МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
СТАРООСКОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ ИМ. А.А. УГАРОВА
(филиал) федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский технологический университет «МИСИС»
СТИ НИТУ «МИСИС»

Рабочая программа утверждена
решением Ученого совета
СТИ НИТУ «МИСИС»
от «20» июня 2023 г.
протокол № 5

Рабочая программа дисциплины

Технология программирования

Закреплена за кафедрой	**Кафедра автоматизированных и информационных систем управления**
Направление подготовки	09.03.02 Информационные системы и технологии
Профиль	Информационные системы и технологии
Квалификация	**Бакалавр**
Форма обучения	**Очная**
Общая трудоемкость	**4 ЗЕТ**

Формы контроля в семестр:

Экзамен 2

Часов по учебному плану	144
в том числе:	
аудиторные занятия	51
самостоятельная работа	57
часов на контроль	36

Распределение часов дисциплины по семестрам

Семестр	2		Итого	
Вид занятий	УП	РП	УП	РП
Лекции	17	17	17	17
Лабораторные	17	17	17	17
Практические	17	17	17	17
Контактная работа	51	51	51	51
Сам. работа	57	57	57	57
Часы на контроль	36	36	36	36
Итого:	144	144	144	144

Год набора 2020 г.
В редакции 2023 г.

Программу составила:
доцент каф. АИСУ, кандидат физико-математических наук, доцент

Михайлюк Екатерина Андреевна
Должность, уч.ст., уч.зв ФИО полностью *подпись*

Рабочая программа дисциплины
Технология программирования
наименование

Разработана в соответствии с ОС ВО НИТУ «МИСИС»:
Самостоятельно устанавливаемый образовательный стандарт высшего образования – бакалавриат Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСИС» по направлению подготовки 09.03.02 Информационные системы и технологии (приказ от 05.03.2020 г. № 95 о.в.)

Составлена на основании учебного плана 2020 года набора.
09.03.02 Информационные системы и технологии,
Профиль: Информационные системы и технологии, утвержденного Ученым советом СТИ НИТУ «МИСИС» 22.06.2020 г., протокол № 23.

Рабочая программа одобрена на заседании кафедры Автоматизированных и информационных систем управления

Протокол от «24» мая 2023 г. № 3

И.о. зав. кафедрой АИСУ Д.А. Полещенко
аббревиатура наименования кафедры *подпись* *И.О. Фамилия*
«24» мая 2023 г.

Руководитель ОПОП ВО
И.о. зав. каф. АИСУ, кандидат технических наук, доцент Д.А. Полещенко
подпись *И.О. Фамилия*

«24» мая 2023 г.

1. ЦЕЛИ ОСВОЕНИЯ
Цель дисциплины – формирование у студентов основных принципов и методов программирования, используемых на различных этапах разработки программного обеспечения сложных компьютерных системах, а так же формирование навыков организации процессов программной разработки; развитие умений использования основных современных средств и методов программирования, технологий проектирования алгоритмов и разработки программных систем. **Задачи дисциплины:** • изучение основных этапов решения задачи на ЭВМ, критериев качества программного обеспечения, методов спецификации программ; • изучение основных методов и средств разработки алгоритмов и программ, приемов структурного программирования, представления структурных алгоритмов; • изучение принципов и приобретение навыков проектирования архитектуры и разработки функциональных модулей программных продуктов, разработки программной документации в соответствии со стандартами; • изучение принципов и приобретение практических навыков реализации, отладки, тестирования и документирования программного обеспечения.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ, ПРАКТИКИ, НИР) В СТРУКТУРЕ ОПОП ВО	
Цикл (раздел) ОП:	Б1.О.
2.1	**Требования к предварительной подготовке обучающихся:**
2.1.1	Информатика
2.1.2	Компьютерное обеспечение специальности.
2.2	**Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее:**
2.2.1	Объектно-ориентированное программирование
2.2.2	Проектирование информационных систем

3. РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫЕ С СООТВЕТСТВУЮЩИМИ КОМПЕТЕНЦИЯМИ, КОТОРЫЕ ДОЛЖНЫ БЫТЬ СФОРМИРОВАНЫ У ОБУЧАЮЩИХСЯ	
УК -2 Способен: **- анализировать продукцию, процессы и системы;** **- ставить задачи в области, соответствующей профилю подготовки;** **- применять системный подход к решению поставленных задач с помощью соответствующих аналитических, вычислительных и экспериментальных методов**	
Знать:	УК-2: З-1 - основные современные технологии программирования
Уметь:	УК-2: У-1 - формулировать цели разработки программных средств по заданным критериям и ограничениям
Владеть:	УК-2: В-1 - методами разработки программных систем
УК - 4 Способен: **- осуществлять поиск литературы, критически используя научные базы данных, профессиональные стандарты и регламенты, нормы безопасности и другие источники информации;** **- осуществлять поиск, критический анализ и синтез информации;** **- осуществлять моделирование, анализ и экспериментальные исследования для решения проблем в профессиональной области**	
Знать:	УК-4: З-1 профессиональные стандарты и регламенты при разработке алгоритмов и программ
Уметь	УК-4: У-1 проводить исследования, использовать научные базы данных при выборе технологий программирования
Владеть	УК-4: В-1 методами анализа и синтеза информации при разработке программных средств
УК - 5 Способен демонстрировать: **- практические навыки для решения задач и реализации проектов, в области, соответствующей профилю подготовки;** **- знания требований безопасности жизнедеятельности, безопасности окружающей среды, экономические и технологические ограничения в области, соответствующей профилю подготовки;** **- знание экономических, организационных и управленческих вопросов (управление проектом, управление рисками и управление изменениями и др.)**	
Знать:	УК-5: З-1: знать требования безопасности, экономические и технологические ограничения в проектах разработки программных средств
Уметь	УК-5: У-1: применять различные методы разработки программных средств при решении профессиональных задач
Владеть	УК-5: В-1: методами управления проектами в проектах разработки программных средств
ОПК -6 Способен: **- разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий.**	

- применять системный подход к решению поставленных задач с помощью соответствующих аналитических, вычислительных и экспериментальных методов
УК-2: З-1 - основные современные системы программирования
УК-2: У-1 - формулировать цели разработки алгоритмов по заданным критериям и ограничениям
УК-2: В-1 - методами разработки программных систем на языках высокого уровня

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Код занятия	Наименование разделов и тем /вид занятия/	Семестр	Количество часов	Компетенции	Литература и электронные ресурсы	Примечание
	Раздел 1. Основы программирования на C/C++.					
1.1	Технология программирования. Основные понятия и подходы. /лек/	2	1	УК-2: З-1 УК-4: З-1 УК-5: З-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	
1.2	Использование библиотечных процедур. /лек/	2	2	УК-2: З-1 УК-4: З-1 УК-5: З-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	
1.3	Основные виды, этапы проектирования и жизненный цикл программных продуктов. /лек/	2	2	УК-2: З-1 УК-4: З-1 УК-5: З-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	
1.4	Разработка и анализ алгоритмов /пз/	2	6	УК-2: У-1 УК-4: У-1 УК-5: У-1 ОПК-6: У-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	Текущий контроль: ПЗ-1 В1-2
1.5	Выполнение программы простой структуры. Вычисление выражений с использованием стандартных функций» /лаб/	2	6	УК-2: У-1 УК-4: У-1 УК-5: У-1 УК-2: В-1 УК-4: В-1 УК-5: В-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	Текущий контроль: ЛР1-З
1.6	Использование библиотечных процедур. /ср/	2	20	УК-2: У-1 УК-4: У-1 УК-5: У-1 ОПК-6: В-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	

	Раздел 2. Приемы обеспечения технологичности программных продуктов.					
2.1	Определение требований к программному обеспечению и исходных данных для его проектирования. /лек/	2	2	УК-2: З-1 УК-4: З-1 УК-5: З-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	
2.2	Анализ требований и определение спецификаций программного обеспечения /лек/	2	2	УК-2: З-1 УК-4: З-1 УК-5: З-1 ОПК-6: З-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Э1	
2.3	Проектирование программного обеспечения при структурном/объектно-ориентированном подходе. /лек/	2	2	УК-2: З-1 УК-4: З-1 УК-5: З-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Э1	
2.4	Разработка и анализ алгоритмов /пз/	2	6	УК-2: У-1 УК-4: У-1 УК-5: У-1 УК-2: В-1 УК-4: В-1 УК-5: В-1 ОПК-6: В-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	Текущий контроль: ПЗ-2 В 1-10
2.5	Выполнение программы простой структуры. Вычисление выражений с использованием стандартных функций» /лаб/	2	6	УК-2: У-1 УК-4: У-1 УК-5: У-1 УК-2: В-1 УК-4: В-1 УК-5: В-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	Текущий контроль: ЛР 4-6
2.6	Основные элементы языка C++ /ср/	2	20	УК-2: У-1 УК-4: У-1 УК-5: У-1 ОПК-6: У-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	
	Раздел 3. Разработка пользовательских интерфейсов					
3.1	Типы пользовательских интерфейсов и этапы их разработки. /лек/	2	2	УК-2: З-1 УК-4: З-1 УК-5: З-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2	

					Э1	
3.2	Тестирование программных продуктов. Отладка программного обеспечения. /лек/	2	2	УК-2: З-1 УК-4: З-1 УК-5: З-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Э1	
3.3	Составление программной документации. /лек/	2	2	УК-2: З-1 УК-4: З-1 УК-5: З-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Э1	
3.4	Разработка и анализ алгоритмов /пз/ Выполнение программы простой структуры. Вычисление выражений с использованием стандартных функций» /лаб/	2	5	УК-2: У-1 УК-4: У-1 УК-5: У-1 УК-2: В-1 УК-4: В-1 УК-5: В-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	Текущий контроль: ЛР4-7
3.5	Разработка и анализ алгоритмов /пз/	2	5	УК-2: У-1 УК-4: У-1 УК-5: У-1 УК-2: В-1 УК-4: В-1 УК-5: В-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	ПЗ-2 В11-20
3.6	Создание виджета. /ср/	2	17	УК-2: У-1 УК-4: У-1 УК-5: У-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	
	Часы на контроль. / Контроль/	2	36	УК-2: З-1 УК-2: У-1 УК-2: В-1 УК-4: З-1 УК-4: У-1 УК-4: В-1 УК-5: З-1 УК-5: У-1 УК-5: В-1 УК-2: З-1 УК-2: У-1 УК-2: В-1	Л1.1 Л.1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Э1	

5. ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ

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

5.1.1. Перечень вопросов для подготовки к экзамену.
(промежуточный контроль успеваемости)
Раздел 1. Основы программирования на С/С++.

1. Алгоритм. Понятие. Определение. Свойства. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
2. Способы описания алгоритмов. Правила описания схем алгоритмов. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
3. Разновидности структур алгоритмов. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
4. Лексемы языка С/С++. Идентификаторы. Операции. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
5. Лексемы языка С/С++. Ключевые слова. Константы. (УК-2: З-1, УК-4: З-1,УК-5: З-1)

6. Переменные. Описание переменных. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
7. Переменные. Типы данных в С/С++.(УК-2: З-1, УК-4: З-1,УК-5: З-1)
8. Структура программы. Команда присваивания в С/С++.(УК-2: З-1, УК-4: З-1,УК-5: З-1)
9. Ввод-вывод данных. Функции scanf,printf. Формат выводимых данных. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
10. Ввод-вывод данных. Стандартные потоки ввода и вывода. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
11. Алгоритм линейной структуры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
12. Операции отношений. Логические операции в С. Таблицы истинности. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
13. Поразрядные операции в С. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
14. Структура IF(если- то- иначе) и ее программирование. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
15. Структура switch(выбор) и ее программирование. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
16. Оператор безусловного перехода GOTO. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
17. Алгоритмы циклической итерационной структуры. Оператор цикла While. Примеры использования. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
18. Алгоритмы циклической итерационной структуры. Оператор цикла do…while. Примеры использования. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
19. Алгоритмы циклической итерационной структуры. Оператор цикла For… . Примеры использования. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
20. Операторы breakicontinue. Примеры использования. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
21. Одномерные массивы. Задание массивам первоначальных значений. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
22. Операции над массивами и их совместимость. Ввод-вывод массивов. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
23. Ввод-вывод матриц. Операции над матрицами. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
24. Квадратная матрица. Диагонали матрицы. Пример алгоритма обработки квадратных матриц. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
25. Нахождение максимального (минимального) элемента массива. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
26. Понятие подпрограммы. Описание подпрограммы. Протопит функции. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
27. Понятие о стандартных директивах препроцессора. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
28. Формальные и фактические параметры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
29. Область видимости имен. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
30. Понятие о локальных и глобальных переменных. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
31. Рекурсия. Виды рекурсий. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
32. Обработка строковых данных. Операции со строками. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
33. Понятие структуры. Массивы структур. Обработка структур. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
34. Файлы. Типы файлов. Способы их описания. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
35. Способы доступа к файлам. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
36. Операции ввода/вывода с файлами через буфер. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
37. Текстовые файлы. Стандартные текстовые файла. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
38. Двоичные файлы. Функции ввода и вывода данных. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
39. Использование файлового указателя при программировании операций с файлами. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
40. Стандартные функции для работы с файлами в стиле С. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
41. Работа с файлами в стиле С++. Запись в файл. Чтение файлов. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
42. Перечисляемые типы данных. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
43. Интегрированные среды разработки. Интерфейс среды. Характеристика проекта. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
44. Основные компоненты панели Standard. Компиляция и выполнение проекта. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
45. Сортировка методом простого выбора. Суть. Алгоритм. Эффективность. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
46. Сортировка методом простого обмена. Суть. Алгоритм. Эффективность. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
47. Сортировка методом прямого включения. Суть. Алгоритм. Эффективность. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
48. Быстрая сортировка. Суть. Алгоритм. Эффективность. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
49. Модификация метода Прямого включения. Сортировка Шелла. Суть. Алгоритм. Эффективность(УК-2: З-1, УК-4: З-1,УК-5: З-1).
50. Модификация метода пузырька. Шейкерная сортировка. Суть. Алгоритм. Эффективность. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
51. Карманная (корзинная) сортировка. Суть. Алгоритм. Эффективность. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
52. Алгоритмы поиска информации. Линейный поиск. Суть. Алгоритм. Эффективность. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
53. Алгоритмы поиска информации. Бинарный поиск. Суть. Алгоритм. Эффективность(УК-2: З-1, УК-4: З-1,УК-5: З-1)
54. Понятие рекурсии и основные определения. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
55. Общие сведения о динамическом распределении памяти и динамических переменных. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
56. Указатели, объявление ссылочных переменных. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
57. Управление выделением и освобождением динамической памяти на логическом уровне. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
58. Создание однонаправленного списка. Добавление элемента в список. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
59. Создание однонаправленного списка. Удаление элемента из списка. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)

60. Создание двунаправленного списка. Добавление и удаление элемента из начала списка. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
61. Создание двунаправленного списка. Удаление элемента из списка по ключу. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
62. Понятие стека. Примеры использования стека в программировании. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
63. Основные приемы и особенности работы со стеками. Добавление элемента в стек. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
64. Основные приемы и особенности работы со стеками. Удаление элемента из стека. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
65. Способы организации очереди. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
66. Обработка очередей. Добавление элемента в очередь. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
67. Обработка очередей. Удаление элемента из очереди. Примеры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
68. Понятие Дерево. Бинарное дерево поиска. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
69. Бинарное Дерево. Формирование Дерева. Добавление узла в Дерево поиска. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
70. Бинарное Дерево. Рекурсивный и нерекурсивный поиск узла в Дереве. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
71. Бинарное Дерево. Алгоритмы Обхода Дерева. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
72. Бинарное Дерево. Удаление узла из Дерева. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
73. Обработка Бинарного Дерева. Подсчет узлов дерева. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
74. Обработка Бинарного Дерева. Определение высоты дерева. (УК-2: З-1, УК-4: З-1,УК-5: З-1)

Раздел 2. Приемы обеспечения технологичности программных продуктов.
(УК-2: З-1, УК-4: З-1,УК-5: З-1)

75. Цели и задачи технологий разработки ПО. Особенности современных крупных проектов разработки ПО. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
76. Структурный подход к проектированию ПО. Сущность структурного подхода. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
77. Объектно-ориентированная разработка программ. Объектно-ориентированные языки программирования. (УК-2: З-1, УК-4: З-1,УК-5: З-1) Объектно-ориентированные методологии разработки программных систем. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
78. Каскадная модель жизненного цикла ПС: содержание этапов, область применения, достоинства и недостатки. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
79. Эволюционная модель жизненного цикла ПС: последовательность действий, область применения, достоинства и недостатки. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
80. Спиральная модель разработки ПО: содержание этапов создания ПС, область применения, достоинства и недостатки. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
81. Инкрементальная модель разработки ПО. Развитие инкрементального подхода. XP-процессы. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
82. Понятие программного проекта. Управление программным проектом. План и содержание его разделов. Составление сетевого графика работ. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
83. Состав и структура коллектива разработчиков программного продукт, их функции. Составление расписания (PERT-диаграммы) (УК-2: З-1, УК-4: З-1,УК-5: З-1)
84. Управление документацией разработки программного продукта.
85. Рациональный Унифицированный Процесс. Динамические аспекты процессов: структура ЖЦ, стадии, итерации и контрольные точки. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
86. Рациональный Унифицированный Процесс. Статическое содержание процесса: виды деятельности (технологические операции), рабочие продукты, исполнители и дисциплины (технологические процессы).
87. Внешнее описание программного средства и спецификация. Виды требований к ПО: системные, функциональные, характеристики качества. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
88. Методы определения и формализация требований к ПО.
89. Понятие качества ПО и его многоуровневая модель. Характеристики и атрибуты качества. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
90. Разработка требований к ПО: формирование и анализ, документирование, аттестация. Управление. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
91. Алгоритмическая декомпозиция. Модульное программирование. Характеристики программного модуля. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
92. Модели архитектур с различными способами обмена данными: репозиторий, «клиент-сервер». (УК-2: З-1, УК-4: З-1,УК-5: З-1)
93. Архитектуры с различными моделями управления. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
94. Событийно-управляемые архитектуры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
95. Модели архитектур с различными подходами к обработке данных: непрерывная обработка, каналы и фильтры. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
96. Объектно-ориентированная декомпозиция. Общая характеристика объектов. Виды отношений между объектами. Агрегация. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
97. Абстрагирование. Общая характеристика классов. Виды отношений между классами. Ассоциации классов. Наследование. Полиморфизм. Агрегация. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
98. Повторное использование компонентов. Инкапсуляция. Интерфейсы. Компонентная объектная модель (COM). (УК-2: З-1, УК-4: З-1,УК-5: З-1)

Раздел 3. Разработка пользовательских интерфейсов

(УК-2: З-1, УК-4: З-1,УК-5: З-1)

99. Принципы проектирования пользовательского интерфейса. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
100. Структурное тестирование. Покрытие операторов, ветвей, условий. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
101. Функциональное тестирование. Метод эквивалентного разбиения, граничных значений, причинно-следственных (функциональных) диаграмм. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
102. Тестирование интеграции компонентов ПО: нисходящее и восходящее. Понятие драйвер и заглушка. Стохастическое тестирование. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
103. Разработка программной документации. С-документация и П-документация. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
104. Отладка ПО: цели и методы. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
105. Управление конфигурацией ПО. Системы контроля версий. Регрессионное тестирование. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
106. Аттестация ПО. Оценка качества ПО. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
107. Инструментальные средства разработки ПО. Автоматизация разработки ПО. CASE-средства. (УК-2: З-1, УК-4: З-1,УК-5: З-1)
108. Сопровождение ПО. Основные подходы: с целью исправления ошибок, адаптации и изменения функциональных возможностей. Решение проблемы эволюции ПО – рефакторинг, реинженерия, реверсная инженерия. (УК-2: З-1, УК-4: З-1,УК-5: З-1)

5.1.2. Перечень контрольных практических заданий, используемых при формировании экзаменационных билетов (промежуточный контроль успеваемости)

(УК-2: У-1, УК-4: У-1,УК-5: У-1)

1.	Дана последовательность целых чисел из n элементов. Найти: - среднее арифметическое; - (максимальное значение; - количество отрицательных элементов; -номер минимального элемента; -количество четных чисел; - минимальный из четных элементов этой последовательности.
2.	Дана последовательность целых чисел, за которой следует 0. Найти: - среднее арифметическое; - (максимальное значение; - количество отрицательных элементов; -номер минимального элемента; -количество четных чисел; - минимальный из четных элементов этой последовательности.
3.	Найти сумму чисел Фибоначчи, меньших заданного числа Q.
4.	Напечатать N простых чисел.
5.	Дан массив целых чисел. Найти: - среднее арифметическое; - максимальное значение; - номер минимального элемента.
6.	Дан массив целых чисел. Реверсировать массив.
7.	Дан массив целых чисел. Поменять местами пары элементов в массиве: 1 и 2, 3 и 4, 5 и 6 и т. д.
8.	Циклически сдвинуть массив на К элементов влево.
9.	Найти первое вхождение элемента К в массив целых чисел.
10.	Удалить из динамической матрицы строку с номером К.
11.	Дана строка символов, состоящая из слов, слова разделены между собой пробелами. Удалить из строки все слова, начинающиеся с цифры.
12.	Сформировать динамический массив строк. Удалить из него строку с заданным номером.
13.	Заданы координаты сторон треугольника. Если такой треугольник существует, то найти его площадь. Решить задачу с использованием функций.
14.	Дан массив int a[100]. Удалить из массива все четные элементы.
15.	Дан массив int *a.Удалить из массива все элементы, совпадающие с первым элементом, используя динамическое выделение памяти.
16.	Найти количество цифр в строке символов, используя функции.

17.	Удалить из бинарного файла, в котором записаны целые числа все четные элементы.
18.	Добавить в бинарный файл, в который записаны элементы типа struct Student {char name[20];int age;}; К элементов после элемента с заданной фамилией.
19.	Удалить из текстового файла все четные строки.
20.	struct Date {char Month[15]; int Day;} В файле содержатся даты типа Date. Заменить все даты, у которых поле Month равно “Май”, ”Июнь” или ”Июль” на даты, у которых поле Day не меняется, а поле Month меняется на “Август”.
21.	Добавить порядковый номер в каждую строку текстового файла.
22.	В текстовом файле заменить все строки, начинающиеся с буквы ‘f’ на строки, начинающиеся с буквы ‘a’.
23.	Удалить из текстового файла все нечетные строки.
24.	Циклически сдвинуть массив на К элементов вправо.
25.	Дан массив целых чисел. Найти: - количество отрицательных элементов; - количество четных чисел; - минимальный из четных элементов этого массива.

5.2. Перечень работ, выполняемых в процессе изучения дисциплины

В процессе изучения дисциплины «Технология программирования» обучающийся должен выполнить и защитить:

5.2.1. Комплект вопросов для защиты лабораторных работ (текущий контроль успеваемости)

(УК-2: У-1, УК-4: У-1,УК-5: У-1, УК-2: В-1, УК-4: В-1,УК-5: В-1)

Лабораторная работа 1

Выполнение программы простой структуры. Вычисление выражений с использованием стандартных функций».

1.Какую структуру имеет программа на языке С++?
2.Какова роль функции void main() ?
3.Как формируется исходный код программы?

Лабораторная работа 2

Использование основных операторов языка С++.

1.Что называют управляющими конструкциями программы?
2.Составные операторы?
3.Операторы выбора?
4.Операторы циклов?
5. Операторы перехода?

Лабораторная работа 3

Работа с одномерными массивами.

1.Определение массива в С++?
2.Для чего предназначены указатели?
3.Одномерные массивы?
4.Как происходит обработка элементов массива?
5.Классы задач при обработке массива?
6.Сортировка массивов?

Лабораторная работа 4

Функции и массивы в С++.

1.Определение функции в С++?
2.Параметры функции. Способы передачи параметров в функцию?
3.Локальные и глобальные переменные в С++. Особенности и использования при программировании?
4.Какие элементы можно передавать в качестве параметров функции?

Лабораторная работа 5

Динамические массивы.

1.Для чего используются динамические массивы в С++?
2.Как происходит обработка элементов массива?
3.Классы задач при обработке динамического массива?

Лабораторная работа 6

Массивы структур и массивы строк.

1.Определение структуры в С++?
2.Какого типа могут быть элементы структуры?
3.Как происходит обработка элементов структуры?
4.Как сформировать динамический массив из элементов структурного типа массива?

Лабораторная работа 7

Функции в С++.
1.Определение функции в С++?
2.Функции с начальными значениями параметров (по умолчанию)?
3.Порядок работы с шаблонами функций?
4.Как использовать указатели функций?
5.Особенности применения функций с переменным числом параметров?
6.Какова цель перегрузки функций?

Лабораторная работа 8

Динамические структуры данных.
1.Понятие однонаправленных списков в С++?
2.Понятие двунаправленных списков в С++?
3.Как происходит обработка очередей в С++?
4.Особенности обработки стека?
Понятие бинарного дерева в С++?

5.2.2. Практические работы:

Практическая работа № 1 Варианты (1-3) (ОПК-3.1: У-1, ОПК-3.1: У-2, ОПК-3.1: В-1, УК-7.1: У-1, УК-7.1: У-2, УК-7.2: У-1, ПК-5.1: У-1)

Практическая работа № 2 Варианты (1-20) (ОПК-3.1: У-1, ОПК-3.1: У-2, ОПК-3.1: В-1, УК-7.1: У-1, УК-7.1: У-2, УК-7.2: У-1, ПК-5.1: У-1)

5.1.3. Комплект вопросов для защиты практических работ (текущий контроль успеваемости)

(УК-2: У-1, УК-4: У-1,УК-5: У-1, УК-2: В-1, УК-4: В-1,УК-5: В-1)

Практическая работа № 1. Вариант 1

Ответить на вопросы в письменной форме:
1. Как записывается и выполняется команда выбора?
2. Какая разница между условиями, записанными после слова while и repeat для одной и той же задачи?
3. Как организовать цикл с уменьшением счетчика?
4. Какого типа выражения допустимы в качестве начального и конечного значений?
5. Как необходимо записать несколько операторов, чтобы они выполнялись в цикле for?
6. Как описывается одномерный массив?
7. Как описывается двумерный массив?
8. Что можем использовать в качестве индекса?
9. Что фактически указывает тип индекса?
10. Какой тип может быть типом индекса?
11. Придумать задания под приведенные блок-схемы.

Практическая работа № 1. Вариант 2

Ответить на вопросы в письменной форме:
1. Как выполняется команда ветвления в неполной форме?
2. Какие логические операции используются в команде ветвления?
3. Как оформляется цикл while?
4. Когда завершается выполнение цикла while?
5. Как организовать выполнение нескольких операторов в цикле while?
6. Управляющие переменные какого типа допустимы в операторе for?
7. Как обращаться к элементам массива?
8. Чем однозначно определяется значение элемента массива?
9. Назовите простые типы используемые в массиве?
10. Как называем элемент массива?
11. Придумать задания под приведенные блок-схемы.

Практическая работа № 1. Вариант 3

Ответить на вопросы в письменной форме:
1. Как выполняется команда ветвления в полной форме?
2. Какие правила оформления команды ветвления существуют?
3. Какие алгоритмы называются циклическими?
4. Какой из операторов цикла является самым универсальным?
5. Для управления циклом Repeat какие функции удобно использовать?
6. Когда используется оператор цикла FOR?
7. Какого типа выражения допустимы в качестве начального и конечного значений?
8. Что такое массив?
9. Где должен быть описан, используемый в программе массив?
10. Какого типа должен быть индекс?
11. Придумать задания под приведенные блок-схемы.

Практическая работа № 2. Вариант 1
1. Чему будет равно значение переменной x после выполнения фрагмента программы?
```
x:=0;
for i:=1 to 5 do
x:=x+1;
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для вычисления функции при заданном значении X

$$Z=\begin{cases} x-0.5 & \text{при } x<0 \\ x & \text{при } 0<x<4 \\ \sin x+\cos x & \text{при } x<4 \end{cases}$$

Практическая работа № 2. Вариант 2
1. Чему будет равно значение переменной f после выполнения фрагмента программы?
```
f:=2; x:=0;
for i:=1 to 5 do
begin
x:=x+1;
f:= f * x;
end;
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для вычисления размера подоходного налога (N). Подоходный налог определяется: 1) зарплата (Z) меньше минимальной заработной платы (M) – не облагается; 2) если выше, то берется 13% от суммы, превышающей минимальную.

Практическая работа № 2. Вариант 3
1. Чему будет равно значение переменной S после выполнения фрагмента программы?
```
x:=0; S:=0;
while x<= 2 do
begin
S:= S + x;
x:=x+2;
end;
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для вывода на экран агрегатного состояния воды (лед, жидкость, пар) в зависимости от введенной температуры.

Практическая работа № 2. Вариант 4
1. Чему будет равно значение переменной S после выполнения фрагмента программы?
```
S:=1; x:=0;
while x> 0 do
begin
S:= S * 2;
x:=x-1;
end;
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для ввода с клавиатуры числа (в диапазоне от 1 до 999), обозначающего денежную единицу, и вывода результата с дописанным словом «рубль», «доллар» в правильной форме. Например, 12 рублей, 1 рубль, 3 рубля.

Практическая работа № 2. Вариант 5
1. Чему будет равно значение переменной S после выполнения фрагмента программы?
```
x:=0; S:=0;
repeat
S:= S + x;
x:=x+2;
until x>2 do
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для определения, является ли треугольник с длинами сторон a, b, c прямоугольным.

Практическая работа № 2. Вариант 6
1. Определить значение переменной y после выполнения следующих элементов программы:
```
Var
x,y : integer:
……————————————
y:=0;
For x:=1 to 9 do y:=y+1;
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для определения – существует ли треугольник с такими сторонами, если заданы три длины отрезков a, b, c.

Практическая работа № 2. Вариант 7
1. Определить значение переменной y после выполнения следующих элементов программы
```
Var
x,y : integer:
```

……
Y := 0;
For x := 5 downto 10 do y := y+x;
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для выдачи на экран названия времени года в зависимости от введенного номера месяца.

Практическая работа № 2. Вариант 8
1. Определить значение переменной n после выполнения следующих элементов программы:
Var
x,y:integer:
……
y:=0;
x:=10;
while x>0 do
Begin
x:=x+2;
y:=y+x;
end;
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для определения какая фигура имеет большую площадь, если заданы радиус круга R и сторона квадрата A.

Практическая работа № 2. Вариант 9
1. Определить значение переменной x после выполнения следующих элементов программы :
Var
x:real;
y:integer:
……
x:=0;
y:=5;
repeat
y:=2*y;
x:=x+y
until y>5;
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для вычисления значения функции y. Если x < 1, то y = -1; если x > 1, то y = x; если x = 1, то y = 1.

Практическая работа № 2. Вариант 10
1. Определить значение переменной y после выполнения следующих элементов программы:
Var
y:real;
x:integer;
……
y: = 0;
For x:= 4 to 6 do y:=y+x;
y:=y/10;
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для определения максимального и минимального значения из трех различных чисел.

Практическая работа № 2. Вариант 11
1. Определить значение переменной y после выполнения следующих элементов программы:
Var
y:real;
k:integer:
……
y:=0;
For k:=2 to 6 do y:=y+k;
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для ввода произвольного числа A; возведения числа A в квадрат, если число отрицательное; извлечение из A квадратного корня, если число положительное; оставить A без изменения, если A=0. Решение задачи выводить на экран с соответствующими комментариями.

Практическая работа № 2. Вариант 12
1. Определить значение переменной y после выполнения следующих элементов программы:
Var
y:real;
k:integer:
……
y:=1;
For k:= 6 downto 3 do y:=y+k;
2. В зависимости от ввода длины в см. 100; 71. 12; 30. 0; 4. 45 и 2. 54 выводить на экран соответственно слова: метр,

аршин, фут, вершок и дюйм. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы на Object Pascal.

Практическая работа № 2. Вариант 13
1. Определить значение переменной y после выполнения следующих элементов программы:
```
Var
y:real;
k:integer:
……
y:=1;
For k:=6 to 3 do y:=y+k;
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для выполнения арифметических действий (+, -, /, *) над вводимыми аргументами (калькулятор).

Практическая работа № 2. Вариант 14
1. Определить значение переменной y после выполнения следующих элементов программы:
```
Var
y:real;
k:integer:
……
y:=1;
For k:=1 to 3 do y:=y+k;
y:=y*10;
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для определения по A-градусной мере угла, является ли он острым, тупым, прямым углом.

Практическая работа № 2. Вариант 15
1. Определить значение переменной s после выполнения следующих элементов программы:
```
Var
S : real;
I : integer:
……
s:=0;
For i:=1 to 4 do
Begin
s:=s*10;
s:=s+i;
end;
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для определения, являются ли значения целочисленных переменных M и N кратными. Если оба кратны 3, то вычислить их сумму, иначе вычислить их произведение.

Практическая работа № 2. Вариант 16
1. Определить значение переменной s после выполнения следующих элементов программы:
```
Var
s:real;
n:integer:
……
s:=0;
For n:=6
downto 3 do
Begin
s:=s+1;
s:=s*2;
end;
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для ввода двух целых чисел, замены первого нулем, если оно меньше или равно второму. Решение задачи выводить на экран с соответствующими комментариями.

Практическая работа № 2. Вариант 17
1. Определить значение переменной s после выполнения следующих элементов программы:
```
Var
s:real;
i:integer:
……
s:=0;
i:=5;
while i> 2 do i:=i-1;
s:=s+i*i;
```
2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для определения, можно ли

составить треугольник из введенных длин 3-х отрезков. Если да, то проверить, является ли он равносторонним или равнобедренным. Решение задачи выводить на экран с соответствующими комментариями.

Практическая работа № 2. Вариант 18
1. Определить значение переменной s после выполнения следующих элементов программы:
```
Var
s:real;
i:integer:
……————————————
s:=0;
i:=2;
repeat
i:=2*i;
s:=s+i
until i>5;
```
2. Каждое из чисел a и b отлично от 0. Если они одинаковых знаков, то заменить меньшее из них большим; если же числа имеют разные знаки, присвоить каждому из них знак числа, меньшего по абсолютной величине. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы.

Практическая работа № 2. Вариант 19
1. Определить значение переменной s после выполнения следующих элементов программы:
```
Var
s:real;
i:integer:
……————————————
s:=0;
i:=1;
while i>1 do
Begin
s:=s+1/i;
i:=i-1;
end;
```
2. Даны треугольники со сторонами: a, b, c и k,l,f. Проверить, равны ли эти треугольники. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы.

Практическая работа № 2. Вариант 20
1. Определить значение переменной s после выполнения следующих элементов программы:
```
Var
s:real;
i:integer:
……————————————
s:=0;
i:=5;
repeat
i:=2*i;
s:=s+i
until i>5;
```
2. Самолет летит из пункта A в пункт B со средней скоростью V. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы для нахождения времени полета t1, если есть встречный ветер, скорость которого V1, и времени t2, если ветра нет. Расстояние между пунктами равно S.

5.2.3. Домашние задания: (ОПК-3.1: У-1, ОПК-3.1: У-2, ОПК-3.1: В-1, УК-7.1: У-1, УК-7.1: У-2, УК-7.2: У-1, ПК-5.1: У-1)
ДЗ 1: Тема: Расчет функций (первая задана с помощью ряда Тейлора, корень второго уравнения необходимо найти при помощи метода Ньютона);
ДЗ 1: Тема: Расчет интеграла методом Симпсона;
ДЗ 2: Тема: Расчет интеграла методом Гаусса;
ДЗ 3: Тема: Расчет интеграла методом прямоугольников;
ДЗ 4: Тема: Расчет интеграла методом трапеций;
ДЗ 5: Тема: Решение систем линейных уравнений методом Гаусса;
ДЗ 6: Тема: Решение систем линейных уравнений методом Крамера;
ДЗ 7: Тема: Решение систем линейных уравнений методом Зейделя;
ДЗ 8: Тема: Решение систем уравнений методом Ньютона;
ДЗ 9: Тема: Решение уравнений методом половинного деления, секущих, Ньютона, хорд;
ДЗ 10: Тема: Сортировка методом простых вставок;
ДЗ 11: Тема: Сортировка методом бинарных вставок;
ДЗ 12: Тема: Сортировка методом слияния;
ДЗ 13: Тема: Сортировка методом выбора;
ДЗ 14: Тема: Сортировка методом пузырька;

ДЗ 15: Тема: Моделирование броуновского движения;
ДЗ 16: Тема: Изображение электронных часов;
ДЗ 17: Тема: Моделирование движения футбольного мяча после удара;

Комплект вопросов и заданий для защиты домашних заданий
(текущий контроль успеваемости)

Домашнее задание ((УК-2: У-1, УК-4: У-1,УК-5: У-1, УК-2: В-1, УК-4: В-1,УК-5: В-1)

Содержание работы: домашнее задание предполагает постановку и решение совокупности аналитических, расчетных, синтетических, исследовательских, оценочных задач, объединенных общностью рассматриваемого объекта..
В ходе решения студент должен:
1. Выполнить постановку задачи в соответствии с вариантом задания;
2. Используя методы математического анализа, создать математическую модель решения поставленной задачи;
3. Сделать схему алгоритма решения задачи с учетом требований ГОСТ;
4. Написать программу на языке С++, реализующую представленную математическую модель в соответствии со схемой алгоритма;
5. Протестировать все ветви работы программы и проанализировать полученные результаты;
6. Написать руководство пользователя;
7. Сделать выводы по работе в целом.

Список ключевых слов, понятий, терминов для тестирования:
1. Программа, программное обеспечение.
2. Задача, приложение.
3. Предметная область.
4. Методология программирования.
5. Технология программирования.
6. Процесс.
7. Программный продукт.
8. ЖЦ программного средства.
9. Язык программирования.
10. Язык программирования низкого уровня.
11. Язык программирования высокого уровня.
12. Машинный язык.
13. Загрузочный модуль.
14. Объектный модуль.
15. Компилятор.
16. Интерпретатор.
17. Транслятор.
18. Декомпилятор.
19. Система программирования.
20. Интегрированная среда разработки.
21. Быстрая разработка приложений.
22. Графический интерфейс пользователя.
23. Интерфейс командной строки.
24. Метод.
25. Событие.
26. Свойство.
27. Объект.
28. Компонент.
29. Кнопка.
30. Надпись.
31. Элемент управления.
32. Флажки.
33. Переключатели.
34. Обработчик события.
35. Пиксель.
36. Пиктограмма.
37. Тип файлов, составляющих проект Delphi.
38. Пять простых компонентов Delphi.
39. Соглашения об именах компонентов.
40. Внутренняя программная документация.
41. Внешняя документация программы.
42. Конкатенация.
43. Интерфейсный раздел модуля.
44. Раздел реализации модуля.
45. Инициализация.
46. Ключевое слово.
47. Синтаксис.
48. Инкапсуляция.
49. Наследование.
50. Полиморфизм.

51. Функция.
52. Подпрограмма.
53. Процедура.
54. Фактический параметр.
55. Формальный параметр.
56. Фокус.
57. Параметр Sender.
58. Параметр Self.
59. Указатель nil.
60. Синтаксическая ошибка.
61. Логическая ошибка.
62. Деструктор.
63. Конструктор.
64. Класс.
65. Объект.
66. Создание объекта.
67. Директива компилятора.

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

Экзаменационный билет включает в себя фундаментальный теоретический вопрос и прикладной теоретический вопрос из установленного перечня контрольных вопросов, используемых при формировании экзаменационных билетов при оценке знаний обучающихся на экзамене по темам, изложенным в разделах 1-3 данной РПД, а также практическое задание из установленного перечня контрольных заданий, используемых при формировании экзаменационных билетов.
Пример экзаменационного билета:

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
СТАРООСКОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ ИМ. А.А. УГАРОВА
(филиал) федерального государственного образовательного учреждения
высшего образования «Национальный исследовательский
технологический университет «МИСиС»
Кафедра «АВТОМАТИЗИРОВАННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ УПРАВЛЕНИЯ»
09.03.02 – Информационные системы и технологии
Профиль 01 – Информационные системы и технологии
Дисциплина **«Технология программирования»**

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Формальные и фактические параметры.
2. Цели и задачи технологий разработки ПО.
3. Задача. Заданы координаты сторон треугольника. Если такой треугольник существует, то найти его площадь. Решить задачу с использованием функций.

«____»_____2020г.

Экзаменатор _____ Е.А. Михайлюк

Утверждено на заседании кафедры АИСУ

Протоколом №___ от _____ 202___ г.

Заведующий кафедрой АИСУ _____ А.И. Глущенко

Билеты в бумажном виде хранятся на кафедре АИСУ и утверждены ее заведующим (или заместителем зав. кафедрой).

5.4. Методика оценки освоения дисциплины

№ п/п	Вид оценочного средства	Критерий	Оценка
1.	Выполнение и защита практических работ	Обучающийся самостоятельно выполняет полное и аргументированное решение индивидуальных заданий, не допустив ошибок. При защите заданий отвечает развернуто и исчерпывающе на все вопросы.	«Отлично»
		Обучающийся практически самостоятельно выполняет полное решение заданий, но не может аргументировать свое решение. При защите заданий допускает незначительные неточности.	«Хорошо»
		Обучающийся в целом правильно решает задание, но не может аргументировать свое решение. При защите заданий допускает значительные неточности. Обучающийся правильно понимает способ решения заданий, но допускает ошибки при их решении. Задание выполнено частично. При защите заданий допускает значительные неточности.	«Удовлетворительно»
		Обучающийся не может решить задание.	«Неудовлетворительно»
2.	Выполнение и защита лабораторных работ	Обучающийся самостоятельно выполняет полное и аргументированное решение индивидуальных заданий, не допустив ошибок. При защите заданий отвечает развернуто и исчерпывающе на все вопросы.	«Отлично»
		Обучающийся практически самостоятельно выполняет полное решение заданий, но не может аргументировать свое решение. При защите заданий допускает незначительные неточности.	«Хорошо»

			Обучающийся в целом правильно решает задание, но не может аргументировать свое решение. При защите заданий допускает значительные неточности.	«Удовлетворительно»	
			Обучающийся правильно понимает способ решения заданий, но допускает ошибки при их решении. Задание выполнено частично. При защите заданий допускает значительные неточности.	«Неудовлетворительно»	
	3.	Выполнение и защита домашнего задания	Обучающийся самостоятельно выполняет полное и аргументированное решение индивидуальных заданий, не допустив ошибок. При защите заданий отвечает развернуто и исчерпывающе на все вопросы.	«Отлично»	
			Обучающийся практически самостоятельно выполняет полное решение заданий, но не может аргументировать свое решение. При защите заданий допускает незначительные неточности.	«Хорошо»	
			Обучающийся в целом правильно решает задание, но не может аргументировать свое решение. При защите заданий допускает значительные неточности. Обучающийся правильно понимает способ решения заданий, но допускает ошибки при их решении. Задание выполнено частично. При защите заданий допускает значительные неточности.	«Удовлетворительно»	
			Обучающийся не может решить задание.	«Неудовлетворительно»	
	4.	Экзамен	Компетенции (УК-2, УК-4,УК-5, ОПК-6) сформированы. Обучающийся демонстрирует: - глубокие знания содержания изученной дисциплины во взаимосвязи с другими дисциплинами; - способность использовать теоретические знания при выполнении практических заданий; - аргументированные, исчерпывающие ответы на все вопросы по билету, а также дополнительные вопросы экзаменатора; - умение выполнять и обосновывать решение практических заданий высокого уровня сложности; - наличие собственной обоснованной позиции по обсуждаемым вопросам; - свободное использование в ответах на вопросы материалов рекомендованной основной и дополнительной литературы.	«Отлично»	
			Компетенции (УК-2, УК-4,УК-5, ОПК-6) сформированы. Обучающийся демонстрирует: - знание основных терминов по содержанию изученной дисциплины; - твердые знания теоретического материала; - умение дать четкие ответы на поставленные вопросы; - умение решать практические задания; - владение основной литературой, рекомендованной программой дисциплины. Допускаются незначительные неточности в ответах на теоретические вопросы и при выполнении практических заданий.	«Хорошо»	
			Компетенции (УК-2, УК-4,УК-5, ОПК-6) сформированы. Обучающийся демонстрирует: - знания теоретического материала по изученной дисциплине;	«Удовлетворительно»	

			- неполные ответы на основные вопросы, допуская ошибки в ответе; недостаточное понимание сущности излагаемых вопросов; - неточные ответы на дополнительные вопросы; - умение выполнять практические задания без грубых ошибок; - недостаточное владение литературой, рекомендованной программой дисциплины.	
			Компетенции (УК-2, УК-4,УК-5, ОПК-6) не сформированы. Обучающийся демонстрирует: - существенные пробелы в знаниях учебного материала; - принципиальные ошибки при ответе на основные вопросы билета, отсутствие знаний и понимания основных терминов и определений; - непонимание сущности дополнительных вопросов в рамках заданий билета; - отсутствие навыка или существенные ошибки при выполнении практических заданий; - незнание литературы, рекомендованной программой дисциплины.	«Неудовлетворительно»

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Рекомендуемая литература

6.1.1 Основная литература

Обозначение	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л 1.1	Давыдов, В.	Технологии программирования C++	НТБ СТИ НИТУ «МИСиС»	СПб. : БХВ-Петербург, 2005
Л 1.2	Калашников, Е. А.	Технологии программирования	НТБ СТИ НИТУ «МИСиС»	МИСИС, 2004.
Л 1.3	Иванова, Г. С.	Технология программирования	НТБ СТИ НИТУ «МИСиС»	М. : МГТУ им. Н.Э. Баумана, 2003
Л 1.4	Хорев, П. Б.	Технологии объектно-ориентированного программирования	НТБ СТИ НИТУ «МИСиС»	М. : Издательский центр "Академия", 2008

6.1.2 Дополнительная литература

Обозначение	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л 2.1	Павловская, Т. А.	C/C++. Программирование на языке высокого уровня	НТБ СТИ НИТУ «МИСиС»	СПб.: Питер, 2001
Л 2.2	Камаев, В. А.	Технологии программирования. Оптимизация программных разработок	НТБ СТИ НИТУ «МИСиС»	Волгоград : Волг. ГТУ, 1998
Л 2.3	Гома, Х.	UML. Проектирование систем реального времени, параллельных и распределенных приложений	НТБ СТИ НИТУ «МИСиС»	М.: ДМК Пресс, 2002

6.1.3 Методические материалы

Обозначение	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л 3.1	Михайлюк Е.А.	Технология программирования: методические указания для самостоятельного выполнения домашнего задания	https://lms.misis.ru/enroll/YG4LAD	Старый Оскол. СТИ НИТУ МИСиС - 2020
Л 3.2	Михайлюк Е.А.	Технология программирования: методические указания к практическим занятиям	https://lms.misis.ru/enroll/YG4LAD	Старый Оскол. СТИ НИТУ МИСиС - 2020
Л 3.3	Михайлюк Е.А.	Технология программирования: методические указания к лабораторным работам	НТБ СТИ НИТУ «МИСиС»	Старый Оскол. СТИ НИТУ МИСиС - 2020
Л 3.4	Михайлюк Е.А.	Технология программирования: курс лекций	НТБ СТИ НИТУ «МИСиС»	Старый Оскол. СТИ НИТУ МИСиС - 2020
6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»				
Э. 1	Технологии программирования: учебное пособие / Г.С. Иванова. – 3 –е. изд., стер. – М.: КНОРУС, 2013. - 336 с. [Электронный ресурс]. - URL: https://avidreaders.ru/read-book/tehnologiya-programmirovaniya.html			
6.3. Перечень программного обеспечения				
П. 1	Dev cpp (свободно распространяемое программное обеспечение);			
П. 2	LabVIEW;			
П. 3	Microsoft Windows;			
П. 4	Microsoft Office;			
П. 5	QtDesigner (свободно распространяемое программное обеспечение);			
П. 6	7- Zip (свободно распространяемое программное обеспечение);			
П. 7	Kaspersky Endpoint Security.			
6.4. Перечень информационных справочных систем и профессиональных баз данных				
	Электронные образовательные ресурсы (ЭОР):			
И. 1	LMS Canvas (приказ НИТУ «МИСиС» № 387 о.в. от 05.06.2018 г. «О применении в учебном процессе ЭОР»)			
И. 2	MS Teams			
И. 3	Федеральный портал «Российское образование»: http://edu.ru			
И. 4	Открытое образование: http://openedu.ru			
И. 5	Российская государственная библиотека: http://www.rsl.ru			
	Электронно-библиотечные системы (ЭБС):			
И. 6	- Электронно-библиотечная система (ЭБС) «Университетская библиотека онлайн»: http://biblioclub.ru			
И. 7	- Электронная библиотека НИТУ «МИСиС»: http://elibrary.misis.ru			
И. 8	- Научная электронная библиотека eLIBRARY.RU: http://elibrary.ru/			
	Профессиональные базы данных и информационные справочные системы:			
И. 9	- Университетская информационная система РОССИЯ: https://uisrussia.msu.ru/			
И. 10	- Электронная библиотека РГБ: диссертации: http://diss.rsl.ru/?menu=disscatalog/			
	Иностранные базы данных (доступ с IP адресов МИСиС):			
И. 11	- аналитическая база (индексы цитирования) Web of Science https://apps.webofknowledge.com			
И. 12	- аналитическая база (индексы цитирования) Scopus https://www.scopus.com/			
И. 13	- наукометрическая система InCites https://apps.webofknowledge.com			
И. 14	- научные журналы издательства Elsevier https://www.sciencedirect.com/			

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Ауд.	Назначение	Оснащение
416	Лаборатория искусственного интеллекта	Перечень основного оборудования, учебно-наглядных пособий: • персональный компьютер - 9 шт.; • экран; • мультимедиа проектор; • комплект учебной мебели на 25 посадочных мест; • лабораторный стенд «исследование сервоприводов».
306	Кабинет для самостоятельной работы	Перечень основного оборудования, учебно-наглядных пособий: 1. Проектор 2. Доска 3. Экран настенный 4. Компьютер – 6 шт 5. Комплект учебной мебели на 20 человек

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

Для успешного освоения дисциплины "Технология программирования" во 2 семестре обучающемуся необходимо:
1. Посещать все виды занятий.
2. Своевременно зарегистрироваться на рекомендованные электронные ресурсы.
3. При возникновении любых вопросов по содержанию курса и организации работы своевременно обращаться к преподавателю (в часы очных консультаций, через MS Teams или LMS Canvas).
4. Отчеты по практическим, лабораторным работам и домашним заданиям рекомендуется выполнять с использованием MS Office, допускается выполнять в рукописном виде.
5. Активно работать с научными базами в сети Интернет.
6. Качественное освоение дисциплины возможно только при систематической самостоятельной работе, что поддерживается системой текущей аттестации.

КОНТРОЛЬ КАЧЕСТВА ОСВОЕНИЯ ДИСЦИПЛИНЫ

Оценка качества подготовки обучающихся проводится с целью контроля освоения обучающимися совокупности компетенций (частей компетенций) на этапе изучения данной дисциплины. Освоение компетенций характеризуются определенными знаниями, умениями и навыками, опытом профессиональной деятельности, которые оцениваются как в процессе изучения дисциплины (текущий контроль успеваемости), так и по завершении изучения дисциплины при проведении промежуточной аттестации обучающихся. Для проведения текущего контроля успеваемости и промежуточной аттестаций обучающихся сформированы оценочные средства.

Текущий контроль успеваемости

Текущий контроль успеваемости обучающихся по дисциплине проводится в форме:
- практические работы (2).
- домашние задания (17).
- лабораторные работы (8).

По результатам выполнения практических работ и домашних заданий обучающиеся оформляют отчеты (по ГОСТ 7.32–2017 Отчет о научно-исследовательской работе).
Структурными элементами отчетов являются:
- титульный лист;
- содержание (по желанию);
- номер варианта, формулировку задания и исходные данные;
- необходимые схемы (исходные и промежуточные);
- расчётные формулы и полученные численные результаты;
- выводы по проделанной работе;
- необходимые для расчётов справочные данные (при необходимости);
- список использованных источников (при необходимости);
- приложения (при необходимости).

Промежуточная аттестация

Учебным планом ОПОП ВО по дисциплине Технология программирования предусматривается промежуточная аттестация в форме экзамена в 2 семестре. Промежуточная аттестация проводится с целью оценки качества освоения обучающимися содержания дисциплины. При проведении промежуточной аттестации обучающийся демонстрирует знания, умения и навыки, приобретенные в процессе изучения дисциплины, которые характеризуют результат освоения совокупности компетенций (частей компетенций) на этапе изучения данной дисциплины.
Результаты обучения по дисциплине, характеризующие освоение совокупности компетенций (части компетенций), при проведении промежуточной аттестации оцениваются по четырех балльной системе.

<u>Система оценивания результатов освоения дисциплины</u>

Порядок, определяющий процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций, определен в Положении «О текущем контроле успеваемости и промежуточной аттестации обучающихся, текущем контроле посещения обучающимися аудиторных занятий в НИТУ «МИСиС» П 239.09- 18, выпуск 2».