. Национальный Открытый Университет «ИНТУИТ» <https://intuit.ru/>
2. Портал CIT Forum <http://citforum.ru/>
3. Информационно-аналитическая система «Статистика» <http://www.ias-stat.ru/>

#### *Электронные библиотечные системы*

4. Каталог библиотеки Красноярского ГАУ -- [www.kgau.ru/new/biblioteka/](http://www.kgau.ru/new/biblioteka/) ;
5. Центральная научная сельскохозяйственная библиотека - [www.cnsnb.ru/](http://www.cnsnb.ru/) ;
6. Научная электронная библиотека "eLibrary.ru" – [www.elibrary.ru](http://www.elibrary.ru/) ;
7. Электронная библиотечная система «Лань» - <https://e.lanbook.com/>
8. Электронно-библиотечная система «Юрайт» - <https://urait.ru/>
9. Электронно-библиотечная система «AgriLib» - <http://ebs.rgazu.ru/>
10. Электронная библиотека Сибирского Федерального университета - <https://bik.sfu-kras.ru/>
11. Национальная электронная библиотека - <https://rusneb.ru/>
12. Электронная библиотечная система «ИРБИС64+» - [http://5.159.97.194:8080/cgi-bin/irbis64r\\_plus/cgiirbis\\_64\\_ft.exe?C21COM=F&I21DBN=IBIS\\_FULLTEXT&P21DBN=IBIS&Z21ID=&S21CNR=5](http://5.159.97.194:8080/cgi-bin/irbis64r_plus/cgiirbis_64_ft.exe?C21COM=F&I21DBN=IBIS_FULLTEXT&P21DBN=IBIS&Z21ID=&S21CNR=5)
13. Электронный каталог Государственной универсальной научной библиотеки Красноярского края - <https://www.kraslib.ru/>

#### *Информационно-справочные системы*

14. справочно-правовая система КонсультантПлюс <http://www.consultant.ru/cons/cgi/online.cgi?req=home;rnd=0.8636296761039928>
15. Информационно-правовой портал «Гарант». <http://www.garant.ru/>

#### *Профессиональные базы данных*

16. Коллективный блог по информационным технологиям, бизнесу и интернету. <https://habr.com/ru/>
17. Форум программистов и сисадминов Киберфорум <https://www.cyberforum.ru/>

#### *Сторонние электронно-образовательные ресурсы*

18. Министерство науки и высшего образования РФ
19. Российское образование
20. Единое окно доступа к образовательным ресурсам
21. Единая коллекция цифровых образовательных ресурсов
22. Федеральный центр информационно-образовательных ресурсов
23. Современная цифровая образовательная среда в РФ
24. <http://window.edu.ru/>
25. [http://window.edu.ru/catalog/resources?p\\_rubr=2.2.75.6](http://window.edu.ru/catalog/resources?p_rubr=2.2.75.6)

### **6.4. Программное обеспечение**

1. Операционная система Windows (академическая лицензия № 44937729 от 15.12.2008).
2. Офисный пакет приложений Microsoft Office (академическая лицензия № 44937729 от 15.12.2008).

3. Программа для создания и просмотра электронных публикаций в формате PDF &#8210; Acrobat Professional (образовательная лицензия № СЕ0806966 от 27.06.2008).
4. Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition. 1000-1499 Node 2 year Educational License (1B08-230201-012433-600-1212 с 01.02.2023 до 09.02.2024).
5. Moodle 3.5.6a (система дистанционного образования) - Открытые технологии договор 969.2 от 17.04.2020.
6. Библиотечная система «Ирбис 64», контракт 37–5–20 от 27.10.2020
7. NetBeans, Свободно распространяемое ПО (Apache License 2.0)
8. Python - среда программирования, свободно распространяемое ПО
9. Visual Studio Community – бесплатная среда разработки программного обеспечения на C++

## Приложение А. Варианты тестовых заданий по дисциплине

КОД (в соответствии с кодификатором)	ТЕСТОВОЕ ЗАДАНИЕ	Ответ
1.1	.....модель является предметом формализации	математическая
1.2	Построение модели исходных данных; построение модели результата, разработка алгоритма, разработка программы, отладка и исполнение программы, анализ и интерпретация результатов – это .....	этапы решения задачи с помощью компьютера
1.3	Процесс построения информационных моделей с помощью формальных языков называется.....	формализацией
1.4	Расписание движения поездов может рассматриваться как пример модели....	табличной
1.5	Математическая модель объекта это -....	совокупность записанных на языке математики формул, отражающих те или иные свойства объекта-оригинала или его поведение
1.6	Натурное (материальное) моделирование – это....	моделирование, при котором в модели узнается моделируемый объект, то есть натурная (материальная) модель всегда имеет визуальную схожесть с объектом-оригиналом
1.7	Система состоит из.....	объектов, которые называются элементами системы
1.8	Упорядоченная совокупность взаимосвязанных и взаимодействующих элементов называется _____	Системой
1.9	Перечислите требования, которые предъявляют к математическим моделям:	Адекватность
1.10	Может ли один объект иметь множество моделей:	да
2.1	Образные модели представляют собой:	зрительные образы объектов, зафиксированные на каком

		либо носители информации
2.2	Какие модели воспроизводят геометрические, физические и другие свойства объектов в материальной форме?	предметные
2.3	Дать определение модели...	материальный или абстрактный заменитель объекта, отражающий существенные с точки зрения цели исследования свойства изучаемого объекта, явления или процесса
2.4	Описание глобальной компьютерной сети Интернет в виде системы взаимосвязанных следует рассматривать как модель.....	сетевую модель
2.5	Последовательность этапов моделирования:	цель, объект, модель, метод, алгоритм, программа, эксперимент, анализ, уточнение
2.6	... методы – это методы, в которых задается некоторое начальное приближение и строится сходящаяся последовательность приближений к точному решению, причем каждое последующее приближение вычисляется с использованием предыдущих.	Итерационные
2.7	Сколько существует основных этапов разработки и исследование моделей на компьютере:	5
2.8	На первом этапе исследования объекта или процесса обычно строится ...	описательная информационная модель
2.9	На какой язык должна быть "переведена" прикладная задача для ее решения с использованием ЭВМ?	формальный математический
2.10	Табличная информационная модель представляет собой:	описание объектов (или их свойств) в виде совокупности значений, размещенных в таблице
3.1	..... модели представляют объекты и процессы в образной или знаковой форме:	информационные
3.2	Рисунки, карты, чертежи, диаграммы, схемы, графики представляют собой:	графические (образные) информационные модели
3.3	Географическую карту следует рассматривать скорее всего как.....	графическую информационную модель
3.4	... - абстрактная динамическая модель,	Имитационная модель

	реализованная на ЭВМ и воспроизводящая в рамках установленных ограничений поведение оригинала в хронологическом порядке	
3.5	Метод ... – численный метод решения системы линейных алгебраических уравнений. Суть метода заключается в нахождении по приближённому значению величины следующего приближения, являющегося более точным	итерации
4.1	... называются имитационные модели, взаимодействие пользователя с которыми сводится только к вводу исходной информации и управлению началом и окончанием работы моделей.	Автоматическими
4.2	..... модель является предметом формализации	математическая
4.3	Тип моделей..... применяется для описания ряда объектов, обладающих одинаковыми наборами свойств:	табличные информационные
4.4	Процесс построения информационных моделей с помощью формальных языков называется.....	формализацией
4.5	... называются имитационные модели, позволяющие исследователю активно управлять ходом моделирования, приостанавливать сеанс моделирования, изменять значения параметров модели, корректировать перечень регистрируемых данных и т. д.	Диалоговыми

## ЭКСПЕРТНОЕ ЗАКЛЮЧЕНИЕ

на фонды оценочных средств по дисциплине «Моделирование и анализ программного обеспечения»

для подготовки специалистов среднего звена по программе ФГОС СПО,  
специальность **09.02.07** «Информационные системы и программирование»  
ФГБОУ ВО Красноярский государственный аграрный университет

Представленные на рецензию фонды оценочных средств оформлены с соблюдением всех требований, предъявляемых к оформлению ФОС по стандартам ФГОС СПО.

Дисциплина «Моделирование и анализ программного обеспечения» является частью учебного плана по подготовке специалистов среднего звена по программе ФГОС СПО, специальность 09.02.07 «Информационные системы и программирование»

Оценочные средства для контроля успеваемости студентов представлены в полном объеме. При помощи фонда оценочных средств осуществляется контроль и управление процессом приобретения студентами необходимых знаний, умений, практического опыта и компетенций, определенных ФГОС СПО.

Представленные оценочные средства по дисциплине стимулируют познавательную деятельность за счет заданий разного уровня сложности, компетентного подхода, формируют навыки само- и взаимопонимания.

Фонды оценочных средств соответствуют обязательному минимуму содержания ФГОС СПО, обеспечивают проведение аттестации студентов учреждений СПО, дают возможность определить соответствие студентов конкретной характеристике.

Представленные ФОС для специальности 09.02.07 «Информационные системы и программирование» могут быть использованы в учебном процессе и соответствуют требованиям ФГОС СПО.

Эксперт:

доцент кафедры Вычислительной техники  
ФГАОУ ВО Сибирский федеральный университет,  
Институт космических и информационных  
технологий, канд. техн. наук



Постников  
Александр  
Иванович