

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



И. Н. Якунина
«20» января 2021 г.

РАБОЧАЯ ПРОГРАММА

по дисциплине Б1.Б.23 Языки программирования

Направление подготовки/специальность: 10.03.01 - Информационная безопасность

Профиль/направленность/специализация: Безопасность компьютерных систем

Уровень высшего образования: бакалавриат

Квалификация: Бакалавр

год набора: 2020

Тамбов, 2021

Авторы программы:

Кандидат педагогических наук, Скворцов Александр Александрович

Кандидат технических наук, Соловьев Денис Сергеевич

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 10.03.01 - Информационная безопасность (уровень бакалавриата) (приказ Министерства образования и науки РФ от «01» декабря 2016 г. № 1515).

Рабочая программа принята на заседании Кафедры математического моделирования и информационных технологий «22» декабря 2020 г. Протокол № 4

Рассмотрена и одобрена на заседании Ученого совета Института математики, физики и информационных технологий, Протокол от «20» января 2021 г. № 1.

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.....	20
5 Методические указания для обучающихся по освоению дисциплины (модуля).....	56
6 Учебно-методическое и информационное обеспечение дисциплины.....	58
7 Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональны	58

1. Цели и задачи дисциплины

1.1 Цель дисциплины – формирование компетенций:

ОПК-3 Способность применять положения электротехники, электроники и схемотехники для решения профессиональных задач

ПК-2 Способность применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач

1.2 Виды и задачи профессиональной деятельности по дисциплине:

- организационно-управленческая
 - осуществление организационно-правового обеспечения информационной безопасности объекта защиты
 - организация работы малых коллективов исполнителей
 - участие в совершенствовании системы управления информационной безопасностью
 - изучение и обобщение опыта работы других учреждений, организаций и предприятий в области защиты информации, в том числе информации ограниченного доступа
 - контроль эффективности реализации политики информационной безопасности объекта защиты
- эксплуатационная
 - установка, настройка, эксплуатация и поддержание в работоспособном состоянии компонентов системы обеспечения информационной безопасности с учетом установленных требований
 - администрирование подсистем информационной безопасности объекта
 - участие в проведении аттестации объектов информатизации по требованиям безопасности информации и аудите информационной безопасности автоматизированных систем

1.3 В результате освоения дисциплины у обучающихся должны быть сформированы следующие компетенции:

Обобщенные трудовые	Код и наименование ко	Знания и умения, необходимые дл
	ОПК-3 Способность приме	Знает и понимает: структуру создания алгоритма программного кода
		Умеет (способен продемонстрировать): проводить анализ кода программ
		Владеет: создания программного кода
	ПК-2 Способность применя профессиональных задач	Знает и понимает: основных возможностей языков программирования и о
		Умеет (способен продемонстрировать): использовать языки программирования для создания пр
		Владеет: программирования в современных средах

1.4 Согласование междисциплинарных связей дисциплин, обеспечивающих освоение компетенций:

ОПК-3 Способность применять положения электротехники, электроники и схемотехники для решения профессиональных задач

№ п/п	Наименование д	Форм	
		Очна	
		(семе	
		3	4
	Адаптационная дисци	+	

2	Основы электро- и ра	+	
3	Физика		+
4	Электроника и схемот	+	

ПК-2 Способность применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач

№ п/п	Наименование д	Форма обучен					
		Очная (семестр)					
		1	2	3	4	5	7
1	Автоматизация деятел			+	+	+	
2	Адаптационная дисци			+	+	+	
3	Алгоритмизация и пр	+	+	+			
4	Основы программиро			+	+	+	
5	Эксплуатационная пр						+

2. Место дисциплины в структуре ОП бакалавриата:

Дисциплина «Языки программирования» относится к базовой части учебного плана ОП по направлению подготовки 10.03.01 - Информационная безопасность.

Дисциплина «Языки программирования» изучается в 5, 6 семестрах.

3. Объем и содержание дисциплины

3.1. Объем дисциплины: 7 з.е.

Очная: 7 з.е.

Вид учебной работы	Очная (всего часов)
Общая трудоёмкость дисциплины	252
Контактная работа	102
Лекции (Лекции)	34
Лабораторные (Лаб. раб.)	68
Самостоятельная работа (СР)	114
Экзамен	36
Зачет	-

3.2. Содержание курса:

№ темы	Название раздела/темы	Вид учебной работы, час.			Формы текущего контроля
		Лек ции	Лаб · раб.	СР	
		О	О	О	
5 семестр					
1	Введение. Структура программы на С. Основные операторы языка С.	2	6	10	Лабораторная работа; Собеседование

2	Массивы. Указатели и ссылки. Указатели и массивы.	2	6	12	Лабораторная работа; Тестирование; Собеседование
3	Символьная информация и строки. Функции в С. Типы данных, определяемые пользователем. Ввод-вывод в С.	2	6	10	Лабораторная работа; Тестирование; Собеседование
4	Объекты и классы. Наследование.	2	6	8	Лабораторная работа; Тестирование; Собеседование
5	Перегрузка операций и преобразование типов.	2	6	8	Лабораторная работа; Собеседование
6	Виртуальные и дружественные функции.	2	6	8	Лабораторная работа; Собеседование
7	Шаблоны и исключения.	2	8	8	Лабораторная работа; Собеседование
8	Стандартная библиотека шаблонов.	4	8	10	Лабораторная работа; Собеседование
6 семестр					
9	Введение в PHP. Основы синтаксиса. Операторы PHP. Условные операторы. Циклы в PHP	2	2	4	Лабораторная работа; Тестирование; Собеседование
10	Обработка запросов с помощью PHP. Функции в PHP. Строки в PHP.	2	2	4	Лабораторная работа; Собеседование
11	Объекты и классы в PHP. Массивы в PHP. Работа с файловой системой	2	2	6	Лабораторная работа; Тестирование; Собеседование
12	Базы данных и СУБД. Введение в SQL	2	2	6	Лабораторная работа; Собеседование
13	Взаимодействие PHP и MySQL.	2	2	4	Лабораторная работа; Собеседование

14	Основы клиент-серверных технологий.	2	2	4	Лабораторная работа; Собеседование
15	Регулярные выражения	2	2	6	Лабораторная работа; Собеседование
16	Использование шаблонов в PHP. Взаимодействие PHP и XML. Работа с FTP средствами PHP.	2	2	6	Лабораторная работа; Собеседование

Тема 1. Введение. Структура программы на С. Основные операторы языка С. (ОПК-3)

Лекция.

Введение. Алгоритм и программа. Свойства алгоритма. Компиляторы и интерпретаторы. Языки программирования. Структура программы на С. Базовые средства языка С. Состав языка. Константы в С. Типы данных в С. Переменные. Знаки операций в С. Выражения. Ввод и вывод данных. Функции printf, scanf. Поточковый ввод и вывод (cin, cout). Основные операторы языка С. Базовые конструкции структурного программирования. Оператор «выражение». Составные операторы. Операторы выбора. Операторы циклов. Операторы перехода.

Лабораторные работы.

Лабораторная работа.

Написать две программы на языке С для расчета значений переменных у и z по заданным формулам (таблица). В первой программе использовать для ввода функцию scanf, для вывода – функцию printf. Во второй программе использовать операторы потокового ввода-вывода cin и cout. Определить разность между значениями у и z. В программе предусмотреть ввод исходных данных с экрана дисплея. Предварительно вычислите ожидаемые значения у и z с помощью калькулятора. Убедитесь, что значения, вычисленные с помощью калькулятора, совпадают с результатами, которые получаются в результате работы программы.

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».
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. Массивы. Указатели и ссылки. Указатели и массивы. (ПК-2)

Лекция.

Массивы. Определение массива в С. Обработка одномерных массивов. Перебор массива по одному элементу. Формирование псевдодинамических массивов. Использование датчика случайных чисел для формирования массива. Перебор массива по два элемента. Классы задач по обработке массивов. Сортировка массивов. Поиск в отсортированном массиве. Указатели и ссылки. Понятие указателя. Динамические переменные. Операции с указателями. Понятие ссылки. Правила работы со ссылками. Указатели и массивы. Одномерные массивы и указатели. Многомерные массивы и указатели. Динамические массивы.

Лабораторные работы.

Лабораторная работа.

Задание 1: Составить алгоритм решения задачи с одномерным массивом согласно своему варианту и написать программу на языке С.

Задание 2: Составить алгоритм решения задачи с двумерным массивом согласно своему варианту и написать программу на языке С.

Составить алгоритм решения задачи с использованием указателей и динамических массивов в качестве аргументов функций согласно своему варианту и написать программу на языке C. В отчете предоставить блок-схемы всех функций, текст программы.

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?

41. Что такое указатель?
42. Какой объем памяти занимает указатель?
43. Что является значением переменной-указателя?
44. Как проинициализировать указатель?
45. Что такое NULL?
46. Что такое указатель на void? Зачем нужны такие указатели?
47. Какие операции допустимы при работе с указателями?
48. Чем отличается унарная операция "&" от унарной "*" ?
49. Совместимость типов указателей.
50. Можно ли получить адрес указателя?
51. Можно ли указателю присвоить его же адрес?
52. Почему к указателю на void нельзя применить операцию разыменования?
53. Как работают операции инкремента и декремента, примененные к указателям?
54. Каков результат операции вычитания, примененной к указателям одного типа?
55. Какой спецификатор типа используется при выводе адреса на экран с помощью функции printf()?
56. В чем отличие записи (float *) а от (float) * а, если а – указатель на целое число?
57. В чем отличие записи *a++ от (*a)++, если а – некоторый указатель, отличный от void*?
58. Как описать указатель на начало массива?
59. Как описать указатель на указатель?
60. Когда и зачем может повторно использоваться операция разыменования?
61. Какие переменные называют динамическими? Чем они отличаются от статических переменных?
62. С какой целью используют динамические переменные?
63. Как обращаются к динамическим переменным?
64. Какие бывают указатели? Как их описать?
65. Каким образом можно выделять память для динамических переменных и освободить её?
66. Как определить, выделена память или нет?
67. Почему нельзя забывать освобождать выделенную память?
68. Чему равно значение указателя после освобождения области памяти, на которую он указывал?
69. Как разместить в динамической памяти массив?
70. Какими способами можно разместить в динамической памяти матрицу?

Тема 3. Символьная информация и строки. Функции в С. Типы данных, определяемые пользователем. Ввод-вывод в С. (ПК-2)

Лекция.

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

Лабораторная работа.

Написать две программы согласно индивидуальному варианту. В случае, когда указано, что задается строка до точки, обрезать строку, отбросив все символы после точки, если точки нет, то считать, что строка ограничена точкой. Использовать функции библиотеки string.

Составить алгоритм решения задачи с использованием функции согласно своему варианту и написать программу на языке C. В отчете предоставить блок-схемы функции согласно задания и основной функции.

Написать программу на языке C решения задачи согласно своему варианту. В программе создать соответствующую заданию структуру, можно написать функции, реализующие операции со структурами. В программе можно использовать массив переменных созданного типа структуры.

Во всех заданиях по вариантам составить две программы. Первая программа должна формировать двоичный файл. Вторая программа должна считать данные из двоичного файла, выполнить соответствующие вычисления и записать их результаты в текстовый файл.

В программах необходимо использовать только динамические структуры. Согласно своему варианту выполнить следующие действия:

С помощью текстового редактора создать текстовый документ, в который занести исходную информацию. Рекомендуется вначале занести количество записей, а затем последовательно все записи, причем тестовые поля вводить в отдельной строке, а числовые – можно в отдельной строке или в одной через пробел или разделитель табуляцию.

Написать первую программу, которая считывает информацию из созданного текстового файла и записывает ее в двоичный файл.

Написать вторую программу, которая считывает информацию из двоичного файла, реализует поставленную задачу. После повторного чтения двоичного файла результат работы выводится в результирующий текстовый файл.

Все текстовые и двоичные файлы должны располагаться или в личной папке или в папке проекта.

Задания для самостоятельной работы.

1. Способы организации строк.
2. Что такое Си-строка? Чем она отличается от массива символов?
3. Что такое нуль-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?
10. Какие функции для работы со строками Вы можете назвать?
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?
21. Где может быть использована структура?
22. Как называются элементы структуры?
23. Как определить структуру?
24. Может ли структура содержать элементы одного типа?
25. Как обратиться к полям структуры?
26. Как обратиться к элементу массива структур?
27. Какое ключевое слово начинает определение структуры?
28. Как обратиться к полям структуры через указатель на нее?
29. При помощи какой операции формируется доступ к полям структуры?
30. Какая разница между определениями структуры и массива?
31. Опишите объект, для которого надо определить структуру и определите ее.
32. Что такое файл с точки зрения информатики и вычислительной техники?
33. Что такое файл с точки зрения языка программирования C?
34. Какие виды файлов Вы знаете?
35. В чем отличие текстового файла от бинарного и от чего зависит наличие этого различия?
36. Что такое поток?
37. Что такое признак окончания файла? Для каких файлов он существует?
38. Как инициализировать поток?
39. Что такое структурный тип?
40. Какая функция отвечает за открытие потока? Какие у нее параметры?
41. Какие режимы открытия файла Вы знаете? В чем их отличия друг от друга?
42. Какие ошибки могут возникнуть при открытии файла?
43. В каком случае указатель на поток принимает значение `NULL`?
44. Какая функция отвечает за закрытие файла? Какие у нее параметры?
45. Зачем закрывать файл?
46. Сколько раз можно открыть файл в программе?
47. Какие функции используются для ввода-вывода данных при работе с текстовыми файлами?
48. Какие функции используются для ввода-вывода данных при работе с бинарными файлами?
49. В чем отличие функции `fprintf()` от функции `fread()`, а функции `fscanf()` от `fwrite()`?
50. Как определить текущую позицию указателя на поток?
51. Как переместить указатель на поток из текущей позиции в заданную?
52. За что отвечает константа `EOF`? Как и где ее можно использовать?
53. Какие основные функции для работы с файлами Вы знаете?
54. С помощью каких функций можно осуществлять побайтовую обработку файлов?
55. Для чего используются функции `rename()` и `remove()`? Нужно ли при работе с ними объявлять файловый указатель?
56. Можно ли удалить или переименовать открытый файл?

Тема 4. Объекты и классы. Наследование. (ПК-2)

Лекция.

Определение класса (поля, методы, доступ к членам класса). Определение методов в классе и вне класса. Конструкторы и деструкторы. Объекты как аргументы методов и доступ к их членам. Конструктор копирования по умолчанию. Размещение в памяти членов объектов одного класса. Способы создания общих полей (свойств). Константные методы, их аргументы и константные объекты.

Базовый и производный классы. Конструкторы производного класса. Базовые методы класса. Иерархия классов. Общее и частное наследование. Уровни наследования. Множественное наследование. Неопределенность при множественном наследовании. Включение: классы в классах.

Лабораторные работы.

Лабораторная работа.

Написать программу, в которой создаются и разрушаются объекты по вариантам, определенного пользователем класса. Выполнить исследование вызовов конструкторов и деструкторов.

Написать программу, в которой создается иерархия классов по вариантам. Включить полиморфные объекты в связанный список, используя статические компоненты класса. Показать использование виртуальных функций.

Задания для самостоятельной работы.

1. Что такое класс, объект?
2. Что такое переменная экземпляра?
3. Какая разница между списком аргументов и списком параметров?
4. Что такое конструктор и как он определяется?
5. Что такое конструктор копирования и его использование?
6. Что такое наследование?
7. Какие классы называются базовыми, а какие производными?
8. Как объявить производный класс?
9. Для чего в базовом классе применяется ключевое слово `protected`?
10. Как работают конструкторы и деструкторы при создании производных объектов?
11. Как выполнить инициализацию базового класса при создании производного объекта?
12. Передаются ли по наследству производные члены-данные и функции через поколения?

Тема 5. Перегрузка операций и преобразование типов. (ПК-2)

Лекция.

Перегрузка операций. Перегрузка унарных операций. Перегрузка бинарных операций. Перегрузка операций индексации массива. Преобразование типов. Преобразование типов от основного к пользовательскому. Преобразование типов от пользовательского к основному. Преобразование типов от пользовательского к пользовательскому. Функция в исходном объекте. Функция в объекте назначения. Предотвращение преобразования типа от основного к пользовательскому с помощью конструктора.

Лабораторные работы.

Лабораторная работа.

Написать программу на языке C++, реализующую один из вариантов задания. В программе должны присутствовать как перегруженные операторы, так и пользовательские преобразования между типами данных.

Задания для самостоятельной работы.

1. Для чего в C++ применяется перегрузка операций?
2. Истинно ли следующее утверждение: операция `>=` может быть перегружена?
3. Сколько аргументов требуется для определения перегруженной унарной операции?
4. Сколько аргументов требуется для определения перегруженной бинарной операции?
5. Чем отличается действие операции `++` в префиксной форме от её действия в постфиксной форме?
6. Истинно ли следующее утверждение: перегруженная операция всегда требует на один аргумент меньше, чем количество операндов?
7. Когда перегружается операция арифметического присваивания, то результат

8. Передается объекту справа от операции
9. Передается объекту слева от операции
10. Передается объекту, вызвавшему операцию
11. Должен быть возвращен
12. Какой механизм преобразования от определенного пользователем класса к встроенному типу может быть использован в языке C++?
13. Какой механизм преобразования от встроенного типа данных к определенному пользователем может быть использован в языке C++?
14. Истинно ли следующее утверждение: компилятор не выдаст сообщение об ошибке, если вы перегрузите операцию * для выполнения деления?
15. Если объект objA принадлежит классу A, объект objB принадлежит классу B, и требуется записать objA = objB, поместив при этом функцию преобразования в класс A, то какую разновидность процедуры преобразования типа можно использовать?
16. Существуют ли операции, которые нельзя перегружать?
17. Что такое конструктор преобразования?
18. Для чего используется ключевое слово explicit?

Тема 6. Виртуальные и дружественные функции. (ПК-2)

Лекция.

Полиморфизм. Виртуальные функции. Абстрактные классы и чистые виртуальные функции. Виртуальные деструкторы. Виртуальные базовые классы. Дружественные функции. Статические функции. Инициализация копирования и присваивания. Перегрузка оператора присваивания. Указатель this.

Лабораторные работы.

Лабораторная работа.

Определить и реализовать класс – абстрактный тип данных. Определить и реализовать операции над данными этого класса. Написать и выполнить программу полного тестирования этого класса.

Задания для самостоятельной работы.

1. Различают 4 элемента определения функции: тип, имя, список параметров, тело. Какими элементами могут отличаться экземпляры одной виртуальной функции, находящиеся в разных производных классах?
2. Какая функция называется другом класса?
3. В чем особенности передачи параметров дружественной функции при перегрузке унарных и бинарных операций?

Тема 7. Шаблоны и исключения. (ПК-2)

Лекция.

Определение функций-шаблонов. Явная перегрузка функций-шаблонов. Ограничения на функции-шаблоны. Классы-шаблоны. Обработка исключений. Использование нескольких инструкций catch. Перехват всех исключений. Перехват всех исключений. Повторная генерация исключений.

Лабораторные работы.

Лабораторная работа.

Создать шаблон заданного класса и использовать его для данных различных типов.

Задания для самостоятельной работы.

1. Что такое шаблоны и с какой целью они используются?
2. Какого типа шаблоны используются в программах?
3. Как оформляются шаблоны функций?
4. Какие требования предъявляются к фактическим параметрам шаблонов?

5. Какие преимущества программы обеспечиваются при использовании шаблонов классов?

Тема 8. Стандартная библиотека шаблонов. (ПК-2)

Лекция.

Сущности STL. Последовательные контейнеры. Ассоциативные контейнеры. Итераторы. Алгоритмы в STL (поиск, сортировка, удаление, другие функции). Функциональный объект.

Лабораторные работы.

Лабораторная работа.

- 1) Получить практические навыки создания объектов-групп и использования методов-итераторов.
- 2) Написать три программы с использованием STL. Первая и вторая программы должны продемонстрировать работу с контейнерами STL, третья – использование алгоритмов STL.

Задания для самостоятельной работы.

1. Какие есть контейнеры в C++?
2. За чем нужны контейнеры?
3. В чем отличие массивов от vector/list?
4. Что такое итераторы? Для чего они используются?
5. Как итерироваться по list, map, vector?
6. В чем отличие между stack/queue, list/vector?

Тема 9. Введение в PHP. Основы синтаксиса. Операторы PHP. Условные операторы. Циклы в PHP (ПК-2)

Лекция.

Введение в PHP. История языка PHP. Возможности PHP (краткий перечень платформ, протоколов, баз данных, приложений электронной коммерции и функций, которые поддерживаются PHP). Области применения PHP (как серверное приложение, в командной строке, создание GUI приложений); Способы использования. Установка и настройка программного обеспечения, необходимого для работы с PHP.

Основной синтаксис PHP. Способы разделения инструкций, создания комментариев. Переменные, константы и типы данных, операторы. Математические операторы и математические функции PHP. Оператор присвоения в PHP. Операторы отношения в PHP. Логические операторы в PHP. Поразрядные операторы в PHP. Строковые операторы в PHP. Оператор подавления ошибок в PHP. Операторы увеличения и уменьшения в PHP. Условный оператор IF в PHP. Операторы сравнения PHP. Логические операторы PHP. Оператор ELSE в PHP. Оператор ELSEIF в PHP. Тернарный оператор PHP. Оператор SWITCH в PHP. Цикл FOR в PHP. Циклы WHILE в PHP. Цикл DO... WHILE в PHP. Цикл FOREACH в PHP.

Лабораторные работы.

Лабораторная работа.

Создание заготовки письма.

Задания для самостоятельной работы.

Задания для самостоятельной работы.

1. Возможности PHP.
2. Области применения PHP.
3. Способы использования.
4. Установка и настройка программного обеспечения.

5. Разделение конструкций.
6. Комментарии.
7. Переменные, константы и операторы.
8. Вывод информации.
9. Типы данных.
10. Математические операторы и математические функции PHP.
11. Операторы увеличения и уменьшения в PHP.
12. Операторы сравнения PHP.
13. Оператор ELSE.
14. Оператор ELSEIF
15. Оператор SWITCH.
16. Цикл FOR.
17. Циклы WHILE.
18. Цикл DO... WHILE

Тема 10. Обработка запросов с помощью PHP. Функции в PHP. Строки в PHP. (ПК-2)

Лекция.

Способы отправки данных на сервер и их обработка с помощью PHP. Основные понятия клиент-серверных технологий. Понятие HTML-формы и отправка данных с ее помощью. Краткая характеристика методов POST и GET. Механизм получения данных из HTML-форм и их обработка с помощью PHP. Определение функций в PHP, их синтаксис, создание и вызов. Аргументы функций в PHP. Передача аргументов по значению и по ссылке, значение аргументов по умолчанию и значения, возвращаемые функцией (функция return()). Функции для обработки строк в PHP. Функции, определяемые пользователем. Переменные функции. Внутренние функции. Как в PHP передать данные в функцию. Что такое строка? Обработка переменных внутри строк. Вывод строк. Форматированный вывод строк. Длина строки в PHP. Поиск элемента в строке. Поиск подстроки в строке. Чистка строк. Замена вхождения подстроки. Разделение и соединение строки. Функции для обработки строк в PHP. Использование строковых функций в PHP.

Лабораторные работы.

Лабораторная работа.

Создание формы для регистрации пользователей на сайте.

1. Отправка «универсального письма» всем зарегистрировавшимся.

Задания для самостоятельной работы.

Задания для самостоятельной работы.

1. Отправка данных с помощью HTML-формы.

2. Методы POST и GET.

Тема 11. Объекты и классы в PHP. Массивы в PHP. Работа с файловой системой (ПК-2)

Лекция.

Понятия класса и объекта. Определение и использование классов. Понятие расширения класса. Константы классов. Автоматическая загрузка классов. Конструкторы и деструкторы. Область видимости. Наследование. Создание массивов в PHP. Модификация элементов массива в PHP. Удаление элементов массива в PHP. Перебор элементов массива в PHP. Функции для работы с массивами в PHP. Сортировка массивов в PHP. Навигация по массивам в PHP. Преобразование строк в массивы и наоборот в PHP. Извлечение переменных из массивов в PHP. Слияние и разделение массивов в PHP. Сравнение массивов в PHP. Добавление элементов массива. Удаление элементов массива. Обработка данных в массивах PHP. Ассоциативные массивы. Многомерные массивы в PHP. Использование циклов в многомерных массивах в PHP. Операторы над массивами в PHP. Создание файла. Закрытие соединения с файлом. Запись данных в файл. Чтение данных из файла. Проверка существования файла. Копирование и переименование файлов в PHP. Удаление файла. Загрузка файла на сервер. Получение информации о файлах в PHP. Файловый указатель в PHP.

Лабораторные работы.

Лабораторная работа..

Автоматическая генерация по желанию пользователя представителей классов статей или личностей, а также их отображения на странице браузера.

Загрузка файла на сервер с помощью web-интерфейса.

Обработка массивов данных

Задания для самостоятельной работы.

Задания для самостоятельной работы.

1. Понятия класса и объекта.

2. Определение и использование классов.

Конструкторы и деструкторы

1. Создание массивов в PHP (удаление, перебор, функции).

2. Сортировка массивов в PHP (навигация, сравнение).

3. Особенности работы с многомерными массивами.

1. Создание файлов.

2. Чтение данных из файла.

3. Удаление файла.

Проверка наличия файла на сервере

Тема 12. Базы данных и СУБД. Введение в SQL (ПК-2)

Лекция.

Работа с базами данных MySQL в PHP. Введение в SQL. Базы данных: основные понятия. Язык запросов SQL: операции выбора, добавления, изменения и удаления строки, а также операции создания, изменения и удаления таблицы. Соединение PHP с сервером базы данных MySQL. Создание и удаление базы данных MySQL. Создание и удаление таблиц MySQL. База данных MySQL. Использование PhpMyAdmin для взаимодействия с базой данных MySQL.

Лабораторные работы.

Лабораторная работа.

1. Проектирование базы данных.
2. Создание базы данных MySQL.
3. Установка соединения с базой данных.
4. Создание страницы для добавления записей базы данных

Задания для самостоятельной работы.

Задания для самостоятельной работы.

1. Базы данных: основные понятия.

Создание и удаление базы данных MySQL

Тема 13. Взаимодействие PHP и MySQL. (ПК-2)

Лекция.

Способы взаимодействия PHP и СУБД MySQL. Установка соединения с базой данных, функции отправки запросов и обработка ответов (mysql_connect, mysql_query, mysql_result, mysql_num_rows, mysql_close).

Лабораторные работы.

Создание страницы для удаления и редактирования записей базы данных

Задания для самостоятельной работы.

1. Способы взаимодействия PHP и СУБД MySQL.
2. Функции отправки запросов.

Тема 14. Основы клиент-серверных технологий. (ОПК-3)

Лекция.

Типы серверов. Использование HTML форм для передачи данных на сервер. Обработка запросов с помощью PHP. Задания для самостоятельной работы

Лабораторные работы.

Лабораторная работа.

1. Типы серверов.
2. Обработка запросов с помощью PHP

Задания для самостоятельной работы.

Задания для самостоятельной работы.

Обработка запросов с помощью PHP.

Тема 15. Регулярные выражения (ПК-2)

Лекция.

Понятие регулярного выражения, реализация механизма регулярных выражений в языке PHP. Синтаксис регулярных выражений. Модификаторы PCRE. Более сложные конструкции регулярных выражений. Изучение вопросов обеспечения безопасности в сети и использованию для этих целей механизма сессий. Инициализация сессий, передача идентификатора пользователя, регистрация переменных сессии, уничтожение сессии. Настройка сессий в файлах `php.ini`, `httpd.conf`, `.htaccess`.

Лабораторные работы.

Задача контекстного поиска

Авторизация пользователя с помощью механизма сессий.

Задания для самостоятельной работы.

1. Понятие регулярного выражения.
2. Квантификаторы
 1. Регистрация переменных сессии.
 2. Уничтожение сессии.
3. Настройка сессий в файлах `php.ini`.

Тема 16. Использование шаблонов в PHP. Взаимодействие PHP и XML. Работа с FTP средствами PHP. (ПК-2)

Лекция.

Понятие шаблона и его использование в языке программирования PHP, а также классы шаблонов `FastTemplate` и `Smarty`. XML: основные понятия. Объектная модель XML-документа. Расширения `SAX` и `DOMXML`. Установка расширения `DOMXML`. Взаимодействие PHP и XML. Основные возможности PHP при работе с FTP. Соединение с FTP-сервером. Регистрация на FTP-сервере. Закрывание соединения. Загрузка файлов с сервера

Лабораторные работы.

Генерация списка документов по таблице базы данных

Установка расширения `DOMXML`

Задания для самостоятельной работы.

1. Использование шаблонов в PHP.
1. XML: основные понятия.
2. Взаимодействие PHP и XML посредством `DOMXML`.
3. Функции `ftp_connect()`, `ftp_get()`, `ftp_login()`, `ftp_quit()`

4. Контроль знаний обучающихся и типовые оценочные средства

4.1. Распределение баллов:

5 семестр

- посещаемость – 10 баллов
- текущий контроль – 60 баллов

- контрольные срезы – 3 среза по 10 баллов каждый
- премиальные баллы – 20 баллов

Распределение баллов по заданиям:

№ т мы	Название т	Формы	Мах. ко	Методика проведения занятия и оце
1.	Введение. Ст	Лабора	4	Лабораторные работы выполняются по тематике практических за 4 балла – лабораторная работа выполнена в полном объёме, студен 3 балла – лабораторная работа выполнена, но имеет некоторые нет 2 балла - лабораторная работа в целом выполнена, однако в проце
		Собесе	4	Собеседование предполагает организацию беседы преподавателя с - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения п - своевременность и эффективность использования наглядных пос - использование дополнительного материала; - рациональность использования времени, отведенного на задание 4 балла – студент умеет сопоставить полученную при подготовке к 2 балла - студент умеет применять полученную при подготовке к п Если студент не владеет проблематикой практического занятия, не
2.	Массивы. У	Лабора	4	Лабораторные работы выполняются по тематике практических за 4 балла – лабораторная работа выполнена в полном объёме, студен 3 балла – лабораторная работа выполнена, но имеет некоторые нет 2 балла - лабораторная работа в целом выполнена, однако в проце
		Тестиров ание(кон трольны й срез)	10	Тестирование подразумевает 10 вопросов. За прохождение тестир - 90 % - 10 баллов; - 65 % - 5 баллов; - 50 % - 2 балла; - менее 50 % - балл не начисляется
		Собесе	3	Собеседование предполагает организацию беседы преподавателя с - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения п - своевременность и эффективность использования наглядных пос - использование дополнительного материала; - рациональность использования времени, отведенного на задание 3 балла – студент умеет сопоставить полученную при подготовке к 1 балл - студент умеет применять полученную при подготовке к пр Если студент не владеет проблематикой практического занятия, не
3.	Символьная инф в С.	Лабора	4	Лабораторные работы выполняются по тематике практических за 4 балла – лабораторная работа выполнена в полном объёме, студен 3 балла – лабораторная работа выполнена, но имеет некоторые нет 2 балла - лабораторная работа в целом выполнена, однако в проце
		Тестиров ание(кон трольны й срез)	10	Тестирование подразумевает 10 вопросов. За прохождение тестир - 90 % - 10 баллов; - 65 % - 5 баллов; - 50 % - 2 балла; - менее 50 % - балл не начисляется.

		Собесе	3	Собеседование предполагает организацию беседы преподавателя с - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения п - своевременность и эффективность использования наглядных пос - использование дополнительного материала; - рациональность использования времени, отведенного на задание 3 балла – студент умеет сопоставить полученную при подготовке к 1 балл - студент умеет применять полученную при подготовке к пр Если студент не владеет проблематикой практического занятия, не
4.	Объекты и	Лабора	3	Лабораторные работы выполняются по текущему разделу или тем 3 балла – лабораторная работа выполнена в полном объеме, студен 2 балла – лабораторная работа выполнена, но имеет некоторые нет 1 балл - лабораторная работа в целом выполнена, однако в процес
		Тестиров ание(кон трольны й срез)	10	Тестирование подразумевает 10 вопросов. За прохождение тестир - 90 % - 10 баллов; - 65 % - 5 баллов; - 50 % - 2 балла; - менее 50 % - балл не начисляется.
		Собесе	3	Собеседование предполагает организацию беседы преподавателя с - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения п - своевременность и эффективность использования наглядных пос - использование дополнительного материала; - рациональность использования времени, отведенного на задание 3 балла – студент умеет сопоставить полученную при подготовке к 1 балл - студент умеет применять полученную при подготовке к пр Если студент не владеет проблематикой практического занятия, не
5.	Перегрузка	Лабора	4	Лабораторные работы выполняются по тематике практических за 4 баллов – лабораторная работа выполнена в полном объеме, студе 3 баллов – лабораторная работа выполнена, но имеет некоторые не 2 балла - лабораторная работа в целом выполнена, однако в проце
		Собесе	4	Собеседование предполагает организацию беседы преподавателя с - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения п - своевременность и эффективность использования наглядных пос - использование дополнительного материала; - рациональность использования времени, отведенного на задание 4 балла – студент умеет сопоставить полученную при подготовке к 2 балла - студент умеет применять полученную при подготовке к п Если студент не владеет проблематикой практического занятия, не
6.	Виртуальнь	Лабора	4	Лабораторные работы выполняются по тематике практических за 4 балла – лабораторная работа выполнена в полном объеме, студен 2 балла – лабораторная работа выполнена, но имеет некоторые нет 1 балл - лабораторная работа в целом выполнена, однако в процес

		Собесе	4	Собеседование предполагает организацию беседы преподавателя с студентом по заданию. Критерии оценки: - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения проблемы; - своевременность и эффективность использования наглядных пособий; - использование дополнительного материала; - рациональность использования времени, отведенного на задание. 4 балла – студент умеет сопоставить полученную при подготовке к занятию информацию с информацией, полученной в ходе занятия; 2 балла - студент умеет применять полученную при подготовке к занятию информацию при выполнении задания; 1 балл - студент не владеет проблематикой практического занятия, не может ответить на вопросы преподавателя.
7.	Шаблоны	Лабора	4	Лабораторные работы выполняются по тематике практических занятий. Критерии оценки: 4 балла – лабораторная работа выполнена в полном объеме, студент умеет применять полученные знания; 2 балла – лабораторная работа выполнена, но имеет некоторые неточности; 1 балл - лабораторная работа в целом выполнена, однако в процессе выполнения студент не применяет полученные знания.
		Собесе	4	Собеседование предполагает организацию беседы преподавателя с студентом по заданию. Критерии оценки: - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения проблемы; - своевременность и эффективность использования наглядных пособий; - использование дополнительного материала; - рациональность использования времени, отведенного на задание. 4 балла – студент умеет сопоставить полученную при подготовке к занятию информацию с информацией, полученной в ходе занятия; 2 балла - студент умеет применять полученную при подготовке к занятию информацию при выполнении задания; 1 балл - студент не владеет проблематикой практического занятия, не может ответить на вопросы преподавателя.
8.	Стандартн	Лабора	4	Лабораторные работы выполняются по тематике практических занятий. Критерии оценки: 4 балла – лабораторная работа выполнена в полном объеме, студент умеет применять полученные знания; 3 балла – лабораторная работа выполнена, но имеет некоторые неточности; 2 балла - лабораторная работа в целом выполнена, однако в процессе выполнения студент не применяет полученные знания.
		Собесе	4	Собеседование предполагает организацию беседы преподавателя с студентом по заданию. Критерии оценки: - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения проблемы; - своевременность и эффективность использования наглядных пособий; - использование дополнительного материала; - рациональность использования времени, отведенного на задание. 4 балла – студент умеет сопоставить полученную при подготовке к занятию информацию с информацией, полученной в ходе занятия; 2 балла - студент умеет применять полученную при подготовке к занятию информацию при выполнении задания; 1 балл - студент не владеет проблематикой практического занятия, не может ответить на вопросы преподавателя.
9.	Посещаемость		10	10 баллов – стопроцентное посещение занятий студентом 7-9 баллов – посещаемость студента составляет не менее 80 % занятий 4-6 баллов – посещаемость студента составляет не менее 50 % занятий 1-3 балла – посещаемость студента составляет не менее 25 % занятий
10.	Премияльные б		20	Дополнительные премиальные баллы могут быть начислены: - за проект, выполненный по заказу работодателя и реализованный; - 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 из области дисциплины – 10 баллов; - участие в выставке по тематике изучаемой дисциплины – 20 баллов; - публикация статьи по тематике изучаемой дисциплины в сборнике научных трудов – 10 баллов.
11.	Индивидуальные зад		20	Провести анализ по одной статье из журналов по рекомендуемой литературе; Пройти тестирование (90 вопросов) по всему курсу дисциплины.
12.	Итого за семестр		100	

6 семестр

- посещаемость – 10 баллов
- текущий контроль – 40 баллов
- контрольные срезы – 2 среза по 10 баллов каждый
- премиальные баллы – 20 баллов
- ответ на экзамене: не более 30 баллов

Распределение баллов по заданиям:

№ т мы	Название т	Формы	Мах. ко	Методика проведения занятия и оце
1.	Введение в РН	Лабора	3	Лабораторные работы выполняются по тематике практических за 3 балла – лабораторная работа выполнена в полном объеме, студен 2 балла – лабораторная работа выполнена, но имеет некоторые нет 1 балл - лабораторная работа в целом выполнена, однако в процес
		Тестиров ание(кон трольны й срез)	10	Тестирование подразумевает 10 вопросов. За прохождение тестиро - 90 % - 10 баллов; - 65 % - 5 баллов; - 50 % - 2 балла; - менее 50 % - балл не начисляется.
		Собесе	2	Собеседование предполагает организацию беседы преподавателя с - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения п - своевременность и эффективность использования наглядных пос - использование дополнительного материала; - рациональность использования времени, отведенного на задание 2 балла – студент умеет сопоставить полученную при подготовке к 1 балл - студент умеет применять полученную при подготовке к пр Если студент не владеет проблематикой практического занятия, не
2.	Обработка за	Лабора	3	Лабораторные работы выполняются по тематике практических за 3 балла – лабораторная работа выполнена в полном объеме, студен 2 балла – лабораторная работа выполнена, но имеет некоторые нет 1 балл - лабораторная работа в целом выполнена, однако в процес
		Собесе	2	Собеседование предполагает организацию беседы преподавателя с - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения п - своевременность и эффективность использования наглядных пос - использование дополнительного материала; - рациональность использования времени, отведенного на задание 2 балла – студент умеет сопоставить полученную при подготовке к 1 балл - студент умеет применять полученную при подготовке к пр Если студент не владеет проблематикой практического занятия, не
3.	Объекты и кл	Лабора	3	Лабораторные работы выполняются по тематике практических за 3 балла – лабораторная работа выполнена в полном объеме, студен 2 балла – лабораторная работа выполнена, но имеет некоторые нет 1 балл - лабораторная работа в целом выполнена, однако в процес
		Тестиров ание(кон трольны й срез)	10	Тестирование подразумевает 6 вопросов. За прохождение тестиро - 90 % - 10 баллов; - 65 % - 5 баллов; - 50 % - 2 балла; - менее 50 % - балл не начисляется.

		Собесе	2	Собеседование предполагает организацию беседы преподавателя с - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения п - своевременность и эффективность использования наглядных пос - использование дополнительного материала; - рациональность использования времени, отведенного на задание 2 балла – студент умеет сопоставить полученную при подготовке к 1 балл - студент умеет применять полученную при подготовке к пр Если студент не владеет проблематикой практического занятия, не
4.	Базы данных	Лабора	3	Лабораторные работы выполняются по тематике практических за 3 балла – лабораторная работа выполнена в полном объеме, студен 2 балл – лабораторная работа выполнена, но имеет некоторые нет 1 балл - лабораторная работа в целом выполнена, однако в процес
		Собесе	2	Собеседование предполагает организацию беседы преподавателя с - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения п - своевременность и эффективность использования наглядных пос - использование дополнительного материала; - рациональность использования времени, отведенного на задание 2 балла – студент умеет сопоставить полученную при подготовке к 1 балл - студент умеет применять полученную при подготовке к пр Если студент не владеет проблематикой практического занятия, не
5.	Взаимодейс	Лабора	3	Лабораторные работы выполняются по тематике практических за 3 балла – лабораторная работа выполнена в полном объеме, студен 2 балла – лабораторная работа выполнена, но имеет некоторые нет 1 балл - лабораторная работа в целом выполнена, однако в процес
		Собесе	2	Собеседование предполагает организацию беседы преподавателя с - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения п - своевременность и эффективность использования наглядных пос - использование дополнительного материала; - рациональность использования времени, отведенного на задание 2 балла – студент умеет сопоставить полученную при подготовке к 1 балл - студент умеет применять полученную при подготовке к пр Если студент не владеет проблематикой практического занятия, не
6.	Основы кл	Лабора	3	Лабораторные работы выполняются по тематике практических за 3 балла – лабораторная работа выполнена в полном объеме, студен 2 балла – лабораторная работа выполнена, но имеет некоторые нет 1 балл - лабораторная работа в целом выполнена, однако в процес

		Собесе	2	Собеседование предполагает организацию беседы преподавателя с студентом. Критерии оценки: - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения проблемы; - своевременность и эффективность использования наглядных пособий; - использование дополнительного материала; - рациональность использования времени, отведенного на задание. 2 балла – студент умеет сопоставить полученную информацию при подготовке к занятию; 1 балл - студент умеет применять полученную информацию при подготовке к занятию. Если студент не владеет проблематикой практического занятия, не получает баллов.
7.	Регулярные	Лабораторные	3	Лабораторные работы выполняются по тематике практических занятий. 3 балла – лабораторная работа выполнена в полном объеме, студент умеет применять полученные знания; 2 балла – лабораторная работа выполнена, но имеет некоторые неточности; 1 балл - лабораторная работа в целом выполнена, однако в процессе выполнения студент не применяет полученные знания.
		Собесе	2	Собеседование предполагает организацию беседы преподавателя с студентом. Критерии оценки: - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения проблемы; - своевременность и эффективность использования наглядных пособий; - использование дополнительного материала; - рациональность использования времени, отведенного на задание. 2 балла – студент умеет сопоставить полученную информацию при подготовке к занятию; 1 балл - студент умеет применять полученную информацию при подготовке к занятию. Если студент не владеет проблематикой практического занятия, не получает баллов.
8.	Использование	Лабораторные	3	Лабораторные работы выполняются по тематике практических занятий. 3 балла – лабораторная работа выполнена в полном объеме, студент умеет применять полученные знания; 2 балла – лабораторная работа выполнена, но имеет некоторые неточности; 1 балл - лабораторная работа в целом выполнена, однако в процессе выполнения студент не применяет полученные знания.
		Собесе	2	Собеседование предполагает организацию беседы преподавателя с студентом. Критерии оценки: - правильность ответа по содержанию; - полнота и глубина ответа; - сознательность ответа; - логика изложения материала; - рациональность использованных приемов и способов решения проблемы; - своевременность и эффективность использования наглядных пособий; - использование дополнительного материала; - рациональность использования времени, отведенного на задание. 2 балла – студент умеет сопоставить полученную информацию при подготовке к занятию; 1 балл - студент умеет применять полученную информацию при подготовке к занятию. Если студент не владеет проблематикой практического занятия, не получает баллов.
9.	Посещаемость		10	10 баллов – стопроцентное посещение занятий студентом 7-9 баллов – посещаемость студента составляет не менее 80 % занятий; 4-6 баллов – посещаемость студента составляет не менее 50 % занятий; 1-3 балла – посещаемость студента составляет не менее 25 % занятий.
10.	Премияльные баллы		20	Дополнительные премиальные баллы могут быть начислены: - за проект, выполненный по заказу работодателя и реализованный; - 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 из области дисциплины – 10 баллов; - участие в выставке по тематике изучаемой дисциплины – 20 баллов; - публикация статьи по тематике изучаемой дисциплины в сборнике – 10 баллов.

11.	Ответ на экзамен	30	Оценка «удовлетворительно»- студент имеет достаточный минимум знаний по основным вопросам дисциплины; усвоение основной и дополнительной литературы, а также по основным вопросам, выходящим за пределы программы. Оценка «хорошо» – достаточно полные и систематизированные знания по основным вопросам дисциплины, а также по основным вопросам, выходящим за пределы программы. Оценка «отлично» – систематизированные и глубокие знания по основным вопросам дисциплины, а также по основным вопросам, выходящим за пределы программы.
12.	Индивидуальные задания	20	Провести анализ по одной статье из журналов по рекомендуемой литературе. Прохождение тестирования (90 вопросов) по всему курсу дисциплины.
13.	Итого за семестр	100	

Итоговая оценка по экзамену 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
85 - 100 баллов	Отлично
70 - 84 баллов	Хорошо
50 - 69 баллов	Удовлетворительно
Менее 50	Неудовлетворительно

4.2 Типовые оценочные средства текущего контроля

Лабораторная работа

Тема 1. Введение. Структура программы на С. Основные операторы языка С.

База данных. Общие понятия. Достоинства и недостатки

- СУБД. Комплекс языковых и программных средств
- Приложения для работы с базами данных
- Типы БД. Масштабирование
- Программное обеспечение для создания приложений баз данных
- Подключение к базе данных в RAD Delphi. Технология BDE
- Использование Database Desktop для создания баз данных.
- Подключение к базе данных в RAD Delphi. Технология ADO
- Подключение к базе данных в RAD Delphi. Технологии MDAC, OLE DB, ODBC
- Транзакции
- Провайдеры данных ADO.

Тема 2. Массивы. Указатели и ссылки. Указатели и массивы.

Задание 1: Составить алгоритм решения задачи с одномерным массивом согласно своему варианту и написать программу на языке С.

Задание 2: Составить алгоритм решения задачи с двумерным массивом согласно своему варианту и написать программу на языке С.

Составить алгоритм решения задачи с использованием указателей и динамических массивов в качестве аргументов функций согласно своему варианту и написать программу на языке С. В отчете предоставить блок-схемы всех функций, текст программы.

Тема 3. Символьная информация и строки. Функции в С. Типы данных, определяемые пользователем. Ввод-вывод в С.

Написать две программы согласно индивидуальному варианту. В случае, когда указано, что задается строка до точки, обрезать строку, отбросив все символы после точки, если точки нет, то считать, что строка ограничена точкой. Использовать функции библиотеки string.

Составить алгоритм решения задачи с использованием функции согласно своему варианту и написать программу на языке C. В отчете предоставить блок-схемы функции согласно задания и основной функции.

Написать программу на языке C решения задачи согласно своему варианту. В программе создать соответствующую заданию структуру, можно написать функции, реализующие операции со структурами. В программе можно использовать массив переменных созданного типа структуры.

Во всех заданиях по вариантам составить две программы. Первая программа должна формировать двоичный файл. Вторая программа должна считать данные из двоичного файла, выполнить соответствующие вычисления и записать их результаты в текстовый файл.

В программах необходимо использовать только динамические структуры. Согласно своему варианту выполнить следующие действия:

С помощью текстового редактора создать текстовый документ, в который занести исходную информацию. Рекомендуется вначале занести количество записей, а затем последовательно все записи, причем тестовые поля вводить в отдельной строке, а числовые – можно в отдельной строке или в одной через пробел или разделитель табуляцию.

Написать первую программу, которая считывает информацию из созданного текстового файла и записывает ее в двоичный файл.

Написать вторую программу, которая считывает информацию из двоичного файла, реализует поставленную задачу. После повторного чтения двоичного файла результат работы выводится в результирующий текстовый файл.

Все текстовые и двоичные файлы должны располагаться или в личной папке или в папке проекта.

Тема 4. Объекты и классы. Наследование.

Написать программу, в которой создаются и разрушаются объекты по вариантам, определенного пользователем класса. Выполнить исследование вызовов конструкторов и деструкторов.

Написать программу, в которой создается иерархия классов по вариантам. Включить полиморфные объекты в связанный список, используя статические компоненты класса. Показать использование виртуальных функций.

Тема 5. Перегрузка операций и преобразование типов.

Написать программу на языке C++, реализующую один из вариантов задания. В программе должны присутствовать как перегруженные операторы, так и пользовательские преобразования между типами данных.

Тема 6. Виртуальные и дружественные функции.

Определить и реализовать класс – абстрактный тип данных. Определить и реализовать операции над данными этого класса. Написать и выполнить программу полного тестирования этого класса.

Тема 7. Шаблоны и исключения.

Создать шаблон заданного класса и использовать его для данных различных типов.

Тема 8. Стандартная библиотека шаблонов.

- 1) Получить практические навыки создания объектов-групп и использования методов-итераторов.
- 2) Написать три программы с использованием STL. Первая и вторая программы должны демонстрировать работу с контейнерами STL, третья – использование алгоритмов STL.

Тема 9. Введение в PHP. Основы синтаксиса. Операторы PHP. Условные операторы. Циклы в PHP
Создание заготовки письма.

Тема 10. Обработка запросов с помощью PHP. Функции в PHP. Строки в PHP.

Создание формы для регистрации пользователей на сайте. 1. Отправка «универсального письма» всем зарегистрировавшимся.

Тема 11. Объекты и классы в PHP. Массивы в PHP. Работа с файловой системой

Автоматическая генерация по желанию пользователя представителей классов статей или личностей, а также их отображения на странице браузера. Загрузка файла на сервер с помощью web-интерфейса. Обработка массивов данных

Тема 12. Базы данных и СУБД. Введение в SQL

1. Проектирование базы данных. 2. Создание базы данных MySQL. 3. Установка соединения с базой данных. 4. Создание страницы для добавления записей базы данных

Тема 13. Взаимодействие PHP и MySQL.

Создание страницы для удаления и редактирования записей базы данных

Тема 14. Основы клиент-серверных технологий.

1. Типы серверов. 2. Обработка запросов с помощью PHP

Тема 15. Регулярные выражения

Авторизация пользователя с помощью механизма сессий.

Тема 16. Использование шаблонов в PHP. Взаимодействие PHP и XML. Работа с FTP средствами PHP.

Генерации списка документов по таблице базы данных

Установка расширения DOMXML

Собеседование

Тема 1. Введение. Структура программы на C. Основные операторы языка C.

1. Из каких частей состоит программа на C?
2. Чем определение отличается от объявления?
3. Перечислить этапы создания исполняемой программы на языке C.
4. Что такое препроцессор?
5. Что такое директива препроцессора? Привести примеры директив препроцессора.
6. Составить программу, которая печатает текст «Моя первая программа на C».
7. Из каких элементов состоит естественный язык? Что является аналогами этих элементов в C?
8. Что такое лексема? Привести примеры лексем в языке C.
9. Что такое идентификатор? Правила записи идентификаторов.
10. Что такое константа? Как константа обрабатывается компилятором?
11. Какие типы констант существуют в C. 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 shortint?
15. Чем отличаются типы данных int и unsignedint?
16. Перечислить все типы данных, которые существуют в C. 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. Массивы. Указатели и ссылки. Указатели и массивы.

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?
41. Что такое указатель?
42. Какой объем памяти занимает указатель?
43. Что является значением переменной-указателя?
44. Как проинициализировать указатель?
45. Что такое NULL?
46. Что такое указатель на void? Зачем нужны такие указатели?
47. Какие операции допустимы при работе с указателями?
48. Чем отличается унарная операция "&" от унарной "*" ?
49. Совместимость типов указателей.
50. Можно ли получить адрес указателя?
51. Можно ли указателю присвоить его же адрес?
52. Почему к указателю на void нельзя применить операцию разыменования?
53. Как работают операции инкремента и декремента, примененные к указателям?
54. Каков результат операции вычитания, примененной к указателям одного типа?
55. Какой спецификатор типа используется при выводе адреса на экран с помощью функции printf()?
56. В чем отличие записи (float *) а от (float) * а, если а – указатель на целое число?
57. В чем отличие записи *a++ от (*a)++, если а – некоторый указатель, отличный от void*?
58. Как описать указатель на начало массива?
59. Как описать указатель на указатель?
60. Когда и зачем может повторно использоваться операция разыменования?
61. Какие переменные называют динамическими? Чем они отличаются от статических переменных?
62. С какой целью используют динамические переменные?

63. Как обращаются к динамическим переменным?
64. Какие бывают указатели? Как их описать?
65. Каким образом можно выделять память для динамических переменных и освобождать её?
66. Как определить, выделена память или нет?
67. Почему нельзя забывать освобождать выделенную память?
68. Чему равно значение указателя после освобождения области памяти, на которую он указывал?
69. Как разместить в динамической памяти массив?
70. Какими способами можно разместить в динамической памяти матрицу?

Тема 3. Символьная информация и строки. Функции в С. Типы данных, определяемые пользователем. Ввод-вывод в С.

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 *stotcharst[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?
 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?
 21. Где может быть использована структура?
 22. Как называются элементы структуры?
 23. Как определить структуру?
 24. Может ли структура содержать элементы одного типа?

25. Как обратиться к полям структуры?
26. Как обратиться к элементу массива структур?
27. Какое ключевое слово начинает определение структуры?
28. Как обратиться к полям структуры через указатель на нее?
29. При помощи какой операции формируется доступ к полям структуры?
30. Какая разница между определениями структуры и массива?
31. Опишите объект, для которого надо определить структуру и определите ее.
32. Что такое файл с точки зрения информатики и вычислительной техники?
33. Что такое файл с точки зрения языка программирования C?
34. Какие виды файлов Вы знаете?
35. В чем отличие текстового файла от бинарного и от чего зависит наличие этого различия?
36. Что такое поток?
37. Что такое признак окончания файла? Для каких файлов он существует?
38. Как инициализировать поток?
39. Что такое структурный тип?
40. Какая функция отвечает за открытие потока? Какие у нее параметры?
41. Какие режимы открытия файла Вы знаете? В чем их отличия друг от друга?
42. Какие ошибки могут возникнуть при открытии файла?
43. В каком случае указатель на поток принимает значение NULL?
44. Какая функция отвечает за закрытие файла? Какие у нее параметры?
45. Зачем закрывать файл?
46. Сколько раз можно открыть файл в программе?
47. Какие функции используются для ввода-вывода данных при работе с текстовыми файлами?
48. Какие функции используются для ввода-вывода данных при работе с бинарными файлами?
49. В чем отличие функции `fprintf()` от функции `fread()`, а функции `fscanf()` от `fwrite()`?
50. Как определить текущую позицию указателя на поток?
51. Как переместить указатель на поток из текущей позиции в заданную?
52. За что отвечает константа EOF? Как и где ее можно использовать?
53. Какие основные функции для работы с файлами Вы знаете?
54. С помощью каких функций можно осуществлять побайтовую обработку файлов?
55. Для чего используются функции `rename()` и `remove()`? Нужно ли при работе с ними объявлять файл указатель?
56. Можно ли удалить или переименовать открытый файл?

Тема 4. Объекты и классы. Наследование.

1. Что такое класс, объект?
2. Что такое переменная экземпляра?
3. Какая разница между списком аргументов и списком параметров?
4. Что такое конструктор и как он определяется?
5. Что такое конструктор копирования и его использование?
6. Что такое наследование?
7. Какие классы называются базовыми, а какие производными?
8. Как объявить производный класс?
9. Для чего в базовом классе применяется ключевое слово `protected`?
10. Как работают конструкторы и деструкторы при создании производных объектов?
11. Как выполнить инициализацию базового класса при создании производного объекта?
12. Передаются ли по наследству производные члены-данные и функции через поколения?

Тема 5. Перегрузка операций и преобразование типов.

1. Для чего в C++ применяется перегрузка операций
2. Истинно ли следующее утверждение: операция \geq может быть перегружена?
3. Сколько аргументов требуется для определения перегруженной унарной операции?
4. Сколько аргументов требуется для определения перегруженной бинарной операции?
5. Чем отличается действие операции $++$ в префиксной форме от её действия в постфиксной форме?
6. Истинно ли следующее утверждение: перегруженная операция всегда требует на один аргумент чем количество операндов?
7. Когда перегружается операция арифметического присваивания, то результат
8. Передается объекту справа от операции
9. Передается объекту слева от операции
10. Передается объекту, вызвавшему операцию
11. Должен быть возвращен
12. Какой механизм преобразования от определенного пользователем класса к встроенному типу : использован в языке C++?
13. Какой механизм преобразования от встроенного типа данных к определенному пользователем использован в языке C++?
14. Истинно ли следующее утверждение: компилятор не выдаст сообщение об ошибке, если перегрузите операцию $*$ для выполнения деления?
15. Если объект `objA` принадлежит классу A, объект `objB` принадлежит классу B, и требуется записать поместив при этом функцию преобразования в класс A, то какую разновидность процес преобразования типа можно использовать?
16. Существуют ли операции, которые нельзя перегружать?
17. Что такое конструктор преобразования?
18. Для чего используется ключевое слово `explicit`?

Тема 6. Виртуальные и дружественные функции.

1. Различают 4 элемента определения функции: тип, имя, список параметров, тело. Какими элементами могут отличаться экземпляры одной виртуальной функции, находящиеся в разных производных классах
2. Какая функция называется другом класса?
3. В чем особенности передачи параметров дружественной функции при перегрузке унарных и бинарных операций?

Тема 7. Шаблоны и исключения.

1. Что такое шаблоны и с какой целью они используются?
2. Какого типа шаблоны используются в программах?
3. Как оформляются шаблоны функций?
4. Какие требования предъявляются к фактическим параметрам шаблонов?
5. Какие преимущества программы обеспечиваются при использовании шаблонов классов?

Тема 8. Стандартная библиотека шаблонов.

1. Какие есть контейнеры в C++?
2. За чем нужны контейнеры?
3. В чем отличие массивов от `vector/list`?
4. Что такое итераторы? Для чего они используются?
5. Как итерироваться по `list`, `map`, `vector`?

Тема 9. Введение в RHP. Основы синтаксиса. Операторы RHP. Условные операторы. Циклы в

1. Возможности RHP.

2. Области применения PHP.
3. Способы использования.
4. Установка и настройка программного обеспечения.
5. Разделение конструкций.
6. Комментарии.
7. Переменные, константы и операторы.
8. Вывод информации.
9. Типы данных.
10. Математические операторы и математические функции PHP.
11. Операторы увеличения и уменьшения в PHP.
12. Операторы сравнения PHP.
13. Оператор ELSE.
14. Оператор ELSEIF
15. Оператор SWITCH.
16. Цикл FOR.
17. Циклы WHILE.
18. Цикл DO... WHILE

Тема 10. Обработка запросов с помощью PHP. Функции в PHP. Строки в PHP.

1. Отправка данных с помощью HTML-формы.
2. Методы POST и GET.

Тема 11. Объекты и классы в PHP. Массивы в PHP. Работа с файловой системой

1. Понятия класса и объекта.
2. Определение и использование классов.

Конструкторы и деструкторы

1. Создание массивов в PHP (удаление, перебор, функции).
2. Сортировка массивов в PHP (навигация, сравнение).
3. Особенности работы с многомерными массивами.

1. Создание файлов.
2. Чтение данных из файла.
3. Удаление файла.

Проверка наличия файла на сервере

Тема 12. Базы данных и СУБД. Введение в SQL

1. Базы данных: основные понятия.

Создание и удаление базы данных MySQL

Тема 13. Взаимодействие PHP и MySQL.

1. Способы взаимодействия PHP и СУБД MySQL.
2. Функции отправки запросов.

Тема 14. Основы клиент-серверных технологий.

Обработка запросов с помощью PHP.

Тема 15. Регулярные выражения

1. Понятие регулярного выражения.
2. Квантификаторы
 1. Регистрация переменных сессии.

2. Уничтожение сессии.
3. Настройка сессий в файлах `php.ini`.

Тема 16. Использование шаблонов в PHP. Взаимодействие PHP и XML. Работа с FTP средствами

1. Использование шаблонов в PHP.
1. XML: основные понятия.
2. Взаимодействие PHP и XML посредством DOMXML.
3. Функции `ftp_connect()`, `ftp_get()`, `ftp_login()`, `ftp_quit()`

Тестирование

Тема 2. Массивы. Указатели и ссылки. Указатели и массивы.

1. Укажите оператор выбора в языке C ++.
 - A. CASE
 - B. choice
 - C. switch ... case +
 - D. default
2. Укажите, в котором выражении правильно применяется операция в 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.
7. 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+

8. 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 '.'

9. 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. +

10. 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;

Тема 3. Символьная информация и строки. Функции в C. Типы данных, определяемые пользователем. Ввод-вывод в C.

1. 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.

2. Укажите правильный вариант объявления указателя в C ++.

A. int x;

B. int & x;

C. int * x; +

D. ptrx;

3. Укажите, в котором 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;

4. 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.

5. 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);` +
6. 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];`
7. Укажите запись экранированного символа языке C ++.
- A. 'F'
 B. «\ 022»
 C. 'Ю'
 D. '\ t' +
8. 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.
9. В одном 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];`
10. В программе на языке C ++ объявлены такие переменные `int x, y`; Выражение позволяет вычислить остаток от деления этих переменных?
- A. `x% y` +
 B. `x div y`
 C. `x mod y`
 D. `x / y`

Тема 4. Объекты и классы. Наследование.

1. С помощью чего реализуется принцип полиморфизма в C ++?
- A. наличия множественного наследования.
 B. наличия виртуальных методов. +
 C. Использование виртуального наследования.
 D. наличия абстрактных классов.
2. В программе описано класс и объект `class A {public: int a, b, c; }; A * obj`; Как обратиться к атрибуту `c`?
- A. `obj.c`
 B. `obj-> c` +
 C. `obj A -> -> c`
 D. `obj->A.c`
3. Какая из перечисленных функций не может быть конструктором?
- A. `void String ()` +
 B. `String ();`
 C. `String (String & s)`
 D. `String (const int a)`
4. Отметьте правильное утверждение для абстрактного класса для языка C ++.

- A. Класс, у которого все методы чисто виртуальные, называется абстрактным.
- B. Абстрактный базовый класс навязывает определенный интерфейс всем производным из него классам.
- C. Невозможно создать объект абстрактного класса. +
- D. В абстрактном классе не описываются методы вообще.
5. Если в программе на языке C ++ в производном классе переопределена операция new то ...
- A. все объекты этого класса и все объекты классов, выведенных из него, будут использовать эту операцию независимо от зоны видимости, в которой она переопределена.
- B. производные от этого класса могут использовать глобальную операцию применив операцию базовый_класс :: new. +
- C. операцию new нельзя переопределить.
- D. в любом случае эта операция будет доступна только в пределах класса-потомка.
6. Какой из перечисленных методов может быть конструктором для класса String в языке C ++?
- A. String * String ();
- B. void String ();
- C. String (String & s); +
- D. const String (int a);
7. Какая функция, не будучи компонентом класса, имеет доступ к его защищенным и внутренним компонентам?
- A. Шаблонная.
- B. Полиморфная.
- C. Дружеская. +
- D. Статическая.
8. Вызовет данный код ошибку компиляции?
- ```
class Rectangle public: int a, b; int sum (); int square (); ~ Rect (); };
```
- A. Ошибки нет, все записано верно.
- B. Ошибка является: имя деструктора должно совпадать с именем класса. +
- C. Ошибка является: имя деструктора не может начинаться с маленькой буквы.
- D. Ошибка является: никакой идентификатор в C ++ не может начинаться со знака «~».
9. Укажите правильное объявление виртуального метода, который принимает одно целочисленное значение и возвращает void.
- A. virtual void SomeFunction (int x); +
- B. void SomeFunction (int x) virtual;
- C. virtual SomeFunction (int x);
- D. virtual void SomeFunction (int \* x);
10. Укажите правильное использование оператора friend.
- A. class A {int \_friendCountPass (); private: short i;};
- B. class A {public: friend int H :: CountPass (); private: short i;};
- C. class A {public: int A1 :: CountPass (); friend: short i;};
- D. class A {public: friend int H :: q; short i;};

#### Тема 9. Введение в РНР. Основы синтаксиса. Операторы РНР. Условные операторы. Циклы в РНР

1. Принцип объектно-ориентированного программирования, заключается в объединении атрибутов и методов объекта с целью обеспечения сохранности данных, называется:
- A. Наследование.
- B. Сочетание.
- C. Инициализация.
- D. Инкапсуляция. +
2. В программе описано абстрактный класс A и производный от этого класса класс A1. Какой из записей заведомо неверный?

A. `A * a = new A; +`

B. `A1 a1;`

C. `A1 a1; A & a = a1;`

D. `A1 a1; A1 a2;`

3. Выберите верное утверждение о деструктор класса в C++.

A. Деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить.

B. Деструктор принимает в качестве параметра указатель `this`.

C. Деструктор не содержит параметров. +

D. Деструктор принимает в качестве параметра тот объект, который нужно уничтожить.

4. Укажите правильный вариант доступа к членам объектов (язык C++), описанных следующим образом: `class my {char s;public: double Z; int f (int c, int d) {return c + d;}; } T1, T2;`

A. `T1.Z = 23.1; +`

B. `T2-> f (2,1);`

C. `T1.s = '#';`

D. `my.T2-> s = 'L';`

5. Какой из вариантов записи абстрактного класса в C++ является правильным?

A. `abstract class A {virtual int f () = 0;};`

B. `class A {virtual int f () = 0;}; +`

C. `class A {virtual int f () = 0;} abstract;`

D. `class A {virtual int f ();};`

## Тема 11. Объекты и классы в PHP. Массивы в PHP. Работа с файловой системой

1. Для чего НЕ может использоваться PHP?

1) для создания операционных систем;

2) для создания скриптов, выполняющихся в командной строке;

3) для создания клиентских GUI-приложений.

2. В чём разница между `echo` и `print`?

1) `print` можно использовать как часть выражения, а `echo` нет;

2) `echo` можно использовать как часть выражения, а `print` нет;

3) между этими функциями нет никакой разницы.

3. Какие из перечисленных тегов непригодны для открытия и закрытия PHP блока?

1) `<% %>;`

2) `<? ?>;`

3) <?= ?>;

4) <! !>;

5) <?php ?>.

4. При вызове функции обязательно указывать:

1) имя функции;

2) список всех параметров функции;

3) конструкцию return.

5. Какие служебные символы используются для обозначения начала и конца блока кода?

1) <>

2) ( )

3) { }

4) Beginend

6. Какими знаками заканчивается большинство строк кода в Си++?

1) . (точка)

2) , (запятая)

3) ; (точка с запятой)

4) : (двоеточие)

4.3 Промежуточная аттестация по дисциплине проводится в форме зачета, экзамена

### **Типовые вопросы зачета (ОПК-3, ПК-2)**

1. Введение.

2. Структура программы на С.

3. Основные операторы языка С.



4. Массивы.
5. Указатели и ссылки.
6. Указатели и массивы
7. Символьная информация и строки.
8. Функции в С.
9. Типы данных, определяемые пользователем.
10. Ввод-вывод в С
11. Объекты и классы.
12. Наследование
13. Перегрузка операций и преобразование типов
14. Виртуальные и дружественные функции
15. Шаблоны и исключения
16. Стандартная библиотека шаблонов

### **Типовые задания для зачета (ОПК-3, ПК-2)**

1. Последовательный контейнер — вектор. Присваивание векторов. Напишите законченную программу, в которой с помощью подходящих конструкторов создайте три вектора  $v1$ ,  $v2$ ,  $v3$  с элементами целого типа, размерами соответственно 5, 7, 6 и одинаковыми значениями элементов соответственно 1, 2, 3. Выведите на экран размеры векторов, значения их элементов и выполните присваивание  $v3=v2-v1$ . После этого вновь выведите на экран размеры векторов и значения их элементов. Используйте только средства стандартной библиотеки языка C++ (поточный ввод-вывод, класс `vector`).
2. Создайте программу с классом `Student`, который включает в себя следующие данные-элементы о студентах университета: - ФИО; - год поступления; - курс; - номер группы; - размер стипендии; - оценки по  $N$  предметам. Номер для каждого создаваемого студента должен запрашиваться, а массив оценок за последнюю сессию создается в динамической памяти операцией `new`. В состав класса входят следующие функции-члены класса: - конструктор с параметрами; - деструктор; - функция просмотра текущего состояния объектов `PRINT ()`; - функция-друг, которая вычисляет средний балл и выводит ФИО студента, у которого он максимальный; - виртуальная функция просмотра текущего объекта `print ()`. Производный класс `Student_1` содержит следующие элементы: - общественная работа; - процент надбавки к стипендии; - переопределенную функцию вывода данных об общественниках.
3. Разработать класс, объект которого реализует «пользовательский» тип данных. Обеспечить его произвольную размерность за счет использования в объекте динамических структур данных. Разработать необходимые конструкторы, деструктор, конструктор копирования, а также методы, обеспечивающие изменение отдельных составных частей объекта (например, коэффициентов полинома) и вывод его содержимого.

4. Опишите класс `fraction`, у которого поля `x` и `y` задают числитель и знаменатель обыкновенной дроби. Перегрузите для этого класса арифметические операции сложения, вычитания, умножения и деления так, чтобы они могли оперировать как с объектами класса, так и с числами (то есть выполнять например, не только действие  $3/4 + 2/5$ , но и  $1/2 + 4$  или  $2 * 5/6$ ). Также перегрузите операции сравнения `==` и `!=`. Продемонстрируйте работу класса.
5. Опишите классы `PointXY` и `PointPolar`, объекты которых задают декартовы и полярные координаты точки на плоскости. Перегрузите для этих классов операции сложения, вычитания и умножения как скалярного произведения, так, чтобы в них могли участвовать объекты как одного, так и обоих классов. Кроме того, задайте функцию преобразования одного класса в другой (для обоих классов).

### Типовые вопросы экзамена (ОПК-3, ПК-2)

1. Переменные в PHP.
2. Типы данных PHP.
3. Операторы PHP (Математические операторы и математические функции, строковые операторы).
4. Операторы сравнения PHP.
5. Логические операторы PHP.
6. Оператор ELSE в PHP (ELSEIF, SWITCH).
7. Цикл FOR в PHP.
8. Циклы WHILE в PHP.
9. Цикл DO... WHILE в PHP.
10. Цикл FOREACH в PHP.
11. Функции для обработки строк в PHP (Использование строковых функций в PHP).
12. Создание массивов в PHP (удаление, перебор, функции).
13. Сортировка массивов в PHP (навигация, сравнение).
14. Функции в PHP, их синтаксис, создание и вызов.
15. Использование PHP в связке с HTML.
16. Получаем данные от элементов формы с помощью PHP.
17. Проверка данных формы с помощью PHP.
18. Взаимодействие PHP и MySQL (`mysql_connect`, `mysqli_connect`, `mysql_close`, `mysqli_close`, `mysql_query`, `mysqli_query`).
19. Взаимодействие PHP и MySQL (`mysql_fetch_array`).
20. Сессии в PHP (пример открытия и закрытия).

### Типовые задания для экзамена (ОПК-3, ПК-2)

1. Установка и настройка ПО.
2. Создание формы для регистрации пользователей на сайте.
3. Обработка массивов данных.
4. Создание базы данных MySQL. Установка соединения с базой данных.
5. Создание страницы для добавления записей базы данных.
6. Загрузка файла на сервер с помощью web-интерфейса.

#### 4.4. Шкала оценивания промежуточной аттестации

##### Зачет

| Оценка | Компет | Дескрипторы (уровни) – основные признаки |
|--------|--------|------------------------------------------|
|--------|--------|------------------------------------------|

|                                 |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| «зачтено»<br>(50 - 100 баллов)  | ОПК-3 | Демонстрирует высокий уровень знаний основных теоретических положений изучаемой дисциплины. Свободно ориентируется в схемотехнике большинства электронных устройств. Демонстрирует знание основ информационной и библиографической культуры. Способен продемонстрировать решение стандартных задач профессиональной деятельности с применением ИКТ.<br>Практическое задание выполнено полностью.<br>Ответ построен логично, материал излагается четко, ясно, хорошим языком, аргументировано. На вопросы отвечает кратко, аргументировано, уверенно, по существу. |
|                                 | ПК-2  | Знает платформы для создания АРМ разработки программ обеспечения на популярных языках высокого уровня. Умеет методы абстрагирования и управления современных языков программирования для описания и решения конкретных прикладных задач. Владеет методами программирования, разработки эффективных алгоритмов решения прикладных задач.                                                                                                                                                                                                                           |
| «не зачтено»<br>(0 - 49 баллов) | ОПК-3 | Демонстрирует не достаточный уровень знаний основных теоретических положений изучаемой дисциплины. Не ориентируется в схемотехнике основных электронных устройств. Демонстрируется не достаточное знание основ информационной и библиографической культуры. Не способен продемонстрировать решение стандартных задач профессиональной деятельности с применением ИКТ.<br>Практическое задание не выполнено.<br>Неуверенно и логически непоследовательно излагает материал. Неправильно отвечает на поставленные вопросы или затрудняется с ответом                |
|                                 | ПК-2  | Не знает стандартные и пользовательские типы данных и метод обработки. Не умеет разработать алгоритм и реализовать программу, выбрав наиболее подходящий метод и язык программирования. Владеет разработкой алгоритмов, используя общие схемы, методы, приемы построения алгоритмов.                                                                                                                                                                                                                                                                              |

### Экзамен

| Оценка                         | Компет | Дескрипторы (уровни) – основные признаки                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| «отлично»<br>(85 - 100 баллов) | ОПК-3  | Демонстрирует высокий уровень знаний основных теоретических положений изучаемой дисциплины. Свободно ориентируется в схемотехнике большинства электронных устройств. Демонстрирует знание основ информационной и библиографической культуры. Способен продемонстрировать решение стандартных задач профессиональной деятельности с применением ИКТ. Практическое задание выполнено полностью. Ответ построен логично, материал излагается четко, ясно, хорошим языком, аргументировано. На вопросы отвечает кратко, аргументировано, уверенно, по существу. |

|                                            |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                            | ПК-2  | Знает платформы для создания АРМ разработки программ обеспечения на популярных языках высокого уровня. Умеет использовать методы абстрагирования и управления современными языками программирования для описания и решения конкретных прикладных задач. Владеет методами программирования, разработкой эффективных алгоритмов решения прикладных задач.                                                                                                                                                                                                                   |
| «хорошо»<br>(70 - 84 балла)                | ОПК-3 | Демонстрирует достаточный уровень знаний основных теоретических положений изучаемой дисциплины. Достаточно свободно ориентируется в схемотехнике основных электронных устройств. Демонстрирует достаточные знания основ информационной и библиографической культуры. Способен продемонстрировать решение стандартных задач профессиональной деятельности с применением ИКТ. Практическое задание выполнено полностью с незначительными недочетами.                                                                                                                        |
|                                            | ПК-2  | Знает принципы структурного и модульного программирования, принципы разработки сложных программных систем, в том числе правила разработки интерфейса. Умеет строить формальную модель системы (подсистемы) по ее описанию в терминах предметной области. Владеет основными методами разработки машинных алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов.                                                                                                                                                          |
| «удовлетворительно»<br>(50 - 69 баллов)    | ОПК-3 | Демонстрирует не достаточный уровень знаний основных теоретических положений изучаемой дисциплины. Слабо ориентируется в схемотехнике основных электронных устройств. Демонстрируется недостаточное знание основ информационной и библиографической культуры. Не способен продемонстрировать решение стандартных задач профессиональной деятельности с применением ИКТ. Практическое задание выполнено не полностью. Ответ не всегда логично выстроен, материал излагается без применения научной терминологии. Вопросы, задаваемые преподавателем, вызывают затруднения. |
|                                            | ПК-2  | Знает стандартные и пользовательские типы данных и методы обработки. Умеет разработать алгоритм и реализовать программу, выбрав наиболее подходящий метод и язык программирования. Владеет разработкой алгоритмов, используя общие схемы, методы и приемы построения алгоритмов.                                                                                                                                                                                                                                                                                          |
| «неудовлетворительно»<br>(менее 50 баллов) | ОПК-3 | Демонстрирует не достаточный уровень знаний основных теоретических положений изучаемой дисциплины. Не ориентируется в схемотехнике основных электронных устройств. Демонстрируется не достаточное знание основ информационной и библиографической культуры. Не способен продемонстрировать решение стандартных задач профессиональной деятельности с применением ИКТ. Практическое задание не выполнено. Неуверенно и логично, непоследовательно излагает материал. Неправильно отвечает на поставленные вопросы или затрудняется с ответом.                              |

|  |      |                                                                                                                                                                                                                                                                                      |
|--|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | ПК-2 | Не знает стандартные и пользовательские типы данных и метод обработки. Не умеет разработать алгоритм и реализовать программу, выбрав наиболее подходящий метод и язык программирования. Владеет разработкой алгоритмов, используя общие схемы, методы, приемы построения алгоритмов. |
|--|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## 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 с содержанием рабочей программы дисциплины (РПД), которая определяет содержание, объем, а также порядок изучения 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 информационно-справочные системы».

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

В разделе «Учебно-методическое и информационное обеспечение дисциплины» указана рекомендованная 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 лекционный материал и дополнить его с учетом рекомендованной дополнительной литературы;
- при самостоятельном изучении теоретической темы сделать конспект, используя рекомендованные 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, необходима при подготовке к устному опросу на семинарских занятиях, к дебатам, тестированию, экзамену. Она включает проработку лекционного материала и рекомендованных источников и литературы по темам лекций.

Конспект лекции должен содержать реферативную запись основных вопросов лекции, в том числе с вопросами, размещенными в системе MOODLe презентации, основных источников и литературы по темам, выходящим на каждый вопрос. Конспект может быть выполнен в рамках распечатки выдачи презентаций лекций на отдельной тетради по предмету. Он должен быть аккуратным, хорошо читаемым, не содержать лишнюю 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 опосредованным (содержать лишь основные ключевые позиции), но при этом позволяющим дать полный ответ по вопросу. Он 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 практики занятия с целью более обстоятельного выявления их знаний по определенному разделу, теме, проблем. 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: [Изд-во ТГУ], 2011

### 6.2 Дополнительная литература:

1. Андреев А.А. Программирование на языке Delphi : курс лекций. - Тамбов: [Изд-во ТГУ], 2009. - 254 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. Бобровский С.И. Delphi 7 : Учебный курс. - СПб.: Питер, 2005. - 735 с.

### 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](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 программное обеспечение:

Операционная система "Альт Образование"

Microsoft Windows 10

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.prilib.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](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.