МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В. Н. Татищева»
(Астраханский государственный университет им. В. Н. Татищева)

СОГЛАСОВАНО
Руководитель ОПОП
А. В. Григорьев
«05» мая 2025 г.

УТВЕРЖДАЮ
И.о. зав. кафедрой информационных технологий
О.Н. Выборнова
«05» мая 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

Составитель	**Карпенко А.В., старший преподаватель кафедры цифровых технологий**
Направление подготовки / специальность	**09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА**
Направленность (профиль) ОПОП	**ПРИКЛАДНАЯ ИНФОРМАТИКА В СОЦИАЛЬНЫХ НАУКАХ**
Квалификация (степень)	**бакалавр**
Форма обучения	**очная**
Год приема	**2023**
Курс	**3**
Семестр(ы)	**5**

Астрахань – 2025 г.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1. Целями освоения дисциплины (модуля) «Технология программирования» являются ознакомление с алгоритмами и процессами решения задач, с событийно-управляемым и параллельным программированием; с прикладными программными интерфейсами, основными структурами данных, методами оценивания эффективности алгоритмов и обоснования их корректности.

1.2. Задачи освоения дисциплины (модуля): «Технология программирования»: ознакомить с основными конструкциями программирования; основными структурами данных; и с объектно-ориентированным программированием; процедурным программированием; дать представление об этапах создания программного продукта в рамках жизненного цикла, о современном состоянии технологий разработки программного продукта; познакомить с существующими подходами к оценке качества процессов создания программного обеспечения, способах и приёмах отладки и тестирования, проектирования архитектуры программного продукта и выбора критериев хорошей архитектуры, типах пользовательского интерфейса.

В результате изучения дисциплины обучаемы должен:

- Знать: принципы проектирования программных систем; организацию процесса проектирования программного обеспечения; методы отладки и тестирования программ; методы декомпозиции и абстракции при программировании.
- Уметь: использовать методы декомпозиции и абстракции при проектировании; проектировать пользовательские интерфейсы; применять средства разработки программного обеспечения.
- Владеть: методами проектирования программного обеспечения при структурном и объектно-ориентированном подходе; методами совместной разработки приложений.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

2.1. Учебная дисциплина (модуль) «Технология программирования» относится к циклу дисциплин базовой части и осваивается в пятом семестре.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы следующие знания, умения и навыки, формируемые предшествующими учебными дисциплинами (модулями):

- Информатика

  Знания: базовые понятия информатики и вычислительной техники; понятие информационной системы и информационной технологии; технические и программные средства реализации информационных процессов; основные устройства, входящие в состав ЭВМ, их назначение и характеристики; формы представления и преобразования информации в компьютере.

  Умения: применять вычислительную технику для решения практических задач; разработать алгоритм поставленной задачи.

  Навыки работы на персональном компьютере.

- Основы программирования

  Знания: основные структуры данных, используемые в языках программирования; структуру программ; нахождение значения выражения.

  Умения: создавать схему алгоритма для задачи; решать задачи с помощью условных операторов, циклов.

  Навыки и (или) опыт деятельности: в области применения функций, работы с файлами.

- Алгоритмы и структуры данных

  Знания: способы применения различных структур данных для решения определённых задач; теория графов.

  Умения: использовать возможности структур при построении алгоритмов решения задач.

Навыки и (или) опыт деятельности: решении задач оптимизации.

2.3. Последующие учебные дисциплины (модули) и (или) практики для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной (модулем):

- Программирование микроконтроллеров;
- WEB - технологии.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОПОП ВО по данному направлению подготовки (специальности):

а) общепрофессиональных (ОПК):

ОПК-7. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем.

Таблица 1. Декомпозиция результатов обучения

Код компетенции	Код и наименование индикатора достижения компетенции[1]	Планируемые результаты обучения по дисциплине (модулю)		
		Знать (1)	Уметь (2)	Владеть (3)
ОПК-7	ОПК-7.1 Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем	- принципы инсталлирования программного и аппаратного обеспечения для информационных и автоматизированных систем	- выполнять параметрическую настройку информационных и автоматизированных систем	- навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины в соответствии с учебным планом составляет 5 зачетных единиц (180 часов).

Трудоемкость отдельных видов учебной работы студентов очной формы обучения приведена в таблице 2.1.

Таблица 2.1. Трудоемкость отдельных видов учебной работы по формам обучения

Вид учебной и внеучебной работы	для очной формы обучения
Объем дисциплины в зачетных единицах	5
Объем дисциплины в академических часах	180
Контактная работа обучающихся с преподавателем (всего), в том числе (час.):	54
- занятия лекционного типа, в том числе:	18
- практическая подготовка (если предусмотрена)	0
- занятия семинарского типа (семинары, практические, лабораторные), в том числе:	36

[1] Указываются в соответствии с утвержденными в ОПОП ВО

Вид учебной и внеучебной работы	для очной формы обучения
- практическая подготовка (если предусмотрена)	0
- в ходе подготовки и защиты курсовой работы[2]	18
- консультация (предэкзаменационная)[3]	0
- промежуточная аттестация по дисциплине[4]	0
Самостоятельная работа обучающихся (час.)	108
Форма промежуточной аттестации обучающегося (зачет/экзамен), семестр (ы)	экзамен – 5 семестр

Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий и самостоятельной работы, для каждой формы обучения представлено в таблице 2.2.

[2] Числовые данные в данной строке соответствуют трудоемкости, указанной в учебном плане в столбце «КР/КП» Если курсовая работа не предусмотрена – необходимо удалить строку «Контактная работа в ходе подготовки и защиты курсовой работы».

[3] Числовые данные в данной строке соответствуют трудоемкости, указанной в учебном плане в столбце «Конс. (для гр.)»

[4] Числовые данные в данной строке соответствуют трудоемкости, указанной в учебном плане в столбце «КПА»

Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий и самостоятельной работы, для каждой формы обучения представлено в таблице 2.2.

Таблица 2.2. Структура и содержание дисциплины (модуля)
для очной формы обучения

Раздел, тема дисциплины (модуля)	Контактная работа, час.							СР, час.	Итого часов	Форма текущего контроля успеваемости, форма промежуточной аттестации
	Л		ПЗ		ЛР		КР / КП			
	Л	в т.ч. ПП	ПЗ	в т.ч. ПП	ЛР	в т.ч. ПП				
Семестр 1.										
Тема 1. Применение технологий программирования	***2***				***8***		***4***	***16***	***30***	Устный опрос на экзамене
Тема 2. Этапы исследования предметной области	***2***				***4***		***1***	***14***	***21***	Лабораторная работа 1; устный опрос на экзамене
Тема 3. Методологии разработки	***2***				***8***		***1***	***16***	***27***	Лабораторная работа 2;
Тема 4. Критерии качества программного обеспечения	***2***				***4***		***4***	***16***	***26***	Лабораторная работа 3;
Тема 5 Проектирование и архитектура программного обеспечения	***2***				***4***		***4***	***18***	***28***	Лабораторная работа 4;
Тема 6. Диаграммы UML	***2***				***2***		***1***	***6***	***11***	Лабораторная работа 5;
Тема 7. Отладка и тестирование	***2***				***4***		***2***	***14***	***22***	Лабораторная работа 6;
Тема 8. Типы пользовательских интерфейсов и этапы их разработки	***4***				***2***		***1***	***8***	***15***	Лабораторная работа 7;
Консультации										
Контроль промежуточной аттестации										**Экзамен**
ИТОГО за семестр:	***18***				***36***		***18***	***108***	***180***	
Итого за весь период	***18***				***36***		***18***	***108***	***180***	

Условные обозначения:

Л – занятия лекционного типа; ПЗ – практические занятия, ЛР – лабораторные работы; КР – курсовая работа; СР – самостоятельная работа по отдельным темам

Таблица 3 – Матрица соотнесения разделов, тем учебной дисциплины (модуля) и формируемых компетенций

Раздел, тема дисциплины (модуля)	**Кол-во часов**	**Код компетенции**	**Общее количество компетенций**
		ОПК-7	
Тема 1. Применение технологий программирования	**30**	+	**1**
Тема 2. Этапы исследования предметной области	**21**	+	**1**
Тема 3. Методологии разработки	**27**	+	**1**
Тема 4. Критерии качества программного обеспечения	**26**		**1**
Тема 5 Проектирование и архитектура программного обеспечения	**28**	+	**1**
Тема 6. Диаграммы UML	**11**	+	**1**
Тема 7. Отладка и тестирование	**22**	+	**1**
Тема 8. Типы пользовательских интерфейсов и этапы их разработки	**15**	+	**1**
Итого	**180**		

Краткое содержание каждой темы дисциплины.

Тема 1. Применение технологий программирования

Данная тема является вводной и даёт определение понятию «Технология программирования». Идёт описание всех основных этапов разработки программного обеспечения включая описание типов программного обеспечения. Это не только жизненный цикл, но технологичные способы достижения цели – создания программного продукта. Уделяется внимание таким инструментам как фреймворки и библиотеки при разработке программного обеспечения. Приводится краткая характеристика тем которые будут проведены в рамках этой учебной дисциплины.

В рамках этой темы приготовлены несколько практических заданий для самостоятельного изучения которые служат подготовкой к выполнению лабораторных работ:

- Технологии описания данных – здесь уделяется внимание процессам сериализации и дессериализации данных в формате Json.
- Разработка RestFul-сервиса – это основной блок практических упражнений в рамках дисциплины. Здесь готовится ядро проекта который станет результатом курсовой работы.

Тема.2 Этапы исследования предметной области

Даётся оперение основным этапам разработки программного обеспечения:

1. Формирование требований
2. Разработка концепции
3. Техническое задание
4. Эскизный проект

5. Технический проект
6. Рабочая документация
7. Поставка или ввод в эксплуатацию
8. Сопровождение

Документ технического задания является финальной точкой прохождения этапов проектирования и достаточно строгим документом, потому что регламентируется ГОСТ. ТЗ это финальная точка по которой происходит реализация. В момент когда написано ТЗ, ничего больше менять нельзя. ТЗ это финальная, чёткая формализация того как вы общались с заказчиком. Кроме того, оно идёт в комплекте с договором где указаны сроки разработки и стоимость.

В рамках этой темы приготовлены практические задания для самостоятельного изучения которые служат подготовкой к выполнению лабораторных работ:

- Принципы ООП – рассматриваются особенности объектно-ориентированного программирования в рамках разрабатываемого проекта для курсовой работы.

Тема 3. Методологии разработки

Существует много различных методологий программирования. В данной теме даётся характеристика наиболее известным из них: Процедурное программирование, Объектно-ориентированное программирование, Функциональное программирование, Логическое программирование. Упоминаются нисходящий и восходящий подходы.

В рамках этой темы приготовлены практические задания для самостоятельного изучения которые служат подготовкой к выполнению лабораторных работ:

- Документирование кода – изучение технологии автоматической генерации документации программного продукта. Уделяется внимание библиотекам способным осуществить эту операции.

Тема 4. Критерии качества программного обеспечения

Гибкость, легкость, с которой программа может быть изменена, важна, потому что исходный код обычно развивается с течением времени. Через исходный код вы сообщаете свой дизайн любым программистам, которым в будущем когда-либо понадобится исправить ошибку, внести улучшения или просто повторно использовать ваш код. Вы рассказываете программистам, как ваш код должен работать, как его следует использовать и как лучше всего его изменить. Один из наиболее важных аспектов гибкости программы – читаемость исходного кода. Во многих случаях в реальном мире исходный код программы является документацией. Когда другие работают над вашим исходным кодом, чтобы добавить новые функции или исправить ошибки, они читают ваш исходный код, чтобы понять, что он делает.

В рамках этой темы приготовлены практические задания для самостоятельного изучения которые служат подготовкой к выполнению лабораторных работ:

- Исключения – программный продукт может считаться эффективным в том числе благодаря грамотному использованию пользовательских исключений

Тема 5. Проектирование и архитектура программного обеспечения

Эффективность системы является одним из основных критериев и её характеризуют:

- Надёжность
- Безопасность
- Производительность
- Масштабируемость

Высокая сопряжённость и слабая связность Когда говорят, что в системе должна быть высокая сопряженность, это означает эта сопряжённость должна быть внутри модуля. Он должен быть плотным, консистентным и не содержать ничего лишнего. Каждый модуль выполняет свою функцию – одну. Если он выполняет много функций, то его нужно разделить

на несколько подмодулей. Любой элемент, который вносится в модуль, должен иметь непосредственную связь с другими частями этого модуля.

При этом должна присутствовать слабая связность. Это означает что между модулями должна быть слабая связь. Чем меньше они друг с другом связаны тем лучше. Они должны быть независимы, в идеале. Важно отменить, что такое никогда не произойдёт на практике.

В рамках этой темы приготовлены практические задания для самостоятельного изучения которые служат подготовкой к выполнению лабораторных работ:

- Валидация данных – валидация и верефикация данных позволяет добиться в том числе и безотказности программного продукта. Эта практика так же нацелена на подготовку к выполнению лабораторной работы.

Тема 6. Диаграммы UML

На данном занятии речь идёт о нотации UML. Умение создания диаграмм является важным навыком при разработке программного обеспечения. Существует много различных видов диаграмм которые служат целям донесения информации о ходе разработки программного продукта до разных заинтересованных лиц. Будь то разработчики графического интерфейса или проектировщики СУБД

В рамках этой темы приготовлены практические задания для самостоятельного изучения которые служат подготовкой к выполнению лабораторных работ:

- Работа с ресурсами и многопоточность – в рамках этих заданий даётся представление о способах разрешения коллизий и состояния гонки.

Тема 7. Отладка и тестирование

На данном занятии речь идёт о том какие существуют методы и средства отладки – так называемые стандартные техники. Soft («софтверный») - это запуск внутри некой IDE, где вы смотрите на содержимое переменных, ставите точки останова, проходите код по шагам. Т.е. вы с помощью специального софта проводите отладку вашего программного обеспечения. «Железный» это распространено для всевозможных специализированных плат (Arduino). Есть специальные платы (дебаггеры) которые подключаются в исследуемое «железо». Удалённый отладчик-отладка через сеть.

Классификация ошибок при отладке: синтаксические ошибки (причиной таких ошибок могут быть неправильно написанные ключевые слова, неверно примененные разделители или недопустимые комбинации операторов); ошибки в структуре программы (ошибки такого рода появляется в результате неправильного написания многострочного оператора); ошибки, возникающие во время выполнения программы (это ошибки, возникающие во время работы программы, например, при выполнении деления на ноль или при попытки чтения из несуществующего на диске файла). Методы отладки программного обеспечения – статические и динамические.

В рамках этой темы приготовлены практические задания для самостоятельного изучения которые служат подготовкой к выполнению лабораторных работ:

- Логирование данных – создание логов это технология программирования способная помочь в достижении решения проблемы как для программиста так и для конечного пользователя.
- Модульное тестирование – это занятие позволит тщательно тестировать программу. Цель - научиться спроектировать и реализовать тестовый пример с помощью Unit-тестирования

Тема 8. Типы пользовательских интерфейсов и этапы их разработки

Интерфейс — общая граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил взаимодействия (управления, контроля и т. д.) между элементами системы.

Пользовательский интерфейс (UI) — это способ, которым вы выполняете какую-либо задачу с помощью какого-либо продукта, а именно совершаемые вами действия и то, что вы получаете в ответ.

Типы пользовательских интерфейсов и этапы их разработки:

- Создание мокапа
- User Flow Diagram (карта экранов)
- Утверждение структуры
- Выбор стиля UI
- Интерактивный прототип
- Утверждение результата

В рамках этой темы приготовлены практические задания для самостоятельного изучения которые служат подготовкой к выполнению лабораторных работ:

- Паттерны проектирования – шаблон проектирования - это повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста. В рамках этого занятия предлагаются шаблоны пригодные для применения в рамках разрабатываемого курсового проекта.
- Принципы SOLID и антипаттерны – здесь речь идёт о «вредных советах» - т.е. о том какие наиболее частые ошибки совершаются в рамках проектирования и разработки. Принципы SOLID напротив же позволяют избегать подобных ошибок.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРЕПОДАВАНИЮ И ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1. Указания для преподавателей по организации и проведению учебных занятий по дисциплине (модулю)

Электронный учебно-методический комплекс размещённый на образовательном портале Moodle, включает теоретические материалы, порядок выполнения лабораторных работ, список рекомендованной литературы.

Студенты выполняют лабораторные работы и прикрепляют свой ответ на образовательном портале Moodle. После проверки преподавателем, выставляется оценка или оставляется комментарий с замечаниями и рекомендациями.

5.2. Указания для обучающихся по освоению дисциплины (модулю)

Самостоятельная работа студентов подразумевает чтение и анализ технической литературы по предмету, документации на программное обеспечение, самостоятельное создание схемы алгоритма для задачи, проведение отладки и тестирования созданных модулей, выполнение индивидуального домашнего задания по одной из выбранных предметных областей. Практические задания представленные в курсе направлены в том числе на самостоятельную работу и помогают в выполнении лабораторных работ.

Сами лабораторные работы являются составными частями для выполнения курсовой работы. Для каждого обучающегося на платформе Moodle опубликована тема его курсовой работы. Суть каждой лабораторной работы заключается в самостоятельном выполнении практических примеров и изучении материалов представленных в рамках этой учебной дисциплины на платформе Moodle

Таблица 4. Содержание самостоятельной работы обучающихся

Вопросы, выносимые на самостоятельное изучение	Кол-во часов	Формы работы
Тема 1. Применение технологий программирования.	16	Выполнения заданий «Технологии описания

Процессы сериализации и дессериализации. Создание схемы данных Json.		данных» и начало работы с заданием «Разработка RestFul-сервиса»
Тема 2. Этапы исследования предметной области. Согласно полученной теме курсовой работы производится анализ предметной области, описание модели данных и создание ER-диаграммы будущего проекта.	14	Продолжение работы с заданием «Разработка RestFul-сервиса». Выполнение лабораторной работы № 1. Описание предметной области.
Тема 3. Методологии разработки. Изучение особенностей объектно-ориентированного программирования в Python (наследование, инкапсуляция, полиморфизм)	16	Продолжение работы с заданием «Разработка RestFul-сервиса». Выполнение практических заданий «Принципы ООП». Выполнение лабораторной работа №2 – использование знаний принципов ООП в проектировании.
Тема 4. Критерии качества программного обеспечения. Способы достижения гибкости программного продукта. Примеры «плохого кода».	16	Выполнение лабораторной работы № 3. К разрабатываемому проекту добавляется документация кода.
Тема 5 Проектирование и архитектура программного обеспечения. Практика в анализе критических сегментов разрабатываемого программного продукта. На их основе создаются пользовательские исключения и обработчики таких исключений	18	Выполнение лабораторной работы № 4. К разрабатываемому проекту добавляются классы обработки пользовательских исключений.
Тема 6. Диаграммы UML. Проектирование различных видов диаграмм (диаграмма вариантов использования, SADT, DFD) в рамках разрабатываемого проекта. Понятие многопоточности и параллельного программирования (семафоры и замки).	6	Выполнение лабораторной работы № 5. К разрабатываемому проекту добавляется валидация данных
Тема 7. Отладка и тестирование. Библиотеки для модульного тестирования (PyTest). Вывод в лог рещультатов тестирования. Формат Problem details.	14	Лабораторная работа №6. К разрабатываемому проекту добавляется обработчики создания лога.
Тема 8. Типы пользовательских интерфейсов и этапы их разработки. Применение антипаттернов в разработке.	8	Лабораторная работа №7.Заключение разработки проекта для курсовой работы. Демонстрация работоспособности.

5.3. Виды и формы письменных работ, предусмотренных при освоении дисциплины (модуля), выполняемые обучающимися самостоятельно

В качестве работ, выполняемых обучающимися самостоятельно используются лабораторные работы и практическое задание.

На информационном портале Moodle в темах дисциплины размещены задания для выполнения лабораторных работ и практического задания. Лабораторная работа заключается в последовательном выполнении шагов описанных в методических указаниях, например цепочка действий при ручной отладке кода или создании архитектуры проекта. В практическом задании, например, требуется описать вариант технического задания по теме закреплённой за обучающимся.

Итоговым результатом является курсовой проект состоящий из последовательного выполнения лабораторных работ. Лабораторные работы выполняются после самостоятельного выполнения практических заданий на занятиях.

Созданные согласно заданиям программы, архивируется и прикрепляется в виде ответа на задание.

6. ОБРАЗОВАТЕЛЬНЫЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

При реализации различных видов учебной работы по дисциплине могут использоваться электронное обучение и дистанционные образовательные технологии.

В рамках реализации компетентностного подхода в соответствии с требованиями ФГОС ВО в учебном процессе предусмотрены активные и интерактивные формы проведения занятий.

Основой для выстраивания аудиторных занятий является лабораторные работы. Это самостоятельная работа учащегося, выполненная с помощью консультаций преподавателя. Основное отличие такой деятельности — это то, что студент, прежде всего, получают практические навыки в области программирования.

6.1. Образовательные технологии

Цели курса достигаются путём сочетания комплекса методов обучения, включающих самостоятельную работу студентов через платформу интерактивного обучения «Moodle» и лабораторные работы, выполняемые на ЭВМ.

В процессе обучения используются мультимедийные презентации. Для проверки промежуточных знаний студентов применяется электронное тестирование.

Студенты выполняют задания по разработке одного пункта из предложенного списка изучаемых вопросов, обосновывают правильность работы, и демонстрируют работу на примерах. В процессе выполнения лабораторных работ достигаются следующие цели:

- закрепляются теоретические познания, полученные на лекциях, актуализируется их практическая значимость, закрепляется мотивация к освоению курса;
- студент вникает в последовательность построения программных конструкций;
- приобретаются навыки программирования;
- формируется навык выявления ошибочных и нештатных ситуаций и реагирования на них.

Лабораторные работы, выполняются самостоятельно, а возникающие при их выполнении проблемы разрешаются в рамках консультации через платформу «Moodle» или очно на лабораторных занятиях.

Во время самостоятельной работы студенты должны написать программы по выбранным задачам и затем представить их на практических занятиях. Текущий контроль усвоения материала осуществляется в виде проверки выполнения заданий и написанных алгоритмов с учетом их обоснования и вычисленной сложности.

В рамках изучения дисциплины предусмотрено использование в учебном процессе следующих активных и интерактивных форм проведения занятий:

- проведение дискуссий.

Таблица 5 – Образовательные технологии, используемые при реализации учебных занятий

Раздел, тема дисциплины (модуля)	Форма учебного занятия		
	Лекция	Практическое занятие, семинар	Лабораторная работа
Тема 1. Применение технологий программирования	Тематические дискуссии, анализ конкретных ситуаций	Не предусмотрено	Не предусмотрено
Тема 2. Этапы исследования предметной области	Тематические дискуссии, анализ конкретных ситуаций	Не предусмотрено	Лабораторная работа 1
Тема 3. Методологии разработки	Тематические дискуссии, анализ конкретных ситуаций	Не предусмотрено	Лабораторная работа 2
Тема 4. Критерии качества программного обеспечения	Тематические дискуссии, анализ конкретных ситуаций	Не предусмотрено	Лабораторная работа 3
Тема 5 Проектирование и архитектура программного обеспечения	Тематические дискуссии, анализ конкретных ситуаций	Не предусмотрено	Лабораторная работа 4
Тема 6. Диаграммы UML	Тематические дискуссии, анализ конкретных ситуаций	Не предусмотрено	Лабораторная работа 5
Тема 7. Отладка и тестирование	Тематические дискуссии, анализ конкретных ситуаций	Не предусмотрено	Лабораторная работа 6
Тема 8. Типы пользовательских интерфейсов и этапы их разработки	Тематические дискуссии, анализ конкретных ситуаций	Не предусмотрено	Лабораторная работа 7

6.2. Информационные технологии

Методическая поддержка дисциплины обеспечивается использованием дистанционных технологий. Студентам предлагается информационный ресурс, расположенный по адресу: http://www.moodl.aspu.ru/.

Доступ студентов к учебным ресурсам осуществляется по учетной записи и паролю после регистрации на курс «Технология программирования» на период обучения по данной дисциплине. На сервере размещен методический материал по данной дисциплине, в содержание которого входит теоретический материал, задания на выполнение лабораторно-практических работ, вопросы к экзамену.

При реализации различных видов учебной и внеучебной работы используются следующие информационные технологии:

- использование виртуальной обучающей среды (или системы управления обучением LMS Moodle) http://moodle.asu.edu.ru (размещение учебно-методического материала, публикация заданий для предоставления студентами выполненных работ) как элемента интерактивного

взаимодействия участников образовательного процесса (технологии дистанционного обучения);

- использование ресурсов ЭБС и сети Internet, как источников информации.

При реализации различных видов учебной и внеучебной работы используются и иные информационные системы, сервисы и мессенджеры.

6.3. Программное обеспечение, современные профессиональные базы данных и информационные справочные системы

6.3.1. Программное обеспечение

Интегрированная среда разработки среда разработки для языка программирования Python. Среда разработки Python используются в рассматриваемом курсе для создания RestFul-приложения являющегося результатом курсовой работы по закреплённой за обучающимся темой.

6.3.2. Современные профессиональные базы данных и информационные справочные системы

1. Электронный каталог Научной библиотеки АГУ на базе MARKSQL НПО «Информ-систем»: https://library.asu.edu.ru.
2. Электронная библиотека «Астраханский государственный университет» собственной генерации на электронной платформе ООО «БИБЛИОТЕХ»: https://biblio.asu.edu.ru.
3. Электронный каталог «Научные журналы АГУ»: http://journal.asu.edu.ru/.
4. Универсальная справочно-информационная полнотекстовая база данных периодических изданий ООО «ИВИС»: http://dlib.eastview.com/
5. Электронно-библиотечная система elibrary. http://elibrary.ru
6. Справочная правовая система КонсультантПлюс: http://www.consultant.ru
7. Информационно-правовое обеспечение «Система ГАРАНТ»: http://garant-astrakhan.ru

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

7.1. Паспорт фонда оценочных средств

При проведении текущего контроля и промежуточной аттестации по дисциплине «Технология программирования» проверяется сформированность у обучающихся компетенций, указанных в разделе 3 настоящей программы. Этапность формирования данных компетенций в процессе освоения образовательной программы определяется последовательным освоением дисциплин и прохождением практик, а в процессе освоения дисциплины – последовательным достижением результатов освоения содержательно связанных между собой разделов, тем.

Таблица 6 – Соответствие разделов, тем дисциплины (модуля), результатов обучения по дисциплине (модулю) и оценочных средств

Контролируемый раздел, тема дисциплины (модуля)	Код контролируемой компетенции (компетенций)	Наименование оценочного средства
Тема 1. Применение технологий программирования	ОПК-7	Вопросы для обсуждения; устный опрос на экзамене
Тема 2. Этапы исследования предметной области	ОПК-7	Лабораторная работа 1; устный опрос на экзамене

Тема 3. Методологии разработки	ОПК-7	Лабораторная работа 2; устный опрос на экзамене
Тема 4. Критерии качества программного обеспечения	ОПК-7	Лабораторная работа 3; устный опрос на экзамене
Тема 5 Проектирование и архитектура программного обеспечения	ОПК-7	Лабораторная работа 4; устный опрос на экзамене
Тема 6. Диаграммы UML	ОПК-7	Лабораторная работа 5; устный опрос на экзамене
Тема 7. Отладка и тестирование	ОПК-7	Лабораторная работа 6; устный опрос на экзамене
Тема 8. Типы пользовательских интерфейсов и этапы их разработки	ОПК-7	Лабораторная работа 7; устный опрос на экзамене

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

В системе Moodle балл за выполнение лабораторно-практической работы выставляется в 100-балльной шкале комплексно с учетом степени подготовки студента к выполнению работы, объема выполненной работы на занятии и оформлении отчета в соответствии с перечисленными критериями. Для восстановления итоговой оценки, за каждую лабораторную работу полученные студентами баллы пересчитываются по шкале в соответствии с БАРС.

Таблица 7 – Показатели оценивания результатов обучения в виде знаний

Шкала оценивания	Критерии оценивания
5 «отлично»	Все задания лабораторной работы выполнены в полном объёме. Программа работает верно, на всех вариантах тестовых данных. Алгоритмы в коде программы реализованы корректно.
4 «хорошо»	В программе реализованы все функции, заявленные в задании лабораторной работы. Программа не работает корректно на всех вариантах входных данных.
3 «удовлетворительно»	В разработанной программе отсутствует реализация всех функций, заявленных в задании лабораторной работы. Программа не работает корректно на всех вариантах входных данных
2 «неудовлетворительно»	Разработанная согласно заданию лабораторной работы, программа не предоставлена либо не запускается

Таблица 8 – Показатели оценивания результатов обучения в виде умений и владений

Шкала оценивания	Критерии оценивания
5 «отлично»	студент демонстрирует системные теоретические знания, владеет терминологией, делает аргументированные выводы и обобщения, приводит примеры, показывает свободное владение монологической речью и способность быстро реагировать на уточняющие вопросы.
4 «хорошо»	студент демонстрирует прочные теоретические знания, владеет терминологией, делает аргументированные выводы и обобщения, приводит примеры, показывает свободное владение монологической речью, но при этом делает несущественные ошибки, которые быстро исправляет самостоятельно или при незначительной коррекции преподавателем.
3 «удовлетворительно»	студент демонстрирует неглубокие теоретические знания, проявляет слабо сформированные навыки анализа явлений и процессов, недостаточное умение делать аргументированные выводы и приводить примеры, показывает недостаточно свободное владение монологической речью, терминологией, логичностью и последовательностью изложения, делает ошибки, которые может исправить только при коррекции преподавателем.
2 «неудовлетворительно»	студент демонстрирует незнание теоретических основ предмета, не умеет делать аргументированные выводы и приводить примеры, показывает слабое владение монологической речью, не владеет терминологией, проявляет отсутствие логичности и последовательностью изложения, делает ошибки, которые не может исправить даже при коррекции преподавателем, отказывается отвечать на занятии.

Преподаватель, реализующий дисциплину (модуль), в зависимости от уровня подготовленности, обучающихся может использовать иные формы, методы контроля и оценочные средства, исходя из конкретной ситуации.

7.3. Контрольные задания и иные материалы, необходимые для оценки результатов обучения по дисциплине (модулю)

Тема 2. Этапы исследования предметной области

Лабораторная работа №1 «Описание предметной области»

Подготовить обоснование и техническое задание на проект согласно закреплённой теме. т.е. требуется написать отчёт с анализом предметной области по варианту. В выходном документе требуется описать модели сущностей в табличном виде, изобразить ER-диаграмму, описать классы сущностей. Дать описание функциям разрабатываемого проекта.

Тема 3. Методологии разработки

Лабораторная работа №2 «Проектирование архитектуры веб-приложения»

Согласно анализу проведенному вами в рамках первой лабораторной работы и полученному отчёту, требуется:

1) Создать базу данных по разработанной вами ER-диаграмме и заполнить её тестовыми данными

2) Описать модели вашей предметной области. Можете использовать data class или например BaseModel (или другое)

3) Описать классы работы с базой данных (сервисные классы в наших методичках)

4) Описать классы-контроллеры для каждой сущности. Вы можете использовать любой доступны вам способ создания сервера приложения Flask (если есть желание, то другой фреймворк, например Django) с подходом через ООП или используя Blueprint (см."Разработка

RestFul-сервиса. Часть 4."). Основные нюансы были рассмотрены нами в уже открытых методических рекомендациях.

5) Проверить работу каждого HTTP-метода вашего приложения через Postman.

Тема 4. Критерии качества программного обеспечения

Лабораторная работа №3 «Документация кода»

В рамках данной работы требуется оформить выходной документ с документацией кода созданный на основе комментариев в коде. Документацию кода вы пишите для проекта который вы уже создали в лабораторной работе №2. Подробности и примеры представлены вам в теме "Документирование кода (Практика №4)".

Т.е. на основе полученных знаний вы оформляете каждый класс, каждый метод вашего проекта любым стандартным стилем.

При этом - вы можете использовать не только стандартные стили представленные в методических рекомендация, но и применить сторонние библиотеки и фреймфорки для генерации "красивого" выходного документа. Здесь вы вольны выбрать то что больше подходит вам. Но задача лабораторной не меняется - вы должны получить полную документацию вашего ПО.

Тема 5. Проектирование и архитектура программного обеспечения

Лабораторная работа №4 «Обработка пользовательских исключений»

Суть данной работы заключается в более подробном анализе предметной области согласно вашему варианту и выявлению возникновения потенциальных исключительных событий.

В теме "Исключения (Практика №5)" подробно приводятся примеры обработки таких событий - изучите этот материал. Для вашего проекта теперь требуется написать соответствующие обработки. Самостоятельно проанализируйте условия их возникновения в вашей предметной области (дубликаты, размер коллекции и т.д.).

Сделайте это для всех сущностей вашего проекта и для всех операций (получение, добавление и т.д) где это необходимо

На основе этого анализа создайте пользовательские классы для обработки этих исключений, используйте их и протестируйте.

Тема 6. Диаграммы UML

Лабораторная работа №5 «Валидация данных»

В рамках данной работы вам требуется написать код валидирующий пользовательский ввод данных. Для этого используйте схемы данных (json или xml). Подробности продемонстрированы в теме "Валидация данных (Практика №6)". В зависимости от того как вы организовали свой проект (например, на основе Blueprint) результат у вас будет соответственно разный. У вас свои сущности согласно вариантам и свои правила для валидации полей и их набора. Вы сами определяете правила, но они должны полностью соответствовать вашей предметной области и не допускать попадание в вашу систему ошибочных данных. В рамках нашей работы вы можете сделать валидацию только для тех методов которые предполагали получение json - т.е. это Post и Put. При этом вы можете сделать так, чтобы все операции требовали некий json - здесь на ваше усмотрение.

Допускается использовать сторонние инструменты для валидации - фреймворки и библиотеки, при условии, что вы сможете объяснить как они работают.

Эта работа является продолжением темы "Исключения" и вы продолжаете писать документацию для новых методов и классов. Ответ пользователю в случае провала валидации требуется отдавать в формате "Problem details".

Тема 7. Отладка и тестирование

Лабораторная работа №6 «Логирование данных»

В рамках данной работы вам требуется реализовать механизм логирования в вашем проекте. Подробности продемонстрированы в теме "Логирование данных (Практика №7)". В зависимости от того как вы организовали свой проект (например, на основе Blueprint) результат у вас будет соответственно разный. У вас свои сущности согласно вариантам и свои правила для выбора - какая информация важна и должна быть залогирована. Вы сами определяете формат журнала: подробности, уровень и т.д. При этом ваш лог должен быть хорошо читаемым.

Для ваших сущностей и вашей предметной области вы определяете отдельные файлы лога, например:

1)для контроллера вы логируете невалидный json

2)для сервиса вы логируете запрос на вставку в базу данных

3)при желании вы можете добавить IP-адрес с которого пришёл запрос GET

и т.д.

Допускается использовать сторонние инструменты для логирования - фреймворки и библиотеки, при условии, что вы сможете объяснить как они работают.

Эта работа является продолжением темы "Исключения" и "Валидация" и вы продолжаете писать документацию для новых методов и классов.

Тема 8. Типы пользовательских интерфейсов и этапы их разработки

Лабораторная работа №7 «Модульное тестирование»

В рамках данной работы требуется реализовать тестовые функции для каждого метода вашего проекта.

Вы можете воспользоваться материалами из темы "Модульное тестирование (Практика №9)" - т.е. применить unittest или pytest если вы работаете с Python или другой фреймворк.

В вашем проекте есть база данных, значит вы должны обеспечить её неизменность для каждого тестового прогона. Вы можете добиться этого разными способами, например, как мы показали это в практике - через скрипт создания базы данных и её заполнения при каждом запуске приложения.

Для каждого метода в вашем проекте вы должны удостовериться, что он работает корректно. Поэтому вы должны создать по 2 теста каждой категории для каждого метода вашего проекта. Например, у вас есть метод добавления в базу данных новой записи, значит вы:

1) создаёте два разных достоверных объекта данных для проверки

2) создаёте два разных заведомо недостоверных объекта для проверки

Итого - каждый метод проверяется 4 раза. Этого может быть две тестовые функции в каждой из которых два раза вызывается assert.

Данная работа очень объёмная и от того крайне важная. Это последняя лабораторная работа, которая дополняет ваш проект. Теперь он будет считаться завершённым и в этом виде его нужно прикрепить в виде архива.

Курсовой проект дисциплины «Технология программирования»

Выполнение всех лабораторных работ является критерием выполнения курсового проекта. Документация кода происходит после прохождения каждого нового этапа. В итоге получается проект:

- Содержащий полно-заполненную базу данных согласно варианту

- Для каждой сущности реализованы базовые операции: добавление новой записи, вывод на экран одной записи, вывод на экран всех записей, редактирование одной записи, удаление одной записи, удаление нескольких записей.
- Сущностей (таблиц базы данных) должно быть не менее 3 штук для реализации отношений многие ко многим или один ко многим в рамках реляционных баз данных.
- Каждая операция должна иметь защиту в виде обработки пользовательских исключений.
- Вывод на экран в случае ошибки должен быть оформлен в виде Problem Details.
- В случае возникновения исключительного события должен формироваться лог.
- В проекте должны быть классы модельного тестирования с демонстрацией работы.

Таким образом результатом реализации курсового проекта является Rest-сервис содержащий все выполненные лабораторные работы. Примерное содержание такого проекта представлено на основе примера «Дисциплины преподавателей» и описано далее.

Предметная область: Дисциплины преподавателей

Объекты: Преподаватели, Дисциплины, Связь преподавателей и дисциплин

Примечание: Одну дисциплину могут вести несколько преподавателей. У каждого преподавателя может быть множество дисциплин в нагрузке.

Формальное описание сущностей:

Объект **Преподаватель** может иметь характеристики:

- код преподавателя (уникальный идентификатор)
- ФИО
- Номер телефона

Дисциплина – характеристики:

- код дисциплины (уникальный идентификатор)
- название дисциплины
- вид дисциплины (например, базовая часть, практика, НИР)

Связь преподавателей и дисциплины:

- Код записи (уникальный идентификатор)
- Ссылка на преподавателя
- Ссылка на дисциплину

Описание сущностей в табличном виде:

- Таблица «*Преподаватель*»

Название поля	Тип данных	Описание
ID(PK)	INTEGER	Уникальный идентификатор преподавателя. Первичный ключ.
FIO	TEXT	ФИО преподавателя
PHONE	TEXT	Номер телефона преподавателя

- Таблица «*Дисциплина*»

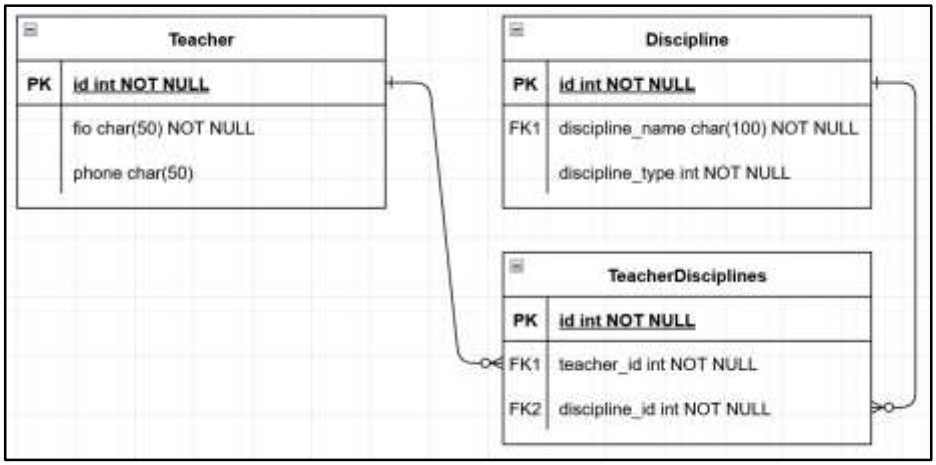
Название поля	Тип данных	Описание
ID(PK)	INTEGER	Уникальный идентификатор дисциплины. Первичный ключ.
DISCIPLINE_NAME	TEXT	Название дисциплины
DISCIPLINE_TYPE	INTEGER	Код вида дисциплины, например 1 – базовая, 2 – практика, 3 – НИР

- Таблица «*Связь преподавателей и дисциплины*»

Название поля	Тип данных	Описание
ID(PK)	INTEGER	Уникальный идентификатор записи. Первичный ключ.
DISCIPLINE_ID (FK)	INTEGER	Код дисциплины. Внешний ключ.
TEACHER_ID (FK)	INTEGER	Код преподавателя. Внешний ключ.

ER-диаграмма:

По описанию таблиц, представляющих объекты вашей предметной области необходимо изобразить ER-диаграмму любым доступным для вас способом (с помощью Draw.io). Пример ER-диаграммы:



Характеристики (состав полей сущностей) автоматизируемых объектов определяются студентом самостоятельно. Обязательной характеристикой объекта является его уникальный идентификатор – т.е. первичный ключ для каждой сущности.

Описание классов сущностей

Согласно созданным таблицам требуется описать программные классы с перечислением полей и указанием типов данных. В рамках дисциплины «Технология программирования» рекомендуется использовать язык **Python**. Например:

```
from dataclasses import dataclass
@dataclass
class Teacher:
    id: int
    fio: str
```

```
from dataclasses import dataclass
@dataclass
class Discipline:
    id: int
    name: str
    type: int
```

```
from dataclasses import dataclass
@dataclass
class TeeacherDiscipline:
    id: int
    teacher_id: int
    discipline_id: int
```

Обратите внимание, что во всех вариантах заданий объекты различных категорий находятся в иерархической зависимости. Например, у каждой страны может быть несколько городов, при этом один город принадлежит только одной стране.

Примерный перечень тем курсовых работ по дисциплине «Технология программирования»

№	Предметная область	Объекты	Примечание
1	Автосалон	Производители автомобилей, Марки, Связь производителя с маркой автомобиля	Марки автомобилей сгруппированы по производителям. У каждого производителя имеется множество марок.
2	Библиотека	Авторы, Книги, Связь автора и книги.	Книги в библиотеке сгруппированы по авторам. У каждого автора имеется множество книг. У книги может быть несколько авторов.
3	Отдел кадров	Подразделения, Сотрудники, Связь подразделения и сотрудника.	Имеется множество подразделений предприятия. В каждом подразделении работает множество сотрудников. Сотрудник может числиться не сколькими подразделениями.
4	Учебный отдел	Группы, Студенты, Связь группы и студента.	Имеется множество учебных групп. Каждая группа включает в себя множество студентов.
5	Табель успеваемости (учет оценок)	Студент, Дисциплина, Оценка студента по дисциплине	Студент обучается на множестве дисциплин. У студента может быть одна оценка по каждой дисциплине.
6	Агентство новостей	Категории новостей, Новости, Связь категории и новостной статьи.	Новости сгруппированы по категориям. У каждой категории имеется множество новостей. Новостная статья может иметь несколько категорий.
7	Продуктовый магазин	Категория продукта, Продукт, Связь категории и продукта	Продукты в магазине сгруппированы по категориям. Для каждой категории определено множество продуктов. У продукта может быть несколько категорий.

№	Предметная область	Объекты	Примечание
8	Футбол	Команды, Игроки, Связь игрока и команды.	Имеется множество футбольных команд. Для каждой команды определено множество игроков. Игрок может переходить из одной команды в другую.
9	Музыкальный магазин	Исполнители, Альбомы, Связь исполнителя и альбом.	В музыкальном магазине альбомы сгруппированы по исполнителям. Для каждого исполнителя задано множество альбомов. У альбома может быть несколько исполнителей.
10	Аэропорт	Авиакомпании, Самолёт, Рейсы	Имеется множество авиакомпаний и множество самолётов. Рейс является использованием авиакомпанией некоего борта.
11	Файловая система	Папки, Файлы, Связь файлов и папок.	Имеется множество папок (независимых друг от друга). Для каждой папки определено множество файлов. Файл может быть перемещён в любую папку.
12	Расписание занятий	Дни недели, Дисциплины, Занятия (связь дня недели и дисциплины)	Имеются рабочие недели: числитель и знаменатель. Для каждого учебного дня определен перечень занятий. Дисциплина может преподаваться в разные дни учебного недели.
13	Записная книжка	Календарные дни, Утвержденные праздники, Мероприятия (связь календарного дня и праздника)	Имеется множество дней. Для каждого дня определен перечень мероприятий. В один день может быть множество праздников.
14	Видеомагазин	Жанры, Фильмы, Связь жанров и фильмов	Имеется множество жаров. Для каждого жанра определен перечень фильмов. У фильма может быть несколько жанров.
15	Железная дорога	Железнодорожные станции, Поезда, Связь поезда и станции.	Имеется множество железных дорог. В ведомстве каждой дороги находится множество станций. Разные поезда останавливаются на множестве станций. На станции останавливается множество поездов.
16	Склад	Секции, Товары, Связь секции и товара.	Товары на складе сгруппированы по секциям. Для каждой секции задано множество товаров. В секции может быть множество товаров. Один товар принадлежит конкретной секции.
17	Учебные курсы	Студенты, Учебные курсы, Связь студентов и учебных курсов.	Имеется множество курсов. Один студент может быть записан на множество курсов. На один курс может быть записано множество студентов.
18	Заказ товаров	Товары, Заказы, Связь товаров и заказов.	Имеется множество товаров и множество заказов. Один заказ может

№	Предметная область	Объекты	Примечание
			содержать множество продуктов. Один продукт может входить в множество заказов.
19	Поликлиника	Пациенты, Врачи, Связь врачей и пациентов.	Один врач может лечить многих пациентов. Один пациент может наблюдаться у многих врачей (по разным специализациям)
20	Группы в социальных сетях	Пользователи, Группы, Связь пользователя и групп.	Один пользователь может состоять во многих группах. В одной группе может состоять много пользователей.
21	Кулинария	Рецепты, Ингредиенты, Связь ингредиентов и группы.	В одном рецепте используется множество ингредиентов. Один ингредиент (например, «соль») может использоваться в множестве рецептов.
22	Проекты	Сотрудники, Проекты, Связь сотрудников и проектов.	Над одним проектом может работать множество сотрудников. Один сотрудник может быть назначен на множество проектов.
23	Музыкальные подборки	Музыкальные композиции, Подборки, Связь композиций и подборки.	Одна музыкальная подборка содержит множество композиций. Одна композиция может быть добавлена во множество музыкальных подборок разными пользователями.
24	Аптеки	Аптека, Лекарство, Наличие (связь лекарства и аптеки).	Одна аптека продаёт множество лекарств. Одно и то же лекарство может продаваться во многих аптеках.
25	Спортивные состязания	Спортсмен, Соревнование, Участие (связь спортсмена и соревнования).	Один спортсмен может участвовать во многих соревнованиях. В одном соревновании участвуют многие спортсмены.
26	Резюме пользователей	Соискатель, Навык, Связь соискателя и навыка (компетенция).	Один соискатель владеет множеством навыков. Один навык (например, «Python») есть у множества соискателей.
27	Аллергии	Ингредиент, Аллергия, Противопоказание (связь ингредиента и аллергии).	Один ингредиент может вызывать множество аллергий (арахис вызывает сыпь, отёк). Одна аллергия может вызываться множеством ингредиентов (отёк может возникнуть из-за арахиса, морепродуктов).
28	Аптечный рецепт	Пациент, Лекарство, Назначение (связь лекарства и пациента).	Одному пациенту могут назначить множество лекарств. Одно лекарство может быть назначено множеству пациентов.
29	Поставки	Поставщики, Компоненты, Связь поставщика и компонента.	Один поставщик может поставлять множество компонентов. Один компонент может закупаться у множества поставщиков.

№	Предметная область	Объекты	Примечание
30	Серверная инфраструктура	Сервер, Сервис, Связь (развёртывание).	На одном сервере может быть развёрнуто множество сервисов. Один сервис может быть развёрнут на множестве серверов (для отказоустойчивости или балансировки нагрузки).

Перечень вопросов и заданий, выносимых на экзамен / зачёт / дифференцированный зачёт

Таблица 9 – Примеры оценочных средств с ключами правильных ответов

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
Код и наименование проверяемой компетенции ОПК-7. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем.				
1.	Задание закрытого типа	Какого вида операции наследования в ООП не существует… 1) одиночное 2) многоуровневое 3) иерархическое 4) вложенное	4	1
2.		Абстрактный класс это… 1) класс из которого нельзя создать экземпляр 2) класс у которого нет конструктора по-умолчанию 3) класс все поля которого все методы должны быть приватными 4) класс у которого не может быть наследников	1	1
3.		Какого типа паттернов проектирования не существует? 1)порождающие 2) структурные 3) побуждающие 4) поведенческие	3	1
4.		Главная задача UI-дизайнера это	3	1

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		1) составление технического задания 2) изучить целевую аудиторию продукта и выделить сценарии использования 3) нарисовать красивый интерфейс под разную вёрстку 4) помочь пользователю быстро и без стресса понять, как пользоваться продуктом		
5.		Результатом, который должен быть получен на этапе спецификации при анализе проблемной области должен стать 1)письменный документ, определяющий требования к программной системе 2) прототип пользовательского интерфейса 3) кодовая база (документация кода) 4) User-flow диаграмма	1	1
6.	Задание открытого типа	Этапы работы компилятора.	Лексический анализ - процесс аналитического разбора входной последовательности символов на распознанные группы — лексемы, с целью получения на выходе идентифицированных последовательностей, называемых «токенами». Синтаксический анализ - процесс сопоставления линейной последовательности лексем (слов, токенов) естественного	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			или формального языка с его формальной грамматикой. Семантический анализ - этап в последовательности действий алгоритма автоматического понимания текстов, заключающийся в выделении семантических отношений, формировании семантического представления текстов. Оптимизация Генерация кода - часть процесса компиляции, когда специальная часть компилятора, кодогенератор, конвертирует синтаксически корректную программу в последовательность инструкций, которые могут выполняться на машине.	
7.		Нисходящий и восходящий подход к программированию.	При нисходящем подходе проблема разбита на более мелкие блоки, которые в дальнейшем могут быть разбиты на еще более мелкие блоки. Каждый блок называется модулем. Каждый модуль является самодостаточным блоком, в котором есть все необходимое для выполнения своей задачи. При восходящем подходе проектирование системы начинается с самого низкого уровня компонентов, которые затем соединяются между собой, чтобы получить компоненты более	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			высокого уровня. Этот процесс продолжается до тех пор, пока не будет создана иерархия всех компонентов системы. Однако в реальном сценарии очень трудно с самого начала знать все компоненты самого низкого уровня. Таким образом, подход "снизу вверх" используется только для очень простых задач.	
8.		Что происходит при анализе прикладной области во время формирования требований?	Уровень знания прикладной области, ограничивает максимальный уровень качества ПО, которое можно сделать. Если понимать все тонкости, то получится прекрасное приложение. На данном этапе изучаются производственные детали объекта автоматизации. Вся информация и технологических цепочках и процессах попадает на этом этапе в разрабатываемую информационную систему. Например, программисту занимающемуся разработкой на 1С важно иметь познания в бухгалтерии и т.п.	5
9.		Критерии плохой архитектуры программного продукта.	Жёсткость – есть антипод гибкости, означает что ПО тяжело изменить. Хрупкость. Этот критерий захватывает и гибкость и расширяемость. Если вам нужно что-то изменить в системе, жёсткость говорит вам «это сделать тяжело». Хрупкость говорить, что даже если что-то было тяжело настроить, то всё ещё и	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			развалится в другом месте. Если система не расширяемая, но вы смогли как-то это преодолеть, то половина модулей может работать некорректно. В идеале должно быть просто вносить изменения и это ничего не должно ломать в уже написанном коде. Неподвижность. Связано с возможностью повторного использования. Если у вас система неподвижна, то возможности повторного использования нет. Это означает что некий модуль из системы нельзя вырезать и запустить «stand alone». Это требует сделать кучу лишних действий. Идеально изъятый модуль, выступает в виде «чёрного ящика», который принимает вход в формате «А», выдаёт в формате «Б». Другими словами, чтобы запустить систему целиком, нужно запустить все модули. Чтобы запустить отдельный модуль, не нужно «тащить» всю систему следом.	
10.		Инкрементная и Итеративная методологии.	В инкрементной модели основные требования определены изначально, однако детали могут дорабатываться в процессе разработки. Продукт можно рано вывести на рынок. В итеративной модели начальные требования определены, дополнительные требования могут изменяться в процессе. Модуль, написанный один раз, может изменяться многократно.	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			Подходит для очень больших проектов. Если ваш проект рассчитан на 5 лет, но уже через год хочется посмотреть, как же он будет выглядеть и работать и нужен ли он рынку В этих подходах проходят те же самый шаги, что и в каскадных моделях, за исключением внедрения и поддержки. Как привило в итеративных моделях происходит некая оценка результатов. Вы приносите заказчику вашу версию, и он либо: • закрывает ваш проект (в худшем случае) • либо говорит, что продукт можно выпустить на рынок • либо что его можно выпустить в качестве альфаверсии • либо говорит что нужно поправить недочёт, и вот тогда выпустить на рынок. Т.е. на этапе тестирования, шаги зацикливаются. Другими словами ваш проект будет крутиться по итерациям или инкрементациям и в какой-то момент выйдет в релиз или нет.	

Полный комплект оценочных материалов по дисциплине (модулю) (фонд оценочных средств) хранится в электронном виде на кафедре, утверждающей рабочую программу дисциплины (модуля), и в Центре мониторинга и аудита качества обучения.

7.4 Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю)

Итоговая оценка по промежуточной аттестации выставляется в соответствии с Положением АГУ о балльно-рейтинговой системе (БАРС). Итоговая оценка складывается из баллов, полученных студентов за текущую успеваемость в течение семестра и баллов, полученных студентом на зачетном занятии/экзамене.

В течение семестра студент может набрать максимально 50 баллов за выполнение аудиторной и самостоятельной работы. На экзамене студент может набрать максимально 50 баллов.

Экзамен проходит в форме устного собеседования со студентом по билетам, составленным из вопросов (п. 7.3). Один билет включает в себя 2 вопроса. Выбор билета осуществляется в случайном порядке. На подготовку студенту отводится не менее 40 мин. Во время проведения экзамена студенту запрещено пользоваться сотовым телефоном и иными средствами связи, персональным компьютером, сетью Интернет, заготовленными заранее ответами и т.п.

Таблица 10 – Технологическая карта рейтинговых баллов по дисциплине (модулю)

№ п/п	Контролируемые мероприятия	Количество мероприятий / баллы	Максимальное количество баллов	Срок представления
Основной блок				
1.	*Выполнение всех лабораторных работ*	6/50	50	До конца семестра
Всего			**50**	
Блок бонусов				
2.	*Своевременное выполнение всех заданий*	6/10	10	
Всего			**10**	
Дополнительный блок**				
3.	*Экзамен*	1/50	50	
Всего			**50**	
ИТОГО			**100**	

Таблица 11 – Система штрафов (для одного занятия)

Показатель	Балл
Пропуск занятия без уважительной причины	-1 балл

Таблица 12 – Шкала перевода рейтинговых баллов в итоговую оценку за семестр по дисциплине (модулю)

Сумма баллов	Оценка по 4-балльной шкале	
90–100	5 (отлично)	Зачтено
85–89	4 (хорошо)	
75–84		
70–74		
65–69	3 (удовлетворительно)	
60–64		
Ниже 60	2 (неудовлетворительно)	Не зачтено

При реализации дисциплины (модуля) в зависимости от уровня подготовленности обучающихся могут быть использованы иные формы, методы контроля и оценочные средства, исходя из конкретной ситуации.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1. Основная литература

1. Камаев, В.А. Технологии программирования : Доп. М-вом образования РФ в качестве учебника для вузов по специальности "Информатика и вычислительная техника" . - М. : Высш. шк., 2005. - 359 с.
2. Кергаль, И. Методы программирования на Бейсике (с упражнениями) / пер. с фр. А.И. Бряндинского; Под ред. Ю.М. Баяковского. - М. : Мир, 1991. - 288 с.
3. Давыдов, В.Г. Технологии программирования С++ : рек. УМО вузов РФ по образованию в области радиотехники, электроники, биомедицинской техники и автоматизации в качестве учеб. пособ. для студентов вузов, обучающихся по специальности 210100 "Управление и информатика в технических системах". - СПб. : БХВ-Петербург, 2005. - 672 с.+1 электрон. диск (CD-ROM).
4. Камаев, В.А. Технологии программирования : доп. М-вом образования и науки РФ в качестве учебника для студентов вузов ... "Информатика и вычислительная техника". - изд. 2-е ; перераб. и доп. - М. : Высш. шк., 2006. - 454 с.
5. Макарова, Н.В. Информатика : рек. УМО ... в качестве учебника для студентов вузов, обучающихся по направлениям подготовки бакалавров "Системный анализ и управление" и "Экономика и управление". - СПб. : Питер, 2013. - 573, [3] с. : ил. - (Учебник для вузов. Стандарт третьего поколения).

8.2. Дополнительная литература

6. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс]учебное пособие / Златопольский Д. М. - 3-е изд. (эл.). - М. : БИНОМ, 2015. Режим доступа: http://www.studentlibrary.ru/book/ISBN9785996329328.html
7. Программирование [Электронный ресурс]: учеб. пособие / Зайцев М.Г. - Новосибирск : Изд-во НГТУ, 2015. Режим доступа: http://www.studentlibrary.ru/book/ISBN9785778226265.html
8. Программирование на Clojure [Электронный ресурс]монография / Эмерик Ч., Карпер Б., Гранд К. - М. : ДМК Пресс, 2015. Режим доступа: http://www.studentlibrary.ru/book/ISBN9785970602997.html
9. Программирование в алгоритмах [Электронный ресурс] / С.М. Окулов. - 5-е изд. (эл.). - М. : БИНОМ, 2014. - (Развитие интеллекта школьников). Режим доступа: http://www.studentlibrary.ru/book/ISBN9785996323111.html
10. Программирование [Электронный ресурс]учебное пособие / Н.А. Давыдова, Е.В. Боровская. - М. : БИНОМ, 2015. Режим доступа: http://www.studentlibrary.ru/book/ISBN9785996326471.html

8.3. Интернет-ресурсы, необходимые для освоения дисциплины (модуля)

1. Электронный каталог Научной библиотеки АГУ на базе MARKSQL НПО «Информ-систем».
   https://library.asu.edu.ru
2. Электронная библиотека «Астраханский государственный университет» собственной генерации на электронной платформе ООО «БИБЛИОТЕХ».
   https://biblio.asu.edu.ru
3. Электронно-библиотечная система (ЭБС) ООО «Политехресурс» «Консультант студента». Многопрофильный образовательный ресурс «Консультант студента» является электронной библиотечной системой, предоставляющей доступ через сеть Интернет к учебной литературе и дополнительным материалам, приобретенным на основании прямых договоров с правообладателями. Каталог в настоящее время содержит около 15000 наименований.
   www.studentlibrary.ru.

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для проведения лабораторных занятий необходима аудитория, оснащенная компьютерными рабочими местами студентов и доступом в Интернет.

Рабочая программа дисциплины (модуля) при необходимости может быть адаптирована для обучения (в том числе с применением дистанционных образовательных технологий) лиц с ограниченными возможностями здоровья, инвалидов. Для этого требуется заявление обучающихся, являющихся лицами с ограниченными возможностями здоровья, инвалидами, или их законных представителей и рекомендации психолого-медико-педагогической комиссии. Для инвалидов содержание рабочей программы дисциплины (модуля) может определяться также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).