

МИНПРОСВЕЩЕНИЯ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
"Тульский государственный педагогический университет им. Л.Н. Толстого"
(ФГБОУ ВО "ТГПУ им. Л.Н. Толстого")

**МОДУЛЬ "АЛГОРИТМИЗАЦИЯ И
ПРОГРАММИРОВАНИЕ"**
Методы и технологии программирования

рабочая программа дисциплины (модуля)

Закреплена за кафедрой	институт передовых информационных технологий
ОПОП	Направление 09.03.03 Прикладная информатика направленность (профиль) Прикладная информатика в здравоохранении
Квалификация	Бакалавр
Год начала подготовки	2023
Форма обучения	очная
Общая трудоемкость	6 з.е.

Виды контроля по семестрам:
экзамен 4
курсовая работа 4

Семестр(Курс.Номер семестра на курсе)	4(2.2)		Итого	
	УП	РПД	УП	РПД
Лекции	34	34	34	34
Лабораторные	34	34	34	34
Итого ауд.	68	68	68	68
КСР	4	4	4	4
Контактная работа	72	72	72	72
Сам. работа	108	108	108	108
Часы на контроль	36	36	36	36
Практическая подготовка	0	0	0	0
Семинары	0	0	0	0
Консультации	0	0	0	0
Итого трудоемкость в часах	216	216	216	216

Программу составил(и):

к.т.н., доцент, Шмелев Алексей Николаевич

Рабочая программа дисциплины

Методы и технологии программирования

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

составлена на основании учебного плана:

Направление 09.03.03 Прикладная информатика
направленность (профиль) Прикладная информатика в здравоохранении
утвержденного Учёным советом вуза от 27.10.2022 протокол № 13.

РПД утверждена Учёным советом университета
протокол от 27.10.2022 г. № 13

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель преподавания дисциплины - изучение и практическое освоение общих принципов и современных методов технологии программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.О.06
2.1	Требования к предварительной подготовке обучающегося:
1.	Алгоритмы и структуры данных
2.	Базы данных
3.	Веб-программирование
4.	Операционные системы
5.	Практикум по программированию
6.	Учебная практика по получению первичных навыков научно-исследовательской работы
7.	Численные методы
8.	Объектно-ориентированное программирование
9.	Программирование
10.	Учебная ознакомительная практика
11.	Алгоритмизация и программирование
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
1.	Математические основы моделирования экономических процессов
2.	Программирование мобильных приложений на языке Kotlin
3.	Производственная эксплуатационная практика
4.	Администрирование информационных систем
5.	Машинное обучение
6.	Производственная практика: научно-исследовательская работа

3. СООТНЕСЕНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) С ИНДИКАТОРАМИ ДОСТИЖЕНИЯ КОМПЕТЕНЦИЙ

3.1 Компетенции обучающегося и индикаторы их достижения:

ОПК-5: Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	
ОПК-5.1	Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем
	Знает принципы и методы анализа, проектирования и разработки сложных программных средств
ОПК-5.2	Умеет выполнять параметрическую настройку информационных и автоматизированных систем
	умеет ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы; проектировать, конструировать и отлаживать программные средства в соответствии с заданными критериями качества и стандартами
ОПК-5.3	Владеет навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем
	имеет навык применения библиотек классов и инструментальных средствах при проектировании и разработке программного обеспечения
ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения	
ОПК-7.1	Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий
	Основные стандарты в области инфокоммуникационных систем и технологий, в том числе стандарты Единой системы программной документации Основы объектно-ориентированного подхода к программированию
ОПК-7.2	Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ
	Работать с современными системами программирования, включая объектно-ориентированные Применять математический аппарат для анализа сложности алгоритмов Проектировать, конструировать и отлаживать программные средства в соответствии с заданными критериями качества и стандартами
ОПК-7.3	Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач

анализа предметной области сложных программных систем на концептуальном, логическом, математическом и алгоритмическом уровнях.	
3.2 Результаты обучения по дисциплине:	
В результате освоения дисциплины обучающийся должен:	
	Знать:
3.1	Основные стандарты в области инфокоммуникационных систем и технологий, в том числе стандарты Единой системы программной документации
3.2	Основы объектно-ориентированного подхода к программированию.
3.3	Принципы и методы анализа, проектирования и разработки сложных программных средств
	Уметь:
У.1	Ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы
У.2	Работать с современными системами программирования, включая объектно-ориентированные
У.3	Применять математический аппарат для анализа сложности алгоритмов
У.4	Проектировать, конструировать и отлаживать программные средства в соответствии с заданными критериями качества и стандартами
	Владеть:
В.1	применения библиотек классов и инструментальных средствах при проектировании и разработке программного обеспечения.
В.2	анализа предметной области сложных программных систем на концептуальном, логическом, математическом и алгоритмическом уровнях.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	Содержание
	Тема 1. Введение в технологии программирования ПО.				
1.1	Введение в технологии программирования ПО /Лек/	4	10	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	Системный подход при разработке ПС. Модели разработки: каскадная, с промежуточным контролем, спиральная и т.д.; CASE и RAD-технологии. Жизненный цикл программного сред-ства. Управление проектом, планирование и распределение ресурсов, контроль исполнения сроков. Парадигмы программирования: визуальная, функциональная, процедурная, объектно-ориентированная и т.д. Объектно-ориентированная парадигма: понятия объекта, класса объектов; основные понятия объектно-ориентированного программирования (инкапсуляция, наследование и полиморфизм); классы и объекты; интерфейсы и реализация.
1.2	Введение в технологии программирования ПО /Лаб/	4	8	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	Системный подход при разработке ПС. Модели разработки: каскадная, с промежуточным контролем, спиральная и т.д.; CASE и RAD-технологии. Жизненный цикл программного сред-ства. Управление проектом, планирование и распределение ресурсов, контроль исполнения сроков. Парадигмы программирования: визуальная, функциональная, процедурная, объектно-ориентированная и т.д. Объектно-ориентированная парадигма: понятия объекта, класса объектов; основные понятия объектно-ориентированного программирования (инкапсуляция, наследование и полиморфизм); классы и объекты; интерфейсы и реализация.

1.3	Введение в технологии программирования ПО /Ср/	4	28	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	Системный подход при разработке ПС. Модели разработки: каскадная, с промежуточным контролем, спиральная и т.д.; CASE и RAD-технологии. Жизненный цикл программного сред-ства. Управление проектом, планирование и распределение ресурсов, контроль исполнения сроков. Парадигмы программирования: визуальная, функциональная, процедурная, объектно-ориентированная и т.д. Объектно-ориентированная парадигма: понятия объекта, класса объектов; основные понятия объектно-ориентированного программирования (инкапсуляция, наследование и полиморфизм); классы и объекты; интерфейсы и реализация.
	Тема 2. Организация процесса проектирования программного обеспечения.				
2.1	Организация процесса проектирования программного обеспечения. /Лек/	4	8	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	Системный подход при разработке ПС. Модели разработки: каскадная, с промежуточным контролем, спиральная и т.д.; CASE и RAD-технологии. Тестирование и оценка качества. Управ-ление проектом, планирование и распределение ресурсов, контроль исполнения сроков. Исполь-зование декомпозиции и абстракции при проектировании ПО. Спецификация процедур и данных. Внешняя и внутренняя спецификации. Декомпозиция задачи. Методы проектирования структуры ПО. Методы защиты программ и данных.
2.2	Организация процесса проектирования программного обеспечения. /Лаб/	4	8	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	Системный подход при разработке ПС. Модели разработки: каскадная, с промежуточным контролем, спиральная и т.д.; CASE и RAD-технологии. Тестирование и оценка качества. Управ-ление проектом, планирование и распределение ресурсов, контроль исполнения сроков. Исполь-зование декомпозиции и абстракции при проектировании ПО. Спецификация процедур и данных. Внешняя и внутренняя спецификации. Декомпозиция задачи. Методы проектирования структуры ПО. Методы защиты программ и данных.
2.3	Организация процесса проектирования программного обеспечения. /Ср/	4	26	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	Системный подход при разработке ПС. Модели разработки: каскадная, с промежуточным контролем, спиральная и т.д.; CASE и RAD-технологии. Тестирование и оценка качества. Управ-ление проектом, планирование и распределение ресурсов, контроль исполнения сроков. Исполь-зование декомпозиции и абстракции при проектировании ПО. Спецификация процедур и данных. Внешняя и внутренняя спецификации. Декомпозиция задачи. Методы проектирования структуры ПО. Методы защиты программ и данных.
	Тема 3. Технологии коллективной разработки программного обеспечения.				

3.1	Технологии коллективной разработки программного обеспечения /Лек/	4	8	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	<p>Библиотеки стандартных компонентов, библиотеки объектов. Проектирование интерфейса с пользователем. Структуры диалога; поддержка пользователя; многооконные интерфейсы; при-меры реализации интерфейсов с пользователем с использованием графических пакетов. «Заглуш-ки». «Маленькие хитрости» в программировании. Статические, полустатические и динамические типы данных. Простые и составные типы данных, операция квалификации. Технологии распреде-ленных вычислений: RPC, RMI, Corba, DCOM.</p> <p>Инструментальная среда разработки. Библиотека VCL. Средства поддержки проекта. От-ладчики. CASE-технология. UML-диаграммы.</p> <p>Обзор и классификация средств поддержки коллективной разработки ПО. Программные средства планирования и управления процессом разработки. Сетевые графики и диаграммы рабо-чего процесса. Сценарии выполнения работ. Применение систем управления документами. CASE-технологии.</p>
3.2	Технологии коллективной разработки программного обеспечения /Лаб/	4	8	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	<p>Библиотеки стандартных компонентов, библиотеки объектов. Проектирование интерфейса с пользователем. Структуры диалога; поддержка пользователя; многооконные интерфейсы; при-меры реализации интерфейсов с пользователем с использованием графических пакетов. «Заглуш-ки». «Маленькие хитрости» в программировании. Статические, полустатические и динамические типы данных. Простые и составные типы данных, операция квалификации. Технологии распреде-ленных вычислений: RPC, RMI, Corba, DCOM.</p> <p>Инструментальная среда разработки. Библиотека VCL. Средства поддержки проекта. От-ладчики. CASE-технология. UML-диаграммы.</p> <p>Обзор и классификация средств поддержки коллективной разработки ПО. Программные средства планирования и управления процессом разработки. Сетевые графики и диаграммы рабо-чего процесса. Сценарии выполнения работ. Применение систем управления документами. CASE-технологии.</p>

3.3	Технологии коллективной разработки программного обеспечения /Ср/	4	26	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	<p>Библиотеки стандартных компонентов, библиотеки объектов. Проектирование интерфейса с пользователем. Структуры диалога; поддержка пользователя; многооконные интерфейсы; при-меры реализации интерфейсов с пользователем с использованием графических пакетов. «Заглуш-ки». «Маленькие хитрости» в программировании. Статические, полустатические и динамические типы данных. Простые и составные типы данных, операция квалификации. Технологии распреде-ленных вычислений: RPC, RMI, Corba, DCOM.</p> <p>Инструментальная среда разработки. Библиотека VCL. Средства поддержки проекта. От-ладчики. CASE-технология. UML-диаграммы.</p> <p>Обзор и классификация средств поддержки коллективной разработки ПО. Программные средства планирования и управления процессом разработки. Сетевые графики и диаграммы рабо-чего процесса. Сценарии выполнения работ. Применение систем управления документами. CASE-технологии.</p>
	Тема 4. Методы отладки и тестирования программ.				
4.1	Методы отладки и тестирования программ. /Лек/	4	8	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	<p>Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Те-стирование на этапе проектирования. Тестирование "белого ящика" на стадии кодирования. Ре-грессионное тестирование. Тестирование "черного ящика". Разработка тестов.</p> <p>Документация, создаваемая в процессе разработки программных средств. ЕСПД. Пользователь-ская документация программных средств. Документация по сопровождению программных средств. Стандарт ISO 9126. Модель качества. Характеристики и субхарактеристики качества программного средства. Метрики качества программного средства. Оценивание характеристик качества программных средств.</p>
4.2	Методы отладки и тестирования программ. /Лаб/	4	10	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	<p>Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Те-стирование на этапе проектирования. Тестирование "белого ящика" на стадии кодирования. Ре-грессионное тестирование. Тестирование "черного ящика". Разработка тестов.</p> <p>Документация, создаваемая в процессе разработки программных средств. ЕСПД. Пользователь-ская документация программных средств. Документация по сопровождению программных средств. Стандарт ISO 9126. Модель качества. Характеристики и субхарактеристики качества программного средства. Метрики качества программного средства. Оценивание характеристик качества программных средств.</p>

4.3	Методы отладки и тестирования программ. /Ср/	4	28	Л1.1 Л1.2Л2.1 Л2.2 Л2.3	Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Тестирование на этапе проектирования. Тестирование "белого ящика" на стадии кодирования. Ре-грессионное тестирование. Тестирование "черного ящика". Разработка тестов. Документация, создаваемая в процессе разработки программных средств. ЕСПД. Пользователь-ская документация программных средств. Документация по сопровождению программных средств. Стандарт ISO 9126. Модель качества. Характеристики и субхарактеристики качества программного средства. Метрики качества программного средства. Оценивание характеристик качества программных средств.
4.4	/КСР/	4	4		

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Типовые задания для проведения текущего контроля

Примеры заданий для практических занятий

Разработать модель системы, построенную на архитектуре "клиент/сервер", автоматизирующую работу инфокоммуникационной системы учебного центра.

Методологической основой разработки является унифицированный процесс RUP. Каждое практическое занятие посвящено определенным фазам и основным потокам работ.

Разработать следующие диаграммы:

- вид с точки зрения функциональности – диаграммы прецедентов;
- вид с точки зрения проектирования – диаграммы классов (для структурного моделирования) и диаграммы взаимодействия (для моделирования поведения).

1. Практическое занятие 1. Спецификация требований и создание вариантов использования.
2. Практическое занятие 2. Разработка диаграммы взаимодействия.
3. Практическое занятие 3. Реализация иерархии классов

Примеры тестовых заданий

1. Жизненный цикл ПС — это период времени, который:

- а) начинается с момента пуска ПС в эксплуатацию и заканчивается при его изъятии;
- б) устанавливает ПС на определенном компьютере;
- в) начинается в момент завершения разработки ПС и заканчивается в момент приобретения средством полной несовместимости с современными ОС;
- г) начинается с момента принятия решений о необходимости создания ПС и заканчивается в момент его полного изъятия из эксплуатации .

2. Процесс приобретения ПС Включает следующее действие заказчика:

- а) тестирование;
- б) кодирование;
- в) надзор за деятельностью поставщика;
- г) проектирование архитектуры ПС.

3. Процесс поставки ПС включает следующее действие поставщика:

- а) подготовка договора;
- б) анализ требований к системе;
- в) анализ требований к ПС;
- г) квалификационное тестирование ПС.

4. Процесс проектирования не включает в себя:

- а) интеграцию системы;
- б) подготовительную работу;
- в) кодирование и тестирование системы;
- г) подготовку заявочного предложения.

5. Стадия создания ПС - это часть процесса создания ПС:

- а) заканчивающаяся завершением работ по кодированию ПС;
- б) завершающая работы по тестированию ПС;
- в) ограниченная некоторыми временными рамками и заканчивающаяся созданием (выпуском) конкретного продукта,

определяемого заданными для данной стадии требованиями;

г) заканчивающаяся после поставки ПС потребителям.

6. В состав жизненного цикла ПС не входит:

а) проектирование

б) реклама,

в) тестирование;

г) эксплуатация и сопровождение.

7. В принципы технологии создания ПС не входит:

а) многоуровневое тестирование ПС с использованием средств автоматизации тестирования;

б) структурированное построение программного средства и его компонентов;

в) эффективное использование ресурсов ЭВМ, на которых реализуется ПС;

г) использование интерактивного режима со средствами автоматизации и работы профессиональных ЭВМ.

8. Проекты, в которых требования формулируются пользователем называются проектами:

а) управляемые пользователем;

б) утверждаемые пользователем;

в) независимые от пользователя.

9. В задачи, возникающие в процессе разработки требований, не входит:

а) выявление пространственно-временных ограничений, налагаемых на систему, а также средств системы, которые в будущем могут претерпеть изменения;

б) выявление наличия информации, необходимой для выполнения планируемых функций;

в) обеспечение полноты и точности определения функций, подлежащих выполнению программным средством, и их взаимосвязи;

г) определение эффективности использования ресурсов системы.

10. Документ, являющийся результатом разработки требований должен быть достаточно полным, чтобы

а) обеспечить лёгкость серьёзных модификаций и пересмотра требований;

б) со 100% - ой гарантией исключить любые модификации и пересмотр требований;

в) впоследствии при разработке исключить серьёзные модификации и пересмотр требований.

11. Цели проекта – это цели, которые должны быть достигнуты

а) в процессе кодирования;

б) после окончания работы над ПС;

в) в процессе кодирования;

г) в процессе проектирования.

12. Цели ПС, рассматриваемые с точки зрения пользователя не включают следующую ин-формацию:

а) безопасность;

б) эффективность;

в) универсальность;

г) определение пользователя.

13. Какая информация включается в детальный внешний проект функции только по требованию самого пользователя

а) замечания по программированию;

б) описание входных данных;

в) преобразование системы;

г) характеристика надёжности.

14. Модуль – это отдельная, функционально законченная программная единица, которая:

а) может быть частью программы, но не может применяться самостоятельно;

б) не может применяться самостоятельно, либо быть частью программы;

в) может применяться самостоятельно, но не может быть частью программы;

г) может применяться самостоятельно, либо быть частью программы.

15. Признаком модуля не является следующий принцип:

а) модуль имеет определенную логическую структуру;

б) модуль реализует одну или несколько функций;

в) модуль используется в одном или нескольких контекстах;

г) модуль реализует всегда только одну функцию и используется только в одном контексте.

16. Связность – это мера:

а) взаимозависимости модулей по данным;

б) зависимости частей модуля;

в) независимости частей модуля;

г) независимости модулей друг от друга по данным.

17. Какой принцип из ниже-перечисленных относится к свойствам программного модуля:

- а) работа ПМ не должна зависеть от того, какому ПМ предназначены выходные данные;
- б) желательно наличие у ПМ спецификации;
- в) ПМ возвращает управление тому ПМ, какой укажет при разработке программист;
- г) не допускается вызов программным модулем другого ПМ.

18. Одним из шагов проектирования и кодирования логики модулей не является:

- а) предварительное тестирование;
- б) компиляция модулей;
- в) проверка правильности программы;
- г) проверка правильности внешних спецификаций модуля.

19. Основные рекомендации к стилю программирования не включают в себя:

- а) требование повышения эффективности программы путем ее усовершенствования до завершения отладки;
- б) использование программистом факторов языка программирования;
- в) стремление программиста повысить эффективность программы в результате тщательного анализа структур данных;
- г) требование удобочитаемости программы.

20. Стил программирования – это:

- а) собрание правил, которых обязан придерживаться каждый программист при написании приложений;
- б) четко очерченный кодекс «статей» программирования на наиболее распространенных языках (C++, Java и.т.п.);
- в) набор приемов и методов программирования, позволяющих создавать корректные, эффективные и доступные для чтения и понимания программы;
- г) выбор определенного языка программирования разработчиком при постановке перед ним определенной задачи.

21. Цель тестирования ПС – это:

- а) доказательство факта «безошибочности» программы;
- б) получение программы, функционирующей с требующимися характеристиками в заданной области изменения входных данных;
- в) выявление как можно большего числа ошибок в программе;
- г) конкретизация внешних взаимодействий программного средства без детализации внутреннего устройства.

22. Аттестация как одна из форм тестирования представляет собой:

- а) выполнение программ в реальной среде;
- б) попытку найти ошибку, выполняя программу в тестовой или моделируемой среде;
- в) авторитетное ее подтверждение правильности программы
- г) попытку найти ошибку безотносительно к внешней среде

23. Одной из аксиом тестирования является следующий принцип:

- а) крайне желательно, чтобы тестирование ПС выполнял сам его разработчик;
- б) хороший тест - тот, который демонстрирует правильную работу программы
- в) в целях облегчения тестирования программы можно провести в ней незначительные изменения;
- г) обязательно наличие в тесте описания ожидаемых выходных данных.

24. Основная цель любой документации ПС:

- а) соответствие запросам некоторых категорий технического персонала;
- б) использование ее для подготовки кадров;
- в) информировать заинтересованного читателя о функциях ПС на таком детальном уровне, который обеспечивает понимание описываемых функций;
- г) использовать ее на любом этапе проектирования.

25. Документация, которая включает подробные технические описания для специалистов, ведущих проектирование, разработку и сопровождение ПС – это:

- а) технологическая документация
- б) эксплуатационная документация
- в) исследовательская документация

26. Показатель качества – это:

- а) измеряемое числовое значение в виде некоторой целевой функции, характеризующее степень выполнения программного средства назначению;
- б) свойство, обуславливающее пригодность продукции удовлетворять определенные потребности в соответствии с назначением;
- в) количественная характеристика одного или нескольких свойств продукции, составляющих ее качество, рассматриваемая применительно к определенным условиям ее создания, эксплуатации и потребления;
- г) объективная особенность, которая проявляется при создании, эксплуатации или потребления;

27. Модели надежности ПС подразделяются на:

- а) эмпирические и аналитические;

- б) внутренние и внешние;
- в) функциональные и конструктивные;
- г) стохастические и категорийные.

28. Модель надежности ПО – математическая модель, построенная для оценки зависимости надежности ПС от заранее известных или оцененных в ходе ... параметров:

- а) проектирования;
- б) тестирования;
- в) программирования;
- г) эксплуатации.

5.2. Типовые задания для проведения промежуточной аттестации

Вопросы к зачету:

1. Технология программирования и основные этапы ее развития.
2. Жизненный цикл и этапы разработки программного обеспечения.
3. Модели жизненного цикла программного обеспечения.
4. Технология RAD. Ускорение разработки программного обеспечения.
5. Основные качественные и экономические критерии программного обеспечения. Серия стандартов ISO 9000.
6. Проектирование программного обеспечения при структурном подходе.
7. Унифицированный язык моделирования UML. Проектирование классов.
8. Разработка пользовательских интерфейсов.
9. Тестирование программных продуктов.
10. Отладка программного обеспечения.
11. Метрики качества программного средства.
12. Оценивание характеристик качества программных средств.

5.3. Перечень видов оценочных средств

1. Практические задания.
2. Тестовые задания.
3. Вопросы к зачету.

5.4. Процедура применения оценочных материалов

Промежуточная аттестация может проводиться с применением электронного обучения и (или) дистанционных образовательных технологий в соответствии с "Порядком проведения промежуточной аттестации с применением электронного обучения и /или дистанционных образовательных технологий".

Оценочные материалы представлены в Приложении к РПД (файл "ОМД Технологии программирования.pdf").
Методические указания по дисциплине представлены в Приложении к РПД (файл "Методические указания Технологии программирования.pdf").

Критерии оценивания компетенций формируются на основе балльно-рейтинговой системы с помощью всего комплекса методических материалов, определяющих процедуры оценивания знаний, умений, навыков и опыта деятельности, характеризующих данный этап формирования компетенций.

Отметка «зачтено» выставляется, если студент в целом за семестр набрал от 41 до 100 баллов (с учетом баллов, набранных на промежуточной аттестации (зачете)).

Отметка «не зачтено» выставляется, если студент в целом за семестр набрал менее 41 балла (с учетом баллов, набранных на промежуточной аттестации (зачете)).

Отметка «зачтено» ставится, если студент освоил программный материал всех разделов, последователен в изложении программного материала, достаточно последовательно и логически стройно его излагает, умеет увязывать теорию с практикой, успешно прошел текущий контроль успеваемости по дисциплине, продемонстрировал индивидуальные знания, умениями и навыки практической работы.

Отметка «не зачтено» ставится, если студент не знает значительной части программного материала, допускает существенные ошибки, непоследователен в его изложении, не прошел текущий контроль успеваемости, не в полной мере владеет необходимыми знаниями, умениями и навыками при выполнении лабораторных работ, то есть студент не может продолжить обучение без дополнительной подготовки по соответствующей дисциплине.

Промежуточная аттестация может проводиться с применением электронного обучения и (или) дистанционных образовательных технологий в соответствии с «Порядком проведения промежуточной аттестации с применением электронного обучения и /или дистанционных образовательных технологий».

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

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л1.1	Терехов А. Н.	Технология программирования	, 2007	http://biblioclub.ru/index.php?page=book&id=23349_1
Л1.2	Громов Ю. Ю., Иванова О. Г., Кулаков Ю. В., Минин Ю. В., Однолько В. Г.	Методы программирования: учебное пособие	Тамбов ГТТУ, 2012	http://biblioclub.ru/index.php?page=book&id=43708_9

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л2.1	Гаврилов М. В.	Информатика и информационные технологии: учебник для прикладного бакалавриата	Москва : Издательство Юрайт, 2019	https://www.biblio-online.ru/bcode/43177_2
Л2.2	Лавлинский В. В., Коровина О. В.	Технология программирования на современных языках программирования	, 2012	http://biblioclub.ru/index.php?page=book&id=14245_3
Л2.3	Сорокин А. А.	Объектно-ориентированное программирование: учебное пособие (курс лекций)	Ставрополь: СКФУ, 2014	http://biblioclub.ru/index.php?page=book&id=45769_6

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	ИКТ [Электронный ресурс] : федеральный образовательный портал / ФГАУ ГНИИ ИТТ "Информатика". - М. : [б. и.], 2003. - Загл. с титул. экрана.
Э2	СITForum.ru [Электронный ресурс] : образовательный портал / "ЦИТ Форум". - [Б. м. : б. и.], 1997. - Загл. с титул. экрана.

6.3. Информационные технологии

6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения

1.	Операционная система ROSA Enterprise Linux Desktop № RL00450-1-110518-01. RL00450-1-110518-17 от 11 мая 2018 г.
2.	Операционная система Microsoft Windows Professional 7 Russian. Лицензия №48497058 от 13.05.2011 г., договор № Пр/16/6 от 05 апреля 2016 г.
3.	Операционная система Microsoft Windows 10 Professional Russian. Контракт № ПР/ФЕН/15/18 от 23.10.2015 г., договор № Пр/16/6 от 05 апреля 2016 г.
4.	Программное обеспечение Microsoft Office Enterprise 2007 Russian. Лицензия №46138962 от 16.11.2009
5.	Программное обеспечение Microsoft Office 2013 Professional. Контракт № 405535 от 2 ноября 2015 года, контракт № ПР/ФЕН/15/18 от 23.10.2015 г.
6.	Программа для распознавания текста ABBYY FineReader 9.0 Corporate Edition. Лицензионный сертификат - код позиции AF90-3U1V25-102, ABBYY FineReader 9.0 Corporate Edition Volume License Concurrent от 28 июля 2009 г.
7.	Файловый архиватор 7z. Свободно распространяемое ПО
8.	Браузеры Google Chrome, Mozilla, Opera. Свободно распространяемое ПО
9.	Текстовый редактор NotePad++. Свободно распространяемое ПО
10.	Программа просмотра файлов формата RPD Adobe Acrobat Reader DC. Свободно распространяемое ПО
11.	Файловый менеджер Far manager. Свободно распространяемое ПО
12.	Среда программирования и набор инструментов для программирования. MinGW 0.6.3 Свободно распространяемое ПО

6.3.2 Перечень информационных справочных систем и профессиональных баз данных

1.	Компьютерная информационно-правовая система «Гарант»
2.	Портал «Информационно-коммуникационные технологии в образовании» (http://www.ict.edu.ru)

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Ауд.	Назначение	Оборудование и технические средства обучения	Вид
------	------------	--	-----

Ауд.	Назначение	Оборудование и технические средства обучения	Вид
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лек
4-303	Помещение для самостоятельной работы	аудиоколонки, кондиционер, маркерная доска, столы компьютерные, столы учебные, компьютерная техника с возможностью подключения сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета	Пр
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Ср
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	КСР
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Зачёт

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Дисциплина «Технологии программирования» направлена на формирование у студентов знаний, умений и владений в области использования современных технологий программирования, применения методов построения корректных и эффективных алгоритмов и структур данных.

В результате изучения данной дисциплины студенты должны научиться решать сложные задачи с помощью компьютера, определять и реализовывать необходимые при решении структуры данных, раскладывать сложные задачи на независимые подзадачи, выбирать оптимальные реализации моделей для решения прикладных задач

Преподавание дисциплины должно включать в себя следующие образовательные технологии:

- 1) Организация лекций с использованием при необходимости мультимедийных технологий;
- 2) Использование в ходе практических занятий дидактических материалов в виде: опорных конспектов по теоретической составляющей занятий, файлов с примерами программ и т.п.
- 3) Использование ресурсов LMS MOODLE с целью организации процесса систематизации, приобретения и контроля знаний;
- 4) Организация самостоятельной работы с целью усиления приобретения навыков самостоятельного построения моделей, разработки и написания программ;
- 5) Формирование у студентов убежденности в необходимости последовательного освоения следующих этапов в образовательной деятельности:
 - a. ознакомься с содержанием и теоретическими основами изучаемой темы;
 - b. рассмотри, обсуди с другом и протестируй задачу, решенную кем-то;
 - c. реши самостоятельно задачу, подобную рассмотренной ранее;
 - d. реши самостоятельно задачу по изучаемой теме.