

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет информатики и информационных технологий

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Объектно-ориентированное программирование

**Кафедра информационных систем и
технологий программирования**

Образовательная программа
09.03.02 Информационные системы и технологии

Профиль подготовки
Прикладная информатика в экономике

Уровень высшего образования
бакалавриат

Форма обучения

Очная

Статус дисциплины:
входит в часть ОПОП, формируемую участниками образовательных отношений

Махачкала, 2020

Рабочая программа дисциплины «Объектно-ориентированное программирование» составлена в 2020 году в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии (уровень бакалавриата) от 19.09.2017 № 922.

Разработчик: кафедра информационных систем и технологий программирования, доц. Баммаева Г.А.

Рабочая программа дисциплины одобрена:

на заседании кафедры ИИСТ от «26» 02 2020 г., протокол № 7
Зав.кафедрой Исмиханов З.Н.
(подпись)

на заседании Методической комиссии факультета ИИСТ
от «12» 03 2020 г., протокол № 8
Председатель Ахмедова З.Х.
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением «08» 03 2020 г. [подпись]
(подпись)

Аннотация рабочей программы дисциплины

Дисциплина «Объектно-ориентированное программирование» входит в вариативную часть образовательной программы бакалавриата по направлению 09.03.02 – Информационные системы и технологии.

Дисциплина реализуется на факультете Информатики и информационных технологий кафедрой Информационных систем и технологий программирования.

Содержание дисциплины охватывает круг вопросов, связанных с общими принципами построения и использования языков программирования; средствам описания данных; средствам описания действий; абстрактным типам данных. Задачи дисциплины – дать знания основ объектно-ориентированного программирования, алгоритмизации и средств описания данных, а также технологии программирования.

Дисциплина нацелена на формирование следующих компетенций выпускника: профессиональных - ПК-2, ПК-6.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические занятия, лабораторные занятия, самостоятельная работа.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме устного опроса, и промежуточный контроль в форме контрольной работы.

Объем дисциплины 2 зачетных единиц, в том числе в академических часах по видам учебных занятий

Семес тр	Учебные занятия						СРС	Форма промежуточной аттестации (зачет, дифференциро ванный зачет, экзамен
	в том числе							
	Контактная работа обучающихся с преподавателем							
	Все го	из них						
Лекц ии		Лаборатор ные занятия	Практич еские занятия	КСР	консульт ации			
2	72	18	36	18	36		72	экзамен

1. Цели освоения дисциплины

Целью преподавания дисциплины «Объектно-ориентированное программирование» является обучить студентов общим принципам построения и использования языков программирования; средствам описания данных; средствам описания действий; абстрактным типам данных.

Задачи дисциплины – дать знания основ объектно-ориентированного программирования, алгоритмизации и средств описания данных, а также технологии программирования.

Объектно-ориентированное программирование является мощным средством, позволяющим как моделировать объекты реального мира, так и генерировать виртуальные объекты, существующие только в электронной среде.

Лабораторные занятия должны включать рассмотрение конкретных приемов как для создания отдельных объектных классов, так и построения среды с объектами различных классов, в структуру которых инкапсулированы возможности обмена данными и управления как между собой, так и с внешними объектами.

В качестве примера функционирования и расширения объектно-ориентированной среды на лабораторных занятиях рассматривается среда Windows, её классы и интерфейсы. В результате студент должен уметь пользоваться готовыми объектно-ориентированными конструкциями, расширяя их структуру и функционал и уметь разрабатывать в стиле объектного подхода собственные конструкции произвольной структурной сложности.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Объектно-ориентированное программирование» входит в вариативную часть дисциплин по выбору учебного плана направления 09.03.02 Информационные системы и технологии.

Знание дисциплины «Объектно-ориентированное программирование» является важной составляющей общей программистской культуры и навыков программирования выпускника. Эти знания необходимы как при проведении теоретических исследований в различных областях математики, так и при решении практических задач из разнообразных прикладных областей, таких, как математический анализ; дискретная математика; информатика и программирование; информационные системы и технологии.

Список дисциплин, знание которых необходимо для изучения курса данной дисциплины.

Для успешного освоения данного курса студент должен иметь элементарные знания по школьному по дисциплине «Информатика и программирование».

Список дисциплин, для изучения которых необходимы знания данного курса.

1. Разработка программных приложений
2. Корпоративные информационные системы
3. Бухгалтерские информационные системы

4. Программная инженерия

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).

Код и наименование компетенции из ФГОС ВО	Код и наименование индикатора достижения компетенций	Планируемые результаты обучения
ПК-2 способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение	ИД1.ПК-2.1. Знать: принципы разработки программного обеспечения, концепции и понятия объектно-ориентированного подхода к программированию, механизмы его реализации в языке программирования ИД1.ПК-2.2 Уметь: создавать приложения на различных языках программирования, использовать основные принципы объектно-ориентированного подхода при написании программ; проектировать и реализовывать программы со сложной иерархией классов и объектов. ИД1.ПК-2.3 Владеть: навыками анализа поставленных задач, проектирования и разработки приложений, приемами разработки программных комплексов для решения прикладных задач, методами использования современных технологий программирования, тестирования и документирования программных комплексов	Знать основы и особенности разработки внедрения и адаптации различных видов прикладного программного обеспечения. Уметь разрабатывать, внедрять и адаптировать различные виды прикладного программного обеспечения Владеть навыками разработки, внедрения и адаптации различных видов прикладного программного обеспечения
ПК-6 Способность программировать приложения и создавать программные прототипы решения прикладных задач.	ИД1.ПК-6.1. Знать: проблемы и процессы анализа предметной области программных решений современные подходы анализа предметной области программных решений. ИД1.ПК-6.2. Уметь: разрабатывать программные приложения для предметной области. ИД1.ПК-6.3. Владеть: практически	Знать методологии, применяемые для внедрения и адаптации прикладных ИС, основные настройки CASE-средств Уметь внедрять, адаптировать и настраивать CASE средства, а также осуществлять планирование процессов внедрения, адаптации и настройки прикладных ИС Владеть навыками внедрения, адаптации и настройки CASE-средств, планирования процессов внедрения и адаптации прикладных

	навыками использования языков программирования для создания программные прототипов решения прикладных задач	ИС
--	---	----

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 2 зачетные единицы, 72 академических часа.

4.2. Структура дисциплины.

№	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы и самостоятельная работа, в час.				Самостоятельная работа	Формы контроля
				лекции	Практические занятия	Лабораторные занятия	Контроль Самост. работа		
	Модуль 1								
1.	Теоретические основы ООП. Переход от структурного программирования к объектно-ориентированному.	1	1-2	2	2	2		2	Устный опрос Выполнение индивидуального задания
2.	Основные принципы и этапы ООП. Особенности ООП	1	3-4	2	2	2		2	Устный опрос Выполнение индивидуального задания
3.	Теоретические основы ООП	1	5-6	2	2	2		2	Устный опрос Выполнение индивидуального задания
4.	Объектно-ориентированные возможности современных языков программирования. Структура проекта Delphi.	1	7-8	2	2	2		2	Устный опрос Выполнение индивидуального задания
	Модуль 2								
5.	Особенности объектно-ориентированного программирования в Delphi.	1	9-10	2	2	2		2	Устный опрос Выполнение индивидуального задания
6.	«Создание собственных компонент	1	11-12	2	2	2		2	Устный опрос Выполнение индивидуального задания
7.	Обработка исключительных ситуаций. Классификация ошибок. Классы исключений.	1	13-14	2	2	2		4	Устный опрос Выполнение индивидуального задания
8.	«Создание DLL-библиотек в	1	15-16	4	2	4		4	Устный опрос

Delphi»								Выполнение индивидуального задания
Итого часов:			18	16	18		20	

4.3. Содержание дисциплины, структурированное по темам (разделам).

4.3.1. Содержание лекционных занятий по дисциплине

Модуль 1

ТЕМА 1. Теоретические основы ООП. ПК-2, ПК-6

1. Переход от структурного программирования к объектно-ориентированному.

2. Исторический аспект возникновения объектно-ориентированного подхода в программировании.

3. Принципы структурного программирования.

4. Структурная декомпозиция.

5. Виды структурирования данных.

6. Модульный подход.

ТЕМА 2. Теоретические основы ООП. ПК-2, ПК-6

1. Основные принципы и этапы ООП.

2. Принципы абстрагирования, ограниченного доступа, модульности, иерархичности, типизации, параллелизма, устойчивости.

3. Обзор этапов разработки программного обеспечения в стиле ООП.

4. Обзор реализации ООП в различных средах программирования.

ТЕМА 3. Теоретические основы ООП. ПК-2, ПК-6

1. Классы и объекты.

2. Объектная декомпозиция.

3. Объектные сообщения, классы.

4. Средства разработки и описания классов.

5. Ограничение доступа.

ТЕМА 4. Объектно-ориентированные возможности современных языков программирования. ПК-2, ПК-6

Структура проекта Delphi.

Модуль 2

ТЕМА 5. Особенности объектно-ориентированного программирования в Delphi. ПК-2, ПК-6

Основой объектно-ориентированного программирования является объект. Объект состоит из трёх основных частей:

1) Имя (например, автомобиль);

2) Состояние, или переменные состояния (например, марка автомобиля, цвет, масса, число мест и т. д.);

3) Методы, или операции, которые выполняют некоторые действия над объектами и определяют, как объект взаимодействует с окружающим миром. Класс - сложная структура, включающая в себя описание данных, процедуры и

функции, которые могут быть выполнены над объектом. Объекты называются экземплярами класса.

ТЕМА 6. Тема: «Создание собственных компонент» ПК-2, ПК-6

Основные этапы создания компонент.

Выбор базового класса.

Создание модуля компонента.

Тестирование и установка компонента Ресурсы компонента Основные этапы создания компонент Delphi предоставляет возможность программисту создать свой собственный компонент, поместить его на одну из вкладок палитры компонентов и использовать при разработке приложений точно так же, как и другие компоненты Delphi.

Процесс создания компонента может быть представлен как последовательность следующих этапов:

- Выбор базового класса
- Создание модуля компонента
- Тестирование компонента
- Добавление компонента в пакет компонентов

ТЕМА 7: Обработка исключительных ситуаций. ПК-2, ПК-6

1. Классификация ошибок.
2. Классы исключений.
3. Предотвращение и обработка ошибок.
4. Классификация ошибок

Ошибки, которые могут быть в программе, принято делить на три группы:

- синтаксические;
- ошибки времени выполнения;
- алгоритмические.

Тема 8: «Создание DLL-библиотек в Delphi» ПК-2, ПК-6

1. Понятие и особенности DLL-библиотек.
2. Структура проекта DLL.
3. Подключение DLL-библиотек.

Динамически подключаемые библиотеки (DLL - Dynamic Link Library) - это универсальный механизм интегрирования в разрабатываемую программу процедур и функций, написанных другими программистами и, в общем случае, на других, чем Object Pascal, языках программирования.

4.3.2. Содержание лабораторно-практических занятий по дисциплине.

Лабораторная работа №1. ПК-2, ПК-6

Интегрированная среда объектно-ориентированного программирования BORLAND DELPHI. Разработка приложений.

Лабораторная работа №2. ПК-2, ПК-6

Программирование линейных и разветвляющихся алгоритмов.

Лабораторная работа №3. ПК-2, ПК-6

Программирование циклических алгоритмов.

Лабораторная работа №4. ПК-2, ПК-6

Программирование с использованием подпрограмм.

Лабораторная работа №5. ПК-2, ПК-6

Использование визуальных компонентов для программирования массивов.

Лабораторная работа №6. ПК-2, ПК-6

Построение диаграмм и графиков функций.

5. Образовательные технологии

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

1. Компьютерные классы с набором лицензионного базового программного обеспечения для проведения лабораторных занятий;

Лекции с применением мультимедийных материалов, мультимедийная аудитория;

Тестирование в системе компьютерного адаптивного тестирования ИС "Деканат"-дистанционное взаимодействие с обучаемыми с целью управления процессом обучения и контроля полученных знаний.

ИС "Рейтинг студентов" - учет учебной деятельности студентов с использованием рейтингового метода оценивания.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

Учебно-методические материалы к курсу:

Учебно-методические материалы к курсу размещены на сайтах ДГУ www.Eor.dgu.ru и ftp.icc.dgu.ru

Самостоятельная работа студентов (СРС) включает контролируемую и внеаудиторную самостоятельную работу, направлена на повышение качества обучения, углубление и закрепление знаний студента, развитие аналитических навыков по проблематике учебной дисциплины, активизацию учебно-познавательной деятельности студентов и снижение аудиторной нагрузки. Часть программного материала выносится для самостоятельного внеаудиторного изучения с последующим текущим или итоговым контролем знаний на занятиях или экзамене. Контроль СРС и оценка ее результатов организуется как самоконтроль (самооценка) студента, а также как контроль и оценка со стороны преподавателя, например, в ходе собеседования. Баллы, полученные по СРС студентом, обязательно учитываются при итоговой аттестации по курсу. Формы контроля СРС включают: тестирование; устную беседу по теме с преподавателем; выполнение индивидуального задания и др.

Роль студента в СРС - самостоятельно организовывать свою учебную работу по предложенному преподавателем, методически обеспеченному плану. СРС по курсу учитывает индивидуальные особенности слушателей и включает не только задания, связанные с решением типовых задач, но также творческие

задания, требующие самостоятельно «добывать» знания из разных областей, группировать и концентрировать их в контексте конкретной решаемой задачи. Технология обучения предусматривает выработку навыков презентации результатов выполненного индивидуального задания и создание условий для командной работы над комплексной темой с распределением функций и ответственности между членами коллектива. Оценка результатов выполнения индивидуального задания осуществляется по критериям, известным студентам, отражающим наиболее значимые аспекты контроля за выполнением этого вида работ.

Самостоятельная работа студентов обеспечена наличием на факультете лабораторий: «Информационные технологии в экономике и образовании», «Сетевая безопасность» SECURITY-CISCO-3, «Криптографические системы», «Системы мониторинга информационной безопасности» и достаточным количеством ПЭВМ. В этой лаборатории студенты используют учебные материалы, расположенные в сети ДГУ и осуществляют поиск тематической информации в глобальной компьютерной сети Internet. К услугам студентов также ресурсы научно-технической библиотеки ДГУ, имеющей ЭБД литературных источников и ИПС для организации поиска по ней, ресурсы «Интернет центра» и «Вычислительного центра».

Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоёмкость, а.ч.	Формируемые компетенции
	Очная	
Текущая СРС		
работа с лекционным материалом, с учебной литературой	8	ПК-2 ПК-6
самостоятельное изучение разделов дисциплины	8	ПК-2 ПК-6
выполнение домашних заданий, домашних контрольных работ	8	ПК-2 ПК-6
подготовка к практическим занятиям	10	ПК-2 ПК-6
подготовка к контрольным работам, коллоквиумам, зачётам	8	ПК-2 ПК-6
Творческая проблемно-ориентированная СРС		
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	10	ПК-2 ПК-6
исследовательская работа, участие в конференциях, семинарах, олимпиадах	10	ПК-2 ПК-6
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей	10	ПК-2 ПК-6

на основе собранных данных		
Итого СРС:	72	

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы. Перечень компетенций с указанием этапов их формирования приведен в описании образовательной программы.

Коды компетенции из ФГОС ВО	Код и наименование индикатора достижения компетенций	Планируемые результаты обучения	Процедура освоения
ПК-2	Способность разрабатывать и адаптировать прикладное программное	<p>Знать: принципы разработки программного обеспечения, концепции и понятия объектно-ориентированного подхода к программированию, механизмы его реализации в языке программирования</p> <p>Уметь: создавать приложения на различных языках программирования, использовать основные принципы объектно-ориентированного подхода при написании программ; проектировать и реализовывать программы со сложной иерархией классов и объектов.</p> <p>Владеть: навыками анализа поставленных задач, проектирования и разработки приложений, приемами разработки программных комплексов для решения прикладных задач, методами использования современных технологий программирования, тестирования и документирования программных комплексов</p>	Устный опрос
ПК-6	ПК-6. Способность программировать приложения и создавать программные прототипы решения	<p>Знать: проблемы и процессы анализа предметной области программных решений современные подходы анализа предметной области программных решений.</p> <p>Уметь: разрабатывать</p>	Устный опрос

	прикладных задач.	программные приложения для предметной области. Владеть: практическими навыками использования языков программирования для создания программных прототипов решения прикладных задач	
--	-------------------	---	--

7.2. Типовые контрольные задания

Примерный перечень вопросов к промежуточному контролю или зачету по всему изучаемому курсу:

Вопросы для контроля модуль 1

1. Понятие технологии программирования
2. Элементы языка программирования
3. Стандартные типы данных в C++
4. Алфавит языка Delphi
5. Переменные и константы
6. Рекурсия

Тесты к модулю 1

1. Языки Pascal, BASIC и C являются _____ языками, в то время как Delphi является _____ языком.
2. В качестве образца по отношению к объекту выступает:
 1. метод;
 2. класс;
 3. операция;
 4. значение.
3. Какие из перечисленных ниже причин являются главными для использования объектно-ориентированных языков?
 1. возможность создания собственных типов данных;
 2. простота операторов объектно-ориентированных языков по сравнению с процедурными языками;
 3. наличие средств для автокоррекции ошибок в объектно-ориентированных языках;
 4. объектно-ориентированные программы легче концептуализируются.
4. Объединение данных и функций называется _____
5. Возможность выполнения оператором или функцией различных действий в зависимости от типа операндов называется _____
6. _____ - это коллекция переменных, скомбинированная с набором связанных функций

7. Термин «_____» не имеет отношения к объектно-ориентированному программированию:

1. Инкапсуляция;
2. Индексация;
3. Наследование;
4. Полиморфизм.

8. Какие из указанных утверждений истинны (перечислить):

1. Множественное наследование является частью любого языка ООП
2. ООП не применяется в интерпретирующих системах разработки
3. С помощью перегрузки можно расширять стандартный набор бинарных операций
4. Объект класса содержит данные, а также методы для работы с данными
5. Объект класса содержит данные, а также указатели на методы для работы с данными
6. Методы класса являются частью объекта и загружаются в память конструктором каждый раз при создании объекта класса
7. Методы класса хранятся в памяти только в одном экземпляре независимо от количества созданных объектов класса

9. Полиморфизм позволяет:

1. Использовать функциональность базового класса
2. Создавать методы с одинаковым именем, но с разной реализацией
3. Вызвать метод именно того класса, экземпляром которого является объект
4. Создавать параметризованные классы (шаблоны)

10. Какие модификаторы доступа к членам класса существуют в C++?

1. public
2. internal
3. final
4. protected
5. static
6. private

11. _____ имеет такое же отношение к _____, как стандартный тип данных к переменной этого типа.

12. Утверждение: поля класса должны быть закрытыми

1. утверждение истинно, так как данные должны быть скрытыми
2. утверждение ложно, так, как и данные, и функции могут быть как скрытыми, так и общедоступными

13. Операция *стрелка* (операция доступа к члену класса) объединяет следующие два элемента (слева направо):

1. член класса и объект класса

2. объект класса и член этого класса
3. объект класса и класс
4. класс и член этого класса

14. Имя конструктора совпадает с именем _____.

15. Пусть определены три объекта и один метод класса. В памяти содержится _____ копий полей класса и _____ копий метода класса

16. Классы полезны потому, что:

1. собирают вместе все аспекты, касающиеся отдельного реального объекта
2. адекватно моделируют объекты реального мира
3. не занимают памяти, если не используются
4. защищают свои данные от доступа со стороны других классов

17. Размер объекта класса в памяти определяется

1. суммой размеров методов класса и переменных-членов
2. суммой размеров методов класса
3. суммой размеров переменных-членов
4. не зависит от суммы размеров методов класса и переменных-членов
5. нет правильного ответа

18. Конструктор класса - это метод, который вызывается при создании объекта для (перечислить)

1. выделения памяти под динамические члены класса
2. выделения памяти под статические члены класса
3. инициализации свойств объекта
4. загрузки методов класса в память

19. Класс может содержать члены-свойства _____ типа.

1. базового
2. производного
3. пользовательского
4. произвольного

20. _____ - это специальная компонентная функция, которую можно явно включать в определение класса для инициализации объектов.

21. Сопоставьте:

1. Класс
 2. Объект
 3. Метод
 4. Структура
- a. -это принадлежащая классу функция.
 - b. -это экземпляр класса.

- c. -задает некоторую структурированную совокупность типизированных данных и позволяет определить набор операций над ними.
- d. -объединенное в единое целое множество поименованных элементов разных типов.

Вопросы для контроля модулю 2

- 1.Идентификаторы
- 2.Простые типы данных
- 3.Операции и выражения
- 4.Циклы for, while, repeat.
- 5.Одномерные массивы символов
- 6.Процедуры и функции для работы со строками

Тесты к модулю 2

1. Разделение программы на функции:
 1. является ключевым методом объектно-ориентированного программирования;
 2. упрощает представление программы;
 3. сокращает размер программного кода;
 4. ускоряет процесс выполнения программы.
2. Двумя основными компонентами объекта являются _____ и _____, которые манипулируют ими.
3. В C++ функцию, входящую в состав класса, называют:
 1. функция-член класса;
 2. оператор класса;
 3. функция класса;
 4. метод класса.
4. _____ отображают объекты реального мира точнее, чем функции.
5. Если язык обеспечивает возможность создания пользовательских типов данных, то говорят, что язык называется:
 1. наследуемым;
 2. инкапсулируемым;
 3. перегруженным;
 4. расширяемым.
6. Операция, выполняющая заданные действия над пользовательским типом данных, называется:
 1. полиморфической;
 2. инкапсулированной;
 3. классифицированной;
 4. перегруженной.
7. Все члены класса - данные и методы - являются по умолчанию
 1. открытыми

2. закрытыми
 3. защищёнными
 4. абстрактными
 5. виртуальными
8. Инкапсуляция обеспечивает возможность
1. вложения классов друг в друга
 2. сокрытия данных в теле класса
 3. наследования свойств и методов класса
 4. создания перегруженных функций
9. Наследование – это:
1. Включение в один объект экземпляра другого объекта
 2. Включение в один объект ссылки на другой объект
 3. Включение функциональности одного класса в другой
 4. Переопределение некоторых функций одного класса в другом
10. Состояние объекта определяется: (2)
1. Перечнем всех свойств данного объекта
 2. Перечнем всех свойств данного объекта и текущими значениями каждого из этих свойств
 3. Реализацией методов
 4. Модификаторами доступа к полям и методам объекта
11. Для чего необходимо определение класса?
1. определение класса описывает действия, которые будут выполняться над его элементами
 2. определение класса описывает, как будут выглядеть объекты после их создания
 3. определение класса создает объекты этого класса
 4. определение класса описывает взаимодействие его объектов с внешней средой
12. В определении класса члены класса с ключевым словом `private` доступны:
1. любой функции программы
 2. только открытым членам класса
 3. в случае, если вам известен пароль
 4. методам этого класса
13. Конструктор вызывается автоматически в момент _____ объекта.
14. Утверждение: класс может иметь более одного конструктора с одним и тем же именем
1. утверждение верно
 2. утверждение неверно
15. Методу класса всегда доступны данные:

1. класса, членом которого он является
 2. любого объекта класса, членом которого он является
 3. объекта, членом которого он является
 4. класса, объявленного открытым
16. Посылка сообщения объекту эквивалентна
1. доступу к конкретному полю объекта
 2. вызову одного из его методов
 3. вызову конструктора класса, объектом которого он является
 4. доступу целиком к объекту
17. Для инициализации переменных-членов класса применяется
1. деструктор
 2. параметр
 3. конструктор
 4. оператор присваивания
 5. начало объявления класса
18. _____ - другое имя уже существующего объекта.
1. идентификатор;
 2. ключевое слово;
 3. массив;
 4. указатель;
 5. ссылка.
19. При создании объекта класса в памяти выделяется область, достаточная для хранения
1. Всех свойств класса
 2. Всех методов класса
 3. Всех свойств и методов класса
 4. Всех свойства и указателей на методы класса
20. Ключевое слово `protected` означает, что
1. общедоступные члены, доступны для всех функций-членов;
 2. защищенные члены, доступны для функций-членов производных объектов;
 3. собственные члены, не доступны для внешних обращений;
 4. защищенные члены, доступны для всех функций-членов;
21. Язык C++ позволяет изменить видимость объектов с помощью операции _____
22. _____ - это производный структурированный тип, введенный программистом на основе уже существующих типов.
23. Атрибуты класса могут быть
1. только целыми числами
 2. любыми встроенными типами
 3. любого определенного в программе типа

24. Методы класса определяют

1. какие операции можно выполнять с объектами данного класса
2. какие значения может принимать переменная данного класса
3. каким образом можно создавать объекты данного класса

25. Все члены класса являются по умолчанию

1. абстрактными
2. внешними
3. внутренними
4. защищенными
5. виртуальными

26. Свойство объекта – это

1. совокупность данных объекта
2. совокупность данных и событий объекта
3. совокупность методов и событий объекта
4. совокупность данных и методов их чтения и записи

27. В Borland C++ Builder под объектом понимается

1. совокупность свойств, которые может иметь объект
2. совокупность свойств и методов, на которые может реагировать объект
3. совокупность свойств, методов и событий, на которые может реагировать объект
4. совокупность свойств и событий, на которые может реагировать объект
5. совокупность методов и событий, на которые может реагировать объект

28. Головной файл проекта

1. создается программистом для каждого приложения
2. содержит функцию WinMain
3. создается средой Borland C++ Builder для каждого приложения
4. содержит объявления, используемые в приложении
5. содержит обработчики различных событий, используемых в приложении

29. Среди перечисленных выберите правильные утверждения

1. В реализацию функций-элементов обязательно нужно добавлять ссылку на класс с помощью операции (::).
2. Функции-элементы объявляются и описываются с применением опции компилятора `_fastcall`
3. В реализацию функций-элементов не нужно добавлять ссылку на класс с помощью операции (::)
4. В функциях-элементах обращение к другим функциям-элементам и данным-элементам того же класса может осуществляться без указания на объект.
5. В функциях-элементах обращение к другим функциям-элементам и данным-элементам того же класса нужно осуществлять с указателем на

объект.

6. В функциях, не являющихся элементами класса данного объекта, доступ к методам и свойствам осуществляется через указатель на объект с помощью операции (->)
7. В функциях, не являющихся элементами класса данного объекта, доступ к методам и свойствам осуществляется без указателя на объект

30. На страницах Палитры компонентов размещены

1. визуальные компоненты
2. часто используемые компоненты
3. все компоненты как визуальные, так и невизуальные
4. компоненты, включенные в процессе установки Borland C++ Builder
5. компоненты, включенные как в процессе установки, так и добавленные самим программистом

7.4. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

а) Критерии оценивания компетенций (результатов). Программой дисциплины в целях проверки прочности усвоения материала предусматривается проведение различных форм контроля:

1. «Входной» контроль определяет степень сформированности знаний, умений и навыков обучающегося, необходимым для освоения дисциплины и приобретенным в результате освоения предшествующих дисциплин.

2. Тематический контроль определяет степень усвоения обучающимися каждого раздела (темы в целом), их способности связать учебный материал с уже усвоенными знаниями, проследить развитие, усложнение явлений, понятий, основных идей.

3. Межсессионная аттестация – рейтинговый контроль знаний студентов, проводимый в середине семестра.

4. Рубежной формой контроля является тестирование. Изучение дисциплины завершается контрольной работой, проводимой в виде письменного опроса с учетом текущего рейтинга. Неявка студента на промежуточный контроль в установленный срок без уважительной причины оценивается нулевым баллом.

Повторная сдача в течение семестра не разрешается. Дополнительные дни отчетности для студентов, пропустивших контрольную работу по уважительной причине, подтвержденной документально, устанавливаются преподавателем дополнительно.

Лабораторные занятия, пропущенные без уважительной причины, должны быть отработаны до следующей контрольной точки, если сдаются позже, то оцениваются в 1 балл.

Итоговой формой контроля знаний, умений и навыков по дисциплине является Экзамен. Экзамен проводится в форме тестирования. При соответствии ответа учащегося на экзамене 37 более чем 51 % критериев из этого списка выставляется оценка «удовлетворительно», 66% – 85% оценка

«хорошо», 86% и выше оценка «отлично».

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.

Основная литература

1. Бобровский С. И. Delphi 7: учеб. курс/ Санкт-Петербург: ПИТЕР, 2008 .-736 с.
2. Культин Н. Программирование в Delphi7. Издательство: БХВ: Петербург, 2010г. - 224 с.
3. Корняков В. Н., Программирование документов и приложений MS Office в Delphi., СПб.: БХВ-Петербург, 2005. — 496 с: ил.
4. Осипов Д. Базы данных и Delphi. Теория и практика. - СПб.: БХВ–Петербург, 2011. - 752 с.

Дополнительная литература

1. Баженова И.Ю. Delphi 7. Библия программиста. – М.: КУДИЦ-ОБРАЗ, 2012. – 448 с.
2. Белов В. В., Чистякова В. И. Программирование в Delphi: процедурное, объектно-ориентированное, визуальное. – М.: «Горячая Линия-Телеком», 2009. - 240 с.
3. Лавлинский В. В. , Коровина О. В. Технология программирования на современных языках программирования [Электронный ресурс] : учеб. пособие / В. В. Лавлинский, О. В.Коровина. – Электрон. текстовые дан. - Воронеж: Воронежская государственная лесотехническая академия, 2012. – Режим доступа: <http://biblioclub.ru/index.php?page=book&id=142453&sr=1> (дата обращения: 01.10.2014).
4. Объектно-ориентированное программирование [Электронный ресурс]: - Режим доступа: <http://www.intuit.ru/catalog/se/objectprog/>

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

1. <http://dgu.ru> – Портал доступа к электронным образовательным ресурсам ДГУ;
2. <http://biblioclub.ru> – Электронно-библиотечная система «Университетская библиотека онлайн»;
3. <http://znanium.com> – Электронно-библиотечная система издательства «Инфра».
4. <http://www.rsdn.ru/>
5. <http://www.intuit.ru>
6. <http://citforum.ru>

10. Методические указания для обучающихся по освоению дисциплины.

При освоении всех разделов дисциплины необходимо сочетание всех форм учебной деятельности: изучение лекционного материала, выполнение

заданий на практических занятиях и лабораторных работах, как с использованием компьютера, так и без него, самостоятельная работа с рекомендуемой литературой и использование методических указаний, консультации преподавателя при выполнении расчетно-графических работ.

После каждого лекционного занятия студенты должны повторить материал лекции по конспектам, а перед каждым очередным занятием - освежить в памяти материал предыдущего.

Самостоятельная работа ориентирует студентов на углубленное изучение и осмысление тем учебного курса.

При подготовке к лабораторной работе студент должен изучить рекомендуемые материалы. Если в задании на лабораторную работу есть непонятные неясные моменты, необходимо задать вопросы преподавателю.

По каждой лабораторной работе необходимо подготовить отчет, в котором отразить все основные действия, выполняемые в процессе лабораторной работы, а также результаты, полученные при выполнении лабораторной работы.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

Использование презентаций в виде слайдов, выполненных в Microsoft Office PowerPoint, Интернет-ресурсы, мультимедиа, электронная почта для коммуникации со студентами.

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Для выполнения лабораторных работ используется компьютерное оборудование с установленными программными продуктами MS Office, ABC Pascal, Borland Delphi, Microsoft Visual Studio, C++.

Аудиторные занятия проводятся в компьютерных классах с доступом к сети Интернет.

У студентов имеется доступ учебным лабораториям:

1. «Информационные технологии в экономике и образовании».
2. «Сетевая безопасность» SECURITY-CISCO-3.
3. «Криптографические системы».
4. «Системы мониторинга информационной безопасности».

Студентам также доступны ресурсы научно-технической библиотеки ДГУ, имеющей ЭБД литературных источников и ИПС для организации поиска по ней, а также ресурсы «Интернет центра» и «Вычислительного центра».

