

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.	Кафедра	Математики, физики и информационных технологий
2.	Направление подготовки	09.03.01 Информатика и вычислительная техника
3.	Направленность (профиль)	Виртуальные технологии и дизайн
4.	Дисциплина (модуль)	Б1.О.14.04 Объектно-ориентированное программирование
5.	Форма обучения	очная
6.	Год набора	2021

I. Методические рекомендации

1.1 Методические рекомендации по организации работы студентов во время проведения лекционных занятий

- В ходе лекций преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации для практического занятия и указания для выполнения самостоятельной работы.
- В ходе лекционных занятий обучающемуся необходимо вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание изучаемой дисциплины, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве.
- Желательно оставить в рабочих конспектах поля, на которых делать пометки, подчеркивающие особую важность тех или иных теоретических положений. Рекомендуется активно задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

1.2 Методические указания к выполнению лабораторных работ

- Лабораторные работы сочетают элементы теоретического исследования и практической работы. Выполняя лабораторные работы, обучающиеся лучше усваивают учебный материал, так как многие определения, казавшиеся отвлеченными, становятся вполне конкретными, происходит соприкосновение теории с практикой, что в целом содействует пониманию сложных вопросов науки и становлению обучающихся как будущих специалистов.
- Выполнение лабораторных работ направлено на:
 - обобщение, систематизацию, углубление теоретических знаний по конкретным темам учебной дисциплины;
 - формирование умений применять полученные знания в практической деятельности;
 - развитие аналитических, проектировочных, конструктивных умений;
 - выработку самостоятельности, ответственности и творческой инициативы.
- Лабораторные занятия как вид учебной деятельности должны проводиться в специально оборудованных лабораториях, где выполняются лабораторные работы (задания).
- Форма организации обучающихся для проведения лабораторного занятия – фронтальная, групповая и индивидуальная – определяется преподавателем, исходя из темы, цели, порядка выполнения работы.
- В ходе лабораторных занятий реализуется интерактивная форма взаимодействия – в виде самостоятельных заданий на лабораторных занятиях.

Результаты выполнения лабораторной работы оформляются обучающимися в виде отчета, форма и содержание которого определяются требованиями соответствующей работы.

1.3 Методические рекомендации по организации самостоятельной работы обучающихся

- Самостоятельная работа – планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).
- Самостоятельная работа студентов (далее – СРС) в ВУЗе является важным видом учебной и научной деятельности студента. СРС играет значительную роль в рейтинговой технологии обучения. Обучение в ВУЗе включает в себя две, практически одинаковые по объему и взаимовлиянию части – процесса обучения и процесса самообучения. Поэтому СРС должна стать эффективной и целенаправленной работой студента.
- К современному специалисту общество предъявляет достаточно широкий перечень требований, среди которых немаловажное значение имеет наличие у выпускников определенных способностей и умения самостоятельно добывать знания из различных источников, систематизировать полученную информацию, давать оценку конкретной ситуации. Формирование такого умения происходит в течение всего периода обучения через участие студентов в практических занятиях, выполнение контрольных заданий и тестов, написание курсовых и выпускных квалификационных работ. При этом СРС играет решающую роль в ходе всего учебного процесса.
- В процессе самостоятельной работы студент приобретает навыки самоорганизации, самоконтроля, самоуправления, саморефлексии и становится активным самостоятельным субъектом учебной деятельности.
- Формы самостоятельной работы студентов разнообразны. Они включают в себя:
 - изучение учебной, научной и методической литературы, материалов периодических изданий с привлечением электронных средств официальной, статистической, периодической и научной информации;
 - подготовку докладов и рефератов, написание курсовых и выпускных квалификационных работ;
 - участие в работе студенческих конференций, комплексных научных исследованиях.
- Самостоятельная работа приобщает студентов к научному творчеству, поиску и решению актуальных современных проблем.
- Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение, рекомендованной литературы, активное участие на практических и лабораторных занятиях.

1.4 Методические рекомендации по решению тестовых заданий

- Тестовая система предусматривает вопросы/задания, на которые обучающийся должен дать один или несколько вариантов правильного ответа из предложенного списка ответов. При поиске ответа необходимо проявлять внимательность.
- При отсутствии какого-либо одного ответа на вопрос, предусматривающий множественный выбор, весь ответ считается неправильным.
- Ответы правильные выделяются в тесте подчеркиванием или любым другим допустимым символом.

1.5 Методические рекомендации по подготовке презентации

Алгоритм создания презентации:

- 1 этап – определение цели презентации
- 2 этап – подробное раскрытие информации,
- 3 этап – основные тезисы, выводы.

Следует использовать 10-15 слайдов. При этом:

- первый слайд – титульный, предназначен для размещения названия презентации, имени докладчика и его контактной информации;
- на втором слайде необходимо разместить содержание презентации, а также краткое описание основных вопросов;
- оставшиеся слайды имеют информативный характер.

Обычно подача информации осуществляется по плану: тезис – аргументация – вывод.

Требования к оформлению и представлению презентации:

- Читательность (видимость из самых дальних уголков помещения и с различных устройств), текст должен быть набран 24-30-ым шрифтом.
- Тщательно структурированная информация.
- Наличие коротких и лаконичных заголовков, маркированных и нумерованных списков.
- Каждому положению (идее) надо отвести отдельный абзац.
- Главную идею надо выложить в первой строке абзаца.
- Использовать табличные формы представления информации (диаграммы, схемы) для иллюстрации важнейших фактов, что даст возможность подать материал компактно и наглядно.
- Графика должна органично дополнять текст.
- Выступление с презентацией длится не более 10 минут.

1.6 Методические рекомендации по подготовке доклада (эссе)

Алгоритм создания доклада:

- 1 этап – определение темы доклада
- 2 этап – определение цели доклада
- 3 этап – подробное раскрытие информации
- 4 этап – формулирование основных тезисов и выводов.

Требования к теме доклада (эссе):

Темы докладов формулируются таким образом, чтобы расширить знания студента о конкретном программном продукте или компьютерном устройстве, а также дать представление о возможности и использования в профессиональной деятельности, например:

1. Назначение и возможности редакторов трехмерной графики.
2. Сравнительный анализ возможностей текстовых процессоров пакетов MS Office и LibreOffice.
3. Обзор возможностей настольной издательской системы MS Publisher на примере создания информационного буклета
4. Сублимационная печать. Назначение, преимущества и недостатки.

Требования к оформлению доклада:

1. Объем доклада – 5 страниц (без титульного листа и списка источников).
2. Титульный лист должен быть оформлен по образцу (имеется файл с образцом).
3. Основной текст работы оформлен в соответствии с требованиями, указанными ниже.
4. В случае использования в тексте таблиц и/или рисунков на каждый объект должна быть ссылка в тексте работы. Например, «... основные виды программных средств представлены ниже (см. Таблица 1)» или «... схему передачи информации можно увидеть на рис. 1».
5. Количество источников должно быть не менее трех, на все должны быть ссылки внутри текста.
6. Список используемых источников должен быть оформлен в соответствии с требованиями, указанными ниже.

1.7 Методические рекомендации по подготовке к сдаче зачета

- Зачет осуществляется в рамках завершения изучения дисциплины и позволяет определить качество усвоения изученного материала, а также степень сформированности компетенций.
- Студенты обязаны сдавать зачет в строгом соответствии с утвержденными учебными планами, разработанными согласно образовательным стандартам высшего образования.
- По данной дисциплине зачет принимается по практической части курса. Задания, предлагаемые на зачете, соответствуют оценочным средствам для промежуточной аттестации обучающихся по предмету.
- Студент обязан не только представить правильно выполненные задания, но и защитить свое решение.
- Преподавателю предоставляется право задавать студентам вопросы по решению заданий практической части с целью выявления глубины понимания изученного материала и степени самостоятельности выполнения заданий.
- При явке на зачет студенты обязаны иметь при себе зачетную книжку, которую они предъявляют преподавателю в начале зачета.
- Рекомендуется при подготовке к зачету опираться на следующий план:
 1. Повторить теоретическую часть курса.
 2. После работы над теорией необходимо ответить на вопросы для самоконтроля.
 3. Повторить методы, способы и приемы решения задач по всем темам курса, опираясь на задания, которые решались на практических занятиях и предлагались для самостоятельной работы.
 4. Решить типовые задания по данной теме.

II. План лабораторных занятий

Общее описание содержания разделов и планы лабораторных работ

Модуль 1. Основные принципы объектно-ориентированного программирования, объектно-ориентированная модель.

Основные понятия: Методология ООП. Абстракция данных, инкапсуляция, наследование, полиморфизм, динамический полиморфизм. Модель ООП.

Лабораторные работы:

ЛР 1. Реализация принципов ООП в разных ЯП.

ЛР 2. Сравнение функционала современных объектно-ориентированных языков программирования и сред разработки.

Литература: [1], [2], [3].

Модуль 2. Классы. Объекты. Методы. События.

Основные понятия: Синтаксис объявления класса, создание объекта класса, использование указателя на объект, динамические объекты, использование операции разрешения видимости, неполное объявление класса, спецификаторы доступа. Конструкторы и деструкторы, правила для них, список инициализации элементов. Встраиваемые функции. Функции, дружественные одному классу, функции, дружественные нескольким классам. Понятие и правила перегрузки операторов, перегрузка унарных операторов, перегрузка оператора присваивания, перегрузка бинарных операторов.

Лабораторные работы:

ЛР 1. Создание класса комплексных чисел и действий над ними.

ЛР 3. Создание класса рациональных чисел и действий над ними.

ЛР 4. Создание программы сортировки различного типа чисел и строк.

ЛР 5. Создание функции обмена местами для переменных различных типов

ЛР 6. Создание программ, работающих с линейными однонаправленными списками.

ЛР 7. Создание программ по перегрузке функций.

ЛР 8. Создание программ работающих с файлами и строковыми переменными.

Литература: [1], [2], [3].

Модуль 3. Инструментальные средства ООП.

Основные понятия: Знакомство с классами. Методы и атрибуты классов. Наследование и производные классы. Пользовательские исключения. Свойства, итераторы и функторы. Специальные методы работы с классами в курсе.

Лабораторные работы:

- ЛР 1. Синтаксис языка программирования
 - ЛР 2. Создание классов и объектов
 - ЛР 3. Конструктор / деструктор класса
 - ЛР 4. Полиморфизм и переопределение операторов
 - ЛР 5. Модули и их импорт
- Литература:* [1], [2], [3].

Модуль 4. Разработка графического пользовательского интерфейса средствами ООП.

Основные понятия: Понятие графического пользовательского интерфейса (ГПИ). Требования к ГПИ. Инструментальные средства ООП для реализации ГПИ. Документирование программного продукта.

Лабораторные работы:

- ЛР 1. Приложения «Калькулятор», «Матричный калькулятор», «Комплексный калькулятор».
 - ЛР 2. Приложения «Текстовый процессор», «Браузер».
 - ЛР 3. Игровые приложения разных жанров
- Литература:* [1], [2], [3].

Примеры задач:

1. Описать класс, реализующий стек. Написать программу, использующую этот класс для моделирования Т-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла и с клавиатуры.
2. Описать класс, реализующий бинарное дерево, обладающее возможностью добавления новых элементов, удаления существующих, поиска элемента по ключу, а также последовательного доступа ко всем элементам. Написать программу, использующую этот класс для представления англо-русского словаря. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Предусмотреть возможность формирования словаря из файла и с клавиатуры.
3. Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.
4. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
5. Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
6. Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
7. Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменение размеров, построение наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
8. Составить описание класса для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы массива, возможность задания произвольных границ индексов при создании объекта, возможность выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, вывода на экран элемента массива по заданному индексу, вывода на экран всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
9. Составить описание класса для определения одномерных массивов строк фиксированной длины. Предусмотреть возможность обращения к отдельным строкам массива по индексам, контроль выхода за пределы массива, выполнения операций поэлементного сцепления двух массивов с образованием

нового массива, слияния двух массивов с исключением повторяющихся элементов, вывод на экран элемента массива по заданному индексу и всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

10. Составить описание класса многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Предусмотреть методы для вычисления значения многочлена для заданного аргумента, операции сложения, вычитания и умножения многочленов с получением нового объекта-многочлена, вывод на экран описания многочлена. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
11. Составить описание класса одномерных массивов строк, каждая строка задается длиной и указателем на выделенную для нее память. Предусмотреть возможность обращения к отдельным строкам массива по индексам, контроль выхода за пределы массивов, выполнения операций поэлементного сцепления двух массивов с образованием нового массива, слияния двух массивов с исключением повторяющихся элементов, вывод на экран элемента массива и всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
12. Составить описание класса, обеспечивающего представление матрицы произвольного размера с возможностью изменения числа строк и столбцов, вывода на экран подматрицы любого размера и всей матрицы. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
13. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
14. Описать класс «записная книжка». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
15. Описать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
16. Описать класс «множество», позволяющий выполнять основные операции — добавление и удаление элемента, пересечение, объединение и разность множеств. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
17. Описать класс, реализующий стек. Написать программу, использующую этот класс для отыскания прохода по лабиринту. Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат определяется его координатами в матрице. После отыскания прохода программа печатает найденный путь в виде координат квадратов.
18. Описать класс «предметный указатель». Каждый компонент указателя содержит слово и номера страниц, на которых это слово встречается. Количество номеров страниц, относящихся к одному слову, от одного до десяти. Предусмотреть возможность формирования указателя с клавиатуры и из файла, вывода указателя, вывода номеров страниц для заданного слова, удаления элемента из указателя. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
19. Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.
20. Написать программу, которая считывает текст из файла и выводит на экран только предложения, содержащие заданное с клавиатуры слово.

Примерные темы докладов/рефератов

1. Язык программирования INTERCAL.
2. Язык программирования CaneCode.
3. Язык программирования Ook!
4. Язык программирования COW.
5. Язык программирования Whitespace.

6. Язык программирования Spoon.
7. Язык программирования LOLCODE.
8. Язык программирования Byter.
9. Язык программирования Befunge.
10. Язык программирования Befunge-93.
11. Язык программирования Unefunge.
12. Язык программирования Trefunge.
13. Язык программирования 4DL.
14. Язык программирования Piet.
15. Язык программирования Thue.
16. Язык программирования Unlambda.
17. Язык программирования Smetana.
18. Язык программирования SMITH.
19. Язык программирования Muriel.
20. Язык программирования Malbolge.
21. Язык программирования ALPACA.
22. Язык программирования Chef.
23. Язык программирования Shakespeare.
24. Язык программирования Haifu.
25. Язык программирования ArnoldC.
26. Язык программирования FiM++.
27. Язык программирования Var'aq.
28. Язык программирования P''.
29. Язык программирования QuakeC.
30. Язык программирования MC++.
31. Язык программирования Автокод «Инженер».
32. Язык программирования Cool.
33. Язык программирования GOTO++.
34. Язык программирования Lithp.
35. Язык программирования paranoid.
36. Язык программирования SARTRE.
37. Язык программирования SIMPLE.
38. Язык программирования HQ9+/HQ9++.
39. Язык программирования Whenever.
40. Язык программирования smilesript.