

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный аграрный университет имени императора Петра I»

Экономический факультет

Кафедра Информационного обеспечения и моделирования агроэкономических систем

Высокоуровневые методы информатики и программирования

Методические указания для обучающихся по освоению дисциплины и
самостоятельной работе

Направление подготовки:

44.03.04 Профессиональное обучение (по отраслям)

Профиль:

Информатика, вычислительная техника и компьютерные технологии

Литвинова Л.И. Высокоуровневые методы информатики и программирования: Методические указания для обучающихся по освоению дисциплины и самостоятельной работе (направление подготовки: 44.03.04 Профессиональное обучение (по отраслям) профиль: Информатика, вычислительная техника и компьютерные технологии) / Л.И. Литвинова. – Воронеж: ВГАУ, 2017 – 16 с.

Рецензент: д.и.н., профессор, декан гуманитарно-правового факультета федерального государственного образовательного учреждения высшего образования «Воронежский государственный аграрный университет имени императора Петра I» В.Н. Плаксин

Методические указания рассмотрены и рекомендованы к изданию на заседании кафедры Информационного обеспечения и моделирования агроэкономических систем (протокол № 8 от 10 апреля 2017 г.).

Методические указания рассмотрены и рекомендованы к изданию на заседании методической комиссии гуманитарно-правового факультета (протокол № 9 от 24 мая 2017 г.).

© Л.И. Литвинова

© ФГБОУ ВО Воронежский ГАУ

Содержание

| | |
|--|----|
| ВВЕДЕНИЕ | 4 |
| 1. УКАЗАНИЯ ПО ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОЙ ЧАСТИ ДИСЦИПЛИНЫ | 5 |
| 1.1. Общие сведения | 5 |
| 1.2. Особенности освоения отдельных тем | 5 |
| 2. УКАЗАНИЯ ПО ПОДГОТОВКЕ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ | 10 |
| 2.1. Общие сведения | 10 |
| 2.2. Особенности освоения отдельных тем | 10 |
| 3. УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ..... | 14 |
| 4. ПОДГОТОВКА К ТЕКУЩЕМУ КОНТРОЛЮ ЗНАНИЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ | 15 |
| 4.1. Общие сведения | 15 |
| 4.2. Текущий контроль знаний в форме индивидуальных опросов | 15 |
| 4.3. Текущий контроль знаний в форме тестирования..... | 16 |
| 4.4. Текущий контроль знаний в форме проверки контрольной работы и собеседования со студентом (для заочной формы обучения)..... | 16 |
| 4.5. Промежуточная аттестация в форме зачета | 16 |
| 4.6. Промежуточная аттестация в форме дифференцированного зачета по результатам защиты курсового проекта | 16 |
| 4.7. Промежуточная аттестация в форме экзамена..... | 16 |

ВВЕДЕНИЕ

1. Цель и задачи дисциплины.

Ознакомить обучающихся с методами проектирования программ, основами алгоритмизации и программирования на языках Pascal, Delphi, основными операторами, структурами и типами данных этих языков.

Основными задачами изучения дисциплины являются:

Изучение этапов и уровней разработки программного обеспечения.

Определение понятия критерия качества программы.

Изучение мероприятий по повышению надежности программного обеспечения.

Освоение алгоритмизации задач и основных способов представления алгоритмов.

Изучение основных типов вычислительных процессов.

Освоение основ программирования на языках Pascal и Delphi.

2. Требования к уровню освоения дисциплины.

Дисциплина нацелена на формирование компетенций:

| Компетенция | | Планируемые результаты обучения |
|-------------|---|---|
| Код | Название | |
| ПК-2 | Способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение | Знать: - основные методы и средства разработки алгоритмов и прикладного программного обеспечения. Уметь: - проектировать, разрабатывать и тестировать прикладное программное обеспечение. Иметь навыки: - использования технологий и инструментальных средств разработки прикладного программного обеспечения. |
| ПК-8 | Способность программировать приложения и создавать программные прототипы решения прикладных задач | Знать: - теоретические и методологические аспекты разработки приложений. Уметь: - программировать приложения и создавать программы с использованием инструментальных средств. Иметь навыки: - работы с инструментальными средствами создания приложений и программ. |

1. УКАЗАНИЯ ПО ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОЙ ЧАСТИ ДИСЦИПЛИНЫ

1.1. Общие сведения

Лекция является важнейшей формой усвоения теоретического материала, поскольку в режиме реального времени преподаватель может ответить на любой вопрос, возникающий у студента по ходу восприятия лекционного материала, очень важны и комментарии преподавателя по самым разным вопросам теории и практики изучаемой дисциплины. Часто преподаватель дает на лекции самую актуальную информацию, почерпнуть которую самостоятельно студенту не всегда удастся. Кроме указанных объективных причин, требующих от студента посещения лекций, можно отметить и субъективные причины. Посещение лекций является одним из важнейших факторов, характеризующих отношение студента к учебному процессу в целом, и к данной дисциплине в частности. А при текущем и итоговом контроле знаний удельный вес субъективных критериев у каждого преподавателя довольно высок. Следует помнить, что лекция – это не монолог преподавателя. Вопросы, заданные лектору по изучаемой теме, помогут лучше разобраться в ней не только Вам, но и всем остальным студентам, присутствующим на лекции. Несмотря на то, что каждому студенту предоставляется доступ к компьютерным презентациям всего лекционного материала, рекомендуется делать конспекты лекций, в которых необходимо фиксировать наиболее важные моменты, связанные с освоением того или иного теоретического вопроса. Чтение лекций осуществляется в соответствии с рабочей программой учебной дисциплины и календарным планом, разрабатываемым ведущим курса.

1.2. Особенности освоения отдельных тем

Раздел 1. Методология разработки программных средств

1.1 Программное обеспечение ПЭВМ. Машинный язык и языки высокого уровня.

При освоении материалов по данной теме следует рассмотреть общую характеристику, состав и назначение основных видов программного обеспечения компьютера, изучить понятие машинной программы, язык ассемблера, программы на языке ассемблера, основные категории языков программирования (языки высокого уровня и языки низкого уровня), предназначение языков программирования высокого уровня.

1.2 Основные этапы решения задач на ЭВМ. Понятие о структурном программировании.

При освоении материалов по данной теме следует рассмотреть программирование как теоретическую и практическую деятельность, связанную с созданием программ, основные этапы решения задач на компьютере, часть из которых осуществляется без участия компьютера, необходимо изучить принципы структурного программирования и ее цель, подпрограмму как важный элемент структурного программирования, достоинства структурного программирования, двумерное структурное программирование

1.3 Линейное программирование. Критерии качества программного продукта, дружелюбность, жизненный цикл программы.

При освоении материалов по данной теме необходимо овладеть основой линейного программирования, симплекс-методом решения задач линейного программирования, изучить элементы линейного программирования, определение понятия качества программного обеспечения, характеристики качества по стандарту ISO/IEC 25010:2011, альтернативные подходы, дружелюбность, стандарты, процессы и стадии жизненного цикла программы.

1.4 Пути повышения надежности программного обеспечения. Модификация и сопровождение программы. Тестирование и отладка программы.

При освоении материалов по данной теме следует рассмотреть определение основных понятий надежности программного обеспечения, методы и средства обеспечения надежности, понятия модификации и сопровождения программы, линии сопровождения, цель тестирования и отладки, этапы в плане тестирования, роль контрольных примеров в отладке программы.

Раздел 2. Способы записи алгоритма.

2.1. Понятие алгоритма и его свойства, разновидности структур алгоритмов.

При освоении материалов по данной теме необходимо изучить понятие, свойства и формы представления алгоритмов, подробнее изучить графическую форму представления в виде блок-схемы, ее понятие и основные элементы, типы алгоритмических процессов: линейные, ветвление, цикл.

2.2. Описание линейных и разветвляющихся структур. Полное и неполное ветвление.

При освоении материалов по данной теме необходимо разобрать сущность линейного вычислительного процесса, понятие линейной структуры и блок-схему линейного алгоритма, сущность ветвления, варианты структуры ветвления (полная и неполная формы), графическое представление ветвления, примеры структуры ветвления.

2.3. Алгоритмизация циклических вычислительных процессов, Цикл с предусловием, цикл с постусловием, цикл с параметром.

При освоении материалов по данной теме следует рассмотреть понятие цикла, блок-схему цикла, виды циклов: цикл с параметром, цикл с предусловием, цикл с постусловием, понять работу программы при каждом цикле, изучить примеры структуры цикла.

Раздел 3. Программирование структур данных

3.1. Структура программы на языке Pascal, алфавит языка, идентификаторы, константы, выражения, операции.

При освоении материалов по данной теме следует обратить внимание на формат программы на языке Pascal, основные понятия языка, основные символы, алфавит языка, идентификаторы, константы, выражения, операции, элементарные конструкции (имена, числа и строки), типы данных, переменные. Особое внимание следует обратить на правила написания имен в программе.

3.2. Простейшие операторы ввода-вывода, форматы ввода и вывода, присвоения и расчет числовых значений.

При освоении материалов по данной теме следует рассмотреть использование стандартных процедур ввода и вывода Read и Write для ввода и вывода данных, оперирующих стандартными последовательными файлами INPUT и OUTPUT, пример записи операторов ввода и вывода, формат написания оператора присваивания.

3.3. Стандартные типы данных. Описание типов данных. Стандартные операции над типами данных.

При освоении материалов по данной теме следует изучить стандартные типы данных, арифметические операции, выполняемые над целыми операндами, операции отношения, стандартные (встроенные) функции, применимые к аргументам целого типа и результат выполнения которых имеет целый тип, группу стандартных функций для аргумента целого типа, дающих вещественный результат, функции, применимые к вещественным аргументам, дающие вещественный результат и целый результат.

3.4. Представление основных структур программирования: операторы if, case, for, while, repeat-until.

При освоении материалов по данной теме следует освоить правила работы условного оператора, блок-схему составного оператора, оператор варианта Case, позволяющий сделать выбор из произвольного числа вариантов, инструкции For, While, Repeat-until, с помощью которых реализуются циклические вычисления.

3.5. Процедуры и функции, типы данных определяемые пользователем.

При освоении материалов по данной теме следует обратить внимание на понятия и виды подпрограмм, структуру и задачи, решаемые подпрограммами, виды подпрограмм, понятие процедуры, виды и вызов процедур, формальные и фактические параметры, структуру функции, отличительные особенности функций, строковый тип данных, процедуры и функции обработки строк. Особое внимание следует обратить на основное различие между процедурой и функцией.

3.6. Структурированные типы данных, файлы, виды файлов, процедуры чтения и записи

файлов, доступ к файлам.

При освоении материалов по данной теме необходимо рассмотреть файловый тип Паскаля, связывающий программу с внешними устройствами ЭВМ, характерные особенности файла, виды файлов Паскаля, основные процедуры и функции для работы с файлами, процедуры и функции обработки файлов, функции Паскаля.

3.7. Динамические структуры данных, списки, создание списков, выборка необходимых значений.

При освоении материалов по данной теме следует изучить понятие данных динамической структуры, ссылочный тип или тип указателей, объявление указателей, выделение и освобождение динамической памяти, присваивание значений указателю, операции с указателями, присваивание значений динамическим переменным, рассмотреть описание списка, формирование и просмотр списка, удаление элемента из списка, прохождение списка, методы использования динамических списков, а также освоить динамические объекты сложной структуры.

Раздел 4 Программирование сложных программных комплексов

4.1. Модульные программы, технология разработки модульного программирования, объектно-ориентированное программирование, технология создания программ

При освоении материалов по данной теме следует обратить внимание на понятие модульных программ, на раскрытие сущности технологии разработки модульного программирования, понятие, принципы и основу объектно-ориентированного программирования, языки и объектно-ориентированные системы визуального проектирования, технологию создания программ.

4.2. Программирование рекурсивных алгоритмов, способы конструирования программ. Компоновка библиотеки, основы доказательства правильности

При освоении материалов по данной теме необходимо обратить внимание на понятие рекурсивной программы в языках программирования, ее особенности и пути выполнения, преобразование рекурсивной программы в итеративную (использующую циклы), использование стека для корректного функционирования рекурсивных подпрограмм, основы конструирования программ, конструирование для верификации, стандарты в конструировании, особенности использования библиотек динамической компоновки, отличие статической компоновки от динамической, библиотека среды выполнения С (CRT), стандартная библиотека С++. Особое внимание следует обратить на понятие доказательства частичной правильности, полностью правильной программы, методы доказательства правильности программ, сущность метода индуктивных утверждений, характеристику формальных методов доказательства.

Раздел 5. Знакомство со средой DELPHI

5.1. Среда разработки DELPHI, панель инструментов, окно формы

При освоении материалов по данной теме следует рассмотреть среду DELPHI, предполагающую использование визуального проектирования и событийного программирования (программирование процедур обработки событий визуальных компонентов), применение которых позволяет существенно упростить процесс разработки и создания приложений, изучить назначение кнопок панели инструментов и окно формы, имеющее заголовок, кнопки оконного меню, размерную рамку.

5.2. Окно редактора, панель компонентов, окно менеджера проекта

При освоении материалов по данной теме необходимо обратить внимание на встроенный в среду редактор кода, где прописываются на языке DELPHI детали программы - процедуры обработки событий, палитру компонентов, имеющую ряд страниц, на которых скомпонованы все компоненты, предопределенные в Delphi и окно менеджера проекта.

5.3. Инспектор объектов, окно структуры проекта, основы визуального программирования

При освоении материалов по данной теме следует изучить инспектор объектов (Object Inspector), отображающий выбранный на форме компонент и его тип, список его свойств (Properties) и событий (Events), окно структуры проекта, рассмотреть основы визуального программирования.

Раздел 6 Использование объектно-ориентированного программирования в среде DELPHI

6.1. Общие сведения о классах: записи классов, работа с классами

При освоении материалов по данной теме следует обратить внимание на характеристику концепции объектно-ориентированного программирования, понятие класса, переменные (поля или свойства), соответствующие классу типы (объектные типы), принятое в DELPHI соглашение объявления класса и что, для использования класса в программе, нужно, как минимум, объявить переменную этого типа. Особое внимание следует обратить на существующую четкую границу между понятиями класс и объект.

6.2. Свойства объектов и инкапсуляция, методы, их наследование и полиморфизм

При освоении материалов по данной теме следует рассмотреть свойства объектов как высокоуровневые атрибуты компонентов класса, методы, их отличие от обычных процедур и функций, основные принципы (инкапсуляция, полиморфизм, наследование), на которых базируется концепция объектно-ориентированного программирования, их понятие и сущность

Раздел 7. Обзор основных визуальных компонентов

7.1. Создание формы с помощью компонентов TLabel, TEdit, TMemo, установление им свойств

При освоении материалов по данной теме следует изучить свойства компонентов TLabel, TEdit, TMemo находящихся на вкладке Standard. Особое внимание следует обратить на компонент TEdit, для работы с которым необходимо использовать функции перевода из целочисленной переменной в строковую, перевода переменной из строкового типа в целочисленный, перевода из вещественной переменной в строковую, перевода переменной из строкового типа в вещественный.

7.2. Создание формы с помощью компонентов TButton, TCheckBox, TRadioButton, установление им свойств

При освоении материалов по данной теме следует изучить свойства компонентов TButton, TCheckBox, TRadioButton, находящихся также на вкладке Standard. Особое внимание следует обратить на свойство Состояние флажка (Checked) и Признак доступности кнопки (Enabled) компонента TCheckBox.

7.3. Создание формы с помощью компонентов TListBox, TComboBox, TStringGrid, установление им свойств

При освоении материалов по данной теме следует изучить свойства компонентов TListBox, TComboBox, TStringGrid, находящихся на вкладке Standard. Особое внимание следует обратить на компонент TComboBox, представляющий собой комбинацию поля для редактирования и списка, возможности формирования списка, отображаемого в поле компонента, во время создания формы или во время работы программы.

Раздел 8. Работа с базами данных в DELPHI

8.1. Подключение баз данных к DELPHI. Визуальные и не визуальные компоненты. Компоненты TADOConnection, TADODataSet, TDataSource, установление свойств этим компонентам

При освоении материалов по данной теме следует изучить визуальные и не визуальные компоненты; свойства не визуальных компонентов TADOConnection, обеспечивающего соединение с базой данных (источником данных; TADODataSet, представляющего собой данные, полученные от источника данных, в результате выполнения SQL-запроса; TDataSource, используемого для связи между данными, рассмотреть механизм взаимодействия этих компонентов. Особое внимание следует обратить при подключении базы данных к DELPHI на свойство ConnectionString (строка соединения) компонента TADOConnection.

8.2. Механизмы доступа к данным, компоненты доступа к данным. Компоненты, обеспечивающие отображение данных.

При освоении материалов по данной теме следует обратить внимание на последователь-

ность установления свойств компонентам (TADOConnection, TADODataSet, TDataSource), обеспечивающим доступ к данным, компоненты TDBGrid, TDBText, TDBEdit, TDBMemo, обеспечивающие отображение и редактирование данных. Особое внимание следует обратить на компонент TDBGrid, который обеспечивает отображение данных в режиме таблицы.

8.3. Выбор информации из базы данных, SQL-запрос, способы создания запросов, работа с фильтрами, работа с базой данных в режиме формы

При освоении материалов по данной теме следует освоить SQL-команду Select, обеспечивающую выбор информации из таблицы базы данных; применение шаблонов в критерии запроса (при сравнении строк) вместо конкретного значения; использование компонентов TEdit и TComboBox как полей ввода критерия отбора; режимы отображения данных (таблица и форма); изучить компоненты (TDBEdit, TDBMemo), обеспечивающие просмотр и редактирование полей записей базы данных, установление свойств этим компонентам; рассмотреть компонент TDBNavigator, обеспечивающий перемещение указателя текущей записи к следующей, предыдущей, первой или последней записи, а также выполняющий другие операции в результате выбора соответствующей кнопки.

2. УКАЗАНИЯ ПО ПОДГОТОВКЕ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ

2.1. Общие сведения

Практические занятия – вид учебных занятий, ориентированный на практическое усвоение материала с помощью приборов, инструментов, технических средств обучения, компьютеров и другого специального оборудования.

Обучающая функция практических занятий заключается в освоении студентом практических навыков разработки и реализации приложений из будущей профессиональной деятельности студентов.

Развивающая функция практических занятий реализуется через ориентацию студента на самостоятельное решение отдельных проблем из будущей профессиональной деятельности с помощью специальных методов и инструментов реализации приложений.

Воспитательная функция практических занятий заключена в тесном контакте преподавателя с каждым студентом, позволяющем максимально эффективно воздействовать на мировоззрение студента, на формирование у студентов навыков культуры общения и чувства корпоративной этики.

Организирующая функция практических занятий предусматривает управление самостоятельной работой студентов как в процессе практических занятий, так и после них. В ходе практических занятий осваиваются технологии разработки и реализации программного обеспечения и приложений, которые создают базис для дальнейшей самостоятельной работы студентов, для формирования навыков исследовательской работы, для генерации новых знаний через использование различного рода информационных ресурсов.

Практические занятия по дисциплине «Высокоуровневые методы информатики и программирования» проводятся по подгруппам в компьютерных классах.

Цель практических занятий по дисциплине «Высокоуровневые методы информатики и программирования» заключается в установлении связей теории с практикой в форме экспериментального подтверждения положений теории; обучении студентов умению разрабатывать приложения, анализировать полученные результаты; проведении контроля самостоятельной работы студентов по освоению курса; обучении навыкам профессиональной деятельности.

Основными структурными элементами практических занятий являются:

- обсуждение преподавателем совместно со студентами темы занятий с пояснением ее взаимосвязи с будущей профессиональной деятельностью;
- освоение технологии разработки и реализации программного обеспечения и приложений;
- самостоятельная разработка и реализация программного обеспечения и приложений;
- консультации преподавателя во время занятий;
- обсуждение и оценка полученных результатов;
- письменный или устный отчет студентов о выполнении заданий;
- текущий контроль знаний.

Проведение практических занятий должно осуществляться в соответствии с рабочей программой учебной дисциплины и календарным планом, разрабатываемым ведущим курса.

2.2. Особенности освоения отдельных тем

Раздел 1. Методология разработки программных средств

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Основные этапы решения задач на ЭВМ. Понятие о структурном программировании». Необходимо самостоятельно освоить таблицу знаков операций и стандартных функций алгоритмического языка, правила записи арифметических выражений в алгоритмическом языке и записать математические выражения по правилам программирования.

Раздел 2. Способы записи алгоритма

2.1. Понятие алгоритма и его свойства, разновидности структур алгоритмов.

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Понятие алгоритма и его свойства, разновидности структур алгоритмов». Необходимо

самостоятельно повторить типы алгоритмических процессов и блок-схемы этих типов.

2.2. Описание линейных и разветвляющихся структур. Полное и неполное ветвление.

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Описание линейных и разветвляющихся структур. Полное и неполное ветвление». Необходимо самостоятельно повторить описание линейных и разветвляющихся структур, обратить внимание на полное и неполное ветвление и выполнить следующие задания: составление блок-схем линейных структур, составление блок-схем разветвляющихся структур.

2.3. Алгоритмизация циклических вычислительных процессов, Цикл с предусловием, цикл с постусловием, цикл с параметром.

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Алгоритмизация циклических вычислительных процессов, Цикл с предусловием, цикл с постусловием, цикл с параметром». Необходимо самостоятельно повторить описание циклических структур, обратить внимание на цикл с параметром, цикл с предусловием, цикл с постусловием и выполнить следующие задания: составление блок-схем циклической структуры (цикл с параметром, цикл с предусловием, цикл с постусловием).

Раздел 3. Программирование структур данных

3.1. Структура программы на языке Pascal, алфавит языка, идентификаторы, константы, выражения, операции.

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Структура программы на языке Pascal, алфавит языка, идентификаторы, константы, выражения, операции». Необходимо самостоятельно повторить структура программы, основу языка программирования Pascal для составления простейших задач на Pascal по теме «Линейные алгоритмы».

3.2. Простейшие операторы ввода-вывода, форматы ввода и вывода, присвоения и расчет числовых значений.

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Простейшие операторы ввода-вывода, форматы ввода и вывода, присвоения и расчет числовых значений». Необходимо самостоятельно повторить формат и работу стандартных процедур Read и Write для ввода и вывода данных и выполнить задания по теме «Операторы ввода-вывода».

3.3. Стандартные типы данных. Описание типов данных. Стандартные операции над типами данных.

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Стандартные типы данных. Описание типов данных. Стандартные операции над типами данных». Необходимо самостоятельно повторить стандартные типы данных, арифметические операции, стандартные (встроенные) функции для составления простейших задач на Pascal.

3.4. Представление основных структур программирования: операторы if, case, for, while, repeat-until.

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Представление основных структур программирования: операторы if, case, for, while, repeat-until». Необходимо самостоятельно повторить правила работы условного оператора, оператор варианта Case, инструкции For, While, Repeat-until и выполнить задания по темам: «Ветвление», «Условный оператор IF», «Циклические алгоритмы. Циклы с заданным числом итераций», «Циклические алгоритмы. Циклы с постусловием и предусловием», «Одномерные массивы», «Двумерные массивы».

3.5. Процедуры и функции, типы данных определяемые пользователем.

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Процедуры и функции, типы данных определяемые пользователем». Необходимо самостоятельно повторить виды подпрограмм, виды и вызов процедур, формальные и фактические параметры, структуру функции, строковый тип данных, процедуры и функции обработки строк и выполнить задания по темам «Процедуры и функции», «Строковый тип данных. Работа

со строками».

3.6. Структурированные типы данных, файлы, виды файлов, процедуры чтения и записи файлов, доступ к файлам.

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Структурированные типы данных, файлы, виды файлов, процедуры чтения и записи файлов, доступ к файлам». Необходимо самостоятельно повторить файловый тип Паскаля, виды файлов Паскаля, основные процедуры и функции для работы с файлами, процедуры и функции обработки файлов и выполнить задание по теме «Файлы».

3.7. Динамические структуры данных, списки, создание списков, выборка необходимых значений.

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Динамические структуры данных, списки, создание списков, выборка необходимых значений». Необходимо самостоятельно повторить ссылочный тип или тип указателей, объявление указателей, операции с указателями, описание списка, формирование и просмотр списка и выполнить задания по теме «Списки».

Раздел 4. Программирование сложных программных комплексов

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по темам «Программирование сложных программных комплексов». Необходимо самостоятельно повторить технологии разработки модульного программирования, объектно-ориентированного программирования, технологии создания программ и выполнить задания: по темам «Модульное программирование», «Модуль crt», «Graph», «Составление программ с помощью рекурсии».

Раздел 5. Знакомство со средой DELPHI

5.1 Среда разработки DELPHI, панель инструментов, окно формы

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Первый запуск среды Delphi». Необходимо самостоятельно выполнить команду запуска DELPHI и после загрузки изучить интерфейс DELPHI: главное окно, расположенные в нем строку заголовка, строку меню, его основные команды, панель инструментов, назначение ее кнопок.

5.2 Окно редактора, панель компонентов, окно менеджера проекта

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Первый запуск среды Delphi». Необходимо самостоятельно выполнить команду запуска Delphi и после загрузки изучить интерфейс Delphi: окно редактора кода, палитру компонентов, окно менеджера проекта.

5.3 Инспектор объектов, окно структуры проекта, основы визуального программирования

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по темам «Структура проекта Delphi», «основы визуального программирования». Необходимо самостоятельно выполнить команду запуска DELPHI и после загрузки изучить окно инспектора объектов, окно структуры проекта и освоить основы визуального программирования

Раздел 6. Использование объектно-ориентированного программирования в среде DELPHI

Лабораторные занятия не предусмотрены

Раздел 7. Обзор основных визуальных компонентов

7.1 Создание формы с помощью компонентов TLabel, TEdit, TMemo, установление им свойств

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Компоненты Delphi». Необходимо самостоятельно рассмотреть свойства компонентов TLabel, TEdit, TMemo вкладки Standard палитры компонентов и разработать приложение в среде Delphi с использованием этих компонентов.

7.2 Создание формы с помощью компонентов TButton, TCheckBox, TRadioButton, установление им свойств

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Компоненты Delphi». Необходимо самостоятельно рассмотреть свойства компонентов TButton, TCheckBox, TRadioButton вкладки Standard палитры компонентов и разработать приложения в среде Delphi с использованием этих компонентов.

7.3 Создание формы с помощью компонентов TListBox, TComboBox, TStringGrid, установление им свойств

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Компоненты Delphi». Необходимо самостоятельно рассмотреть свойства компонентов TListBox, TComboBox, TStringGrid вкладки Standard палитры компонентов и разработать приложения в среде Delphi с использованием этих компонентов.

Раздел 8. Работа с базами данных в DELPHI

8.1 Подключение баз данных к DELPHI. Визуальные и не визуальные компоненты. Компоненты TADOConnection, TADODataSet, TDataSource, установление свойств этим компонентам

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Использование баз данных в программирование на DELPHI». Необходимо самостоятельно изучить визуальные и не визуальные компоненты и их свойства, создать с помощью MicroSoft Access базу данных и с использованием компонента TADOConnection подключить базу данных к Delphi, установив для подключения необходимые свойства этому компоненту, указать, если для доступа к базе данных необходим пароль и идентификатор пользователя, присвоить значение False свойству LoginPrompt, если для доступа к базе данных пароль не нужен, необходимо_ проверить подключение (соединение с базой данных настроено правильно).

8.2 Механизмы доступа к данным, компоненты доступа к данным. Компоненты, обеспечивающие отображение данных

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по теме «Использование баз данных в программирование на DELPHI». Необходимо самостоятельно разработать и создать приложение с использованием баз данных, для этого необходимо подключить базу данных (пункт 4.1), затем приступить к настройке компонентов TADODataSet и TDataSource, установив им необходимые свойства, добавить и настроить для отображения данных компоненты TDBGrid (режим таблицы), TDBText, TDBEdit, TDBMemo (режим формы).

8.3 Выбор информации из базы данных, SQL-запрос, способы создания запросов, работа с фильтрами, работа с базой данных в режиме формы

Для выполнения задания по данной теме следует вспомнить материалы лекционного курса по темам «Работа с базами данных. Составление запросов», «Работа с базой данных в режиме формы». Необходимо самостоятельно разработать и создать приложение (многостраничную форму) с использованием баз данных, для этого необходимо выполнить подключение базы данных (пункт 4.2) и на первую страницу для отображения данных добавить компоненты TDBEdit и TDBNavigator (режим формы); на последующие страницы – компонент TDBGrid; для поиска данных использовать компонент TEdit; написать процедуру обработки события (нажатие кнопки), позволяющую выбирать нужную информацию.

3. УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа при изучении дисциплины складывается из самостоятельной работы на аудиторных занятиях и внеаудиторной самостоятельной работы.

Самостоятельная работа предполагает широкое использование различных источников информации (учебников и учебных пособий, специальной научной и научно-популярной литературы, ресурсов глобальной сети Интернет, материалов личных наблюдений и умозаключений и т.д.).

Связь студента с преподавателем при необходимости и в ходе самостоятельной работы может осуществляться по электронной почте, адрес которой преподаватель должен дать студенту на первом же занятии.

Основными видами самостоятельной работы при изучении дисциплины «Высокоуровневые методы информатики и программирования» являются:

- самостоятельная подготовка к практическим занятиям через проработку лекционного материала по соответствующей теме;
- самостоятельное изучение тем теоретического курса, не вошедших в лекционный материал;
- самостоятельное изучение тем практических занятий;
- систематизация знаний путем проработки пройденных лекционных материалов по конспекту лекций, учебникам и пособиям на основании перечня экзаменационных вопросов, тестовых вопросов по материалам лекционного курса и базовых вопросов по результатам освоения тем, вынесенных на практические занятия, приведенных в Фонде оценочных средств по дисциплине;
- подготовка к текущему и итоговому контролю;
- самостоятельное решение задач по заранее освоенным алгоритмам,
- выполнение контрольной работы.

Студенты всех форм обучения самостоятельно изучают все темы дисциплины на основе собственных конспектов лекций, раздаточного материала к лекциям, материалов компьютерных презентаций лекционного курса, основной и дополнительной литературы и других информационных ресурсов.

Все практические задания выполняются как на практических занятиях (в то числе и самостоятельно), так и вне аудиторий.

Систематизацию знаний необходимо осуществлять самостоятельно как в ходе отдельных аудиторных занятий, так и во время внеаудиторной работы. Систематизация знаний проводится на основе проработки собственных конспектов лекций, раздаточного материала к лекциям, материалов компьютерных презентаций лекционного курса, формирования отчета о выполняемых темах лабораторных занятий, изучения основной и дополнительной литературы и поиска необходимой информации в других информационных ресурсах.

В этой связи на каждом лабораторном занятии проводятся опросы студентов с целью как контроля самостоятельной работы, так и с целью побуждения к осознанной работе по целенаправленной систематизации знаний.

Важным аспектом при систематизации знаний являются консультации преподавателя, который на каждом занятии должен обращать внимание студентов на ключевые вопросы каждой темы и на взаимосвязь тем между собой.

4. ПОДГОТОВКА К ТЕКУЩЕМУ КОНТРОЛЮ ЗНАНИЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.1. Общие сведения

Целью текущего контроля знаний со стороны преподавателя является оценка качества освоения студентами данной дисциплины в течение всего периода ее изучения. К главной задаче текущего контроля относится повышение мотивации студентов к регулярной учебной работе, самостоятельной работе, углублению знаний, дифференциации итоговой оценки знаний.

Преподаватель, осуществляющий текущий контроль, на первом занятии доводит до сведения студентов требования и критерии оценки знаний по дисциплине. В целях предупреждения возникновения академической задолженности (либо своевременной ее ликвидации) преподаватель проводит регулярные консультации и иные необходимые мероприятия в пределах учебных часов, предусмотренных учебным планом.

При преподавании данной дисциплины предусматриваются следующие формы текущего контроля знаний: текущий контроль в форме индивидуальных опросов, текущий контроль в форме тестирования, текущий контроль в форме проверки контрольных работ и собеседования со студентом (для студентов заочной формы обучения).

Промежуточная аттестация проводится в форме сдачи зачета и экзамена.

Студент должен с первого занятия помнить, что по каждому разделу дисциплины будет проводиться тестирование по материалам теоретического курса, а по результатам выполненных тем лабораторных занятий будет производиться индивидуальный опрос.

Подготовка к текущему контролю и промежуточной аттестации происходит как в ходе отдельных аудиторных занятий, так и во время внеаудиторной работы.

По итогам выполнения заданий по каждой теме лабораторных занятий необходимо сформировать письменный отчет с результатами каждого задания. При подготовке к защите отчета (сдаче работы) необходимо самостоятельно повторить лекционный материал по данной теме и провести самоконтроль знаний на основании перечня вопросов для самоконтроля по отдельным темам, приведенных в Практикуме по моделированию социально-экономических систем и процессов.

После изучения каждого раздела учебной дисциплины подготовка к тестированию знаний проводится на основании тестовых вопросов, приведенных в Фонде оценочных средств по дисциплине.

К итоговому контролю следует готовиться на основании экзаменационных вопросов, приведенных в Фонде оценочных средств по дисциплине.

4.2. Текущий контроль знаний в форме индивидуальных опросов

Постоянный текущий контроль знаний (после изучения каждой темы и раздела) позволяет студенту систематизировать знания как в разрезе отдельных тем, так и отдельных разделов дисциплины. По итогам каждой темы лабораторных занятий должен быть сформирован отчет с результатами выполнения индивидуального задания. В ходе индивидуального опроса преподаватель должен проверить правильность выполнения задания и уровень освоения студентом данной темы. Вопросы для самоконтроля по отдельным темам лабораторных занятий приведены в Практикуме по моделированию социально-экономических систем и процессов. При индивидуальном опросе преподаватель обращает особое внимание на знание студентами алгоритмов решения экономико-математических задач, технологии разработки экономико-математических моделей, экономической интерпретации всех элементов экономико-математических моделей и результатов решения задач. По результатам опроса по каждой теме студенту выставляется оценка.

Критерии оценки знаний по отдельным темам:

- оценка «отлично» выставляется, если студент выполнил задание полностью и без ошибок, показал полные и глубокие знания по изученной теме, логично и аргументировано ответил на все вопросы по выполненному заданию;

- оценка «хорошо» выставляется, если студент выполнил задание полностью и без ошибок, твердо знает материал по данной теме, грамотно его излагает, не допускает существенных неточностей в ответе, достаточно полно отвечает на вопросы по выполненному заданию;
- оценка «удовлетворительно» выставляется, если студент выполнил задание полностью, но с незначительными ошибками, показал знание только основ материала по данной теме, усвоил его поверхностно, но не допускал при ответе на вопросы грубых ошибок или неточностей;
- оценка «неудовлетворительно» выставляется, если студент выполнил задание полностью, но с грубыми ошибками, не знает основ материала по данной теме, допускает при ответе на вопросы грубые ошибки или неточности.

Студент не аттестуется по данной теме, если задание по теме не выполнено или выполнено не полностью.

Если студент не аттестован хотя бы по одной из тем практических занятий или имеет оценку «неудовлетворительно», то преподаватель, ведущий практические занятия, имеет право не допустить студента до сдачи экзамена.

4.3. Текущий контроль знаний в форме тестирования

Тестирование - форма унифицированного контроля знаний, умений и навыков на основе тестов, стандартизированных процедур проведения тестового контроля, обработки, анализа и представления результатов. Тестирование как форма текущего контроля знаний используется по мере изучения отдельных разделов дисциплины. Также тестирование проводится и после изучения всего курса.

Вопросы тестов приведены в Фонде оценочных средств по дисциплине. Тестирование по разделам дисциплины и в целом по дисциплине проходит в соответствии с графиком тестирования, составляемым на основе календарных планов проведения аудиторных занятий.

На основании аттестации по отдельным темам лабораторных занятий и результатов тестирования преподаватель, ведущий практические занятия, выводит среднюю интегрированную оценку, которой он оценивает результаты освоения дисциплины каждым студентом.

4.4. Текущий контроль знаний в форме проверки контрольной работы и собеседования со студентом (для заочной формы обучения)

Контрольная работа учебным планом не предусмотрена.

4.5. Промежуточная аттестация в форме зачета

Критерием допуска к зачету является выполнение плана практических занятий. Вопросы, выносимые на зачет, приведены в Фонде оценочных средств по дисциплине.

4.6. Промежуточная аттестация в форме дифференцированного зачета по результатам защиты курсового проекта

Курсовой проект учебным планом не предусмотрен.

4.7. Промежуточная аттестация в форме экзамена

Экзамен учебным планом не предусмотрен.