

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ СОЦИАЛЬНЫЙ ИНСТИТУТ»

Утверждаю
Декан факультета

Ж.В. Игнатенко
« 21 » ИЮНЯ 20 20 г.



**Комплект оценочных материалов
по дисциплине**

Основы программирования

основной образовательной программы
по специальности СПО

09.02.03 Программирование в компьютерных системах

Базовый уровень подготовки

год начала подготовки – 2020

Рекомендовано
на заседании кафедры
от « 28 » 10 2020 г.
протокол № 2
Зав. кафедрой _____ Ж.В. Игнатенко

Согласовано
Заведующий выпускающей
кафедрой _____ Ж.В. Игнатенко

Одобрено
на заседании учебно-методической
комиссии факультета
от « 28 » 10 2020 г.
протокол № 2
Председатель УМК _____ Ж.В. Игнатенко

Ставрополь, 2020 г.

Комплект оценочных материалов разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности СПО 09.02.03 Программирование в компьютерных системах базового уровня и рабочей программы дисциплины «Основы программирования»

Разработчик(и):

АНО ВО СКСИ

канд.техн.наук, доцент кафедры
информационных систем и
сервиса

А.В. Чернышов

СОДЕРЖАНИЕ

1. Паспорт комплекта оценочных средств	4
2. Результаты освоения дисциплины, подлежащие проверке	5
3. Оценка освоения дисциплины:	6
3.1. Формы и методы оценивания	6
3.2. Типовые задания для оценки освоения учебной дисциплины.....	6
4. Контрольно-оценочные материалы для промежуточной аттестации по дисциплине	15

1. Паспорт комплекта оценочных средств

В результате освоения дисциплины «Основы программирования» обучающийся должен обладать предусмотренными ФГОС СПО по специальности 09.02.03 Программирование в компьютерных системах базового уровня подготовки следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями:

ПК 1.1. Выполнять разработку спецификаций отдельных компонент.

ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять оптимизацию программного кода модуля.

ПК 3.1 Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

Уметь

У1 работать в среде программирования;

У2 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Знать

31 этапы решения задачи на компьютере;

32 типы данных;

33 базовые конструкции изучаемых языков программирования;

34 принципы структурного и модульного программирования;

35 принципы объектно-ориентированного программирования

Формой аттестации по дисциплине является экзамен.

2. Результаты освоения дисциплины, подлежащие проверке

2.1. В результате аттестации по дисциплине осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих компетенций:

Таблица 1.1

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата	Форма контроля и оценивания
Уметь: работать в среде программирования		
ПК 1.1. ОК 1,2,8.	Умеет выполнять разработку спецификаций отдельных компонент. Умеет понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес. Умеет организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. Умеет самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации	Практическая работа Самостоятельная работа Тестирование
Знать: этапы решения задачи на компьютере		
ПК 3.1 ОК 2.	Знает как анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения Знает как организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество	Практическая работа Самостоятельная работа Тестирование
Уметь: реализовывать построенные алгоритмы в виде программ на конкретном языке программирования		
ПК 1.2. ОК 2.. ОК 3.	Умеет осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля. Умеет организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество Умеет принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	Практическая работа Самостоятельная работа Тестирование
Знать: типы данных		
ОК 7. ОК 8..	Знает как брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий. Знает как самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации	Практическая работа Самостоятельная работа Тестирование
Знать: базовые конструкции изучаемых языков программирования		
ПК 1.5. ОК 8..	Знает как осуществлять оптимизацию программного кода модуля. Знает как самостоятельно определять задачи	Практическая работа Самостоятельная работа

	профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации	Тестирование
Знать: принципы структурного и модульного программирования		
ПК 1.3. ОК 6	Знает как выполнять отладку программных модулей с использованием специализированных программных средств. Знает как работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями	Практическая работа Самостоятельная работа Тестирование
Знать: принципы объектно-ориентированного программирования		
ПК 1.4. ОК 5,9.	Знает как выполнять тестирование программных модулей. Знает как использовать информационно-коммуникационные технологии в профессиональной деятельности Знает как ориентироваться в условиях частой смены технологий в профессиональной деятельности.	Практическая работа Самостоятельная работа Тестирование

3. Оценка освоения дисциплины:

3.1. Формы и методы оценивания

Предметом оценки служат умения и знания, предусмотренные ФГОС по дисциплине «Основы программирования», направленные на формирование общих и профессиональных компетенций.

Проверяемые ОК, умения и знания	Форма контроля
Текущий контроль	
У1, У2	Практическая работа Самостоятельная работа Тестирование
31-35	Практическая работа Самостоятельная работа Тестирование
Промежуточная аттестация	
У1, У2 31-35	Экзамен

3.2. Типовые задания для оценки освоения учебной дисциплины

Типовое тестирование

Тест 1.

1. Стартовая форма проекта языка программирования Delphi:

1. LABEL
2. BUTTON
3. FORM

2. Поле вывода символов на поверхность формы:

1. LABEL
2. IMAGE
3. BUTTON

3. Поле ввода/редактирования строки символов:

1. EDIT
2. FORM
3. BUTTON

4. Командная кнопка:

1. BUTTON
2. EDIT
3. LABEL

5. Поле вывода иллюстраций в формате BMP:

1. LABEL
2. FORM
3. IMAGE

6. Какой тип данных является перечнем всех значений, которые может принимать переменная, относящаяся к данному типу?

- 1) Интервальный
- 2) Перечисляемый
- 3) Запись

7. Какой тип данных является отрезком или частью другого целого типа данных, называемого базовым?

- 1) Интервальный
- 2) Запись
- 3) Перечисляемый

8. Структура данных, состоящая из отдельных именованных компонентов разного типа, называемых полями - это:

- 1) Перечисляемый тип
- 2) Запись
- 3) Интервальный тип

9. Какая инструкция позволяет использовать в тексте программы имена полей без указания имени переменной-записи:

- 1) IF
- 2) FOR
- 3) WITH

10. Как называется переменная, значением которой является адрес другой переменной или структуры данных:

1. Определитель
2. Указатель
3. Распределитель

11. Как называется переменная, память для которой выделяется во время работы программы?

1. Строковая
2. Целая
3. Динамическая

12. Указатели и динамические переменные позволяют создавать сложные динамические структуры данных, которые называются:

- 1) Списки
- 2) Записи
- 3) Прописи

13. Сложная структура, включающая, помимо описания данных, описание процедур и функций, которые могут быть выполнены над объектом:

- 1) Список
- 2) Класс
- 3) Запись

14. Представитель класса, который в Delphi является динамической структурой:

1. Объект
2. Запись
3. Список

15. Процедуры и функции, объявление которых включено в описание класса, выполняющие действия над объектами:

- 1) Метод класса
- 2) Свойство класса
- 3) Действия класса

16. Скрытие полей объекта с целью обеспечения доступа к ним только посредством методов класса

- 1) Полиморфизм
- 2) Наследование
- 3) Инкапсуляция

17. Возможность порожденного класса (потомка) наследовать свойства и методы своего базового, родительского класса – это:

- 1) Порождение
- 2) Наследование
- 3) Добавление

18. Событие создания объекта (формы, элемента управления):

1. CLICK
2. CREATE
3. ENTER






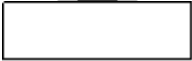


19. Событие происходит при появлении окна в начале работы графической программы

- 1) MOUSEUP
- 2) KEYPRESS
- 3) PAINT

20. Возможность использовать одинаковые имена для методов, входящих в различные классы:

1. Инкапсуляция
2. Полиморфизм
3. Наследование

Тест 2:

- 1.) Алгоритм – это
 - a. Совокупность шагов выполнения какого-либо действия, которые приводят в какому-либо результату, независимо от их очередности
 - b. Процесс решения задачи в виде последовательного выполнения некоторых простых шагов, который повторяется определенное количество раз
 - c. Процесс решения задачи, который должен включать команды, доступные для исполнителя
 - d. **Совокупность правил выполнения определенных действий, обеспечивающих решение задачи или достижение поставленной цели**
- 2.) Свойствами алгоритма **не** являются
 - a. Дискретность и понятность
 - b. Однозначность и результативность
 - c. **Формальность и системность**
 - d. Массовость и дискретность
- 3.) Способы задания алгоритмов
 - a. **Графический, словесный**
 - b. Алгоритмический, математический
 - c. Словесный, физический
 - d. Математический, формальный
- 4.) С помощью какой геометрической фигуры представляется команда «Вычисление» в графическом способе представления алгоритма
 - a. 
 - b. 
 - c. 
 - d. 
- 5.) С помощью какой геометрической фигуры представляется *начало цикла* в графическом способе представления алгоритма
 - a. 
 - b. 
 - c. 
 - d. 
- 6.) Какие существуют типы алгоритмов?
 - a. Прямой, обратный
 - b. Линейный, повторяющийся
 - c. Циклический, прямой
 - d. **Линейный, разветвляющийся**
- 7.) Для перехода из полноэкранного режима в оконный в программе Turbo Pascal используется комбинация клавиш
 - a. Shift + Enter
 - b. Alt + Shift
 - c. Ctrl + Shift
 - d. **Alt + Enter**
- 8.) Алфавит языка программы Turbo Pascal состоит из
 - a. Из латинских букв, дробных чисел, специальных символов

- b. Из русских букв, дополнительных символов и знаков
 - c. **Из латинских букв, арабских цифр, специальных символов**
 - d. Из целых и дробных чисел, дополнительных символов и знаков
- 9.) К каким стандартным типам относятся данные 345; -35,2; F; строка?
- a. Byte, integer, real, char
 - b. **Integer, real, char, string**
 - c. Real, real, char, char
 - d. Boolean, real, char, integer
- 10.) Где перечисляются все переменные, используемые в программе?
- a. В заголовке программы после слова Program
 - b. После служебного слова uses
 - c. **В разделе описания переменных – после слова var**
 - d. В теле программы, между служебными словами begin и end
- 11.) Какой комбинацией клавиш переключаются языки в программе Turbo Pascal?
- a. Клавиши Ctrl+Shift справа – английский, Ctrl+Shift слева – русский язык
 - b. Клавиши Ctrl+Shift – английский, Alt+ Shift – русский язык
 - c. **Клавиши Ctrl+Shift справа – русский, Ctrl+Shift слева – английский язык**
 - d. Клавиши Ctrl+ Alt слева – английский, Ctrl+ Alt справа – русский язык
- 12.) Какой оператор выводит информацию на экран?
- a. **Write(' '), writeln(' ')**
 - b. Writ(), writln()
 - c. Read(), readln()
 - d. Rid(' '), ridln(' ')
- 13.) Выберите оператор присваивания
- a. Real
 - b. Read
 - c. :=
 - d. Writeln
- 14.) Какой командой запускается программа?
- a. Debug – User Screen
 - b. Compile – Compile
 - c. **Run – Run**
 - d. File – Exit
- 15.) Выберите правильно написанный заголовок программы
- a. Program 1_Reshenie;
 - b. Program Reshenie-1;
 - c. **Program Reshenie1;**
 - d. Program Reshenie 1;
- 16.) Выберите правильный вид записи оператора
- a. **Writeln ('Введите числа через пробел');**
 - b. Real ('a,b');
 - c. Writeln (Сумма двух чисел равна, Summa);
 - d. 3*sqr(x):=A;
- 17.) Укажите правильную неполную форму условного оператора
- a. If <условие> then <оператор> else <оператор>
 - b. If < оператор > then < условие >
 - c. **If <условие> then <оператор>**
 - d. If <условие> then < условие >
- 18.) Что необходимо сделать, чтобы выполнялось несколько операторов после условия (if)?
- a. После служебного слова then перечислить в скобках операторы
 - b. После служебного слова else оператор написать в скобках

- c. **Операторы нужно взять в операторные скобки begin и end**
 - d. Перед словом then и после слова else добавить операторные скобки
- 19.) Что такое составной оператор?
- a. Группа операторов перед операторными скобками begin и end
 - b. Группа операторов после операторных скобок begin и end
 - c. **Группа операторов внутри операторных скобок begin и end**
 - d. Группа операторов в разделе описания переменных var
- 20.) Оператор выбора case используется для того, чтобы
- a. **Выбрать одно из нескольких возможных продолжений программы**
 - b. Выбрать последовательность этапов решения программы
 - c. Выбрать операторы, способные продолжить выполнение программы
 - d. Выбрать вариант, который определит ошибку в программе
- 21.) Цикл for удобно использовать тогда,
- a. Когда примерно известно количество повторений
 - b. **Когда точно известно количество повторений**
 - c. Когда неизвестно количество повторений
 - d. Когда повторений в условии задачи нет
- 22.) Циклический алгоритм – это
- a. Выполнение алгоритма программой в определенном порядке
 - b. **Последовательность шагов, повторяющихся при определенном условии**
 - c. Последовательные шаги, следующие друг за другом
 - d. Определенное количество шагов, которое приводит программу к завершению
- 23.) Какой формат записи имеет оператор for с шагом -1?
- a. For i := 1 to 10 do
 - b. For i := 1 dawnto 10 do
 - c. For i := 1 down to 10 do
 - d. **For i := 1 downto 10 do**
- 24.) Если условие верное, то цикл repeat . . . until продолжится?
- a. Да, работа цикла продолжается
 - b. **Нет, это условие на выход из цикла**
 - c. Да, если условие на продолжение верное
 - d. Да, если условие на продолжение ложное
- 25.) Чем отличается цикл while от цикла repeat?
- a. Цикл while – с постусловием, цикл repeat – с предусловием
 - b. **Цикл while – с предусловием, цикл repeat – с постусловием**
 - c. Цикл while – с параметром, цикл repeat – с постусловием
 - d. Циклы while и repeat – циклы с условием

Критерии и шкала оценивания тестирования на этапе текущего контроля

На этапе текущего контроля студентам на практических занятиях, предлагается выполнить тесты по темам дисциплины. Преподаватель определяет студентам исходные данные для подготовки к тестированию, называет темы, вопросы, по которым будут задания в тестовой форме, и источники информации для подготовки.

Каждому студенту отводится на тестирование время, соответствующее количеству тестовых заданий. До окончания теста студент может еще раз просмотреть все свои ответы на задания и при необходимости внести коррективы.

При прохождении тестирования пользоваться конспектами лекций, учебниками, и иными материалами не разрешено.

Оценка «5» ставится, если: студент выполняет правильно 86-100 % тестовых заданий.

Оценка «4» ставится, если: студент выполняет правильно 71-85 % тестовых заданий.

Оценка «3» ставится, если: студент выполняет правильно 51-70% тестовых заданий.

Оценка «2» ставится, если: студент выполняет правильно до 50% тестовых заданий

Типовая практическая работа Работа с формами.

Цель: Овладеть основными приемами работы с формами

Теоретическая часть

Форма - это компонент Form класса TForm. На ее основе конструируется приложение.

Существуют 2 класса форм:

1. Немодальные – те, которые позволяют переключаться в другую форму приложения без своего закрытия;
2. Модальные – те, которые требуют обязательного закрытия перед обращением к другой форме. Примером модальных форм могут служить стандартные диалоги ShowMessage, MessageDlg, InputBox.

Каждое приложение имеет одну главную форму и, возможно, несколько второстепенных. Главная форма загружается автоматически при запуске приложения. Чтобы сделать форму главной надо в файле проекта оператор, отвечающий за создание данной формы поставить на первое место.

Каждая форма имеет две области:

1. Клиентская область – та часть формы, в которой размещаются визуальные компоненты;
2. Неклиентская область – занята рамкой, заголовком формы и строкой главного меню.

Свойства и методы форм:

- visible – позволяет скрывать и отображать данную форму;
- left, top – задают координаты расположения формы на экране;
- width, height – задают размеры формы;
- active – определяет активность формы;
- clientwidth – возвращает ширину клиентской области;
- clientheight – возвращает высоту клиентской области;
- windowstate – определяет состояние отображаемой формы;
- show – отображает форму в немодальном режиме, при этом свойство visible автоматически устанавливается в True;
- showmodal – отображает форму в модальном режиме;
- hide – скрывает форму;
- close – закрывает форму (форма делается невидимой, но не уничтожается);
- destroy – уничтожает форму.

В приложении формы могут взаимодействовать друг с другом тремя способами:

1. Если одна форма выполняет действия с другой формой, то в списке Uses раздела Implementation модуля первой формы должна быть ссылка на модуль второй формы;
2. Delphi позволяет выполнить автоматизированную вставку ссылки. Для этого выбираем File/UseUnit и в появившемся диалоговом окне выбираем нужную форму;
3. Форма может выполнять различные действия с компонентами другой формы. В этом случае при обращении к нужному компоненту указывается форма, которой он принадлежит, и вставляется ссылка на модуль другой формы.

Задание к работе.

1. Написать приложение, состоящее из 2-х форм. На первой расположить:
 - два текстовых поля, в которых задаются координаты вывода второй формы;
 - кнопку «Форма 2», при нажатии на которую открывается вторая форма и располагается в заданном месте экрана;

- кнопку «Движение», при нажатии на которую Форма 2 осуществляет движение по горизонтали в одну и другую стороны до границ экрана.
2. Написать приложение, состоящее из 2-х форм. На первой расположить:
 - два текстовых поля, в которых задаются координаты вывода второй формы;
 - кнопку «Форма 2», при нажатии на которую открывается вторая форма и располагается в заданном месте экрана;
 - кнопку «Движение», при нажатии на которую Форма 2 осуществляет движение по вертикали вверх и вниз до границ экрана
 3. Написать приложение, состоящее из 2-х форм. На первой расположить:
 - два текстовых поля, в которых задаются размеры второй формы;
 - кнопку «Форма 2», при нажатии на которую открывается вторая форма заданного размера;
 - кнопку «Изменить», при нажатии на которую Форма 2 то увеличивает, то уменьшает размеры до определенных пределов. Визуальные компоненты, расположенные на Форме2, закрепить на ней таким образом, чтобы они перемещались вместе с границами формы.
 4. Написать приложение, состоящее из 2-х форм. На первой расположить:
 - четыре текстовых поля, в которых задаются максимально и минимально возможные размеры второй формы;
 - кнопку «Форма 2», при нажатии на которую открывается вторая форма заданного размера;
 - На второй форме расположить кнопку «Изменить», при нажатии на которую Форма1 то увеличивает, то уменьшает размеры до заданных пределов.
 5. Написать приложение, состоящее из 2-х форм. На первой расположить:
 - кнопку «Форма 2», при нажатии на которую открывается вторая форма, размещается в центре экрана и не допускается изменение размеров Формы2;
 - кнопку «Закрыть», при нажатии на которую Форма 2 закрывается;
 - кнопку «Уничтожить», при нажатии на которую Форма2 уничтожается.

Создать главную форму разместить на ней кнопки Button. Каждая кнопка соответствует номеру лабораторной работы и одна кнопка является выходом из приложения. Все компоненты Button должны иметь всплывающую подсказку, указывающую номер лабораторной работы.

Каждая лабораторная работа должна начинаться с загрузки формы на которой в соответствующих компонентах должны быть указаны тема, цель и задание к работе, а также реализованы 2 кнопки: запускающая программу и осуществляющая возврат к главной форме.

Содержание отчета:

1. Номер, тема, цель лабораторной работы.
2. Задание к работе.
3. Программа на языке программирования.
4. Результат выполнения программы.
5. Вывод о проделанной работе.

Критерии и шкала оценивания практических работ

При выполнении практических работ студентам необходимо выполнить всю работу согласно тексту задания. Результаты работы сохранить в файлах. После выполнения задания необходимо преподавателю продемонстрировать результаты работы и быть готовым ответить на вопросы и продемонстрировать выполнение отдельных пунктов задания. Защита практических работ осуществляется в дни и часы, устанавливаемые преподавателем.

Оценка «5» ставится, если: студент самостоятельно и правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя экономические понятия.

Оценка «4» ставится, если: студент самостоятельно и в основном правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя экономические понятия.

Оценка «3» ставится, если: студент в основном решил учебно-профессиональную задачу, допустил несущественные ошибки, слабо аргументировал свое решение, используя в основном экономическими понятиями.

Оценка «2» ставится, если: студент не решил учебно-профессиональную задачу.

Типовая самостоятельная работа

Самостоятельная работа приводит обучающегося к получению нового знания, упорядочению и углублению имеющихся знаний, формированию у него профессиональных навыков и умений.

Самостоятельная работа выполняет ряд функций:

- развивающую;
- информационно-обучающую;
- ориентирующую и стимулирующую;
- воспитывающую;
- исследовательскую.

Виды самостоятельной работы, выполняемые в рамках курса:

1. Проработка и повторение лекционного материала
2. Подготовка к практическим занятиям
3. Подготовка к аттестации

Обучающимся рекомендуется с самого начала освоения курса работать с литературой и предлагаемыми заданиями в форме подготовки к очередному аудиторному занятию. При этом актуализируются имеющиеся знания, а также создается база для усвоения нового материала, возникают вопросы, ответы на которые обучающийся получает в аудитории.

Можно отметить, что некоторые задания для самостоятельной работы по курсу имеют определенную специфику. При освоении курса обучающийся может пользоваться библиотекой вуза, которая в полной мере обеспечена соответствующей литературой. Значительную помощь в подготовке к очередному занятию может оказать имеющийся в учебно-методическом комплексе краткий конспект лекций. Он же может использоваться и для закрепления полученного в аудитории материала.

Самостоятельная работа предусматривает работу с источниками информации, указанным в рабочей программе дисциплины. Необходимо изучить дополнительный материал по темам:

- Языки и системы программирования
- Структурное программирование.
- Модульное программирование
- Основные принципы объектно-ориентированного программирования (ООП)
- Интегрированная среда разработки Delphi.
- Событийно-управляемая модель программирования. Компонентно-ориентированный подход.
- Категории компонентов. Свойства компонентов
- Проектирование объектно-ориентированного приложения
- Классы. Библиотека визуальных компонентов VCL и ее базовые классы.

Иерархия базовых классов

Критерии и шкала оценивания самостоятельной работы

Цель самостоятельной работы – содействие оптимальному усвоению студентами учебного материала, развитие их познавательной активности, готовности и потребности в самообразовании.

Качество выполнения внеаудиторной самостоятельной работы студентов оценивается посредством текущего контроля самостоятельной работы студентов.

Максимальное количество баллов **«отлично»** студент получает, если:

- студент свободно применяет знания на практике, не допускает ошибок в воспроизведении изученного материала, выделяет главные положения в изученном материале и не затрудняется в ответах на видоизмененные вопросы;

- весь объем программного материала усвоен полностью;

- обстоятельно с достаточной полнотой излагает соответствующую тему;

- материал (задание) оформлен аккуратно в соответствии с требованиями;

Оценку **«хорошо»** студент получает, если:

- студент знает весь изученный материал, отвечает без особых затруднений на вопросы преподавателя;

- применяет полученные знания на практике;

- в условных ответах не допускает серьезных ошибок, легко устраняет определенные неточности с помощью дополнительных вопросов преподавателя;

- неполно, но правильно изложено задание;

- может обосновать свой ответ, привести необходимые примеры;

- материал оформлен недостаточно аккуратно и в соответствии с требованиями.

Оценку **«удовлетворительно»** студент получает, если:

- студент обнаруживает освоение основного материала, но испытывает затруднения при его самостоятельном воспроизведении и требует дополнительных дополняющих вопросов преподавателя;

- предпочитает отвечать на вопросы воспроизводящего характера и испытывает затруднения при ответах на воспроизводящие вопросы;

- излагает выполнение задания недостаточно логично и последовательно;

- затрудняется при ответах на вопросы преподавателя;

- материал оформлен не аккуратно или не в соответствии с требованиями.

Оценку **«неудовлетворительно»** студент получает, если:

- у студента имеются отдельные представления об изучаемом материале, но все, же большая часть не усвоена;

- задание изложено неполно, без учета требований к оформлению;

- при изложении были допущены существенные ошибки, т.е. если оно не удовлетворяет требованиям, установленным преподавателем к данному виду работы.

4. Контрольно-оценочные материалы для промежуточной аттестации по дисциплине

Оценка освоения дисциплины предусматривает использование накопительной системы оценивания и проведение экзамена.

КОМ предназначен для контроля и оценки результатов освоения дисциплины Информационные технологии по специальности СПО 090203 Программирование в компьютерных системах базового уровня подготовки:

Умения

У1 работать в среде программирования;

У2 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Знания

- 31 этапы решения задачи на компьютере;
- 32 типы данных;
базовые конструкции изучаемых языков программирования;
- 34 принципы структурного и модульного программирования;
- 35 принципы объектно-ориентированного программирования

ЗАДАНИЯ ДЛЯ ЭКЗАМЕНУЮЩИХСЯ

Вариант1

Инструкция для обучающихся

Промежуточная аттестация по дисциплине проводится в форме устного экзамена по расписанию экзаменационной сессии.

Вопросы к экзамену доводятся до сведения студентов заранее.

Билет к экзамену содержит 3 вопроса.

При подготовке к ответу пользование учебниками, учебно-методическими пособиями, средствами связи и электронными ресурсами на любых носителях запрещено.

Время на подготовку ответа – от 30 до 45 минут.

По истечении времени подготовки ответа, студент отвечает на вопросы экзаменационного билета. На ответ студента по каждому вопросу билета отводится, как правило, 5-7 минут.

После ответа студента преподаватель может задать дополнительные (уточняющие) вопросы в пределах предметной области экзаменационного задания.

После окончания ответа преподаватель объявляет обучающемуся оценку по результатам экзамена, а также вносит эту оценку в экзаменационную ведомость, зачетную книжку.

Задание

1. Дать понятие технологии программирования. Охарактеризовать виды технологий программирования.
2. Дайте понятие структурного программирования. Перечислите основные структуры.
3. Написать приложение на языке Delphi, состоящее из 2-х форм:
 - два текстовых поля, в которых задаются координаты вывода второй формы;
 - кнопку «Форма 2», при нажатии на которую открывается вторая форма и располагается в заданном месте экрана;
 - кнопку «Движение», при нажатии на которую Форма 2 осуществляет движение по вертикали вверх и вниз до границ экрана

Типовые вопросы к экзамену

1. Дать понятие языка программирования. Охарактеризовать эволюцию языков программирования. Перечислить классификацию языков и основные классификационные признаки.
2. Дать понятие технологии программирования. Охарактеризовать виды технологий программирования.
3. Дать понятие системы программирования. Перечислить компоненты системы программирования и назвать их основные функции. Дать понятие исходного, объектного и загрузочного модулей.
4. Охарактеризовать методы программирования: структурный, модульный, объектно-ориентированный. Перечислить достоинства и недостатки методов программирования.
5. Дайте понятие жизненного цикла программного средства и перечислите стадии и фазы жизненного цикла. Назовите и принципы разработки программного обеспечения.
6. Дайте понятие алгоритма. Перечислите свойства алгоритмов и способы описания алгоритмов. Назовите и изобразите графически основные алгоритмические конструкции.

7. Охарактеризуйте логический тип данных. Перечислите логические операции, операции сравнения, применяемые к булевским операндам. Приведите примеры построения логических выражений с использованием операций сравнения и логических операций.
8. Укажите форматы условного оператора и расскажите принцип работы. Приведите примеры использования условного оператора и вложенного оператора условия.
9. Укажите форма топератора варианта и расскажите принцип работы. Приведите пример использования оператора варианта.
10. Укажите форматы оператора цикла с параметром, расскажите принцип работы. Приведите пример циклической программы с использованием цикла с параметром. Расскажите принцип работы вложенных циклов.
11. Охарактеризуйте строковый тип данных. Перечислите операции, применяемые к строковым переменным. Приведите примеры использования этих операций.
12. Перечислите стандартные процедуры и функции для обработки строк и укажите их форматы записи. Приведите примеры использования этих процедур и функций.
13. Дайте понятие структурного программирования. Перечислите основные структуры.
14. Дайте понятие объектно-ориентированного программирования. Назовите и охарактеризуйте основные принципы ООП.
15. Охарактеризуйте событийно-управляемое программирование.
16. Перечислите и охарактеризуйте средства интегрированной среды разработки Delphi.
17. Дайте общую характеристику интегрированной среды разработки (IDE) Delphi. Расскажите структуру окна среды Delphi.
18. Охарактеризуйте визуальную модель Delphi. Дайте понятие объекту и его характеристикам: событиям, методам, свойствам.
19. Дайте характеристику проекта Delphi, перечислите основные файлы проекта.
20. Перечислите и охарактеризуйте этапы разработки приложения в Delphi. Укажите структуру программы в Delphi.
21. Дайте понятие иерархии классов. Охарактеризуйте библиотеку визуальных компонентов VCL и ее базовые классы.
22. Назовите способы ввода с клавиатуры и вывода на экран данных в Delphi. Приведите примеры использования различных компонентов. Укажите назначение и формат функций преобразования.
23. Расскажите назначение и способы использования компонентов StringGrid и Memo для работы с массивами.
24. Расскажите назначение переключателей и списков. Охарактеризуйте компоненты CheckBox, RadioGroup, ListBox, ComboBox.
25. Расскажите назначение и основные свойства контейнера для группы элементов в Delphi.
26. Дайте понятие меню. Назовите виды меню. Укажите способы создания меню приложения в Delphi.
27. Охарактеризуйте страницу компонентов Dialogs. Укажите компоненты для создания типовых диалогов Windows.
28. Опишите основные средства создания мультимедийных приложений в Delphi. Опишите основные компоненты для работы с графикой в Delphi.

КРИТЕРИИ ОЦЕНКИ

Критерии оценивания на этапе экзамена по дисциплине

Экзамен по дисциплине проводится в виде ответа студента по вопросам билета.

Ответ студента на экзамене должен представлять собой связное, логически последовательное сообщение по вопросам билета, показывать его умение применять определения, правила в конкретных случаях.

Критерии оценивания:

- 1) полнота и правильность ответа;
- 2) степень осознанности, понимания изученного;
- 3) владение понятийным аппаратом.

Оценка «отлично» ставится, если:

- студент глубоко и всесторонне усвоил программный материал;
- уверенно, логично, последовательно и грамотно его излагает;
- опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью;
- умело обосновывает и аргументирует выдвигаемые им идеи;
- делает выводы и обобщения;
- свободно владеет системой понятий по дисциплине.

Оценка «хорошо» ставится, если:

- студент твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы;
- не допускает существенных неточностей;
- увязывает усвоенные знания с практической деятельностью;
- аргументирует научные положения;
- делает выводы и обобщения;
- владеет системой понятий по дисциплине.

Оценка «удовлетворительно» ставится, если:

- студент усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы;
- допускает несущественные ошибки и неточности;
- испытывает затруднения в практическом применении знаний;
- слабо аргументирует научные положения;
- затрудняется в формулировании выводов и обобщений;
- частично владеет системой понятий по дисциплине.

Оценка «неудовлетворительно» ставится, если:

- студент не усвоил значительной части программного материала;
- допускает существенные ошибки и неточности при рассмотрении проблем;
- испытывает трудности в практическом применении знаний;
- не может аргументировать научные положения;
- не формулирует выводов и обобщений.