

Федеральное государственное бюджетное образовательное учреждение высшего образования «Тамбовский государственный университет имени Г.Р. Державина»
Институт математики, физики и информационных технологий
Кафедра математического моделирования и информационных технологий

УТВЕРЖДАЮ:
Директор института



Н. Л. Королева
«05» июля 2021 г.

РАБОЧАЯ ПРОГРАММА

по дисциплине Б1.В.ДВ.05.3 "High-level language programming"

Направление подготовки/специальность: 09.03.03 - Прикладная информатика

Профиль/направленность/специализация: Прикладная информатика в
информационной сфере

Уровень высшего образования: бакалавриат

Квалификация: Бакалавр

год набора: 2021

Автор программы:

Кандидат технических наук, Соловьев Денис Сергеевич

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 09.03.03 - Прикладная информатика (уровень бакалавриата) (приказ Министерства образования и науки РФ от «19» сентября 2017 г. № 922).

Рабочая программа принята на заседании Кафедры математического моделирования и информационных технологий «18» мая 2021 г. Протокол № 9

Рассмотрена и одобрена на заседании Ученого совета Института математики, физики и информационных технологий, Протокол от «05» июля 2021 г. № 5.

СОДЕРЖАНИЕ

1. Цели и задачи дисциплины.....	4
2. Место дисциплины в структуре ОП бакалавра.....	5
3. Объем и содержание дисциплины.....	5
4. Контроль знаний обучающихся и типовые оценочные средства.....	17
5. Методические указания для обучающихся по освоению дисциплины (модуля).....	35
6. Учебно-методическое и информационное обеспечение дисциплины.....	37
7. Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы.....	38

1. Цели и задачи дисциплины

1.1 Цель дисциплины – формирование компетенций:

ПК-2 Способен выявлять требования к информационным системам

ПК-6 Способен применять системный подход и математические методы в формализации решения прикладных задач

1.2 Типы задач профессиональной деятельности, к которым готовятся обучающиеся в рамках освоения дисциплины:

- проектный

1.3 Дисциплина ориентирована на подготовку обучающихся к профессиональной деятельности в сфере: 06 Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации информационных систем, управления их жизненным циклом)

1.4 В результате освоения дисциплины у обучающихся должны быть сформированы:

Обобщенные трудовые функции / трудовые функции / трудовые или профессиональные действия (при наличии профстандарта)	Код и наименование компетенции ФГОС ВО, необходимой для формирования трудового или профессионального действия	Индикаторы достижения компетенций
	ПК-2 Способен выявлять требования к информационным системам	Реализует программное обеспечение с использованием современных языков программирования с учётом требований организации
	ПК-6 Способен применять системный подход и математические методы в формализации решения прикладных задач	Применяет системный подход в разработке структуры программного обеспечения для решения прикладных задач

1.5 Согласование междисциплинарных связей дисциплин, обеспечивающих освоение компетенций:

ПК-2 Способен выявлять требования к информационным системам

№ п/п	Наименование дисциплин, определяющих междисциплинарные связи	Форма обучения											
		Очная (семестр)						Заочная (семестр)					
		3	4	5	6	7	8	3	4	5	6	7	9
1	Интернет вещей				+						+		
2	Преддипломная практика						+						+
3	Программирование баз данных		+	+					+	+			
4	Программирование на Java	+	+	+				+	+	+			
5	Программирование на языках высокого уровня		+						+				
6	Технологическая (проектно-технологическая) практика					+						+	

ПК-6 Способен применять системный подход и математические методы в формализации решения прикладных задач

№ п/п	Наименование дисциплин, определяющих междисциплинарные связи	Форма обучения											
		Очная (семестр)					Заочная (семестр)						
		3	4	5	7	8	3	4	5	7	8	9	
1	Научно-исследовательская работа					+							+
2	Преддипломная практика					+							+
3	Программирование на Java	+	+	+			+	+	+				
4	Программирование на языках высокого уровня		+					+					
5	Управление ИТ-проектами				+	+				+	+		

2. Место дисциплины в структуре ОП бакалавриата:

Дисциплина «"High-level language programming"» относится к части, формируемой участниками образовательных отношений, учебного плана ОП по направлению подготовки 09.03.03 - Прикладная информатика.

Дисциплина «"High-level language programming"» изучается в 4 семестре.

3. Объем и содержание дисциплины

3.1. Объем дисциплины:

Вид учебной работы	Очная (всего часов)	Заочная (всего часов)
Общая трудоёмкость дисциплины	72	72
Контактная работа	48	8
Лекции (Лекции)	16	4
Лабораторные (Лаб. раб.)	32	4
Самостоятельная работа (СР)	24	60
Зачет	-	4

3.2. Содержание курса:

№ темы	Название раздела/темы	Вид учебной работы, час.						Формы текущего контроля
		Лекции		Лаб. раб.		СР		
		О	З	О	З	О	З	
4 семестр								
1	Введение	1	1	-	-	1	10	Вопросы для самоподготовки
2	Структура программы на C	1	1	3	-	1	5	Вопросы для самоподготовки
3	Ввод и вывод данных	1	1	Пп 3	-	2	5	Практическое задание для практической подготовки

4	Основные операторы языка С	1	1	Пп 3	-	2	5	Практическое задание для практической подготовки; Тестирование
5	Массивы	1	-	Пп 5	-	3	5	Практическое задание для практической подготовки; Тестирование
6	Указатели и ссылки	1	-	5	-	3	5	Вопросы для самоподготовки
7	Указатели и массивы	1	-	Пп 5	-	3	5	Практическое задание для практической подготовки
8	Символьная информация и строки	1	-	Пп 2	-	2	5	Тестирование; Практическое задание для практической подготовки
9	Функции в С	1	-	Пп 2	-	1	5	Практическое задание для практической подготовки
10	Типы данных, определяемые пользователем	3	-	Пп 2	Пп 2	5	5	Практическое задание для практической подготовки
11	Ввод-вывод в С	4	-	Пп 2	Пп 2	1	5	Практическое задание для практической подготовки

Тема 1. Введение (ПК-2)

Лекция.

Введение. Алгоритм и программа. Свойства алгоритма. Компиляторы и интерпретаторы. Языки программирования.

Задания для самостоятельной работы.

1. Из каких частей состоит программа на С?
2. Чем определение отличается от объявления?
3. Перечислить этапы создания исполняемой программы на языке С.
4. Что такое препроцессор?
5. Что такое директива препроцессора? Привести примеры директив препроцессора.
6. Составить программу, которая печатает текст «Моя первая программа на С».

Тема 2. Структура программы на С (ПК-6)

Лекция.

Структура программы на С. Базовые средства языка С. Состав языка. Константы в С. Типы данных в С. Переменные. Знаки операций в С. Выражения.

Лабораторные работы.

Практическое задание по лабораторной работе 1.

Задания для самостоятельной работы.

1. Из каких элементов состоит естественный язык? Что является аналогами этих элементов в С?
2. Что такое лексема? Привести примеры лексем в языке С.
3. Что такое идентификатор? Правила записи идентификаторов.
4. Что такое константа? Как константа обрабатывается компилятором?
5. Какие типы констант существуют в С. Привести примеры констант разных типов.
6. К какому типу относятся константы 192345, 0x56, 0xCB, 016, 0.7865, .0045, 'c', "x", one, "one", 5, 5.?
7. Что такое тип данных?
8. Чем отличаются типы данных: float и double, char и wchar_t, int и shortint?
9. Чем отличаются типы данных int и unsignedint?
10. Перечислить все типы данных, которые существуют в С. Сколько места в памяти занимают данные каждого типа?
11. На что влияет количество памяти, выделяемое для данных определенного типа?
12. Что такое переменная? Чем объявление переменной отличается от ее определения? Привести примеры определений и объявлений.
13. Что такое класс памяти? Какие классы памяти существуют в С? Привести примеры объявлений и определений переменных разных классов памяти.
14. Что такое выражение? Из чего состоит выражение?
15. Что такое операнд?
16. Какие операции можно применять к целочисленным данным? К вещественным данным? К символьным данным?
17. Что такое отношение?
18. В каком случае отношение считается ложным, а в каком – истинным?
19. Какие операции называются унарными? Привести примеры.

20. Какие операции называются бинарными? Привести примеры.
21. Что такое тернарная операция? Привести пример.
22. Какая разница между постфиксной и префиксной операцией инкремента (декремента)?
23. Какие операции присваивания существуют в C?
24. Привести примеры выражений, содержащих операции присваивания, операции инкремента (декремента), аддитивные и мультипликативные операции. Пояснить, как они будут выполняться.
25. Что такое леводопустимое значение? Привести пример.
26. Чему будет равно значение выражений:

`int z=x/y++`; если `int x=1, y=2`;

`int w=x%++y`, если `int x=1, y=2`;

`int a=++m+n++*sizeof(int)`; если `int m=1, n=2`;

`float a=4*m/0.3*n`; если `float m=1.5; int n=5`;

`int ok=int(0.5*y)<short(x)++`; если `int x=10, y=3`;

Тема 3. Ввод и вывод данных (ПК-2)

Лекция.

Ввод и вывод данных. Функции `printf`, `scanf`. Поточковый ввод и вывод (`cin`, `cout`).

Лабораторные работы.

Практическое задание по лабораторной работе 2.

Задания для самостоятельной работы.

1. Какие библиотеки используются для организации ввода/вывода?
2. С какой библиотекой работают операторы ввода/вывода `printf/scanf`?
3. С какой библиотекой работают операторы ввода/вывода `cin/cout`?
4. Особенности использования двух типов операторов ввода/вывода?

Тема 4. Основные операторы языка C (ПК-6)

Лекция.

Основные операторы языка C. Базовые конструкции структурного программирования. Оператор «выражение». Составные операторы. Операторы выбора. Операторы циклов. Операторы перехода.

Лабораторные работы.

Практическое задание по лабораторной работе 3.

Задания для самостоятельной работы.

1. Чем отличается условная операция от условного оператора?
2. Что такое полная и неполная форма условного оператора?

3. Может ли существовать неполная форма условной операции?
4. Нужно ли писать "else", если при выполнении условия выполняется оператор return?
5. Выражения какого типа могут определять условия в условном операторе или условной операции?
6. Какие значения выражения, определяющего условие, считаются истинными, а какие ложными?
7. Какие операции относятся к операциям отношения?
8. Чем отличается операция "=" от операции "=="?
9. Какие операции относятся к логическим? Каков их приоритет?
10. Какой операцией можно заменить операцию "&&" ?
11. Какой операцией можно заменить операцию "||" ?
12. Чему может быть равно значение выражения отношения или логического выражения?
13. Как правильно сравнить на равенство вещественные числа?
14. Как правильно проверить вхождение значения в некоторый диапазон?
15. Как проверить некоторое целочисленное значение на равенство нулю?
16. Как проверить отличие целочисленного значения от нуля?
17. Когда применяется вложение условных операторов?
18. Как правильно записать вложенные условные операторы?
19. Что такое оператор выбора? Как им пользоваться?
20. Как записать оператор выбора с помощью вложенных условных операторов?
21. Что такое цикл?
22. Какие виды циклов вы знаете?
23. Чем отличается цикл с предусловием от цикла с постусловием?
24. Когда необходимо использовать цикл с предусловием, а когда с постусловием? Приведите примеры.
25. Какие циклы с предусловием существуют в языке C?
26. Сколько операторов содержит в себе тело цикла с предусловием?

27. Как правильно записать цикл с постусловием на языке C?
28. Как задать бесконечный цикл? Зачем он нужен? Как из него выйти?
29. Каким должно быть значение выражения, определяющего условие выполнения цикла, для завершения цикла?
30. Каким должно быть значение выражения, определяющего условие выполнения цикла, для выполнения тела цикла?
31. К чему приведет неправильное задание выражения, определяющего условие выполнения цикла?
32. Может ли тело цикла отсутствовать? Если может, то приведите примеры таких циклов.
33. Чем отличается оператор while от оператора if?
34. Каков порядок действий при выполнении цикла for?
35. Как организовать арифметический цикл с помощью цикла for?
36. Запишите алгоритм, определяемый циклом for, с помощью цикла while.
37. Что такое вложенный цикл?
38. Сколько раз в общей сложности выполняется тело вложенного цикла?
39. Как и когда используются операторы break и continue?
40. Что такое рекуррентные вычисления? Когда они используются? Как их программировать?

Тема 5. Массивы (ПК-2)

Лекция.

Массивы. Определение массива в C. Обработка одномерных массивов. Перебор массива по одному элементу. Формирование псевдинамических массивов. Использование датчика случайных чисел для формирования массива. Перебор массива по два элемента. Классы задач по обработке массивов. Сортировка массивов. Поиск в отсортированном массиве.

Лабораторные работы.

Практическое задание по лабораторной работе 4.

Задания для самостоятельной работы.

1. Что такое массив?
2. Что представляет собой имя массива?
3. Что представляет собой индекс элемента массива?
4. Как можно обратиться к элементу массива?
5. Как получить адрес элемента массива?

6. Как описать указатель на начало массива?
7. Как обратиться к элементу массива через указатель?
8. Существует ли связь между индексом элемента и его значением?
9. Чему равен индекс первого элемента массива?
10. Как проинициализировать массив?
11. Когда можно не указывать количество элементов массива при описании?
12. Какого типа могут быть элементы массива?
13. Какие операции можно производить над целым массивом?
14. Почему при обработке массивов используют циклы?
15. Как поменять местами два элемента массива?
16. Чему равен индекс последнего элемента массива?
17. Возникнет ли ошибка при обращении к элементу массива, индекс которого больше индекса последнего элемента этого массива?
18. Может ли существовать массив из одного элемента? Если может, то как его описать?
19. Что такое сортировка массива?
20. Какие методы сортировки массива Вы знаете?
21. Что такое матрица?
22. Как располагается матрица в памяти компьютера?
23. Как проинициализировать матрицу?
24. Почему при объявлении матрицы с одновременной инициализацией первые скобки можно оставить пустыми, а вторые нет?
25. При обращении к элементу матрицы сначала указывается индекс строки или индекс столбца?
26. Как вывести матрицу таблицей с колонками одинаковой ширины?
27. Что представляет собой имя матрицы?
28. Что представляет собой индекс строки матрицы?

29. Что представляет собой индекс столбца матрицы?
30. Что представляет собой имя строки матрицы?
31. Как получить адрес элемента матрицы?
32. Как получить адрес начала матрицы?
33. Как получить адрес строки матрицы?
34. Чем отличается просмотр элементов матрицы по столбцам от просмотра по строкам?
35. Какова связь между индексами строки и столбца у элементов, лежащих на главной диагонали квадратной матрицы?
36. Какова связь между индексами строки и столбца у элементов, лежащих на побочной диагонали квадратной матрицы?
37. Можно ли просмотреть все элементы матрицы в одном цикле?
38. Как обратиться к некоторому элементу матрицы через указатель на первый элемент матрицы?
39. Как обратиться к элементу матрицы через указатель на начало матрицы?
40. Как описать трехмерный массив и как с ним работать?

Тема 6. Указатели и ссылки (ПК-6)

Лекция.

Указатели и ссылки. Понятие указателя. Динамические переменные. Операции с указателями. Понятие ссылки. Правила работы со ссылками.

Лабораторные работы.

Практическое задание по лабораторной работе 5.

Задания для самостоятельной работы.

1. Что такое указатель?
2. Какой объем памяти занимает указатель?
3. Что является значением переменной-указателя?
4. Как проинициализировать указатель?
5. Что такое NULL?
6. Что такое указатель на void? Зачем нужны такие указатели?
7. Какие операции допустимы при работе с указателями?
8. Чем отличается унарная операция "&" от унарной "*" ?

9. Совместимость типов указателей.
10. Можно ли получить адрес указателя?
11. Можно ли указателю присвоить его же адрес?
12. Почему к указателю на void нельзя применить операцию разыменования?
13. Как работают операции инкремента и декремента, примененные к указателям?
14. Каков результат операции вычитания, примененной к указателям одного типа?
15. Какой спецификатор типа используется при выводе адреса на экран с помощью функции printf()?
16. В чем отличие записи (float *) a от (float) * a, если a – указатель на целое число?
17. В чем отличие записи *a++ от (*a)++, если a – некоторый указатель, отличный от void*?
18. Как описать указатель на начало массива?
19. Как описать указатель на указатель?
20. Когда и зачем может повторно использоваться операция разыменования?

Тема 7. Указатели и массивы (ПК-2)

Лекция.

Указатели и массивы. Одномерные массивы и указатели. Многомерные массивы и указатели. Динамические массивы.

Лабораторные работы.

Практическое задание по лабораторной работе 6.

Задания для самостоятельной работы.

1. Какие переменные называют динамическими? Чем они отличаются от статических переменных?
2. С какой целью используют динамические переменные?
3. Как обращаются к динамическим переменным?
4. Какие бывают указатели? Как их описать?
5. Каким образом можно выделять память для динамических переменных и освобождать её?
6. Как определить, выделена память или нет?
7. Почему нельзя забывать освобождать выделенную память?
8. Чему равно значение указателя после освобождения области памяти, на которую он указывал?
9. Как разместить в динамической памяти массив?

10. Какими способами можно разместить в динамической памяти матрицу?

Тема 8. Символьная информация и строки (ПК-6)

Лекция.

Символьная информация и строки. Библиотечные функции для работы со строками. Примеры работы со строками.

Лабораторные работы.

Практическое задание по лабораторной работе 7.

Задания для самостоятельной работы.

1. Способы организации строк.
2. Что такое Си-строка? Чем она отличается от массива символов?
3. Что такое ноль-терминатор?
4. Чем отличается описание `char *str` от `char str[N]`, где `N` – некоторая константа?
5. Как инициализировать строку?
6. Чем отличается ввод строки с помощью функции `scanf()` от ввода с помощью функции `gets()`?
7. Можно ли использовать операцию присваивания для задания значения строки?
8. Как изменить строку?
9. Что такое конкатенация строк?
10. Какие функции для работы со строками Вы можете назвать?

Тема 9. Функции в С (ПК-2)

Лекция.

Функции в С. Объявление и определение функций. Прототип функции. Параметры функции. Локальные и глобальные переменные. Функции и массивы. Передача одномерных массивов как параметров функции. Передача строк в качестве параметров функций. Передача многомерных массивов в функцию. Функции с начальными (умалчиваемыми) значениями параметров. Подставляемые (`inline`) функции. Функции с переменным числом параметров. Перегрузка функций. Шаблоны функций. Указатель на функцию. Ссылки на функцию.

Лабораторные работы.

Практическое задание по лабораторной работе 8.

Задания для самостоятельной работы.

1. Что такое функция?
2. Что такое функция типа `void`?
3. Что такое прототип функции?
4. Чем отличается прототип функции от вызова функции?
5. Чем отличается описание функции от определения функции?

6. Критерии соответствия формальных и фактических параметров.
7. Соответствие типов формальных и фактических параметров.
8. Зачем нужен оператор return?
9. Может ли в функции быть несколько операторов return?
10. Когда необходимо писать оператор return в функции типа void?
11. Что такое побочный эффект функции?
12. Когда используются формальные параметры-указатели? Какими в этом случае должны быть фактические параметры?
13. Как передать массив в функцию? Как передать в функцию матрицу?
14. Как описать функцию, позволяющую работать и с одномерными массивами, и с матрицами?
15. Что такое параметр-константа? Когда используются такие параметры?
16. Что такое указатель на функцию? Как его описать?
17. Как передать функцию в функцию? Что будет являться формальным параметром, что фактическим? Как обратиться к функции, переданной через параметры?
18. Что такое рекурсия? Когда она применяется? Что такое глубина рекурсии?
19. Как описать функцию с переменным числом параметров? Как использовать такую функцию?
20. Какие параметры могут быть у функции main()? Как отлаживать и тестировать программу, содержащую функцию main() с параметрами?

Тема 10. Типы данных, определяемые пользователем (ПК-6)

Лекция.

Типы данных, определяемые пользователем. Переименование типов. Перечисления. Структуры. Инициализация структур. Присваивание структур. Доступ к элементам структур. Указатели на структуры. Битовые поля. Объединения. Динамические структуры данных. Линейный однонаправленный список. Работа с двунаправленным списком.

Лабораторные работы.

Практическое задание по лабораторной работе 9.

Задания для самостоятельной работы.

1. Где может быть использована структура?
2. Как называются элементы структуры?
3. Как определить структуру?

4. Может ли структура содержать элементы одного типа?
5. Как обратиться к полям структуры?
6. Как обратиться к элементу массива структур?
7. Какое ключевое слово начинает определение структуры?
8. Как обратиться к полям структуры через указатель на нее?
9. При помощи какой операции формируется доступ к полям структуры?
10. Какая разница между определениями структуры и массива?
11. Опишите объект, для которого надо определить структуру и определите ее.

Тема 11. Ввод-вывод в С (ПК-2)

Лекция.

Ввод-вывод в С. Поточный ввод-вывод. Открытие и закрытие потока. Стандартные файлы и функции для работы с ними. Символьный ввод-вывод. Строковый ввод-вывод. Блочный ввод-вывод. Форматированный ввод-вывод. Прямой доступ к файлам. Удаление и добавление элементов в файле.

Лабораторные работы.

Практическое задание по лабораторной работе 10.

Задания для самостоятельной работы.

1. Что такое файл с точки зрения информатики и вычислительной техники?
2. Что такое файл с точки зрения языка программирования С?
3. Какие виды файлов Вы знаете?
4. В чем отличие текстового файла от бинарного и от чего зависит наличие этого различия?
5. Что такое поток?
6. Что такое признак окончания файла? Для каких файлов он существует?
7. Как инициализировать поток?
8. Что такое структурный тип?
9. Какая функция отвечает за открытие потока? Какие у нее параметры?
10. Какие режимы открытия файла Вы знаете? В чем их отличия друг от друга?
11. Какие ошибки могут возникнуть при открытии файла?
12. В каком случае указатель на поток принимает значение NULL?
13. Какая функция отвечает за закрытие файла? Какие у нее параметры?

14. Зачем закрывать файл?
15. Сколько раз можно открыть файл в программе?
16. Какие функции используются для ввода-вывода данных при работе с текстовыми файлами?
17. Какие функции используются для ввода-вывода данных при работе с бинарными файлами?
18. В чем отличие функции `fprintf()` от функции `fread()`, а функции `fscanf()` от `fwrite()`?
19. Как определить текущую позицию указателя на поток?
20. Как переместить указатель на поток из текущей позиции в заданную?
21. За что отвечает константа EOF? Как и где ее можно использовать?
22. Какие основные функции для работы с файлами Вы знаете?
23. С помощью каких функций можно осуществлять побайтовую обработку файлов?
24. Для чего используются функции `rename()` и `remove()`? Нужно ли при работе с ними объявлять файловый указатель?
25. Можно ли удалить или переименовать открытый файл?

4. Контроль знаний обучающихся и типовые оценочные средства

4.1. Распределение баллов:

4 семестр

- посещаемость – 20 баллов
- текущий контроль – 30 баллов
- контрольные срезы – 3 среза по 10 баллов каждый
- премиальные баллы – 20 баллов
- ответ на экзамене: не более 20 баллов

Распределение баллов по заданиям:

№ темы	Название темы / вид учебной работы	Формы текущего контроля / срезы	Мах. кол-во баллов	Методика проведения занятия и оценки

1.	Введение	Вопросы для самоподготовки	2	<p>Методика оценки самоподготовки студентов.</p> <p>2 балла ставится тогда, когда:</p> <ul style="list-style-type: none"> • Студент свободно применяет знания на практике; • Не допускает ошибок в воспроизведении изученного материала; • Студент выделяет главные положения в изученном материале и не затрудняется в ответах на видоизмененные вопросы; • Студент усваивает весь объем программного материала. <p>1 балл ставятся тогда, когда:</p> <ul style="list-style-type: none"> • Студент знает весь изученный материал; • Отвечает без особых затруднений на вопросы преподавателя; • Студент умеет применять полученные знания на практике; • В условных ответах не допускает серьезных ошибок, легко устраняет определенные неточности с помощью дополнительных вопросов преподавателя. <p>0,5 балла ставится тогда, когда:</p> <ul style="list-style-type: none"> • Студент обнаруживает освоение основного материала, но испытывает затруднения при его самостоятельном воспроизведении и требует дополнительных дополняющих вопросов преподавателя; • Предпочитает отвечать на вопросы воспроизводящего характера и испытывает затруднения при ответах на воспроизводящие вопросы. <p>Балл не начисляется тогда, когда:</p> <ul style="list-style-type: none"> • У студента имеются отдельные представления об изучаемом материале, но все же большая часть не усвоена.
2.	Структура программы на С	Вопросы для самоподготовки	2	<p>Методика оценки самоподготовки студентов.</p> <p>2 балла ставится тогда, когда:</p> <ul style="list-style-type: none"> • Студент свободно применяет знания на практике; • Не допускает ошибок в воспроизведении изученного материала; • Студент выделяет главные положения в изученном материале и не затрудняется в ответах на видоизмененные вопросы; • Студент усваивает весь объем программного материала. <p>1 балл ставятся тогда, когда:</p> <ul style="list-style-type: none"> • Студент знает весь изученный материал; • Отвечает без особых затруднений на вопросы преподавателя; • Студент умеет применять полученные знания на практике; • В условных ответах не допускает серьезных ошибок, легко устраняет определенные неточности с помощью дополнительных вопросов преподавателя. <p>0,5 балла ставится тогда, когда:</p> <ul style="list-style-type: none"> • Студент обнаруживает освоение основного материала, но испытывает затруднения при его самостоятельном воспроизведении и требует дополнительных дополняющих вопросов преподавателя; • Предпочитает отвечать на вопросы воспроизводящего характера и испытывает затруднения при ответах на воспроизводящие вопросы. <p>Балл не начисляется тогда, когда:</p> <ul style="list-style-type: none"> • У студента имеются отдельные представления об изучаемом материале, но все же большая часть не усвоена.

3.	Ввод и вывод данных	Практическое задание для практической подготовки	3	<p>Практические задания выполняются по тематике практических занятий. 3 балла – практическое задание выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, свободно отвечает на вопросы используя профессиональную терминологию 1 балл – практическое задание выполнено, но имеет некоторые неточности выполнения, студент владеет представленным материалом, отвечает на заданные вопросы</p>
4.	Основные операторы языка С	Практическое задание для практической подготовки	3	<p>Практические задания выполняются по тематике практических занятий. 3 балла – практическое задание выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, свободно отвечает на вопросы используя профессиональную терминологию 1 балл – практическое задание выполнено, но имеет некоторые неточности выполнения, студент владеет представленным материалом, отвечает на заданные вопросы</p>
		Тестирование(контрольный срез)	10	<p>Тестирование подразумевает 10 вопросов. За прохождение тестирования выставляются следующие баллы: - 90 % - 10 баллов; - 65 % - 5 баллов; - 50 % - 2 балла; - менее 50 % - балл не начисляется.</p>
5.	Массивы	Практическое задание для практической подготовки	3	<p>Практические задания выполняются по тематике практических занятий. 3 балла – практическое задание выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, свободно отвечает на вопросы используя профессиональную терминологию 1 балл – практическое задание выполнено, но имеет некоторые неточности выполнения, студент владеет представленным материалом, отвечает на заданные вопросы</p>

		Тестирование(контрольный срез)	10	Тестирование подразумевает 10 вопросов. За прохождение тестирования выставляются следующие баллы: - 90 % - 10 баллов; - 65 % - 5 баллов; - 50 % - 2 балла; - менее 50 % - балл не начисляется.
6.	Указатели и ссылки	Вопросы для самоподготовки	2	Методика оценки самоподготовки студентов. 2 балла ставится тогда, когда: • Студент свободно применяет знания на практике; • Не допускает ошибок в воспроизведении изученного материала; • Студент выделяет главные положения в изученном материале и не затрудняется в ответах на видоизмененные вопросы; • Студент усваивает весь объем программного материала. 1 балл ставятся тогда, когда: • Студент знает весь изученный материал; • Отвечает без особых затруднений на вопросы преподавателя; • Студент умеет применять полученные знания на практике; • В условных ответах не допускает серьезных ошибок, легко устраняет определенные неточности с помощью дополнительных вопросов преподавателя. 0,5 балла ставится тогда, когда: • Студент обнаруживает освоение основного материала, но испытывает затруднения при его самостоятельном воспроизведении и требует дополнительных дополняющих вопросов преподавателя; • Предпочитает отвечать на вопросы воспроизводящего характера и испытывает затруднения при ответах на воспроизводящие вопросы. Балл не начисляется тогда, когда: • У студента имеются отдельные представления об изучаемом материале, но все, же большая часть не усвоена.
7.	Указатели и массивы	Практическое задание для практической подготовки	3	Практические задания выполняются по тематике практических занятий. 3 балла – практическое задание выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, свободно отвечает на вопросы используя профессиональную терминологию 1 балл – практическое задание выполнено, но имеет некоторые неточности выполнения, студент владеет представленным материалом, отвечает на заданные вопросы
8.	Символьная информация и строки	Тестирование(контрольный срез)	10	Тестирование подразумевает 10 вопросов. За прохождение тестирования выставляются следующие баллы: - 90 % - 10 баллов; - 65 % - 5 баллов; - 50 % - 2 балла; - менее 50 % - балл не начисляется.

		Практическое задание для практической подготовки	3	Практические задания выполняются по тематике практических занятий. 3 балла – практическое задание выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, свободно отвечает на вопросы используя профессиональную терминологию 1 балл – практическое задание выполнено, но имеет некоторые неточности выполнения, студент владеет представленным материалом, отвечает на заданные вопросы
9.	Функции в С	Практическое задание для практической подготовки	3	Практические задания выполняются по тематике практических занятий. 3 балла – практическое задание выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, свободно отвечает на вопросы используя профессиональную терминологию 1 балл – практическое задание выполнено, но имеет некоторые неточности выполнения, студент владеет представленным материалом, отвечает на заданные вопросы
10.	Типы данных, определяемые пользователем	Практическое задание для практической подготовки	3	Практические задания выполняются по тематике практических занятий. 3 балла – практическое задание выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, свободно отвечает на вопросы используя профессиональную терминологию 1 балл – практическое задание выполнено, но имеет некоторые неточности выполнения, студент владеет представленным материалом, отвечает на заданные вопросы

11.	Ввод-вывод в С	Практическое задание для практической подготовки	3	Практические задания выполняются по тематике практических занятий. 3 балла – практическое задание выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, свободно отвечает на вопросы используя профессиональную терминологию 1 балл – практическое задание выполнено, но имеет некоторые неточности выполнения, студент владеет представленным материалом, отвечает на заданные вопросы
12.	Посещаемость		20	20 баллов – студент посетил все 100% занятий 14-19 баллов – студент посетил не менее 80% занятий 8-13 баллов – студент посетил не менее 50% занятий 2-7 балла – студент посетил не менее 25% занятий Если студент посетил менее 25% занятий, баллы не начисляются
13.	Премиальные баллы		20	Дополнительные премиальные баллы могут быть начислены: - за проект, выполненный по заказу работодателя и реализованный на практике – 20 баллов; - постоянная активность во время практических занятий – 10 баллов; - полностью подготовленная к публикации статья по тематике в рамках дисциплины – 10 баллов; - участие с докладом во всероссийской олимпиаде по тематике изучаемой дисциплины – 20 баллов; - участие в выставке по тематике изучаемой дисциплины – 20 баллов; - публикация статьи по тематике изучаемой дисциплины в сборнике студенческих работ / материалах всероссийской конференции / журнале из перечня ВАК – 10 / 15 / 20
14.	Ответ на экзамене		20	8-12 баллов – студент раскрыл основные вопросы и задания билета на оценку «удовлетворительно» 12-16 баллов – студент раскрыл основные вопросы и задания билета на оценку «хорошо», 17-20 баллов – студент раскрыл основные вопросы и задания билета на оценку «отлично».
15.	Индивидуальные задания, с помощью которых можно набрать дополнительные баллы		20	Решение кейса (10 баллов) Прохождение тестирования (30 вопросов) по всему курсу дисциплины (10 баллов)
16.	Итого за семестр		100	

Итоговая оценка по зачету выставляется в 100-балльной шкале и в традиционной четырехбалльной шкале. Перевод 100-балльной рейтинговой оценки по дисциплине в традиционную четырехбалльную осуществляется следующим образом:

100-балльная система	Традиционная система
50 - 100 баллов	Зачтено
0 - 49 баллов	Не зачтено

4.2 Типовые оценочные средства текущего контроля

Вопросы для самоподготовки

Тема 1. Введение

1. Из каких частей состоит программа на С?
2. Чем определение отличается от объявления?
3. Перечислить этапы создания исполняемой программы на языке С.
4. Что такое препроцессор?
5. Что такое директива препроцессора? Привести примеры директив препроцессора.
6. Составить программу, которая печатает текст «Моя первая программа на С».
7. Из каких элементов состоит естественный язык? Что является аналогами этих элементов в С?
8. Что такое лексема? Привести примеры лексем в языке С.
9. Что такое идентификатор? Правила записи идентификаторов.
10. Что такое константа? Как константа обрабатывается компилятором?
11. Какие типы констант существуют в С. Привести примеры констант разных типов.
12. К какому типу относятся константы 192345, 0x56, 0xCB, 016, 0.7865, .0045, 'c', "x", one, "one", 5, 5.?
13. Что такое тип данных?
14. Чем отличаются типы данных: float и double, char и wchar_t, int и short int?
15. Чем отличаются типы данных int и unsigned int?
16. Перечислить все типы данных, которые существуют в С. Сколько места в памяти занимают данные каждого типа?
17. На что влияет количество памяти, выделяемое для данных определенного типа?
18. Что такое переменная? Чем объявление переменной отличается от ее определения? Привести примеры определений и объявлений.
19. Что такое класс памяти? Какие классы памяти существуют в С? Привести примеры объявлений и определений переменных разных классов памяти.
20. Что такое выражение? Из чего состоит выражение?
21. Что такое операнд?
22. Какие операции можно применять к целочисленным данным? К вещественным данным? К символьным данным?
23. Что такое отношение?
24. В каком случае отношение считается ложным, а в каком – истинным?
25. Какие операции называются унарными? Привести примеры.
26. Какие операции называются бинарными? Привести примеры.
27. Что такое тернарная операция? Привести пример.
28. Какая разница между постфиксной и префиксной операцией инкремента (декремента)?
29. Какие операции присваивания существуют в С?
30. Привести примеры выражений, содержащих операции присваивания, операции инкремента (декремента), аддитивные и мультипликативные операции. Пояснить, как они будут выполняться.
31. Что такое леводопустимое значение? Привести пример.
32. Чему будет равно значение выражений:
`int z=x/y++`; если `int x=1, y=2`;
`int w=x%++y`, если `int x=1, y=2`;
`int a=++m+n++*sizeof(int)`; если `int m=1, n=2`;
`float a=4*m/0.3*n`; если `float m=1.5; int n=5`;
`int ok=int(0.5*y)<short(x)++`; если `int x=10, y=3`;
33. Какие библиотеки используются для организации ввода/вывода?
34. С какой библиотекой работают операторы ввода/вывода `printf/scanf`?
35. С какой библиотекой работают операторы ввода/вывода `cin/cout`?

36. Особенности использования двух типов операторов ввода/вывода?

Тема 2. Структура программы на С

- 1 Чем отличается условная операция от условного оператора?
 - 2 Что такое полная и неполная форма условного оператора?
 - 3 Может ли существовать неполная форма условной операции?
 - 4 Нужно ли писать "else", если при выполнении условия выполняется оператор return?
 - 5 Выражения какого типа могут определять условия в условном операторе или условной операции?
 - 6 Какие значения выражения, определяющего условие, считаются истинными, а какие ложными?
 - 7 Какие операции относятся к операциям отношения?
 - 8 Чем отличается операция "=" от операции "=="?
 - 9 Какие операции относятся к логическим? Каков их приоритет?
 - 10 Какой операцией можно заменить операцию "&&" ?
 - 11 Какой операцией можно заменить операцию "||" ?
 - 12 Чему может быть равно значение выражения отношения или логического выражения?
 - 13 Как правильно сравнить на равенство вещественные числа?
 - 14 Как правильно проверить вхождение значения в некоторый диапазон?
 - 15 Как проверить некоторое целочисленное значение на равенство нулю?
 - 16 Как проверить отличие целочисленного значения от нуля?
 - 17 Когда применяется вложение условных операторов?
 - 18 Как правильно записать вложенные условные операторы?
 - 19 Что такое оператор выбора? Как им пользоваться?
 - 20 Как записать оператор выбора с помощью вложенных условных операторов?
 - 21 Что такое цикл?
 - 22 Какие виды циклов вы знаете?
 - 23 Чем отличается цикл с предусловием от цикла с постусловием?
 - 24 Когда необходимо использовать цикл с предусловием, а когда с постусловием? Приведите примеры.
 - 25 Какие циклы с предусловием существуют в языке С?
 - 26 Сколько операторов содержит в себе тело цикла с предусловием?
 - 27 Как правильно записать цикл с постусловием на языке С?
 - 28 Как задать бесконечный цикл? Зачем он нужен? Как из него выйти?
 - 29 Каким должно быть значение выражения, определяющего условие выполнения цикла, для завершения цикла?
 - 30 Каким должно быть значение выражения, определяющего условие выполнения цикла, для выполнения тела цикла?
 - 31 К чему приведет неправильное задание выражения, определяющего условие выполнения цикла?
 - 32 Может ли тело цикла отсутствовать? Если может, то приведите примеры таких циклов.
 - 33 Чем отличается оператор while от оператора if?
 - 34 Каков порядок действий при выполнении цикла for?
 - 35 Как организовать арифметический цикл с помощью цикла for?
 - 36 Запишите алгоритм, определяемый циклом for, с помощью цикла while.
 - 37 Что такое вложенный цикл?
 - 38 Сколько раз в общей сложности выполняется тело вложенного цикла?
 - 39 Как и когда используются операторы break и continue?
- Что такое рекуррентные вычисления? Когда они используются? Как их программировать?

Тема 6. Указатели и ссылки

- 1 Способы организации строк.
- 2 Что такое Си-строка? Чем она отличается от массива символов?
- 3 Что такое ноль-терминатор?
- 4 Чем отличается описание `char *st` от `char st[N]`, где `N` – некоторая константа?
- 5 Как инициализировать строку?
- 6 Чем отличается ввод строки с помощью функции `scanf()` от ввода с помощью функции `gets()`?
- 7 Можно ли использовать операцию присваивания для задания значения строки?
- 8 Как изменить строку?
- 9 Что такое конкатенация строк?

Какие функции для работы со строками Вы можете назвать?

Практическое задание для практической подготовки

Тема 3. Ввод и вывод данных

- 1 Что такое массив?
- 2 Что представляет собой имя массива?
- 3 Что представляет собой индекс элемента массива?
- 4 Как можно обратиться к элементу массива?
- 5 Как получить адрес элемента массива?
- 6 Как описать указатель на начало массива?
- 7 Как обратиться к элементу массива через указатель?
- 8 Существует ли связь между индексом элемента и его значением?
- 9 Чему равен индекс первого элемента массива?
- 10 Как проинициализировать массив?
- 11 Когда можно не указывать количество элементов массива при описании?
- 12 Какого типа могут быть элементы массива?
- 13 Какие операции можно производить над целым массивом?
- 14 Почему при обработке массивов используют циклы?
- 15 Как поменять местами два элемента массива?
- 16 Чему равен индекс последнего элемента массива?
- 17 Возникнет ли ошибка при обращении к элементу массива, индекс которого больше индекса последнего элемента этого массива?
- 18 Может ли существовать массив из одного элемента? Если может, то как его описать?
- 19 Что такое сортировка массива?
- 20 Какие методы сортировки массива Вы знаете?
- 21 Что такое матрица?
- 22 Как располагается матрица в памяти компьютера?
- 23 Как проинициализировать матрицу?
- 24 Почему при объявлении матрицы с одновременной инициализацией первые скобки можно оставить пустыми, а вторые нет?
- 25 При обращении к элементу матрицы сначала указывается индекс строки или индекс столбца?
- 26 Как вывести матрицу таблицей с колонками одинаковой ширины?
- 27 Что представляет собой имя матрицы?
- 28 Что представляет собой индекс строки матрицы?
- 29 Что представляет собой индекс столбца матрицы?
- 30 Что представляет собой имя строки матрицы?
- 31 Как получить адрес элемента матрицы?
- 32 Как получить адрес начала матрицы?

- 33 Как получить адрес строки матрицы?
 - 34 Чем отличается просмотр элементов матрицы по столбцам от просмотра по строкам?
 - 35 Какова связь между индексами строки и столбца у элементов, лежащих на главной диагонали квадратной матрицы?
 - 36 Какова связь между индексами строки и столбца у элементов, лежащих на побочной диагонали квадратной матрицы?
 - 37 Можно ли просмотреть все элементы матрицы в одном цикле?
 - 38 Как обратиться к некоторому элементу матрицы через указатель на первый элемент матрицы?
 - 39 Как обратиться к элементу матрицы через указатель на начало матрицы?
- Как описать трехмерный массив и как с ним работать?

Тема 4. Основные операторы языка C

- 1 Что такое указатель?
 - 2 Какой объем памяти занимает указатель?
 - 3 Что является значением переменной-указателя?
 - 4 Как проинициализировать указатель?
 - 5 Что такое NULL?
 - 6 Что такое указатель на void? Зачем нужны такие указатели?
 - 7 Какие операции допустимы при работе с указателями?
 - 8 Чем отличается унарная операция "&" от унарной "*" ?
 - 9 Совместимость типов указателей.
 - 10 Можно ли получить адрес указателя?
 - 11 Можно ли указателю присвоить его же адрес?
 - 12 Почему к указателю на void нельзя применить операцию разыменования?
 - 13 Как работают операции инкремента и декремента, примененные к указателям?
 - 14 Каков результат операции вычитания, примененной к указателям одного типа?
 - 15 Какой спецификатор типа используется при выводе адреса на экран с помощью функции printf()?
 - 16 В чем отличие записи (float *) a от (float) * a, если a – указатель на целое число?
 - 17 В чем отличие записи *a++ от (*a)++, если a – некоторый указатель, отличный от void*?
 - 18 Как описать указатель на начало массива?
 - 19 Как описать указатель на указатель?
- Когда и зачем может повторно использоваться операция разыменования?

Тема 5. Массивы

- 1 Какие переменные называют динамическими? Чем они отличаются от статических переменных?
 - 2 С какой целью используют динамические переменные?
 - 3 Как обращаются к динамическим переменным?
 - 4 Какие бывают указатели? Как их описать?
 - 5 Каким образом можно выделять память для динамических переменных и освобождать её?
 - 6 Как определить, выделена память или нет?
 - 7 Почему нельзя забывать освобождать выделенную память?
 - 8 Чему равно значение указателя после освобождения области памяти, на которую он указывал?
 - 9 Как разместить в динамической памяти массив?
- Какими способами можно разместить в динамической памяти матрицу?

Тема 7. Указатели и массивы

- 1 Что такое функция?
- 2 Что такое функция типа void?
- 3 Что такое прототип функции?
- 4 Чем отличается прототип функции от вызова функции?
- 5 Чем отличается описание функции от определения функции?
- 6 Критерии соответствия формальных и фактических параметров.
- 7 Соответствие типов формальных и фактических параметров.
- 8 Зачем нужен оператор return?
- 9 Может ли в функции быть несколько операторов return?
- 10 Когда необходимо писать оператор return в функции типа void?
- 11 Что такое побочный эффект функции?
- 12 Когда используются формальные параметры-указатели? Какими в этом случае должны быть фактические параметры?
- 13 Как передать массив в функцию? Как передать в функцию матрицу?
- 14 Как описать функцию, позволяющую работать и с одномерными массивами, и с матрицами?
- 15 Что такое параметр-константа? Когда используются такие параметры?
- 16 Что такое указатель на функцию? Как его описать?
- 17 Как передать функцию в функцию? Что будет являться формальным параметром, что фактическим? Как обратиться к функции, переданной через параметры?
- 18 Что такое рекурсия? Когда она применяется? Что такое глубина рекурсии?
- 19 Как описать функцию с переменным числом параметров? Как использовать такую функцию? Какие параметры могут быть у функции main()? Как отлаживать и тестировать программу, содержащую функцию main() с параметрами?

Тема 8. Символьная информация и строки

1. Где может быть использована структура?
2. Как называются элементы структуры?
3. Как определить структуру?
4. Может ли структура содержать элементы одного типа?
5. Как обратиться к полям структуры?
6. Как обратиться к элементу массива структур?
7. Какое ключевое слово начинает определение структуры?
8. Как обратиться к полям структуры через указатель на нее?
9. При помощи какой операции формируется доступ к полям структуры?
10. Какая разница между определениями структуры и массива?
11. Опишите объект, для которого надо определить структуру и определите ее.

Тема 9. Функции в С

1. Что такое файл с точки зрения информатики и вычислительной техники?
2. Что такое файл с точки зрения языка программирования С?
3. Какие виды файлов Вы знаете?
4. В чем отличие текстового файла от бинарного и от чего зависит наличие этого различия?
5. Что такое поток?
6. Что такое признак окончания файла? Для каких файлов он существует?
7. Как инициализировать поток?
8. Что такое структурный тип?
9. Какая функция отвечает за открытие потока? Какие у нее параметры?
10. Какие режимы открытия файла Вы знаете? В чем их отличия друг от друга?
11. Какие ошибки могут возникнуть при открытии файла?

12. В каком случае указатель на поток принимает значение NULL?
13. Какая функция отвечает за закрытие файла? Какие у нее параметры?
14. Зачем закрывать файл?
15. Сколько раз можно открыть файл в программе?
16. Какие функции используются для ввода-вывода данных при работе с текстовыми файлами?
17. Какие функции используются для ввода-вывода данных при работе с бинарными файлами?
18. В чем отличие функции fprintf() от функции fread(), а функции fscanf() от fwrite()?
19. Как определить текущую позицию указателя на поток?
20. Как переместить указатель на поток из текущей позиции в заданную?
21. За что отвечает константа EOF? Как и где ее можно использовать?
22. Какие основные функции для работы с файлами Вы знаете?
23. С помощью каких функций можно осуществлять побайтовую обработку файлов?
24. Для чего используются функции rename() и remove()? Нужно ли при работе с ними объявлять файловый указатель?
25. Можно ли удалить или переименовать открытый файл?

Тема 10. Типы данных, определяемые пользователем

1. Где может быть использована структура?
2. Как называются элементы структуры?
3. Как определить структуру?
4. Может ли структура содержать элементы одного типа?
5. Как обратиться к полям структуры?
6. Как обратиться к элементу массива структур?
7. Какое ключевое слово начинает определение структуры?
8. Как обратиться к полям структуры через указатель на нее?
9. При помощи какой операции формируется доступ к полям структуры?
10. Какая разница между определениями структуры и массива?
11. Опишите объект, для которого надо определить структуру и определите ее.

Тема 11. Ввод-вывод в С

1. Что такое файл с точки зрения информатики и вычислительной техники?
2. Что такое файл с точки зрения языка программирования С?
3. Какие виды файлов Вы знаете?
4. В чем отличие текстового файла от бинарного и от чего зависит наличие этого различия?
5. Что такое поток?
6. Что такое признак окончания файла? Для каких файлов он существует?
7. Как инициализировать поток?
8. Что такое структурный тип?
9. Какая функция отвечает за открытие потока? Какие у нее параметры?
10. Какие режимы открытия файла Вы знаете? В чем их отличия друг от друга?
11. Какие ошибки могут возникнуть при открытии файла?
12. В каком случае указатель на поток принимает значение NULL?
13. Какая функция отвечает за закрытие файла? Какие у нее параметры?
14. Зачем закрывать файл?
15. Сколько раз можно открыть файл в программе?
16. Какие функции используются для ввода-вывода данных при работе с текстовыми файлами?
17. Какие функции используются для ввода-вывода данных при работе с бинарными файлами?
18. В чем отличие функции fprintf() от функции fread(), а функции fscanf() от fwrite()?

19. Как определить текущую позицию указателя на поток?
20. Как переместить указатель на поток из текущей позиции в заданную?
21. За что отвечает константа EOF? Как и где ее можно использовать?
22. Какие основные функции для работы с файлами Вы знаете?
23. С помощью каких функций можно осуществлять побайтовую обработку файлов?
24. Для чего используются функции rename() и remove()? Нужно ли при работе с ними объявлять файловый указатель?
25. Можно ли удалить или переименовать открытый файл?

Тестирование

Тема 4. Основные операторы языка C

1. Укажите оператор выбора в языке C ++.
 - A. CASE
 - B. choice
 - C. switch ... case +
 - D. default
2. Укажите, в котором выражении правильно применяется операция целочисленных типов (язык C ++).
 - A. int a = 2, b = 0x24, c; c = a + b% a; +
 - B. short x = 0, y = 2; for (y>! = x) {...};
 - C. int c <> 2;
 - D. const char mychar = 2; long s = 3; mychar / = s;
3. В сложных выражениях последовательность выполнения операций определяется ...
 - A. только приоритетом операций.
 - B. только скобками.
 - C. скобками, приоритетом операций, а при одинаковом приоритете ассоциативностью операций. +
 - D. только ассоциативностью операций.
4. В чем разница между фактическими и формальными параметрами функций?
 - A. Формальные параметры могут использоваться только вне тела функции, а фактические — используются как вне функции, так и внутри нее.
 - B. Нет разницы, это одни и те же параметры.
 - C. И формальные, и фактические параметры используются вне тела функции.
 - D. Формальные параметры определены в теле функции, а фактические — значение, с которыми функция вызывается. +
5. В каком файле заголовков определен объект cout?
 - A. stream.h
 - B. sysutils.hpp
 - C. cout.h
 - D. iostream.h +
6. Для чего предназначен оператор continue в языке C ++?
 - A. Пропускает остаток тела цикла и переходит к следующей итерации. +
 - B. Пропускает цикл и переходит к следующему оператору в теле программы.
 - C. Определяет условие продолжения цикла.
 - D. Продолжает выполнение текущей итерации цикла.

Тема 5. Массивы

1. Значение переменной number не лежит между 3 и 6. Укажите правильный вариант записи данного утверждения на языке C ++.

- A. `number > 3 && number < 6`
 B. `!(Number < 3 && number < 6)`
 C. `!(Number < 6 || number > 3)`
 D. `number < 3 || number > 6`
2. Укажите верное утверждение.
- A. Строки в C + представляются как массивы элементов типа `char`, заканчивающиеся символом `'0'`.
 B. Строки в C + представляются как массивы элементов типа `char`, заканчивающиеся символом `'\0'`.
 C. Строки в C + представляются как массивы элементов типа `char`, заканчивающиеся символом `»`.
 D. Строки в C + представляются как массивы элементов типа `char`, заканчивающиеся символом `'.'`
3. Что выполняет операция `++` в языке C ++?
- A. Уменьшает значение операнда на единицу.
 B. Уменьшает значение операнда на два.
 C. Увеличивает значение операнда на два.
 D. Увеличивает значение операнда на единицу.
4. Выберите правильный вариант объявления константной переменной в C +, где `type` — тип данных в C +; `variable` — имя переменной; `value` — константные значения.
- A. `const type variable = value;`
 B. `const type variable = value; +`
 C. `const variable = value;`
 D. `type const variable = value;`
5. В каком случае выражение C ++ будет исчисляться быстрее?
- A. `X = X + Y`
 B. `X += Y +`
 C. `X = Y + X`
 D. Все три выражения будут выполнены с одинаковой скоростью.
6. Укажите правильный вариант объявления указателя в C ++.
- A. `int x;`
 B. `int & x;`
 C. `int * x; +`
 D. `ptr x;`

Тема 8. Символьная информация и строки

1. Укажите, в каком выражении языке C ++ неправильно применяются операции с действительными типами?
- A. `float a = 2, b = 0x24, c; c = a + b% a; +`
 B. `float x = 0, y = 2; if (b > x) {...};`
 C. `long double c = 2;`
 D. `double mychar = 2; float s = 3; mychar /= s;`
2. В программе на языке C ++ есть два объявления переменных `int qwerty; int QWERTY`. Какое из утверждений верно?
- A. Такие имена переменных недопустимы.
 B. Объявления правильные.
 C. Такие объявления недопустимы, потому что мы пытаемся создать две переменные с одинаковыми идентификаторами.
 D. Переменные описываются не по такому принципу.
3. Как средствами языка C ++ выполняется потоковое чтение из двоичного (бинарного) файла? Название потока `a_file`.
- A. `a_file >> z;`

- B. `a_file.in (z);`
 C. `z = a_file.read;`
 D. `a_file.read ((char *) & z, sizeof z); +`
4. В каком из вариантов объявлен двумерный массив в C ++?
 A. `int anarray [20] [20]; +`
 B. `int array [20, 20];`
 C. `array anarray [20] [20];`
 D. `char array [20];`
5. Укажите запись экранированного символа языке C ++.
 A. 'F'
 B. «\ 022»
 C. 'Ю'
 D. '\ t' +
6. Зачем в C ++ используют оператор `return`?
 A. Чтобы задержать работу программы.
 B. Функция, в которой он содержится, завершает свое выполнение и управление возвращается в то место программы, из которого вызывалась данная функция. +
 C. Чтобы организовать цикл.
 D. Чтобы ввести в программу новые значения.
7. В какой из следующих строк, записанных на языке C ++, выполняется обращение к седьмому элементу массива (размер массива равен 10)?
 A. `mas (7);`
 B. `mas [6]; +`
 C. `mas 8;`
 D. `mas [7];`
8. В программе на языке C ++ объявлены переменные `int x, y`. Какое выражение позволяет вычислить остаток от деления этих переменных?
 A. `x% y +`
 B. `x div y`
 C. `x mod y`
 D. `x / y`

4.3 Промежуточная аттестация по дисциплине проводится в форме зачета

Типовые вопросы зачета (ПК-2, ПК-6)

1. Алгоритм и его свойства. Способы записи алгоритма. Программа. Языки программирования. Примеры алгоритмов и программ.
2. Структура программы на языке C. Примеры. Этапы создания исполняемой программы.
3. Состав языка C. Константы и переменные C.
4. Типы данных в C.
5. Выражения. Знаки операций.
6. Основные операторы C (присваивание, составные, выбора, циклов, перехода). Синтаксис, семантика, примеры.

7. Этапы решения задачи. Виды ошибок. Тестирование.
8. Массивы (определение, инициализация, способы перебора).
9. Сортировка массивов (простой обмен, простое включение, простой выбор).
10. Поиск в одномерных массивах (дихотомический и линейный).
11. Указатели. Операции с указателями. Примеры
12. Динамические переменные. Операции new и delete. Примеры.
13. Ссылки. Примеры.
14. Одномерные массивы и указатели. Примеры.
15. Многомерные массивы и указатели. Примеры.
16. Динамические массивы. Примеры.
17. Символьная информация и строки. Функции для работы со строками (библиотечный файл string.h).
18. Функции ввода-вывода (scanf(), printf(), puts(), gets(), putchar(), getchar()).
19. Функции в С. Формальные и фактические параметры. Передача параметров по адресу и по значению. Локальные и глобальные переменные. Примеры.
20. Прототип функции. Библиотечные файлы. Директива препроцессора #include.
21. Передача одномерных массивов в функции. Примеры.
22. Передача многомерных массивов в функции. Примеры.
23. Передача строк в функции. Примеры.
24. Функции с умалчиваемыми параметрами. Примеры.
25. Подставляемые функции. Примеры.
26. Функции с переменным числом параметров. Примеры.
27. Перегрузка функции. Шаблоны функций. Примеры.
28. Указатели на функции. Примеры.
29. Ссылки на функции. Примеры.

30. Типы данных, определяемые пользователем (переименование типов, перечисление, структуры, объединения). Примеры.
31. Структуры. Определение, инициализация, присваивание структур, доступ к элементам структур, указатели на структуры, битовые поля структур.
32. Динамические структуры данных (однонаправленные и двунаправленные списки).
33. Создание списка, печать, удаление, добавление элементов (на примере однонаправленных и двунаправленных списков).
34. Поточковый ввод-вывод в С. Открытие и закрытие потока. Стандартные потоки ввода-вывода.
35. Символьный, строковый, блоковый и форматированный ввод-вывод.
36. Прямой доступ к файлам.
37. Создание бинарных и текстовых файлов, удаление, добавление, корректировка элементов, печать файлов.

Типовые задания для зачета (ПК-2, ПК-6)

1. Определить, попадет ли точка с координатами (x, y) в указанную область.

2. 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.

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

19. Добавить в однонаправленный (двунаправленный) список элемент с заданным номером.

20. Удалить из бинарного файла, в котором записаны целые числа все четные элементы.

21. Добавить в бинарный файл, в который записаны элементы типа

```
struct Student
```

```
{char name[20];int age;};
```

К элементов после элемента с заданной фамилией.

22. Удалить из текстового файла все четные строки.

23. Добавить порядковый номер в каждую строку текстового файла.

24. 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”.

25. В текстовом файле заменить все строки, начинающиеся с буквы ‘f’ на строки, начинающиеся с буквы ‘a’.

4.4. Шкала оценивания промежуточной аттестации

Оценка	Компетенции	Дескрипторы (уровни) – основные признаки освоения (показатели достижения результата)
«зачтено» (50 - 100 баллов)	ПК-2	Способен реализовать программное обеспечение с использованием современных языков программирования с учётом требований организации
	ПК-6	Способен применять системный подход в разработке структуры программного обеспечения для решения прикладных задач
«не зачтено» (0 - 49 баллов)	ПК-2	Не способен реализовать программное обеспечение с использованием современных языков программирования с учётом требований организации
	ПК-6	Не способен применять системный подход в разработке структуры программного обеспечения для решения прикладных задач

5. Методические указания для обучающихся по освоению дисциплины (модуля)

5.1 Методические указания по организации самостоятельной работы обучающихся:

Приступая к изучению дисциплины, в первую очередь обучающимся необходимо ознакомиться содержанием рабочей программы дисциплины (РПД), которая определяет содержание, объем, а также порядок изучения и преподавания учебной дисциплины, ее раздела, части.

Для самостоятельной работы важное значение имеют разделы «Объем и содержание дисциплины», «Учебно-методическое и информационное обеспечение дисциплины» и «Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы».

В разделе «Объем и содержание дисциплины» указываются все разделы и темы изучаемой дисциплины, а также виды занятий и планируемый объем в академических часах.

В разделе «Учебно-методическое и информационное обеспечение дисциплины» указана рекомендуемая основная и дополнительная литература.

В разделе «Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы» содержится перечень профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины.

5.2 Рекомендации обучающимся по работе с теоретическими материалами по дисциплине

При изучении и проработке теоретического материала необходимо:

- просмотреть еще раз презентацию лекции в системе MOODLe, повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной дополнительной литературы;
- при самостоятельном изучении теоретической темы сделать конспект, используя рекомендованные в РПД источники, профессиональные базы данных и информационные справочные системы;
- ответить на вопросы для самостоятельной работы, по теме представленные в пункте 3.2 РПД.
- при подготовке к текущему контролю использовать материалы фонда оценочных средств (ФОС).

5.3 Рекомендации по работе с научной и учебной литературой

Работа с основной и дополнительной литературой является главной формой самостоятельной работы и необходима при подготовке к устному опросу на семинарских занятиях, к дебатам, тестированию, экзамену. Она включает проработку лекционного материала и рекомендованных источников и литературы по тематике лекций.

Конспект лекции должен содержать реферативную запись основных вопросов лекции, в том числе с опорой на размещенные в системе MOODLe презентации, основных источников и литературы по темам, выводы по каждому вопросу. Конспект может быть выполнен в рамках распечатки выдачи презентаций лекций или в отдельной тетради по предмету. Он должен быть аккуратным, хорошо читаемым, не содержать не относящуюся к теме информацию или рисунки.

Конспекты научной литературы при самостоятельной подготовке к занятиям должны содержать ответы на каждый поставленный в теме вопрос, иметь ссылку на источник информации с обязательным указанием автора, названия и года издания используемой научной литературы. Конспект может быть опорным (содержать лишь основные ключевые позиции), но при этом позволяющим дать полный ответ по вопросу, может быть подробным. Объем конспекта определяется самим студентом.

В процессе работы с основной и дополнительной литературой студент может:

- делать записи по ходу чтения в виде простого или развернутого плана (создавать перечень основных вопросов, рассмотренных в источнике);
- составлять тезисы (цитирование наиболее важных мест статьи или монографии, короткое изложение основных мыслей автора);
- готовить аннотации (краткое обобщение основных вопросов работы);
- создавать конспекты (развернутые тезисы).

5.4. Рекомендации по подготовке к отдельным заданиям текущего контроля

Собеседование предполагает организацию беседы преподавателя со студентами по вопросам практического занятия с целью более обстоятельного выявления их знаний по определенному разделу, теме, проблеме и т.п. Все члены группы могут участвовать в обсуждении, добавлять информацию, дискутировать, задавать вопросы и т.д.

Устный опрос может применяться в различных формах: фронтальный, индивидуальный, комбинированный. Основные качества устного ответа подлежащего оценке:

- правильность ответа по содержанию;
- полнота и глубина ответа;
- сознательность ответа;
- логика изложения материала;
- рациональность использованных приемов и способов решения поставленной учебной задачи;
- своевременность и эффективность использования наглядных пособий и технических средств при ответе;

- использование дополнительного материала;
- рациональность использования времени, отведенного на задание.

Устный опрос может сопровождаться презентацией, которая подготавливается по одному из вопросов практического занятия. При выступлении с презентацией необходимо обращать внимание на такие моменты как:

- содержание презентации: актуальность темы, полнота ее раскрытия, смысловое содержание, соответствие заявленной темы содержанию, соответствие методическим требованиям (цели, ссылки на ресурсы, соответствие содержания и литературы), практическая направленность, соответствие содержания заявленной форме, адекватность использования технических средств учебным задачам, последовательность и логичность презентуемого материала;
- оформление презентации: объем (оптимальное количество), дизайн (читаемость, наличие и соответствие графики и анимации, звуковое оформление, структурирование информации, соответствие заявленным требованиям), оригинальность оформления, эстетика, использование возможности программной среды, соответствие стандартам оформления;
- личностные качества: ораторские способности, соблюдение регламента, эмоциональность, умение ответить на вопросы, систематизированные, глубокие и полные знания по всем разделам программы;
- содержание выступления: логичность изложения материала, раскрытие темы, доступность изложения, эффективность применения средств ИКТ, способы и условия достижения результативности и эффективности для выполнения задач своей профессиональной или учебной деятельности, доказательность принимаемых решений, умение аргументировать свои заключения, выводы.

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Основная литература:

1. Хорев П.Б. Объектно-ориентированное программирование : учеб. пособие. - 4-е изд., стер.. - М.: Академия, 2012. - 448 с.
2. Зуев М.С. Программирование на C++ : учеб. пособие для вузов : в 5 ч.. - Тамбов: [Издат. дом ТГУ], 2011
3. Малашонок Г.И., Переславцева О.Н., Рыбаков М.А. Параллельное программирование на OpenMP и Java с приложениями в Math Partner : в 3 ч. : учеб. пособие. - Тамбов: [Издат. дом ТГУ им. Г.Р. Державина], 2014
4. Программирование на языке Delphi : лабораторный практикум: в 2 ч., Ч.1. - Тамбов: Изд-во ТГУ, 2010. - 116 с.

6.2 Дополнительная литература:

1. Программирование на языке Delphi : лабораторный практикум : в 2 ч., Ч.2. - Тамбов: [Издат. дом ТГУ], 2011. - 192 с.
2. Сидляр М.Ю., Максимова Н.А. Программирование в MyPascal2 : учеб.-метод. пособие. - Тамбов: [Издат. дом ТГУ им. Г.Р.Державина], 2012. - 134 с.
3. Биллиг В. А. Основы программирования на C : учебный курс. - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2006. - 485 с. - Текст : электронный // ЭБС «Университетская библиотека онлайн» [сайт]. - URL: <http://biblioclub.ru/index.php?page=book&id=233695>

6.3 Иные источники:

1. Журнал РАН «Программирование» - <http://www.ispras.ru/programming/>
2. Журнал «Кибернетика и программирование» - <http://e-notabene.ru/kp/>
3. Журнал «Практика функционального программирования» - <http://fprog.ru/>

4. Журнал «Компьютеры, Сети, Программирование» - https://books.google.ru/books/about/%D0%96%D1%83%D1%80%D0%BD%D0%B0%D0%BB_%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D1%8B_%D0%A1%D0%B5%D1%82.html?id=gsgnugBhj1cC&redir_esc=y
5. Журнал «БИТ» - <http://bit.samag.ru/>
6. Журнал «Системный администратор» - <http://samag.ru/main/part/2>

7. Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы

Для проведения занятий по дисциплине необходимо следующее материально-техническое обеспечение: учебные аудитории для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещения для самостоятельной работы.

Учебные аудитории и помещения для самостоятельной работы укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы укомплектованы компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду Университета.

Для проведения занятий лекционного типа используются наборы демонстрационного оборудования, обеспечивающие тематические иллюстрации (проектор, ноутбук, экран/ интерактивная доска).

Лицензионное и свободно распространяемое программное обеспечение:

Операционная система "Альт Образование"

LibreOffice

Microsoft Windows 10

Microsoft Office Профессиональный плюс 2007

Профессиональные базы данных и информационные справочные системы:

1. Электронный каталог Фундаментальной библиотеки ТГУ. – URL: <http://biblio.tsutmb.ru/elektronnyij-katalog>
2. Университетская библиотека онлайн: электронно-библиотечная система. – URL: <https://biblioclub.ru>
3. Консультант студента. Гуманитарные науки: электронно-библиотечная система. – URL: <https://www.studentlibrary.ru>
4. Научная электронная библиотека eLIBRARY.ru. – URL: <https://elibrary.ru>
5. Российская государственная библиотека. – URL: <https://www.rsl.ru>
6. Российская национальная библиотека. – URL: <http://nlr.ru>
7. Президентская библиотека имени Б.Н. Ельцина. – URL: <https://www.prlib.ru>
8. Научная электронная библиотека Российской академии естествознания. – URL: <https://www.monographies.ru>
9. Электронная библиотека РФФИ. – URL: <https://www.rfbr.ru/rffi/ru/library>

Электронная информационно-образовательная среда

https://auth.tsutmb.ru/authorize?response_type=code&client_id=moodle&state=xyz

Взаимодействие преподавателя и студента в процессе обучения осуществляется посредством мультимедийных, гипертекстовых, сетевых, телекоммуникационных технологий, используемых в электронной информационно-образовательной среде университета.