

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

УТВЕРЖДАЮ:
Директор института



Н. Л. Королева
«21» июня 2023 г.

РАБОЧАЯ ПРОГРАММА

по дисциплине Б1.О.05 Программирование

Направление подготовки/специальность: 03.03.02 - Физика

Профиль/направленность/специализация: Фундаментальная физика

Уровень высшего образования: бакалавриат

Квалификация: Бакалавр

год набора: 2023

Тамбов, 2023

Автор программы:

Кандидат физико-математических наук, доцент Хлебников Владимир Викторович

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 03.03.02 - Физика (уровень бакалавриата) (приказ Министерства науки и высшего образования РФ от «07» августа 2020 г. № 891).

Рабочая программа принята на заседании Кафедры математического моделирования и информационных технологий «16» июня 2023 г. Протокол № 15

Рассмотрена и одобрена на заседании Ученого совета Института математики, физики и информационных технологий, Протокол от «21» июня 2023 г. № 3.

СОДЕРЖАНИЕ

1. Цели и задачи дисциплины.....	4
2. Место дисциплины в структуре ОП Бакалавриата.....	5
3. Объем и содержание дисциплины.....	5
4. Контроль знаний обучающихся и типовые оценочные средства.....	7
5. Методические указания для обучающихся по освоению дисциплины (модуля).....	24
6. Учебно-методическое и информационное обеспечение дисциплины.....	26
7. Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы.....	26

1. Цели и задачи дисциплины

1.1 Цель дисциплины – формирование компетенций:

ПК-3 Способен осуществлять действия по обработке и анализу научно-технической информации и результатов исследований в области физики при планировании и организации научных исследований и формировании отчетной документации

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

- научно-исследовательский
- педагогический

1.3 Дисциплина ориентирована на подготовку обучающихся к профессиональной деятельности в сферах: 01 Образование и наука (в сферах: реализации образовательных программ среднего общего образования, среднего профессионального образования, высшего образования и дополнительных профессиональных программ; научных исследований и научно-конструкторских разработок), 40 Сквозные виды деятельности в промышленности (в сферах: фундаментальных основ физики живых систем и физико-химической биологии, применения диагностического и лечебного оборудования, участия в инновационных и опытно-конструкторских разработках; эксплуатации электронных приборов и систем различного назначения; мониторинга параметров материалов; мониторинга состояния окружающей среды)

1.4 В результате освоения дисциплины у обучающихся должны быть сформированы:

Обобщенные трудовые функции / трудовые функции / трудовые или профессиональные действия (при наличии профстандарта)	Код и наименование компетенции ФГОС ВО, необходимой для формирования трудового или профессионального действия	Индикаторы достижения компетенций
	ПК-3 Способен осуществлять действия по обработке и анализу научно-технической информации и результатов исследований в области физики при планировании и организации научных исследований и формировании отчетной документации	Осуществляет действия по обработке и анализу научно-технической информации и результатов исследований в области физики при планировании и организации научных исследований и формировании отчетной документации с использованием навыков программирования

1.5 Согласование междисциплинарных связей дисциплин, обеспечивающих освоение компетенций:

ПК-3 Способен осуществлять действия по обработке и анализу научно-технической информации и результатов исследований в области физики при планировании и организации научных исследований и формировании отчетной документации

№ п/п	Наименование дисциплин, определяющих междисциплинарные связи	Форма обучения
		Очная (семестр)
		4

1	Избранные вопросы математической физики	+
---	---	---

2. Место дисциплины в структуре ОП бакалавриата:

Дисциплина «Программирование» относится к части, формируемой участниками образовательных отношений, учебного плана ОП по направлению подготовки 03.03.02 - Физика.

Дисциплина «Программирование» изучается в 2 семестре.

3. Объем и содержание дисциплины

3.1. Объем дисциплины: 3 з.е.

Очная: 3 з.е.

Вид учебной работы	Очная (всего часов)
Общая трудоёмкость дисциплины	108
Контактная работа	48
Лекции (Лекции)	16
Лабораторные (Лаб. раб.)	32
Самостоятельная работа (СР)	60
Зачет	-

3.2. Содержание курса:

№ темы	Название раздела/темы	Вид учебной работы, час.			Формы текущего контроля
		Лек ции	Лаб · раб.	СР	
		О	О	О	
2 семестр					
1	Введение	2	4	12	Собеседование
2	Типы данных и операции	2	4	12	Собеседование
3	Процедурное программирование	4	8	12	Собеседование; Тестирование
4	Классы	4	8	12	Собеседование
5	Абстракция данных	4	8	12	Собеседование; Тестирование

Тема 1. Введение (ПК-3)

Лекция.

Язык программирования C++ . Парадигмы программирования.

Введение в язык C++. История возникновения языка, основные понятия програм-мирования. Понятие парадигм программирования. Структура парадигм. Взаимосвязь па-радигм и модульного программирования.

Тема 2. Типы данных и операции (ПК-3)

Лекция.

Числовые типы. Числовые типы в арифметических операторах (*/+/-). Унарные и би-нарные операторы. Целые типы данных (char, short, int, long). Скалярные типы. Операторы сравнения и логические операторы. Понятие скалярного типа данных. Аналогия скалярного типа с булевым типом. Операторы сравнения. Неабстрактные операции Понятие неабстрактной операции. Методы использования неабстрактной операции. Понятие хэш-функции. Типы определяемые пользователем. Понятие типа определяемого пользователем. Определение класса пользователя. Понятие операторной функции. Методы применения типов определяемых пользователем. Ссылки. Понятие ссылки. Классификация ссылочных типов. Методы применения ссылочных типов. Параметры ссылочного типа. Методы применения ссылочных параметров. Константы .Понятие константы. Методы использования констант. Объявление указателя при использовании констант и ссылочных параметров.

Тема 3. Процедурное программирование (ПК-3)

Лекция.

Функции как модули. Понятие функции, определение модуля, процедуры. Входные и выходные данные функционального модуля. Функциональная декомпозиция. Понятие функциональной декомпозиции. Метод конструирования. Формирования структуры дан-ных. Файловая организация. Объявление заголовочного файла. Функции и глобальные объекты. Объявление вспомогательных функций. . Структурное программирование. При-емы структурного программирования. методы данной технологий. Применение управля-ющих операторов. Перегружаемые и подставляемые функции. Введение перегружаемых и подставляемых функций. Внедрение в программный продукт. Аргументы и возвращае-мые значения. Понятия аргумента и возвращаемой функции. Применение ссылочных, формальных параметров и фактических аргументов в процедурном программировании.

Тема 4. Классы (ПК-3)

Лекция.

Классовые типы. Сложные структуры данных. Функции элементов. Операторные функции. Компоненты данных. Определение классового типа, Доступ к компонентам класса. Объекты объединенного типа. Функциональные компоненты. Понятие функцио-нальной компоненты, применение функционального элемента. Области видимости клас-са. Операторные функции. Предопределенные в языке операторы. Синтаксис их вызова, Взаимосвязи предопределенных операторов. Понятие операторной функции. Примене-ние операторной функции, синтаксис вызова. . Защита доступа и дружественные функ-ции. Функции, объявленные дружественными в определении класса. Доступность эле-ментов. Три уровня защиты. Союзы элементов. Структуры, как другой тип классов. Инициализация и преобразования. Использование конструкторов. Применение классо-вых объектов, являющихся компонентами других классовых объектов. Инициализация конструктора, аргументы для конструктора. Преобразование по типу вызова функции. Указатели на компоненты класса. Элементы класса. Модификатор типа для индикации указателя. Спецификация типа указателя на функциональный элемент.

Тема 5. Абстракция данных (ПК-3)

Лекция.

Комплексные числа. Класс комплексных чисел как абстрактный тип данных. Определе-ние абстрактного интерфейса для пользователя, семантика типа. Применение дополни-тельной семантики типа для комплексных чисел. Строки. Модифицированная версия типа данных «Строки», как абстракция данных. Инициализация типа STRING.Симуляция поведение встроенного оператора присваивания. Разница в семан-тике присваивания инициализации. Упорядоченные последовательности. Два основных направления абстракции данных. Использование общедоступного интерфейса абстракт-ного типа данных как договоренность между реализатором и пользователем типа. Общ-ность. Упорядоченная последовательность целых. Концепция родовых и параметризо-ванных типов. . Абстракция управления.

Функция как композиция двух различных концепций. Отслеживание любой струк-туры данных. Приватность скрытой реализации. Центральные идеи абстракции управле-ния.

4. Контроль знаний обучающихся и типовые оценочные средства

4.1. Распределение баллов:

2 семестр

- текущий контроль – 60 баллов
- контрольные срезы – 2 среза по 10 баллов каждый

Распределение баллов по заданиям:

№ темы	Название темы / вид учебной работы	Формы текущего контроля / срезы	Мак. кол-во баллов	Методика проведения занятия и оценки
1.	Введение	Собеседование	12	12 баллов начисляется студенту, который демонстрирует уверенное владение теоретическим материалом, легко ориентируется в методах и подходах к решению задач, без недочетов, абсолютно правильно выполнил решение проверочной задачи. 5 баллов – студент хорошо ориентируется в теоретических вопросах, и демонстрирует устойчивые навыки в решении задач, выполнил решение проверочной задачи. 2 балла – демонстрирует слабое понимание теоретического материала, не способен применить теоретические навыки в решении практических задач, неудовлетворительно выполнил решение проверочной задачи.
2.	Типы данных и операции	Собеседование	12	12 баллов начисляется студенту, который демонстрирует уверенное владение теоретическим материалом, легко ориентируется в методах и подходах к решению задач, без недочетов, абсолютно правильно выполнил решение проверочной задачи. 5 баллов – студент хорошо ориентируется в теоретических вопросах, и демонстрирует устойчивые навыки в решении задач, выполнил решение проверочной задачи. 2 балла – демонстрирует слабое понимание теоретического материала, не способен применить теоретические навыки в решении практических задач, неудовлетворительно выполнил решение проверочной задачи.
3.	Процедурное программирование	Собеседование	12	12 баллов начисляется студенту, который демонстрирует уверенное владение теоретическим материалом, легко ориентируется в методах и подходах к решению задач, без недочетов, абсолютно правильно выполнил решение проверочной задачи. 5 баллов – студент хорошо ориентируется в теоретических вопросах, и демонстрирует устойчивые навыки в решении задач, выполнил решение проверочной задачи. 2 балла – демонстрирует слабое понимание теоретического материала, не способен применить теоретические навыки в решении практических задач, неудовлетворительно выполнил решение проверочной задачи.
		Тестирование(контрольный срез)	10	Тестирование проводится в виде теста из 10 вопросов. За верный ответ на каждый вопрос студент получает 1 балл

4.	Классы	Собеседование	12	12 баллов начисляется студенту, который демонстрирует уверенное владение теоретическим материалом, легко ориентируется в методах и подходах к решению задач, без недочетов, абсолютно правильно выполнил решение проверочной задачи. 5 баллов – студент хорошо ориентируется в теоретических вопросах, и демонстрирует устойчивые навыки в решении задач, выполнил решение проверочной задачи. 2 балла – демонстрирует слабое понимание теоретического материала, не способен применить теоретические навыки в решении практических задач, неудовлетворительно выполнил решение проверочной задачи.
5.	Абстракция данных	Собеседование	12	12 баллов начисляется студенту, который демонстрирует уверенное владение теоретическим материалом, легко ориентируется в методах и подходах к решению задач, без недочетов, абсолютно правильно выполнил решение проверочной задачи. 5 баллов – студент хорошо ориентируется в теоретических вопросах, и демонстрирует устойчивые навыки в решении задач, выполнил решение проверочной задачи. 2 балла – демонстрирует слабое понимание теоретического материала, не способен применить теоретические навыки в решении практических задач, неудовлетворительно выполнил решение проверочной задачи.
		Тестирование(контрольный срез)	10	Тестирование проводится в виде теста из 10 вопросов. За верный ответ на каждый вопрос студент получает 1 балл
6.	Итого за семестр		80	

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

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

4.2 Типовые оценочные средства текущего контроля

Собеседование

Тема 1. Введение

Вопросы собеседования

1. Привести классификацию языков программирования.
2. Рассказать об эволюции языков программирования.
3. Привести примеры применения языков программирования высокого уровня.
4. Что такое уровень языка программирования?
5. Какие у машинных языков достоинства и недостатки?
6. В чём преимущества алгоритмических языков перед машинными?
7. Какие компоненты образуют алгоритмический язык?
8. Какие понятия используют алгоритмические языки?
9. В какой последовательности возникали известные вам языки программирования?
10. Что такое транслятор, компилятор, интерпретатор?

11. Что такое тип данных?
12. Какие типы данных относятся к простым?
13. Что понимают под языками программирования?
14. Перечислите первые алгоритмические языки.
15. Перечислите фундаментальные структуры данных.
16. Чем отличается компилятор от интерпретатора?
17. Что понимают под интегрированной средой программирования?
18. На каких принципах основывается структурное программирование?
19. Какие основные базовые конструкции структурного программирования?
20. Озвучьте основные принципы объектно-ориентированного программирования.
21. Что такое объект?
22. Как характеризуется объект?
23. Что такое класс?
24. Объясните связь между объектом и классом.
25. Что такое инкапсуляция, наследование, полиморфизм в объектно-ориентированном программировании?
26. Какие символы могут содержать имена переменных?
27. Какие типы переменных допустимы на языке C?
28. Что такое оператор?
29. Какие типы операторов допустимы на C?
30. Для чего используются комментарии в программе?
31. Перечислите основные логические операции.
32. Какие циклы можно организовать на языке C?
33. Формат описания переменных. Привести пример.
34. Запись целых констант.
35. Выражения.
36. Арифметические операции.
37. Операции присваивания.
38. Приоритеты логических операций.
39. Структура простейшей программы на C.
40. Напишите формат и пример условного оператора в полной форме.
41. Напишите формат и пример условного оператора в краткой форме.
42. Формат оператора цикла с параметром, постусловием, предусловием.
43. Формат описания массива.

Тема 2. Типы данных и операции

Вопросы собеседования

1. Привести классификацию языков программирования.
2. Рассказать об эволюции языков программирования.
3. Привести примеры применения языков программирования высокого уровня.
4. Что такое уровень языка программирования?
5. Какие у машинных языков достоинства и недостатки?
6. В чём преимущества алгоритмических языков перед машинными?
7. Какие компоненты образуют алгоритмический язык?
8. Какие понятия используют алгоритмические языки?
9. В какой последовательности возникали известные вам языки программирования?
10. Что такое транслятор, компилятор, интерпретатор?
11. Что такое тип данных?
12. Какие типы данных относятся к простым?

13. Что понимают под языками программирования?
14. Перечислите первые алгоритмические языки.
15. Перечислите фундаментальные структуры данных.
16. Чем отличается компилятор от интерпретатора?
17. Что понимают под интегрированной средой программирования?
18. На каких принципах основывается структурное программирование?
19. Какие основные базовые конструкции структурного программирования?
20. Озвучьте основные принципы объектно-ориентированного программирования.
21. Что такое объект?
22. Как характеризуется объект?
23. Что такое класс?
24. Объясните связь между объектом и классом.
25. Что такое инкапсуляция, наследование, полиморфизм в объектно-ориентированном программировании?
26. Какие символы могут содержать имена переменных?
27. Какие типы переменных допустимы на языке C?
28. Что такое оператор?
29. Какие типы операторов допустимы на C?
30. Для чего используются комментарии в программе?
31. Перечислите основные логические операции.
32. Какие циклы можно организовать на языке C?
33. Формат описания переменных. Привести пример.
34. Запись целых констант.
35. Выражения.
36. Арифметические операции.
37. Операции присваивания.
38. Приоритеты логических операций.
39. Структура простейшей программы на C.
40. Напишите формат и пример условного оператора в полной форме.
41. Напишите формат и пример условного оператора в краткой форме.
42. Формат оператора цикла с параметром, постусловием, предусловием.
43. Формат описания массива.

Тема 3. Процедурное программирование

Вопросы собеседования

1. Привести классификацию языков программирования.
2. Рассказать об эволюции языков программирования.
3. Привести примеры применения языков программирования высокого уровня.
4. Что такое уровень языка программирования?
5. Какие у машинных языков достоинства и недостатки?
6. В чём преимущества алгоритмических языков перед машинными?
7. Какие компоненты образуют алгоритмический язык?
8. Какие понятия используют алгоритмические языки?
9. В какой последовательности возникали известные вам языки программирования?
10. Что такое транслятор, компилятор, интерпретатор?
11. Что такое тип данных?
12. Какие типы данных относятся к простым?
13. Что понимают под языками программирования?
14. Перечислите первые алгоритмические языки.

15. Перечислите фундаментальные структуры данных.
16. Чем отличается компилятор от интерпретатора?
17. Что понимают под интегрированной средой программирования?
18. На каких принципах основывается структурное программирование?
19. Какие основные базовые конструкции структурного программирования?
20. Озвучьте основные принципы объектно-ориентированного программирования.
21. Что такое объект?
22. Как характеризуется объект?
23. Что такое класс?
24. Объясните связь между объектом и классом.
25. Что такое инкапсуляция, наследование, полиморфизм в объектно-ориентированном программировании?
26. Какие символы могут содержать имена переменных?
27. Какие типы переменных допустимы на языке C?
28. Что такое оператор?
29. Какие типы операторов допустимы на C?
30. Для чего используются комментарии в программе?
31. Перечислите основные логические операции.
32. Какие циклы можно организовать на языке C?
33. Формат описания переменных. Привести пример.
34. Запись целых констант.
35. Выражения.
36. Арифметические операции.
37. Операции присваивания.
38. Приоритеты логических операций.
39. Структура простейшей программы на C.
40. Напишите формат и пример условного оператора в полной форме.
41. Напишите формат и пример условного оператора в краткой форме.
42. Формат оператора цикла с параметром, постусловием, предусловием.
43. Формат описания массива.

Тема 4. Классы

Вопросы собеседования

1. Привести классификацию языков программирования.
2. Рассказать об эволюции языков программирования.
3. Привести примеры применения языков программирования высокого уровня.
4. Что такое уровень языка программирования?
5. Какие у машинных языков достоинства и недостатки?
6. В чём преимущества алгоритмических языков перед машинными?
7. Какие компоненты образуют алгоритмический язык?
8. Какие понятия используют алгоритмические языки?
9. В какой последовательности возникали известные вам языки программирования?
10. Что такое транслятор, компилятор, интерпретатор?
11. Что такое тип данных?
12. Какие типы данных относятся к простым?
13. Что понимают под языками программирования?
14. Перечислите первые алгоритмические языки.
15. Перечислите фундаментальные структуры данных.
16. Чем отличается компилятор от интерпретатора?

17. Что понимают под интегрированной средой программирования?
18. На каких принципах основывается структурное программирование?
19. Какие основные базовые конструкции структурного программирования?
20. Озвучьте основные принципы объектно-ориентированного программирования.
21. Что такое объект?
22. Как характеризуется объект?
23. Что такое класс?
24. Объясните связь между объектом и классом.
25. Что такое инкапсуляция, наследование, полиморфизм в объектно-ориентированном программировании?
26. Какие символы могут содержать имена переменных?
27. Какие типы переменных допустимы на языке C?
28. Что такое оператор?
29. Какие типы операторов допустимы на C?
30. Для чего используются комментарии в программе?
31. Перечислите основные логические операции.
32. Какие циклы можно организовать на языке C?
33. Формат описания переменных. Привести пример.
34. Запись целых констант.
35. Выражения.
36. Арифметические операции.
37. Операции присваивания.
38. Приоритеты логических операций.
39. Структура простейшей программы на C.
40. Напишите формат и пример условного оператора в полной форме.
41. Напишите формат и пример условного оператора в краткой форме.
42. Формат оператора цикла с параметром, постусловием, предусловием.
43. Формат описания массива.

Тема 5. Абстракция данных

Вопросы собеседования

1. Привести классификацию языков программирования.
2. Рассказать об эволюции языков программирования.
3. Привести примеры применения языков программирования высокого уровня.
4. Что такое уровень языка программирования?
5. Какие у машинных языков достоинства и недостатки?
6. В чём преимущества алгоритмических языков перед машинными?
7. Какие компоненты образуют алгоритмический язык?
8. Какие понятия используют алгоритмические языки?
9. В какой последовательности возникали известные вам языки программирования?
10. Что такое транслятор, компилятор, интерпретатор?
11. Что такое тип данных?
12. Какие типы данных относятся к простым?
13. Что понимают под языками программирования?
14. Перечислите первые алгоритмические языки.
15. Перечислите фундаментальные структуры данных.
16. Чем отличается компилятор от интерпретатора?
17. Что понимают под интегрированной средой программирования?
18. На каких принципах основывается структурное программирование?

19. Какие основные базовые конструкции структурного программирования?
20. Озвучьте основные принципы объектно-ориентированного программирования.
21. Что такое объект?
22. Как характеризуется объект?
23. Что такое класс?
24. Объясните связь между объектом и классом.
25. Что такое инкапсуляция, наследование, полиморфизм в объектно-ориентированном программировании?
26. Какие символы могут содержать имена переменных?
27. Какие типы переменных допустимы на языке C?
28. Что такое оператор?
29. Какие типы операторов допустимы на C?
30. Для чего используются комментарии в программе?
31. Перечислите основные логические операции.
32. Какие циклы можно организовать на языке C?
33. Формат описания переменных. Привести пример.
34. Запись целых констант.
35. Выражения.
36. Арифметические операции.
37. Операции присваивания.
38. Приоритеты логических операций.
39. Структура простейшей программы на C.
40. Напишите формат и пример условного оператора в полной форме.
41. Напишите формат и пример условного оператора в краткой форме.
42. Формат оператора цикла с параметром, постусловием, предусловием.
43. Формат описания массива.

Тестирование

Тема 3. Процедурное программирование

Тест

1. Какая из приведенных ниже констант относится к восьмеричной.
 (!) 020
 (?) 16
 (?) 0x10
2. Какая из приведенных ниже констант относится к десятичной.
 (!) 16
 (?) 020
 (?) 0x10
3. Какая из приведенных ниже констант относится к шестнадцатеричной.
 (!) 0x10
 (?) 020
 (?) 16
4. Оператор присвоения на языке C++.
 (!) =
 (?) :=
 (?) ==
5. Где приведено не правильное описание комментариев?
 (!) /* комментарии к алгоритму решения */ / краевой задачи */

(?) // комментарии к программе

(?) /* начало алгоритма */

6. Какие типы данных относятся целым?

(!) char, int, short, long.

(?) float, double, long double.

7. Какие типы данных относятся к типам с плавающей точкой?

(!) float, double, long double.

(?) char, int, short, long.

8. Какая переменная относится к переменной перечисляемого типа?

(!) enum.

(?) short

(?) long double

9. Пример описания двумерного массива на языке C++.

(!) int a[2][3];

(?) int a[2..3,1..3];

10. В каком месте следует объявлять переменные?

(!) В любом до их первого использования.

(?) В разделе описания переменных.

(?) Можно не объявлять.

11. Где изображен оператор декремента.

(!) --

(?) ++

(?) **

12. Где изображен оператор инкремента.

(!) ++

(?) --

(?) **

13. Приведите пример оператора, который находит остаток от деления в языке C++.

(!) %

(?) mod

(?) div

14. Приведите пример оператора логического «и» в языке C++.

(!) &&

(?) &

(?) &&&

15. Приведите пример оператора логического «или» в языке C++.

(!) ||

(?) ||||

(?) |||

16. Приведите пример оператора «не равно» в языке C++.

(!) !=

(?) <>

17. Что означает оператор sizeof.

(!) Определяет размер памяти которая соответствует идентификатору или типу.

(?) Определяет конец файла.

18. Что присвоится переменной max в результате выполнения следующего кода $max = (d \leq b) ? b : d;$?

(!) Переменной max присваивается максимальное значение переменных d и b.

(?) Переменной max присваивается минимальное значение переменных d и b.

(?) Ничего не присвоится.

19. Чему будет равно значение переменной b в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int b, c=1;
    b=c++ + c++;
    cout<<b;
return 0;}
```

(!) 2

(?) 4

(?) 3

20. Чему будет равно значение переменной b в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int b, c=1;
    b=++c + c++;
    cout<<b;
return 0;}
```

(!) 4

(?) 3

(?) 2

21. Чему будет равно значение переменной b в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int b, c=1;
    b=++c + ++c;
    cout<<b;
return 0;}
```

(!) 6

(?) 4

(?) 8

22. Чему будет равно значение переменной z в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int t=1, s=2, z, f;
    z=(t++)*5;
    cout<<z;
return 0;}
```

(!) 5

(?) 10

(?) 15

23. Чему будет равно значение переменной z в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int t=1, s=2, z, f;
    z=t++*5;
    cout<<z;
return 0;}
```

(!) 5

(?) 10

(?) 15

24. Чему будет равно значение переменной z в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int t=1, s=2, z, f;
    z=++t*5;
    cout<<z;
    return 0;}
```

(!) 10

(?) 5

(?) 15

25. Перечислите операторы выбора.

(!) if, switch.

(?) for, while, do while.

(?) break, continue, return, goto.

26. Перечислите операторы цикла.

(!) for, while, do while.

(?) if, switch.

(?) break, continue, return, goto.

27. Перечислите операторы перехода.

(!) break, continue, return, goto.

(?) if, switch.

(?) for, while, do while.

28. Приведите пример цикла с пост условием.

(!) do while.

(?) while.

(?) for.

29. Приведите пример цикла с пред условием.

(!) while.

(?) do while.

(?) for.

30. Приведите пример цикла со счетчиком.

(!) for.

(?) do while.

(?) while.

Тема 2. Функции и классы

31. Где, в программе приведенной ниже, находится прототип функции square:

```
#include <iostream.h>
```

```
int square(int);
```

```
main() {
    for (int x = 1; x <= 10; x++)
        cout << square (x) << " ";
```

```
cout << endl; return 0; }
```

```
int square(int y) {return y * y; }
```

(!) int square(int);

(?) int square(int y) {return y * y; }

(?) cout << square (x) << " ";

32. Какой формат описания имеют функции?

(!) тип-возвращаемого-значения имя-функции
 (список-параметров)
 {
 объявления и операторы
 }

(?) тип-возвращаемого-значения имя-функции
 (список-параметров)

(?) имя-функции
 (список-параметров)

33. Перечислите возможные области действия идентификатора.

(!) Файл, функция, блок, прототип функции.

(?) Цикл, ветвления, функция.

(?) Файл, функция.

34. Где описана функция с пустым списком параметров?

(!) void f2(void);

(?) void f2(int x);

(?) void f2(int);

35. Где определена функция, которая не возвращает значение?

(!) void f2(void);

(?) int f2(void);

(?) float f2(void);

36. Где определена встраиваемая функция?

(!) inline float cube (const float s) {return s*s*s; }

(?) float cube (const float s) {return s*s*s; }

(?) double cube (const float s) {return s*s*s; }

37. Что такое указатель?

(!) Указатели — это переменные, которые содержат в качестве своих значений адреса памяти.

(?) Направление, указывающее на переменную.

(?) Адрес памяти.

38. Что такое класс?

(!) Механизм для создания объектов.

(?) Класс памяти.

(?) Область на жестком диске.

39. Какое ключевое слово используется для определения открытых функций и переменных класса?

(!) public

(?) class

(?) private

40. Когда вызывается конструктор класса?

(!) При создании объектов данного класса.

(?) При удалении объектов данного класса.

(?) Никогда.

41. Когда вызывается деструктор класса?

(!) При удалении объектов данного класса.

(?) При создании объектов данного класса.

(?) Никогда.

42. Может ли быть конструктор класса с параметрами?

(!) да

(?) нет

(?) не всегда

43. Что такое наследование классов?

(!) Это механизм, посредством которого один класс может наследовать свойства дру-гого.

(?) Это механизм создания классов.

(?) Это механизм удаления классов.

44. Что такое производный класс?

(!) Наследующий класс

(?) Базовый класс

45. Можно ли объекты классов присваивать друг другу?

(!) Можно, только объекты одинакового типа.

(?) Нельзя.

(?) Можно всегда.

46. Для чего нужна дружественные функции?

(!) Для получения доступа к закрытым членам класса.

(?) Для убыстрения работы.

(?) Для увеличения производительности.

47. Какие операции не могут быть перегружены?

(!) ::

(?) >=

(?) &=

48. Какие операции могут быть перегружены?

(!) ^=

(?) ?:

(?) .*

49. Могут ли быть объекты класса в качестве возвращаемого значения функции?

(!) да

(?) нет

50. Можно ли передавать объекты класса в функцию?

(!) да

(?) нет

Тема 5. Абстракция данных

Тест

1. Какая из приведенных ниже констант относится к восьмеричной.

(!) 020

(?) 16

(?) 0x10

2. Какая из приведенных ниже констант относится к десятичной.

(!) 16

(?) 020

(?) 0x10

3. Какая из приведенных ниже констант относится к шестнадцатеричной.

(!) 0x10

(?) 020

(?) 16

4. Оператор присвоения на языке C++.

(!) =

(?) :=

(?) ==

5. Где приведено не правильное описание комментариев?

(!) /* комментарии к алгоритму решения */ краевой задачи */

(?) // комментарии к программе

(?) /* начало алгоритма */

6. Какие типы данных относятся к целым?

(!) char, int, short, long.

(?) float, double, long double.

7. Какие типы данных относятся к типам с плавающей точкой?

(!) float, double, long double.

(?) char, int, short, long.

8. Какая переменная относится к переменной перечисляемого типа?

(!) enum.

(?) short

(?) long double

9. Пример описания двумерного массива на языке C++.

(!) int a[2][3];

(?) int a[2..3,1..3];

10. В каком месте следует объявлять переменные?

(!) В любом до их первого использования.

(?) В разделе описания переменных.

(?) Можно не объявлять.

11. Где изображен оператор декремента.

(!) --

(?) ++

(?) **

12. Где изображен оператор инкремента.

(!) ++

(?) --

(?) **

13. Приведите пример оператора, который находит остаток от деления в языке C++.

(!) %

(?) mod

(?) div

14. Приведите пример оператора логического «и» в языке C++.

(!) &&

(?) &

(?) &&&

15. Приведите пример оператора логического «или» в языке C++.

(!) ||

(?) ||||

(?) |||

16. Приведите пример оператора «не равно» в языке C++.

(!) !=

(?) <>

17. Что означает оператор sizeof.

(!) Определяет размер памяти которая соответствует идентификатору или типу.

(?) Определяет конец файла.

18. Что присвоится переменной max в результате выполнения следующего кода $max = (d \leq b) ? b : d;$?

(!) Переменной max присваивается максимальное значение переменных d и b.

(?) Переменной `max` присваивается минимальное значение переменных `d` и `b`.

(?) Ничего не присвоится.

19. Чему будет равно значение переменной `b` в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int b, c=1;
    b=c++ + c++;
    cout<<b;
return 0;}
```

(!) 2

(?) 4

(?) 3

20. Чему будет равно значение переменной `b` в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int b, c=1;
    b=++c + c++;
    cout<<b;
return 0;}
```

(!) 4

(?) 3

(?) 2

21. Чему будет равно значение переменной `b` в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int b, c=1;
    b=++c + ++c;
    cout<<b;
return 0;}
```

(!) 6

(?) 4

(?) 8

22. Чему будет равно значение переменной `z` в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int t=1, s=2, z, f;
    z=(t++)*5;
    cout<<z;
return 0;}
```

(!) 5

(?) 10

(?) 15

23. Чему будет равно значение переменной `z` в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int t=1, s=2, z, f;
    z=t++*5;
    cout<<z;
return 0;}
```

(!) 5

(?) 10

(?) 15

24. Чему будет равно значение переменной z в результате выполнения следующего кода:

```
#include <iostream.h>
```

```
main() {
    int t=1, s=2, z, f;
    z=++t*5;
    cout<<z;
    return 0;}
```

(!) 10

(?) 5

(?) 15

25. Перечислите операторы выбора.

(!) if, switch.

(?) for, while, do while.

(?) break, continue, return, goto.

26. Перечислите операторы цикла.

(!) for, while, do while.

(?) if, switch.

(?) break, continue, return, goto.

27. Перечислите операторы перехода.

(!) break, continue, return, goto.

(?) if, switch.

(?) for, while, do while.

28. Приведите пример цикла с пост условием.

(!) do while.

(?) while.

(?) for.

29. Приведите пример цикла с пред условием.

(!) while.

(?) do while.

(?) for.

30. Приведите пример цикла со счетчиком.

(!) for.

(?) do while.

(?) while.

Тема 2. Функции и классы

31. Где, в программе приведенной ниже, находится прототип функции square:

```
#include <iostream.h>
```

```
int square(int);
```

```
main() {
    for (int x = 1; x <= 10; x++)
        cout << square (x) << " ";
```

```
cout << endl; return 0; }
```

```
int square(int y) {return y * y; }
```

```
(!) int square(int);
```

```
(?) int square(int y) {return y * y; }
```

```
(?) cout << square (x) << " ";
```

32. Какой формат описания имеют функции?

- (!) тип-возвращаемого-значения имя-функции
(список-параметров)
{
объявления и операторы
}

(?) тип-возвращаемого-значения имя-функции
(список-параметров)

(?) имя-функции
(список-параметров)

33. Перечислите возможные области действия идентификатора.

(!) Файл, функция, блок, прототип функции.

(?) Цикл, ветвления, функция.

(?) Файл, функция.

34. Где описана функция с пустым списком параметров?

(!) void f2(void);

(?) void f2(int x);

(?) void f2(int);

35. Где определена функция, которая не возвращает значение?

(!) void f2(void);

(?) int f2(void);

(?) float f2(void);

36. Где определена встраиваемая функция?

(!) inline float cube (const float s) {return s*s*s; }

(?) float cube (const float s) {return s*s*s; }

(?) double cube (const float s) {return s*s*s; }

37. Что такое указатель?

(!) Указатели — это переменные, которые содержат в качестве своих значений адреса памяти.

(?) Направление, указывающее на переменную.

(?) Адрес памяти.

38. Что такое класс?

(!) Механизм для создания объектов.

(?) Класс памяти.

(?) Область на жестком диске.

39. Какое ключевое слово используется для определения открытых функций и переменных класса?

(!) public

(?) class

(?) private

40. Когда вызывается конструктор класса?

(!) При создании объектов данного класса.

(?) При удалении объектов данного класса.

(?) Никогда.

41. Когда вызывается деструктор класса?

(!) При удалении объектов данного класса.

(?) При создании объектов данного класса.

(?) Никогда.

42. Может ли быть конструктор класса с параметрами?

(!) да

(?) нет

(?) не всегда

43. Что такое наследование классов?

(!) Это механизм, посредством которого один класс может наследовать свойства дру-гого.

(?) Это механизм создания классов.

(?) Это механизм удаления классов.

44. Что такое производный класс?

(!) Наследующий класс

(?) Базовый класс

45. Можно ли объекты классов присваивать друг другу?

(!) Можно, только объекты одинакового типа.

(?) Нельзя.

(?) Можно всегда.

46. Для чего нужна дружественные функции?

(!) Для получения доступа к закрытым членам класса.

(?) Для убыстрения работы.

(?) Для увеличения производительности.

47. Какие операции не могут быть перегружены?

(!) ::

(?) >=

(?) &=

48. Какие операции могут быть перегружены?

(!) ^=

(?) ?:

(?) .*

49. Могут ли быть объекты класса в качестве возвращаемого значения функции?

(!) да

(?) нет

50. Можно ли передавать объекты класса в функцию?

(!) да

(?) нет

4.3 Промежуточная аттестация по дисциплине проводится в форме зачета

Типовые вопросы зачета (ПК-3)

Типовые вопросы экзамена

1. Парадигмы программирования.

2. Неабстрактные операции

3. Типы определяемые пользователем

4. Указатели и массивы

5. Ссылки

6. Константы

7. Функции как модули

8. Функциональная декомпозиция

9. Файловая организация

10. Структурное программирование

11. Перегружаемые и подставляемые функции

12. Аргументы и возвращаемые значения

13. Классовые типы

14. Компоненты данных

15. Функциональные компоненты
16. Операторные функции
17. Защита доступа и дружественные функции
18. Инициализация и преобразования
19. Указатели на компоненты класса
20. Числовые типы
21. Скалярные типы. Операторы сравнения, логические операторы
22. Неабстрактные операции
23. Типы определяемые пользователем
24. Указатели и массивы
25. Ссылки
26. Константы
27. Функции как модули
28. Функциональная декомпозиция
29. Файловая организация
30. Структурное программирование
31. Перезагружаемые и подставляемые функции
32. Аргументы и возвращаемые значения
33. Классовые типы
34. Компоненты данных
35. Функциональные компоненты
36. Операторные функции
37. Защита доступа и дружественные функции
38. Инициализация и преобразования
39. Указатели на компоненты класса
40. Комплексные числа
41. Строки
42. Упорядоченные выборки
43. Общность
44. Абстракция управления

Типовые задания для зачета (ПК-3)

Не предусмотрено

4.4. Шкала оценивания промежуточной аттестации

Оценка	Компетенции	Дескрипторы (уровни) – основные признаки освоения (показатели достижения результата)
«зачтено» (50 - 100 баллов)	ПК-3	На удовлетворительном уровне осуществляет действия по обработке и анализу научно-технической информации и результатов исследований в области физики при планировании и организации научных исследований и формировании отчетной документации с использованием навыков программирования
«не зачтено» (0 - 49 баллов)	ПК-3	Не способен осуществить действия по обработке и анализу научно-технической информации и результатов исследований в области физики при планировании и организации научных исследований и формировании отчетной документации с использованием навыков программирования

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

5.1 Методические указания по организации самостоятельной работы обучающихся:

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

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

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

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

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

5.2 Рекомендации обучающимся по работе с теоретическими материалами по дисциплине

При изучении и проработке теоретического материала необходимо:

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

5.3 Рекомендации по работе с научной и учебной литературой

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

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

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

В процессе работы с основной и дополнительной литературой студент может:

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

5.4. Рекомендации по подготовке к отдельным заданиям текущего контроля

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

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

- правильность ответа по содержанию;
- полнота и глубина ответа;
- сознательность ответа;
- логика изложения материала;
- рациональность использованных приемов и способов решения поставленной учебной задачи;
- своевременность и эффективность использования наглядных пособий и технических средств при ответе;
- использование дополнительного материала;
- рациональность использования времени, отведенного на задание.

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

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

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Основная литература:

1. Хлебников В.В. Основы программирования на C++ : учеб. пособ.. - Тамбов: Изд-во ТГУ, 2009. - 115 с.
2. Крючин О.В., Хлебников В.В. Основы программирования на языке C : учеб. пособие. - Тамбов: [Издат. дом ТГУ им. Г.Р. Державина], 2015. - 88 с.

6.2 Иные источники:

1. Единое окно доступа к образовательным интернет-ресурсам Федерального портала «Российское образование» - http://window.edu.ru/catalog/?p_rubr=2.1.21%2F
2. Официальный сайт Всероссийского центра изучения общественного мнения (ВЦИОМ) - www.wciom.ru
3. Федеральная служба по надзору в сфере образования и науки - <http://obrnadzor.gov.ru>
4. Вопросы образования - <http://www.ecsocman.edu.ru/vo>

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

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

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

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

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

Лицензионное и свободно распространяемое программное обеспечение:

Kaspersky Endpoint Security для бизнеса - Стандартный Russian Edition. 1500-2499 Node 1 year Educational Renewal Licence

Microsoft Office Профессиональный плюс 2007

7-Zip 9.20

Операционная система Microsoft Windows 10

Adobe Reader XI - Russian

Профессиональные базы данных и информационные справочные системы:

1. Научная электронная библиотека eLIBRARY.ru. – URL: <https://elibrary.ru>
2. Web of Science: политематическая реферативно-библиографическая и наукометрическая база данных . – URL: <https://apps.webofknowledge.com>
3. Scopus: база данных . – URL: <https://www.scopus.com>
4. Электронный каталог Фундаментальной библиотеки ТГУ. – URL: <http://biblio.tsutmb.ru/elektronnyij-katalog>
5. Электронная библиотека ТГУ. – URL: <https://elibrary.tsutmb.ru/>
6. Российская государственная библиотека. – URL: <https://www.rsl.ru>
7. Российская национальная библиотека. – URL: <http://nlr.ru>
8. Президентская библиотека имени Б.Н. Ельцина. – URL: <https://www.prlib.ru>
9. Электронная библиотека РФФИ. – URL: <https://www.rfbr.ru/rffi/ru/library>

Электронная информационно-образовательная среда

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

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