

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. Н. Марьенков

«02» июня 2022 г.

УТВЕРЖДАЮ
Заведующий кафедрой ЦТ



А. Н. Марьенков

«02» июня 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

Составитель

**Карпенко А.В., старший преподаватель кафедры
цифровых технологий**

Направление подготовки /
специальность

**09.03.02 ИНФОРМАЦИОННЫЕ СИСТЕМЫ
И ТЕХНОЛОГИИ**

Направленность (профиль) ОПОП

Квалификация (степень)

бакалавр

Форма обучения

очная

Год приема

2022

Курс

2

Семестр(ы)

4

Астрахань – 2022 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 направлена на приобретение общекультурных компетенций / общепрофессиональных компетенций / профессиональных компетенций.

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-3 - способность применять языки, системы и инструментальные средства программирования в профессиональной деятельности; ОПК-8 - способность к освоению новых образцов программных, технических средств и информационных технологий

б) профессиональных (ПК): ПК-10 - способность применять знания в области электроники и схемотехники, технологий, методов и языков программирования, технологий связи и передачи данных при разработке программно- аппаратных компонентов защищенных автоматизированных систем в сфере профессиональной деятельности.

Таблица 1. Декомпозиция результатов обучения

Код и наименование компетенции	Планируемые результаты обучения по дисциплине (модулю)		
	Знать (1)	Уметь (2)	Владеть (3)
ОПК-3	ИОПК-3.1.1 общие сведения о методах проектирования, документирования, разработки, тестирования и отладки программного обеспечения; ИОПК-3.1.2 основные сведения о базовых структурах данных.	ИОПК-3.2.1 реализовывать базовые алгоритмы с использованием библиотек современных языков программирования; ИОПК-3.2.2 формировать 3 требования и разрабатывать внешние спецификации для разрабатываемого программного обеспечения.	ИОПК-3.3.1 навыками проектирования программного обеспечения с использованием средств автоматизации; ИОПК-3.3.2 навыками разработки программной документации.
ОПК-8	ИОПК-8.1.1 способы инсталляции и механизмы администрирования.	ИОПК-8.2.1 производить выбор программных средств для проектирования, тестирования и документирования программного обеспечения.	ИОПК-8.3.1 навыками проектирования программного обеспечения для различных аппаратно-программных платформ.
ПК-10	ИОПК-10.1.1 структуру, состав и свойства	ИОПК-10.2.1 проектировать структуру и	ИОПК-10.3.1 навыками выбора эффективных

Код и наименование	Планируемые результаты обучения по дисциплине (модулю)		
	Знать (1)	Уметь (2)	Владеть (3)
	информационных процессов, систем и технологий, методы анализа информационных систем, модели представления проектных решений, конфигурации информационных систем.	архитектуру программного обеспечения с использованием современных методологий и средств автоматизации проектирования программного обеспечения.	способов реализации структур данных и конкретных алгоритмов при решении профессиональных задач.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Объем дисциплины (модуля) **4 зачетных единиц**, 144 часа, на контактную работу обучающихся с преподавателем (лекции, лабораторные работы) выделено 54 часов и на самостоятельную работу 90 часов.

Таблица 2. Структура и содержание дисциплины (модуля)

Раздел, тема дисциплины (модуля)	Семестр	Контактная работа (в часах)		Сам. работа	Форма текущего контроля успеваемости, форма промежуточной аттестации [по семестрам]
		Л	ЛР		
Тема 1. Компиляторы	5	1	-	4	Устный опрос; Устный опрос на экзамене
Тема 2. Методологии программирования.	5	1	-	4	Устный опрос; Устный опрос на экзамене
Тема 3. Этапы проектирования	5	2	6	8	Лабораторная работа 1; Устный опрос на экзамене
Тема 4. Документирование кода и гибкость разработки	5	1	-	8	Устный опрос; Устный опрос на экзамене
Тема 5. Критерии хорошей архитектуры	5	1	-	8	Устный опрос; Устный опрос на экзамене
Тема 6. Стандартные техники отладки	5	1	6	6	Лабораторная работа 2; Устный опрос на экзамене
Тема 7. Отладка и тестирование	5	2	-	8	Устный опрос; Устный опрос на экзамене
Тема 8 Исключения	5	1	6	8	Лабораторная работа 3; Устный опрос на экзамене
Тема 9. Модульное тестирование	5	1	6	8	Лабораторная работа 4; Устный опрос на экзамене

Тема 10. Паттерны проектирования	5	2	6	8	Лабораторная работа 5; Устный опрос на экзамене
Тема 11. Методологии разработки	5	1	-	4	Устный опрос; Устный опрос на экзамене
Тема 12. Типы пользовательских интерфейсов и этапы их разработки	5	2	6	8	Лабораторная работа 6; Устный опрос на экзамене
Тема 13. Применение технологий программирования	5	2	-	8	Устный опрос на экзамене
Итого		18	36	90	ЭКЗАМЕН

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	Компетенции			
		ОПК 3	ОПК 8	ПК-10	общее количество компетенций
Тема 1. Компиляторы	5	+	+	+	3
Тема 2. Методологии программирования.	5	+	+	+	3
Тема 3. Этапы проектирования	16	+	+	+	3
Тема 4. Документирование кода и гибкость разработки	9	+	+	+	3
Тема 5. Критерии хорошей архитектуры	9	+	+	+	3
Тема 6. Стандартные техники отладки	13	+	+	+	3
Тема 7. Отладка и тестирование	10	+	+	+	3
Тема 8 Исключения	15	+	+	+	3
Тема 9. Модульное тестирование	15	+	+	+	3
Тема 10. Паттерны проектирования	16	+	+	+	3
Тема 11. Методологии разработки	5	+	+	+	3
Тема 12. Типы пользовательских интерфейсов и этапы их разработки	16	+	+	+	3
Тема 13. Применение технологий программирования	10	+	+	+	3
ИТОГО	144				

Краткое содержание каждой темы дисциплины.

Тема 1. Компиляторы

Данная тема посвящена этапам 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 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.

Тема.2 Методологии программирования.

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

Тема 3. Этапы проектирования ПО

В списке этапов не указана разработка ПО, потому что эти шаги имеют отношение непосредственно к проектированию. Разработка находится там внутри них на определяемой разработчиком шаге:

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 достаточно строгим, 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.

Тема 4. Документирование кода и гибкость разработки

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.

Тема 5. Критерии эффективной архитектуры программных продуктов

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 сопряжённость, 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.

Тема 6. Стандартные техники отладки.

На данном занятии речь идёт об поговорим 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.

Тема 7. Тестирование и отладка программного обеспечения.

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 попытке чтения из несуществующего на диске файла). Методы отладки программного обеспечения – статические и динамические.

Тема 8. Исключения.

Представьте, что на автомобильной дороге есть участок с аварийным мостом с ограниченной грузоподъемностью. Если по нему поедет автомобиль с массой, превышающей грузоподъемность моста, он может разрушиться. Чтобы этого не произошло, дорожная служба заблаговременно устанавливает предупредительные знаки на дороге. Водитель автомобиля, глядя на предупреждающий знак, будет сравнивать массу своего автомобиля с разрешенной для проезда по мосту. Если она превышает ее – он поедет по объездному пути. Возможность предупреждения и разрешения исключительной ситуации в программе для ее продолжения – одна из причин использования исключений в Java. Механизм исключений также позволяет защитить написанный вами код (программный интерфейс) от неправильного использования пользователем за счет валидации (проверки) входящих данных.

Тема 9. Модульное тестирование

Процесс проектирования программного обеспечения, имеет решающее значение для создания эффективной программы. Инженеры-программисты используют мини-процессы проектирования внутри более крупного процесса. Например, инженеры-программисты должны анализировать, проектировать, внедрять и тестировать компоненты тестирования, чтобы правильно тестировать разрабатываемое программное обеспечение. На этапе анализа вы решаете, идентифицируете и формулируете проблему. Вы определяете ограничения и переменные. Обладая этими знаниями, вы переходите к этапу проектирования, на котором вы решаете, какие алгоритмы и управляющие структуры могут работать для решения проблемы. Чтобы реализовать проект, инженер-программист пишет код, используя правильный синтаксис для конкретного используемого языка. Следующий этап, тестирование, имеет решающее значение для процесса проектирования. На этой лекции вы научитесь тщательно тестировать программу на Java. Цель лекции - научиться спроектировать и реализовать тестовый пример с помощью JUnit, инструмента тестирования программного обеспечения Java.

Тема 10. Паттерны проектирования.

Это повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста. Обычно шаблон не является законченным образцом, который может быть прямо преобразован в код; это лишь пример решения задачи, который можно использовать в различных ситуациях. Объектно-ориентированные шаблоны показывают отношения и взаимодействия между классами или объектами, без определения того, какие конечные классы или объекты приложения будут использоваться.

Порождающие шаблоны — шаблоны проектирования, которые абстрагируют процесс инстанцирования. Они позволяют сделать систему независимой от способа создания, композиции и представления объектов. Шаблон, порождающий классы, использует наследование, чтобы изменять инстанцируемый класс, а шаблон, порождающий объекты, делегирует инстанцирование другому объекту.

Структурные шаблоны определяют различные сложные структуры, которые изменяют интерфейс уже существующих объектов или его реализацию, позволяя облегчить разработку и оптимизировать программу.

Тема 11. Методологии разработки.

Этапы развития технологии программирования. Модели жизненного цикла программного обеспечения

- Каскадная модель жизненного цикла программного обеспечения (водопад)
- Каскадная модель с промежуточным контролем (водоворот)
- V модель (разработка через тестирование)
- Модель на основе разработки прототипа
- Спиральная модель жизненного цикла программного обеспечения

Понятие эффективности и технологичности программного обеспечения. Модули и их свойства. Средства описания структурных алгоритмов (псевдокоды, блок-схемы)

Тема 5. 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

Тема 13. Применение технологий программирования

На данном занятии речь идёт про принципы SOLID. Для каждого из них приводится пример программного кода. Так же речь идёт о применимости фреймворков разработки, особенностях REST-сервисов.

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.

Учебные ресурсы и тренинги ресурса Intuit.ru, который обеспечивает большой спектр курсов в области компьютерных наук для образовательных учреждений.

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.

Таблица 4. Содержание самостоятельной работы обучающихся

Вопросы, выносимые на самостоятельное изучение	Кол-во часов	Формы работы
Тема 1. Компиляторы. Методы трансляции кода с языка высокого уровня на низкоуровневый язык	4	Изучение учебной литературы, подготовка к занятиям
Тема 2. Методологии программирования. Способы применения логического программирования в веб-разработке	4	Изучение учебной литературы, подготовка к занятиям
Тема 3. Этапы проектирования. Структура документа технического задания по ГОСТ. Уличение аналогов по выбранной теме в рамках лабораторной работы.	8	Выполнение лабораторной работы. Изучение учебной литературы, подготовка к занятиям
Тема 4. Документирование кода и гибкость разработки. Создание фала документации кода для своей курсовой	8	Изучение учебной литературы,

работы.		подготовка к занятиям
Тема 5. Критерии хорошей архитектуры. Анализ предложенного проекта и предложение способов модифицирования его архитектуры.	8	Изучение учебной литературы, подготовка к занятиям
Тема 6. Стандартные техники отладки. Отработка способов отладки ошибок с помощью разного уровня логгирования.	6	Выполнение лабораторной работы. Изучение учебной литературы, подготовка к занятиям
Тема 7. Отладка и тестирование. Анализ кода на пример логических противоречий.	8	Изучение учебной литературы, подготовка к занятиям
Тема 8 Исключения. Выполнение практического задания по созданию собственных классов пользовательских исключений.	8	Выполнение лабораторной работы. Изучение учебной литературы, подготовка к занятиям
Тема 9. Модульное тестирование. Создание классов модульного тестирования в среде разработки Visual Studio.	8	Выполнение лабораторной работы. Изучение учебной литературы, подготовка к занятиям
Тема 10. Паттерны проектирования. Применение порождающего паттерна Builder вместе с поведенческим паттерном «Стратегия»	8	Выполнение лабораторной работы. Изучение учебной литературы, подготовка к занятиям
Тема 11. Методологии разработки. Изучение автоматизированных систем управления процессом разработки в рамках методологии Agile и Scrum.	4	Изучение учебной литературы, подготовка к занятиям
Тема 12. Типы пользовательских интерфейсов и этапы их разработки. Разработка схем пользовательских маршрутов.	8	Выполнение лабораторной работы. Изучение учебной литературы, подготовка к занятиям
Тема 13. Применение технологий программирования. Модификация кода с учетом выполнения принципов SOLID.	8	Изучение учебной литературы, подготовка к занятиям

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. В практическом задании, требуется описать вариант технического задания по одной из предложенных тем. Созданные согласно заданиям программы, архивируется и прикрепляется в виде ответа на задание.

6. ОБРАЗОВАТЕЛЬНЫЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

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 нестандартных ситуаций и реагирования на них.

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

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. Компиляторы	Лекционное занятие	Не предусмотрено	Фронтальный опрос, выполнение практических заданий, тематические дискуссии
Тема 2. Методологии программирования.	Лекционное занятие	Не предусмотрено	Фронтальный опрос, выполнение

			практических заданий, тематические дискуссии
Тема 3. Этапы проектирования	Лекционное занятие	Не предусмотрено	Лабораторная работа 1;
Тема 4. Документирование кода и гибкость разработки	Лекционное занятие	Не предусмотрено	Тематические дискуссии, анализ конкретных ситуаций
Тема 5. Критерии хорошей архитектуры	Лекционное занятие	Не предусмотрено	Тематические дискуссии, анализ конкретных ситуаций
Тема 6. Стандартные техники отладки	Лекционное занятие	Не предусмотрено	Лабораторная работа 2;
Тема 7. Отладка и тестирование	Лекционное занятие	Не предусмотрено	Тематические дискуссии, анализ конкретных ситуаций
Тема 8 Исключения	Лекционное занятие	Не предусмотрено	Лабораторная работа 3;
Тема 9. Модульное тестирование	Лекционное занятие	Не предусмотрено	Лабораторная работа 4;
Тема 10. Паттерны проектирования	Лекционное занятие	Не предусмотрено	Лабораторная работа 5;
Тема 11. Методологии разработки	Лекционное занятие	Не предусмотрено	Тематические дискуссии, анализ конкретных ситуаций
Тема 12. Типы пользовательских интерфейсов и этапы их разработки	Лекционное занятие	Не предусмотрено	Лабораторная работа 6;
Тема 13. Применение технологий программирования	Лекционное занятие	Не предусмотрено	Тематические дискуссии, анализ конкретных ситуаций

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 «Алгоритмы и структура данных» 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 Eclipse. Среда разработки Java используется в рассматриваемом курсе для изучения алгоритмов и структур данных, а так же принципов объектно-ориентированного программирования.

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 «Технологии и методы 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. Компиляторы	ОПК-3, ОПК-8; ПК-10	Вопросы для обсуждения; устный

		опрос на экзамене
Тема 2. Методологии программирования.	ОПК-3, ОПК-8; ПК-10	Вопросы для обсуждения; устный опрос на экзамене
Тема 3. Этапы проектирования	ОПК-3, ОПК-8; ПК-10	Лабораторная работа 1; устный опрос на экзамене
Тема 4. Документирование кода и гибкость разработки	ОПК-3, ОПК-8; ПК-10	Вопросы для обсуждения; устный опрос на экзамене
Тема 5. Критерии хорошей архитектуры	ОПК-3, ОПК-8; ПК-10	Вопросы для обсуждения; устный опрос на экзамене
Тема 6. Стандартные техники отладки	ОПК-3, ОПК-8; ПК-10	Лабораторная работа 2; устный опрос на экзамене
Тема 7. Отладка и тестирование	ОПК-3, ОПК-8; ПК-10	Вопросы для обсуждения; устный опрос на экзамене
Тема 8 Исключения	ОПК-3, ОПК-8; ПК-10	Лабораторная работа 3; устный опрос на экзамене
Тема 9. Модульное тестирование	ОПК-3, ОПК-8; ПК-10	Лабораторная работа 4; устный опрос на экзамене
Тема 10. Паттерны проектирования	ОПК-3, ОПК-8; ПК-10	Лабораторная работа 5; устный опрос на экзамене
Тема 11. Методологии разработки	ОПК-3, ОПК-8; ПК-10	Вопросы для обсуждения; устный опрос на экзамене
Тема 12. Типы пользовательских интерфейсов и этапы их разработки	ОПК-3, ОПК-8; ПК-10	Лабораторная работа 6; устный опрос на экзамене
Тема 13. Применение технологий программирования	ОПК-3, ОПК-8; ПК-10	Вопросы для обсуждения; 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 3. Этапы проектирования

1. Лабораторная работа №1 «Описание ТЗ»

1) Подготовить обоснование и техническое задание на проект. Показатели технико-экономического обоснования должны быть обоснованы. В качестве варианта предметной области допускается интересная вам область в которой возможна автоматизация работы.

2) _____ Разрешается выбрать любой интересный вам формат поставки: мобильное приложение, десктопное или веб-сайт.

3) Выполненное задание приложить в виде документа формата Word..

Тема 6. Стандартные техники отладки

1. Лабораторная работа №2 «Тестирование и отладка»

1) Выполните пошагово задания начиная от создания или импорта проекта.

_____ Проанализировать ком программ, приведённых в заданиях, проследите какой они должны

давать результат.

2) _____ Организуйте логгирование своего проекта, ознакомьтесь с различными уровнями логгирования.

3) Выполните отладку проекта с помощью установки точек останова.

Тема 8. Исключения

1. Лабораторная работа №3 «Исключения в Java»

1) Изучить синтаксис операторов.

2) Скомпилировать примеры программ с различными типами исключений.

3) Проанализировать ситуации когда допустимы те или иные исключения.

Тема 9. Модульное тестирование

1. Лабораторная работа №4 «JUnit»

1) В рамках данной работы требуется для приложенного проекта создать тестовый класс и реализовать в нём методы тестирования для всех методов исходного класса

2) Выполненный проект прикрепить в формате архива .zip

Тема 10. Паттерны проектирования

1. Лабораторная работа №5 «Паттерны проектирования»

1) Изучить теоретическую составляющую темы. Порождающие и поведенческие паттерны

2) Пошагово создать проекты, продемонстрированные на занятии.

3) Создать собственный проект, реализующий порождающий паттерн проектирования (Factory method или Abstract factory).

Тема 12. Типы пользовательских интерфейсов и этапы их разработки

Требуется:

1) Выбрать уже существующие приложение, веб-сайт или придумать свой собственный проект, например приложение для аренды или покупки жилья.

2) Построить пользовательские пути данного проекта

3) Необходимо представить как блок-схему в нотации UML так и схему с представлением окон(экранов) с указанием и описание всех переходов состояний

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-3, ОПК-8; ПК-10				
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 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	3	1

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		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 или формального языка с его формальной грамматикой. Семантический анализ - этап в последовательности действий алгоритма автоматического понимания текстов, заключающийся в выделении семантических отношений, формировании	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			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 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"	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			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 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	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			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. 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 самые шаги, что и в каскадных моделях, за исключением внедрения и	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			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 студентами 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.	<i>Выполнение всех лабораторных работ</i>	6/50	50	До конца семестра
Всего			50	
Блок бонусов				
2.	<i>Своевременное выполнение всех заданий</i>	6/10	10	
Всего			10	
Дополнительный блок**				
3.	<i>Экзамен</i>	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 C++ : 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).