

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Котова Лариса Анатольевна  
Должность: Директор филиала  
Дата подписания: 14.09.2023 11:30:31  
Уникальный программный ключ:  
10730ffe6b1ed036b744b6e9d97700b86e5c04a7

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский технологический университет «МИСиС»  
Новотроицкий филиал

## Рабочая программа дисциплины (модуля)

# Технологии программирования

Закреплена за подразделением Кафедра математики и естествознания (Новотроицкий филиал)

Направление подготовки 09.03.03 Прикладная информатика

Профиль Прикладная информатика в технических системах

Квалификация	<b>Бакалавр</b>	
Форма обучения	<b>заочная</b>	
Общая трудоемкость	<b>8 ЗЕТ</b>	
Часов по учебному плану	288	Формы контроля на курсах: экзамен 3 курсовой проект 3
в том числе:		
аудиторные занятия	32	
самостоятельная работа	247	
часов на контроль	9	

### Распределение часов дисциплины по курсам

Курс	3		Итого	
	уп	рп		
Лекции	12	34	12	34
Лабораторные	6	34	6	34
Практические	14	34	14	34
Итого ауд.	32	102	32	102
Контактная работа	32	102	32	102
Сам. работа	247	150	247	150
Часы на контроль	9	36	9	36
Итого	288	288	288	288

Программу составил(и):

*к.п.н, доцент, Алферьева Т.И.*

Рабочая программа

**Технологии программирования**

Разработана в соответствии с ОС ВО:

Самостоятельно устанавливаемый образовательный стандарт высшего образования - бакалавриат Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСиС» по направлению подготовки 09.03.03 Прикладная информатика (приказ от 05.03.2020 г. № 95 о.в.)

Составлена на основании учебного плана:

09.03.03 Прикладная информатика, 09.03.03\_22\_Прикладная информатика\_ПрПИВТС\_заоч.rlx Прикладная информатика в технических системах, утвержденного Ученым советом ФГАОУ ВО НИТУ "МИСиС" в составе соответствующей ОПОП ВО 30.11.2021, протокол № 35

Утверждена в составе ОПОП ВО:

09.03.03 Прикладная информатика, Прикладная информатика в технических системах, утвержденной Ученым советом ФГАОУ ВО НИТУ "МИСиС" 30.11.2021, протокол № 35

Рабочая программа одобрена на заседании

**Кафедра математики и естествознания (Новотроицкий филиал)**

Протокол от 24.06.2021 г., №11

Руководитель подразделения доцент, к.ф.м.н. Гюнтер Д.А.

### 1. ЦЕЛИ ОСВОЕНИЯ

1.1	Цели освоения дисциплины: изучение и практическое освоение общих принципов и современных методов технологии программирования.
1.2	
1.3	Задачи:
1.4	- освоение практических навыков по конструированию собственных базовых классов программного проекта для конкретной предметной области на основе объектно-ориентированной методологии;
1.5	- формирование практических навыков разработки, тестирования, отладки и документирования программных проектов на основе объектно-ориентированного подхода.

### 2. МЕСТО В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Блок ОП:		Б1.О
<b>2.1</b>	<b>Требования к предварительной подготовке обучающегося:</b>	
2.1.1	Информационные системы и технологии	
<b>2.2</b>	<b>Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b>	
2.2.1	Математическое моделирование в технических системах	
2.2.2	Моделирование информационных систем и сетей	
2.2.3	Проектирование информационных систем	
2.2.4	Проектирование систем SCADA	
2.2.5	Разработка UI/UX	
2.2.6	Разработка web-приложений	
2.2.7	Решение прикладных задач с использованием MATLAB	
2.2.8	Подготовка к процедуре защиты и защита выпускной квалификационной работы	
2.2.9	Преддипломная практика	
2.2.10	Разработка интернет-приложений на клиентской стороне	
2.2.11	Языки и среды разработки интернет-приложений	

### 3. РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫЕ С ФОРМИРУЕМЫМИ КОМПЕТЕНЦИЯМИ

<b>ПК-3: способностью проектировать ИС в соответствии с профилем подготовки по видам обеспечения</b>
<b>Знать:</b>
ПК-3-31 организацию процесса разработки ПО, включая все стадии, организацию верификации, тестирования и проверки стабильности ПО, управления качеством.
<b>ОПК-7: Способен выбирать и применять методики проектирования и актуальные инструментальные средства, проектировать и разрабатывать алгоритмы и программы, пригодные для практического применения</b>
<b>Знать:</b>
ОПК-7-31 основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий
<b>ПК-3: способностью проектировать ИС в соответствии с профилем подготовки по видам обеспечения</b>
<b>Уметь:</b>
ПК-3-У1 планировать и управлять процессами в ходе выполнения программного проекта, документировать и оценивать качество программных продуктов.
<b>ОПК-7: Способен выбирать и применять методики проектирования и актуальные инструментальные средства, проектировать и разрабатывать алгоритмы и программы, пригодные для практического применения</b>
<b>Уметь:</b>
ОПК-7-У1 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
<b>ПК-3: способностью проектировать ИС в соответствии с профилем подготовки по видам обеспечения</b>
<b>Владеть:</b>
ПК-3-В1 навыками организации команды разработчиков, обучения пользователей информационных систем.
<b>ОПК-7: Способен выбирать и применять методики проектирования и актуальные инструментальные средства, проектировать и разрабатывать алгоритмы и программы, пригодные для практического применения</b>

**Владеть:**

ОПК-7-В1 навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач

**4. СТРУКТУРА И СОДЕРЖАНИЕ**

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Формируемые индикаторы компетенций	Литература и эл. ресурсы	Примечание	КМ	Выполняемые работы
	<b>Раздел 1. Введение, основные этапы проектирования ПО</b>							
1.1	Понятие программного средства. Технология программирования и основные этапы ее развития. Жизненный цикл программного средства. Архитектура ПО. Системный анализ при создании ПС. Системный подход при разработке ПС. Модели разработки. CASE и RAD-технологии. Управление проектом, планирование и распределение ресурсов, контроль исполнения сроков. Тестирование и оценка качества. Управление программными конфигурациями. Сопровождение. Модернизация и масштабирование программного обеспечения. /Лек/	3	6		Л1.1 Л1.2 Л1.3Л2.2 Э1 Э2 Э3 Э4			
1.2	Самостоятельное изучение учебного материала в LMS Canvas: Сведения об истории развития и становления проблематики дисциплины. Классификация программных систем. Стандарты по разработке программного обеспечения. Основные понятия технологии программирования, особенности программного проекта. Способы преодоления сложностей при разработке. /Ср/	3	8		Л1.1 Л1.2 Л1.3Л2.2 Э1 Э2 Э3 Э4			
1.3	Компонентные технологии и разработка распределенного ПО. /Лаб/	3	6		Л1.1 Л1.2 Л1.3Л2.1 Л2.2Л3.2 Э1 Э2 Э3 Э4			
1.4	Управление содержанием проекта и качеством. Метрики ПО. /Пр/	3	8		Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э1 Э2 Э3 Э4			
	<b>Раздел 2. Методы проектирования и разработки ПО</b>							

2.1	Методы проектирования и разработки программного обеспечения. Парадигмы программирования. Основные подходы к разработке программ: процедурное, логическое, функциональное и объектно-ориентированное программирование. Процедурное и декларативное представление данных. /Лек/	3	6		Л1.1 Л1.2 Л1.3Л2.2 Э1 Э2 Э3 Э4			
2.2	Самостоятельное изучение учебного материала в LMS Canvas: Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования. Типы данных в языках программирования. Связь между данными и операциями. Абстрактные типы данных. Понятие интерфейса. /Ср/	3	8		Л1.1 Л1.2 Л1.3Л2.2 Э1 Э2 Э3 Э4			
2.3	Оценка характеристик разработанных программ с помощью метрик. /Пр/	3	8		Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э1 Э2 Э3 Э4			
2.4	Общие принципы построения распределенных систем. /Лаб/	3	4		Л1.1 Л1.2 Л1.3Л2.1 Л2.2Л3.2 Э1 Э2 Э3 Э4			
	<b>Раздел 3. Методология объектно-ориентированного программирования</b>							
3.1	Объектно-ориентированная парадигма: понятия объекта, класса объектов; основные понятия объектно-ориентированного программирования; классы и объекты; интерфейсы и реализация. Принципы объектного подхода: абстрагирование, ограничение доступа, модульность, иерархия, типизация, параллелизм, устойчивость. Технология создания программного кода. Библиотеки стандартных компонентов, библиотеки объектов. /Лек/	3	6		Л1.1 Л1.2Л2.2 Л2.3 Э1 Э2 Э3 Э4			
3.2	Самостоятельное изучение учебного материала в LMS Canvas: Статические, полустатические и динамические типы данных. Простые и составные типы данных, операция квалификации. /Ср/	3	20		Л1.1 Л1.2Л2.2 Л2.3 Э1 Э2 Э3 Э4			

3.3	Проектирование библиотек классов. Виды классов. /Пр/	3	10		Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Э1 Э2 Э3 Э4			
3.4	Проектирование интерфейса с пользователем. Структуры диалога, поддержка пользователя. /Лаб/	3	6		Л1.1 Л1.2Л2.1 Л2.2 Л2.3Л3.2 Э1 Э2 Э3 Э4			
	<b>Раздел 4. Технологические средства разработки ПО</b>							
4.1	Инструментальная среда разработки. Библиотека VCL. Средства поддержки проекта. Отладчики. CASE-технология. CASE-системы, системы ускоренной разработки приложений. Язык объектного моделирования UML. UML-диаграммы. /Лек/	3	6		Л1.1 Л1.2Л2.2 Э1 Э2 Э3 Э4			
4.2	Самостоятельное изучение учебного материала в LMS Canvas: Язык объектного моделирования UML. Основные определения: система, домен, подсистема, элемент, связи, среда. Структура системы, декомпозиция, иерархия элементов. Процессы в системе и потоки информации. Исследование действий. Построение моделей доменов и подсистем, связей и взаимодействия подсистем, взаимодействия объектов, событий, процессов, потоков данных, действий. /Ср/	3	24		Л1.1 Л1.2Л2.2 Э1 Э2 Э3 Э4			
4.3	Разработка UML-диаграмм. /Лаб/	3	6		Л1.1 Л1.2Л2.1 Л2.2Л3.2 Э1 Э2 Э3 Э4			
4.4	Диаграммы объектов. Диаграмма процессов. Выдача заданий для курсовой работы. /Пр/	3	8		Л1.1 Л1.2Л2.1 Л2.2			
	<b>Раздел 5. Технологии коллективной разработки программного обеспечения</b>							

5.1	Обзор и классификация средств поддержки коллективной разработки программного обеспечения. Программные средства планирования и управления процессом разработки. Сетевые графики и диаграммы рабочего процесса. Сценарии выполнения работ, согласование графиков. Применение систем управления документами. /Лек/	3	4		Л1.1 Л1.2Л2.2 Э1 Э2 Э3 Э4			
5.2	Применение систем управления документами. /Лаб/	3	6		Л1.1 Л1.2Л2.1 Л2.2Л3.2 Э1 Э2 Э3 Э4			
5.3	Выполнение курсовой работы. /Ср/	3	40		Л1.1 Л1.2Л2.1 Л2.2Л3.1			
<b>Раздел 6. Методы отладки и тестирования программ</b>								
6.1	Методы отладки и тестирования программ. Инструментальные средства верификации и тестирования программ. Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Тестирование на этапе проектирования. Регрессионное тестирование. Разработка тестов. /Лек/	3	6		Л1.1 Л1.2Л2.2 Э1 Э2 Э3 Э4			
6.2	Самостоятельное изучение учебного материала в LMS Canvas: Планирование и автоматизированная генерация тестов. Сценарии тестирования. Анализаторы профиля выполнения теста. Репозитарий тестов. Контроль показателей качества. Подготовка к защите курсовой работы. Подготовка к экзамену. /Ср/	3	50		Л1.1 Л1.2Л2.2Л3.1 Э1 Э2 Э3 Э4			
6.3	Тестирование и комплексная отладка ПС. /Лаб/	3	6		Л1.1 Л1.2 Л1.3Л2.1 Л2.2Л3.2 Э1 Э2 Э3			
6.4	Проведение экзамена /Экзамен/	3	36		Э1 Э2 Э3 Э4			

**5. ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ**

### 5.1. Вопросы для самостоятельной подготовки к экзамену (зачёту с оценкой)

Вопросы к экзамену (ПК-2-31, ПК-2-32, ПК-2-У1, ПК-2-У2, ПК-3-31, ПК-3-У1, ОПК-7-31, ОПК-7-У1, УК-6-31, УК-6-У1):

1. Понятие программного средства.
2. Технология программирования и основные этапы ее развития.
3. Жизненный цикл программного средства.
4. Архитектура ПО.
5. Системный анализ при создании ПС.
6. Системный подход при разработке ПС.
7. Модели разработки: каскадная, с промежуточным контролем, спиральная и т.д.
8. CASE и RAD-технологии.
9. Управление проектом, планирование и распределение ресурсов, контроль исполнения сроков.
10. Методы проектирования программного обеспечения.
11. Декомпозиции и абстракции при проектировании ПО.
12. Спецификация процедур и данных.
13. Декомпозиция задачи.
14. Методы защиты программ и данных.
15. Парадигмы программирования.
16. Объектно-ориентированная парадигма: понятия объекта, класса объектов;
17. Интерфейсы и реализация.
18. Технология создания программного кода.
19. Библиотеки стандартных компонентов, библиотеки объектов.
20. Проектирование интерфейса с пользователем. Структуры диалога; поддержка пользователя.
21. Статические, полустатические и динамические типы данных.
22. Простые и составные типы данных, операция квалификации.
23. Технологии распределенных вычислений.
24. Обзор и классификация средств поддержки коллективной разработки ПО.
25. Программные средства планирования и управления процессом разработки.
26. Сетевые графики и диаграммы рабочего процесса.
27. Сценарии выполнения работ.
28. Применение систем управления документами.
29. Технологические средства разработки программного обеспечения.
30. Инструментальная среда разработки.
31. Библиотека VCL.
32. Средства поддержки проекта.
33. UML-диаграммы.
34. Методы отладки и тестирования программ.
35. Категории программных ошибок. Типы тестов.
36. Тестирование на этапе планирования.
37. Тестирование на этапе проектирования.
38. Регрессионное тестирование. Разработка тестов.
39. Документирование и оценка качества программных продуктов.
40. Пользовательская документация программных средств.



**5.2. Перечень работ, выполняемых по дисциплине (модулю, практике, НИР) - эссе, рефераты, практические и расчетно-графические работы, курсовые работы, проекты и др.**

Курсовая работа на тему: "Разработка прикладного ПО" (ПК-2-В1, ПК-2-В2, ПК-3-В1, ОПК-7-В1, УК-6-В1).

Целью курсовой работы является приобретение практических навыков проектирования и разработки программной системы применением современных информационных технологий.

В рамках выполнения курсовой работы необходимо разработать один из предложенных ниже проектов.

1. Написать программу, которая позволяет сортировать заданный линейный массив целых чисел различными методами
2. Написать программу, которая обеспечивает выполнение операций сложения, вычитания и умножения над матрицами целых чисел.
3. Написать программу, которая позволяет зашифровывать и расшифровывать с помощью «шифра Цезаря» сообщение, написанное на русском языке.
4. Написать программу, которая позволяет зашифровывать и расшифровывать сообщение с помощью «шифра перестановки».
5. Написать программу, позволяющую сжимать текстовую информацию на русском языке, а затем преобразовывать сжатую информацию в исходное состояние.
6. Написать программу, которая определяет «водящего» в детской игре.
7. Написать программу, моделирующую игру.
8. Написать программу перевода чисел из одной системы счисления в другую.
9. Написать программу, позволяющую выполнять арифметические действия над двоичными или шестнадцатеричными числами.

10. Написать программу для справочной службы железнодорожного вокзала.

Результат выполнения курсовой работы оформляется в виде пояснительной записки, содержащей следующие основные структурные элементы: титульный лист, содержание, цели и задачи работы, описание решаемой задачи, описание входных данных с указанием требований к ним, описание всех выполняемых программой функций, описание основных алгоритмов работы, описание результатов работы программы, описание интерфейса.

Объем расчетно-пояснительной записки курсовой работы – 30-35 стр.

**5.3. Оценочные материалы, используемые для экзамена (описание билетов, тестов и т.п.)**

Формой промежуточной аттестации по дисциплине является экзамен.

Ниже представлен образец билета для экзамена, проводимого в устной форме.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ  
«МИСиС»  
НОВОТРОИЦКИЙ ФИЛИАЛ

Кафедра электроэнергетики и электротехники

БИЛЕТ К ЭКЗАМЕНУ № 0

Дисциплина: «Технологии программирования»  
Направление: 09.03.03 «Прикладная информатика»  
Форма обучения: очная

Форма проведения экзамена: устная

1. Методы проектирования программного обеспечения.
2. Объектно-ориентированная парадигма: понятия объекта, класса объектов.

Составил доцент: \_\_\_\_\_ Р.Е. Мажирина

Зав. кафедрой ЭиЭ: \_\_\_\_\_ Р.Е. Мажирина

«01» сентября 2020 г.

Дистанционно экзамен проводится в LMS Canvas. Экзаменационный тест содержит 30 заданий. На решение отводится 30 минут. Разрешенные попытки - две. Зачитывается наилучший результат.

Образец заданий для экзамена, проводимого дистанционно в LMS Canvas:

1. Среди этапов жизненного цикла ПО нет этапа  
1) кодирования;  
2) интеграции;  
3) архивации;  
4) сопровождения.

2. Критический путь на сетевом графике проекта можно найти
  - 1) методом динамического программирования;
  - 2) при помощи алгоритма Флойда;
  - 3) методом линейного программирования;
  - 4) перебором вариантов.
3. Ленточный график используется для
  - 1) установления зависимостей между видами работ;
  - 2) определения времени выполнения проекта;
  - 3) наглядного представления хода работ;
  - 4) оптимизации распределения ресурсов.
4. . Сложность программы обычно выражается
  - 1) количеством человеко-дней, затраченных на разработку;
  - 2) количеством строк кода;
  - 3) размером исполняемого модуля в байтах;
  - 4) количеством операторов в программе.
5. Наибольшей сложность отличается модель процесса разработки ПО
  - 1) спиралевидная;
  - 2) модель RAD;
  - 3) каскадная;
  - 4) V-образная.
6. Использование настраиваемого представления интерфейса недопустимо в приложениях, предназначенных для
  - 1) программа управления роботом-спасателем;
  - 2) обработки данных;
  - 3) офисных работ;
  - 4) управления производством.
7. Для сопровождения индустриального ПО чаще всего используется
  - 1) техническая спецификация ПО;
  - 2) исходный текст программ;
  - 3) специальное руководство по сопровождению;
  - 4) руководство системного оператора.
8. Количество используемых элементов отображения изменяющихся параметров в интерфейсе не должно
  - 1) влиять на работу оператора;
  - 2) быть слишком маленьким;
  - 3) изменяться в ходе эксплуатации;
  - 4) превышать число Ингве.
9. Мобильный интерфейс нельзя создать при помощи
  - 1) VisualC;
  - 2) Delphi;
  - 3) FLTK;
  - 4) QTdesigner.
10. В библиотеке FLTK к элементам управления, не имеющим метода "value" является
  - 1) slider;
  - 2) roller;
  - 3) counter;
  - 4) browser.
11. Промышленное использование переноса программ, написанных на Delphi в ОС Linux
  - 1) из-за неоптимальности кода используется редко;
  - 2) сейчас используется повсеместно;
  - 3) невозможно из-за несовместимости графических подсистем Windows и Linux;
  - 4) не делается из-за отсутствия компилятора.
12. Для разработки графического пользовательского интерфейса наиболее всего подходит методология
  - 1) функционального программирования;
  - 2) объектно-ориентированного программирования;
  - 3) императивного программирования;
  - 4) логического программирования.
13. Метод черного ящика не применяется чтобы подтвердить
  - 1) правильное обнаружение ошибок и их обработку;

- 2) правильную работу продукта при больших потоках данных;
  - 3) правильный выбор размера критических секций кода;
  - 4) правильность вычислений.
14. Целью тестирования методом белого ящика не является
- 1) отслеживание всех ветвей выполняемого кода;
  - 2) обнаружение ошибок в работе интегрированного продукта;
  - 3) доказательство правильной работы модулей;
  - 4) проверка правильности функционирования интерфейса.
15. При нисходящем способе проектирования на ранних этапах предпочтителен метод
- 1) системного тестирования;
  - 2) черного ящика;
  - 3) белого ящика;
  - 4) интеграционного тестирования.
16. При восходящем способе проектирования на ранних этапах предпочтителен метод
- 1) черного ящика;
  - 2) системного тестирования;
  - 3) интеграционного тестирования;
  - 4) белого ящика.
17. Методика разработки программ, в основе которой лежит понятие объекта как некоторой структуры, описывающей объект реального мира, его поведение, -это ...
- 1) Объектно-ориентированное программирование
  - 2) Объект
  - 3) Инкапсуляция
  - 4) Наследование
18. Объект -это ...
- 1) Возможность объектов с одинаковой спецификацией иметь различную реализацию
  - 2) Возможность при описании класса указывать на его происхождение от другого класса
  - 3) Возможность скрыть внутреннее устройство объекта от его пользователей, предоставив через интерфейс доступ только к тем членам объекта, с которыми клиенту разрешается работать напрямую
  - 4) Некоторая часть окружающего нас мира, которая может быть рассмотрена как единое целое
  - 5) Методика разработки программ, в основе которой лежит понятие объекта как некоторой структуры, описывающей объект реального мира, его поведение
19. Именованные категории, позволяющие группировать сходные объекты, -это ...
- 1) Классы
  - 2) Объекты
  - 3) События
  - 4) Свойства
  - 5) Методы класса
20. Отдельные, четко обозначенные экземпляры некоторого класса, -это ...
- 1) Классы
  - 2) Объекты
  - 3) События
  - 4) Свойства
  - 5) Методы класса
21. Некоторые особые состояния, в которые может попадать объект, -это ...
- 1) Классы
  - 2) Объекты
  - 3) События
  - 4) Свойства
  - 5) Методы класса
22. Характеристики объекта -это ...
- 1) Классы
  - 2) Объекты
  - 3) События
  - 4) Свойства
  - 5) Методы класса
23. При использовании метода главного программиста
- 1) тестированием занимается член команды;

- 2) тестирование не производится из-за ненужности;
  - 3) тестирование выполняет сам главный программист;
  - 4) тестирование выполняет дублер.
24. Какого вида ошибок не существует?
- 1) Синтаксические
  - 2) Алгоритмические
  - 3) Нет правильного ответа
  - 4) Ошибки времени выполнения
  - 5) Структурированные
25. Угроза это
- 1) потенциальное возможное событие, действие, процесс или явление, которое может привести к нанесению хороших данных
  - 2) возможное событие, действие, процесс или явление, которое может привести к сохранности данных
  - 3) потенциальное возможное событие, действие, процесс или явление, которое может привести к нанесению изменению данных
  - 4) потенциальное возможное событие, действие, процесс или явление, которое может привести к нанесению ущерба.
  - 5) нет правильного ответа
26. Алгоритмические ошибки, это ошибки
- 1) ошибки времени компиляции
  - 2) ошибки при выполнении
  - 3) ошибки при подсчете из-за не правильного алгоритма
  - 4) ошибки во время не выполнения
  - 5) ошибки некорректного ввода
27. Синтаксические ошибки, это ошибки
- 1) ошибки времени компиляции
  - 2) ошибки при выполнении
  - 3) ошибки при подсчете
  - 4) ошибки во время не выполнения
  - 5) ошибки некорректного ввода
28. Из каких 2 составляющих состоит надежность программ
- 1) Корректность и Контроль
  - 2) Корректность и Обеспечение
  - 3) Корректность и Устойчивость
  - 4) Устойчивость и Контроль
  - 5) Обеспечение и Контроль
28. Метод Application.Run:
- 1) Создает окно формы, регистрирует входящие в него компоненты
  - 2) Подключает автоматический цикл обработки сообщений Windows к приложению
  - 3) Выполняет подпрограммы раздела инициализации модулей приложения в случае их наличия
29. Набор всех файлов, необходимых для создания приложения, называется...
- 1) Модулем
  - 2) Проектом
  - 3) Задачей
  - 4) Формой
30. Системное программное обеспечение:
- 1) программы для организации совместной работы устройств компьютера как единой системы
  - 2) программы для организации удобной системы размещения программ на диске
  - 3) набор программ для работы устройства системного блока компьютера
31. Является техническим заданием ...
- 1) документ, утвержденный в установленном порядке, определяющий цели, требования и основные исходные данные, необходимые для разработки автоматизированной системы управления
  - 2) основание для разработки системы
  - 3) техническая документация, утвержденная в установленном порядке, содержащая общесистемные проектные решения, алгоритм решения задач
32. Требования, предъявляемые к средствам проектирования ...
- 1) простыми в освоении и применении
  - 2) помогать планировщику
  - 3) технически, программно и информационно совместимыми
  - 4) охватывать в совокупности все этапы жизненного цикла ПО

- 5) экономически целесообразными
33. Критерием выбора алгоритмического языка при проектировании программного обеспечения задачи является ...
- 1) объем памяти, занимаемой разработанной программой
  - 2) синтаксическая и семантическая ясность языка
  - 3) совместимость с другими языками
  - 4) возможность управления файлами
  - 5) решение заказчика проекта
  - 6) время написания программы
34. В системе программирования после обнаружения факта ошибки начинается процесс ее поиска и исправления, который называют ...
- 1) отладкой
  - 2) компиляцией
  - 3) тестированием
  - 4) трассировка

#### 5.4. Методика оценки освоения дисциплины (модуля, практики. НИР)

Критерии оценки ответов на экзамене, проводимом в устной форме

Оценка «Отлично» ставится, если

- на теоретические вопросы даны развернутые ответы, при необходимости изложен математический аппарат (формулы, графики и т.д.) приведены соответствующие схемы, таблицы, рисунки и т.д., правильно решена задача  
- обучающийся хорошо ориентируется в материале, владеет терминологией, приводит примеры, обосновывает, анализирует, высказывает свою точку зрения по анализируемым явлениям и процессам, правильно применяет полученные знания при решении практических задач. Ответы излагаются свободно, уверенно без использования листа устного опроса

Оценка «Хорошо» ставится, если

- на теоретические вопросы даны полные ответы, но имела место неточность в определении каких-либо понятий, явлений и т.д. Задача решена.  
- обучающийся ориентируется в материале хорошо, но допускает ошибки при формулировке, описании отдельных категорий

Оценка «Удовлетворительно» ставится, если

- на теоретические вопросы даны общие неполные ответы  
- обучающийся слабо ориентируется в материале, не может решать задачи, не может привести пример, не может анализировать и обосновывать

Оценка «Неудовлетворительно» ставится, если

- не решена задача и правильный ответ дан на один вопрос (либо ни на один)  
- обучающийся в материале дисциплины практически не ориентируется, т.е. не может дать даже общих сведений по вопросу.

Критерии оценки ответов на экзамене, проводимом в дистанционной форме в LMS Canvas

$90 \leq$  Процент верных ответов  $\leq 100$  - отлично

$75 \leq$  Процент верных ответов  $< 90$  - хорошо

$60 \leq$  Процент верных ответов  $< 75$  – удовлетворительно

Критерии оценки выполнения курсовой работы:

1. Теоретические сведения изложены в достаточном объеме, четко и последовательно
2. Используются выводы (позиции, мнения и др.) известных ученых, профессионалов
3. Исследуются и сравниваются разные подходы, методики, приводятся собственные суждения и выводы
4. Приведено описание решаемой задачи, описание входных данных с указанием требований к ним.
5. Приедено описание всех выполняемых программой функций, описание основных алгоритмов работы, описание результатов работы программы, описание интерфейса.
6. Текст написан грамотно, стилистически выдержан
7. Текст оформлен в соответствии с требованиями

Работа оценивается на отлично, если:

теоретические сведения изложены в достаточном объеме, четко и последовательно, использованы выводы (позиции, мнения и др.) известных ученых, профессионалов, исследуются и сравниваются разные подходы, методики, приводятся собственные суждения и выводы, имеются примеры, даются ссылки на источники, текст написан грамотно, стилистически выдержан и оформлен в соответствии с требованиями.

описание решаемой задачи, описание входных данных с указанием требований к ним приведено полном объеме.

Приедено полное описание всех выполняемых программой функций, описание основных алгоритмов работы, описание результатов работы программы, описание интерфейса, даны пояснения и рисунки.

В целом по работе: расставлены ссылки на источники, текст написан грамотно, стилистически выдержан, оформлен в соответствии с требованиями.

Выполнение работы оценивается как хорошее, если она соответствует всем критериям, перечисленным выше, но в работе отсутствует описание и сравнения разных подходов, методик и т.д. с последующим формированием собственных выводов на данный счет. Описание всех выполняемых программой функций, описание основных алгоритмов работы, описание результатов работы программы, описание интерфейса приведено не полностью, отсутствуют пояснения и рисунки.

В целом по работе: расставлены ссылки на источники, текст написан грамотно, стилистически выдержан, оформлен в соответствии с требованиями.

Выполнение работы оценивается как удовлетворительное, если она соответствует всем критериям, перечисленным выше, но в работе отсутствуют описание и сравнения разных подходов, методик и т.д. с последующим формированием собственных выводов на данный счет. Описание всех выполняемых программой функций, описание основных алгоритмов работы приведено не полностью, имеются неточности. Отсутствует описание результатов работы программы, описание интерфейса.

Если работа допущена до защиты с оценкой «отлично», в процессе защиты студент хорошо владеет материалом, не использует при этом опорных конспектов и т.д., с легкостью отвечает на любой вопрос по курсовой работе, то в этом случае студенту за выполнение курсовой работы ставится оценка «отлично», которая и проставляется в зачетную книжку и в ведомость.

В процессе защиты оценка повышаться не может, т.е. если студент допущен до защиты с оценкой «хорошо», «отлично» он уже в любом случае не сможет получить, а вот «удовлетворительно» может – если при защите возникают определенные трудности с ориентацией в материале, ответами на вопросы по курсовой работе.

Если студент совершенно не владеет материалом курсовой работы, то получает «неудовлетворительно».

Если работа не соответствует критериям выполнения курсовой работы, то оценивается неудовлетворительно и до защиты

не допускается.

**6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ****6.1. Рекомендуемая литература****6.1.1. Основная литература**

	Авторы, составители	Заглавие	Библиотека	Издательство, год, эл. адрес
Л1.1	Д.Р.Кувшинов	Основы программирования: учеб.пособие		М.: Юрайт, Екатеринбург: изд-во УрФУ, 2019,
Л1.2	Гниденко И.Г.	Технологии и методы программирования: учеб.пособие		М.: Юрайт, 2019,
Л1.3	Т.М. Зубкова	Технология разработки программного обеспечения: учебное пособие		Оренбург : ОГУ, 2017, <a href="http://biblioclub.ru/index.php?page=book&amp;id=485553">http://biblioclub.ru/index.php? page=book&amp;id=485553</a>

**6.1.2. Дополнительная литература**

	Авторы, составители	Заглавие	Библиотека	Издательство, год, эл. адрес
Л2.1	В.Т. Николаев, С.В. Купцов, В.Н. Тикменов	Практика программирования в инженерных расчётах: учебное пособие		Москва :Физматлит, 2018, <a href="http://biblioclub.ru/index.php?page=book&amp;id=485295">http://biblioclub.ru/index.php? page=book&amp;id=485295</a>
Л2.2	Милов А.В.	Основы программирования в задачах и примерах: учебный курс		Харьков: Холио, 2002,
Л2.3	Т.В.Куренкова	Основы алгоритмизации и объектно-ориентированного программирования: учебн.пособие № 2056		М.: МИСиС, 2011, <a href="http://elibrary.misis.ru">http://elibrary.misis.ru</a>

**6.1.3. Методические разработки**

	Авторы, составители	Заглавие	Библиотека	Издательство, год, эл. адрес
Л3.1	А.С.Измайлова	Методические указания по выполнению курсовых работ: Методические указания		НФ НИТУ «МИСиС», 2020, <a href="http://www.nf.misis.ru">www.nf.misis.ru</a>
Л3.2	Д.Ю.Плясунов	Технологии программирования (Программирование на Borland Delphi): лабораторный практикум		Новотроицк: НФ МИСИС, 2002,

**6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

Э1	Научная электронная библиотека eLIBRARY	<a href="https://www.elibrary.ru/">https://www.elibrary.ru/</a>
Э2	LMS Canvas	<a href="https://lms.misis.ru">https://lms.misis.ru</a>
Э3	НФ НИТУ МИСиС	<a href="http://nf.misis.ru/">http://nf.misis.ru/</a>
Э4	Университетская библиотека ONLINE	<a href="https://biblioclub.ru/">https://biblioclub.ru/</a>

**6.3 Перечень программного обеспечения****6.4. Перечень информационных справочных систем и профессиональных баз данных**

И.1	<a href="https://standartgost.ru/0/2870/2880-edinaya_sistema_programmnoy_dokumentatsii">https://standartgost.ru/0/2870/2880-edinaya_sistema_programmnoy_dokumentatsii</a> - Единая система программной документации	
И.2		
И.3	<a href="http://www.it-daily.ru">http://www.it-daily.ru</a> – Новости российского ИТ-рынка	
И.4	<a href="http://www.intuit.ru/">http://www.intuit.ru/</a> - Каталог курсов Интернет Университета Информационных Технологий	

**8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ**

Освоение дисциплины предполагает как проведение традиционных аудиторных занятий, так и работу в электронной информационно-образовательной среде НИТУ «МИСиС» (ЭИОС), частью которой непосредственно предназначенной для осуществления образовательного процесса является Электронный образовательный ресурс LMS Canvas. Он доступен по URL адресу <https://lms.misis.ru> и позволяет использовать специальный контент и элементы электронного обучения и дистанционных образовательных технологий. LMS Canvas используется преимущественно для асинхронного взаимодействия между участниками образовательного процесса посредством сети «Интернет».

Чтобы эффективно использовать возможности LMS Canvas, а соответственно и успешно освоить дисциплину, нужно:

- 1) зарегистрироваться на курс. Для этого нужно перейти по ссылке ... Логин и пароль совпадает с логином и паролем от личного кабинета НИТУ МИСиС;
- 2) в рубрике «В начало» ознакомиться с содержанием курса, вопросами для самостоятельной подготовки, условиями допуска к аттестации, формой промежуточной аттестации (зачет/экзамен), критериями оценивания и др.;
- 3) в рубрике «Модули», заходя в соответствующие разделы изучать учебные материалы, размещенные преподавателем. В т.ч. пользоваться литературой, рекомендованной преподавателем, переходя по ссылкам;
- 4) в рубрике «Библиотека» возможно подбирать для выполнения письменных работ (контрольные, домашние работы, курсовые работы/проекты) литературу, размещенную в ЭБС НИТУ «МИСиС»;
- 5) в рубрике «Задания» нужно ознакомиться с содержанием задания к письменной работе, сроками сдачи, критериями оценки. В установленные сроки выполнить работу(ы), подгрузить здесь же для проверки. Удобно называть файл работы следующим образом (название предмета (сокращенно), группа, ФИО, дата актуализации (при повторном размещении)). Например, Экономика\_Иванов\_И.И.\_БМТ-19\_20.04.2020. Если работа содержит рисунки, формулы, то с целью сохранения форматирования ее нужно подгружать в pdf формате.

Работа, подгружаемая для проверки, должна:

- содержать все структурные элементы: титульный лист, введение, основную часть, заключение, список источников, приложения (при необходимости);
- быть оформлена в соответствии с требованиями.

Преподаватель в течение установленного срока (не более десяти дней) проверяет работу и размещает в комментариях к заданию рецензию. В ней он указывает как положительные стороны работы, так замечания. При наличии в рецензии замечаний и рекомендаций, нужно внести поправки в работу, подгрузить ее заново для повторной проверки. При этом важно следить за сроками, в течение которых должно быть выполнено задание. При нарушении сроков, указанных преподавателем возможность подгрузить работу остается, но система выводит сообщение о нарушении сроков. По окончании семестра подгрузить работу не получится;

- 6) в рубрике «Тесты» пройти тестовые задания, освоив соответствующий материал, размещенный в рубрике «Модули»;
- 7) в рубрике «Оценки» отслеживать свою успеваемость;
- 8) в рубрике «Объявления» читать объявления, размещаемые преподавателем, давать обратную связь;
- 9) в рубрике «Обсуждения» создавать обсуждения и участвовать в них (обсуждаются общие моменты, вызывающие вопросы у большинства группы). Данная рубрика также может быть использована для взаимной проверки;
- 10) проявлять регулярную активность на курсе.

Преимущественно для синхронного взаимодействия между участниками образовательного процесса посредством сети «Интернет» используется Microsoft Teams (MS Teams). Чтобы полноценно использовать его возможности нужно установить приложение MS Teams на персональный компьютер и телефон. Старостам нужно создать группу в MS Teams.

Участие в группе позволяет:

- слушать лекции;
- работать на практических занятиях;
- быть на связи с преподавателем, задавая ему вопросы или отвечая на его вопросы в общем чате группы в рабочее время с 9.00 до 17.00;
- осуществлять совместную работу над документами (вкладка «Файлы»).

При проведении занятий в дистанционном синхронном формате нужно всегда работать с включенной камерой.

Исключение – если преподаватель попросит отключить камеры и микрофоны в связи с большими помехами. На аватарках должны быть исключительно деловые фото.

При проведении лекционно-практических занятий ведется запись. Это дает возможность просмотра занятия в случае невозможности присутствия на нем или при необходимости вновь обратиться к материалу и заново его просмотреть.