

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ 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 технологии программирования

Образовательная программа

09.03.03 Прикладная информатика

Профиль подготовки

Прикладная информатика в экономике

Прикладная информатика в менеджменте

Уровень высшего образования

бакалавриат

Форма обучения

очная

Статус дисциплины -

входит в часть ОПОП, формируемую участниками образовательных
отношений

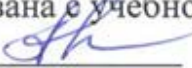
Махачкала, 2020

Рабочая программа дисциплины «Объектно-ориентированное программирование» составлена в 2020 году в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика (уровень бакалавриата) от 19.09.2017 № 922.

Разработчик: кафедра информационных систем и технологий программирования, ст. преп. Магомедова С.Р. 

Рабочая программа дисциплины одобрена:
на заседании кафедры ИСи ТП от «28» 02 2020 г., протокол № 7
Зав.кафедрой Исмиханов З.Н.
(подпись)

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

Рабочая программа дисциплины согласована с учебно-методическим управлением «16» 03 2020 г. 
(подпись)

Аннотация рабочей программы дисциплины

Дисциплина «Объектно-ориентированное программирование» входит в модуль профильной направленности образовательной программы бакалавриата по направлению 09.03.03 Прикладная информатика.

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 технологий программирования.

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

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

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

Объем дисциплины 5 зачетных единиц, в том числе в академических часах по видам учебных занятий

Семестр		Учебные занятия				Форма промежуточной аттестации	
		в том числе					
	Контактная работа обучающихся с преподавателем						СРС, в том числе экзамен
	всего	Из них					
		Лекции	Лабораторные занятия	Практические занятия	Консультация		

4	180	18	36	18		108	экзамен
---	-----	----	----	----	--	-----	---------

1. Цели освоения дисциплины

Целями освоения дисциплины «Объектно-ориентированное программирование» являются: подготовка квалифицированных специалистов, владеющих основами объектно-ориентированного программирования на языках высокого уровня с использованием современных сред разработки программ.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Объектно-ориентированное программирование» входит в модуль профильной направленности образовательной программы бакалавриата по направлению подготовки 09.03.03 Прикладная информатика.

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 экономики и менеджмента, так и при решении практических задач из разнообразных прикладных областей, таких, как информатика и программирование; информационные системы и технологии. Список дисциплин, для изучения которых необходимы знания данного курса: «Базы данных», «Программная инженерия», «Разработка и сопровождение программных приложений».

3. Компетенции обучающего, формируемые в результате освоения дисциплины

Код компетенции из ФГОС ВО	Код и наименование индикатора достижения компетенций	Планируемые результаты обучения
ПК-2 Способность разрабатывать и адаптировать прикладное	ПК-2.1. Демонстрирует знание основных принципов и технологий промышленной разработки программного обеспечения	<i>Знает:</i> принципы разработки программного обеспечения, концепции и понятия объектно-ориентированного подхода

программное обеспечение	ПК-2.2. Демонстрирует умения разрабатывать программное обеспечение с применением инструментов автоматизации сборки, интеграции, тестирования	к программированию, механизмы его реализации в языке программирования <i>Умеет:</i> 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. <i>Владеет:</i> 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
ПК-6	ПК-6.1. Демонстрирует	<i>Знает:</i> проблемы и

Способность программировать приложения и создавать программные прототипы решения прикладных задач.	знание принципов и шаблонов проектирования программного обеспечения, баз данных, программных интерфейсов, основ моделирования предметной области ПК-6.2. Демонстрирует умение выполнять проектирование компонентов программного обеспечения по заданным требованиям в рамках определенной предметной области	процессы анализа предметной области программных решений, современные подходы анализа предметной области программных решений <i>Умеет:</i> разрабатывать программные приложения для предметной области <i>Владеет:</i> практическими навыками использования языков программирования для создания программных прототипов решения прикладных задач
--	--	--

4. Объем, структура и содержание дисциплины

4.1. Объем дисциплины составляет 5 зачетных единиц, 180 академических часа.

4.2. Структура дисциплины.

№ п/ п	Названия разделов	Семестр	Неделя	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)	Самост оатель ная работа	Формы текущего контроля успеваем ости (по неделям
--------------	----------------------	---------	--------	---	-----------------------------------	--

				Лекции	Практические занятия	Лабораторные занятия	Контроль		семестра)
Модуль 1. . Основы объектно-ориентированного программирования на языке C++									
1	Методология объектно-ориентированного программирования	4		2	2	4		4	
2	Операторы языка C++	4		2	2	4		4	
3	Массивы, кортежи и строки	4		2	2	4		4	
	Итого за модуль:			6	6	12		12	
Модуль 2. Классы и объекты языка C++									
1	Классы и объекты. Понятие. Средства описания данных.	4		2	2	4		10	Устный опрос
2	Наследование и полиморфизм	4		2	2	4		10	тест
	Итого за модуль:			4	4	8		20	
Модуль 3. Интерфейсы и структурные типы									
1	Обработка исключений	4		2	2	4		10	Устный опрос
2	Интерфейсы	4		2	2	4		10	Устный опрос
	Итого за модуль:			4	4	8		20	
Модуль 4. Делегаты, события, потоки									
1	Делегаты, события	4		2	2	4		10	
2	Многопоточные приложения	4		2	2	4		10	
	Итого за модуль:			4	4	8		20	

Модуль 5. Подготовка к экзамену									
	Подготовка к экзамену	4					36		
	Итого по модулю:						36		
	Всего часов			18	18	36	36	72	

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

4.3.1. Содержание лекционных занятий по дисциплине

№ п/п	Наименование темы	Трудоемкость	Содержание	Формируемые компетенции	Результаты освоения (знает, Умеет, Владеет)	Технологии обучения
1	Методология объектно-ориентированного программирования	2	Основная концепция объектно-ориентированного подхода (ООП) к проектированию программного обеспечения.	ПК-2.1.	Знать концепции и понятия объектно-ориентированного подхода к программированию	Собеседование
2	Операторы языка C++	2	Состав языка. Комментарии. Переменные. Область видимости переменных, константы. Типы данных. Литералы. Преобразования типов. Арифметические операторы. Операторы отношения и логические операторы. Оператор присваивания. Поразрядные операторы.	ПК-2.1	Знать концепции и понятия объектно-ориентированного подхода к программированию	Устный опрос, тестирование

			Тернарный оператор. Условные операторы. Операторы цикла. Операторы перехода.			
3	Массивы, кортежи и строки	2	Массивы. Многомерные массивы. Ступенчатые (зубчатые) массивы. Класс Array. Массивы в качестве параметров. Кортежи. Строки. Форматирующие строки. Регулярные выражения.	ПК-2.1	Знать концепции и понятия объектно-ориентированного подхода к программированию	Устный опрос
4	Классы и объекты языка C++	2	Класс Object. Создание объектов. Методы. Конструкторы борка мусора и деструкторы. Ключевое слово this. Доступ к членам класса. Модификаторы параметров. Необязательные и именованные аргументы. Рекурсия. Ключевое слово static. Модификаторы доступа в	ПК-2.2	Знать способы и механизмы разработки программного обеспечения	Устный опрос

			аксессуарах. Иерархии классов.			
5	Наследование и полиморфизм	2	Основы наследования. Защищенный доступ и исключение наследования. Конструкторы и наследование. Наследование и сокрытие имен. Ссылки на базовый класс и объекты производных классов. Виртуальные методы, свойства и индексаторы. Абстрактные классы.	ПК-2.2.	Знать способы и механизмы разработки программного обеспечения	Устный опрос, тестиров ание
6	Обработка исключений	2	Основы обработки исключений. Перехват исключений. Класс Exception. Конфигурирован ие состояния исключения. Исключения уровня системы и приложения. Обработка многочисленных исключений. Операторы throw и finally. Исключения, связанные с	ПК-2.2.	Знать способы и механизмы разработки программного обеспечения	Устный опрос, тестиров ание

			поврежденным состоянием (Corrupted State Exceptions). Ключевые слова checked и unchecked.			
7	Интерфейсы	2	Интерфейсные ссылки. Интерфейсные свойства и индексы. Наследование интерфейсов. Явная реализация интерфейса. Структуры. Перечисления.	ПК-6	Знать понятия интерфейсной ссылки, наследования интерфейсов	Устный опрос, тестирование
8	Делегаты, события	2	Групповой вызов и адресация делегируемых методов. Ковариантность и контравариантность делегатов. Делегаты Action и Func. Анонимные методы. Лямбда-выражения. События. Аксессуары событий.	ПК-6.	Знать основные принципы использования делегатов	Устный опрос
9	Многопоточные приложения	2	Класс Thread, асинхронные делегаты	ПК-6	Знать цели использования многопоточных приложений	Устный опрос, тестирование

Лабораторные работы

№ п/п	Наименование темы	Трудоемкость	Содержание	Формируемые компетенции	Результаты освоения (знает, Умеет, Владеет)	Технологии обучения
1	Первые простые приложения в IDE Borland C++ Builder	4	Знакомство со средой разработки и базовыми компонентами			Устный опрос
2	Программирование линейных программ	4	формирование навыка работы со средой программирования IDE Borland C++ Builder			Устный опрос
3	Программирование ветвящихся структур	4	формирование навыка работы со средой программирования C++Builder при разработке приложений с применением оператора выбора, компонентов: radioButton, checkBox, groupBox.			Устный опрос
4	Операторы цикла	4	формирование навыка работы со средой программирования C++ Builder при разработке приложений с применением операторов цикла			Устный опрос
5	Простейшие	4	Разработка			Устный

	классы и объекты		простейших классов, выполнение индивидуальных заданий			опрос, тестирование
6	Разработка классов	4	Разработка простейших классов, выполнение индивидуальных заданий			
7	Классы для работы с динамическим и структурами	4	Описать класс, который используется для представления элементов динамической структуры данных.			Устный опрос
8	Наследование	4	Определить базовый класс и производные от него классы. Предусмотреть передачу аргументов конструкторам базового класса; использование виртуальных и перегруженных функций; обработку исключительных ситуаций.			Устный опрос
9	Потоки, обработка исключительных ситуаций в C++	4	Для соответствующих классов перегрузить операции вставки в поток и извлечения из			Устный опрос

			потока. При динамическом выделении памяти предусмотреть обработку исключения, возникающего при нехватке памяти.			
--	--	--	---	--	--	--

4.3.2. Содержание лабораторно-практических занятий по дисциплине.

Лабораторная работа № 1

Тема: Разработка простейших программ на языке C++ в среде Embarcadero RAD Studio.

Цель работы: Получить основные навыки при создании консольных приложений для Windows в среде Embarcadero RAD Studio .

Общие сведения о среде RAD Studio

Embarcadero RAD Studio — среда быстрой разработки кросс-платформенных приложений для Microsoft Windows фирмы Embarcadero Technologies.

Версии продукта:

- Borland Developer Studio (2002г.) — первая версия с Delphi 7 Borland Developer Studio 1.0.

- CodeGear RAD Studio (2006г.) — добавлены средства разработки БД. При использовании RAD Studio приложения могут подключаться к популярным базам данных, используя быстросействующие функции. Можно создавать многозвенные приложения с серверами Windows и поддержкой клиентов в нескольких операционных системах.

- Embarcadero RAD Studio (2008г.) – добавлены средства, позволяющие эффективно работать с данными для Windows, Mac OS X, .NET, PHP, веб-решений и мобильных устройств. Включена возможность также подключаться к широкому набору данных и служб.

Запуск Embarcadero RAD Studio XE2 для работы в C++ Builder XE

Запуск среды Embarcadero RAD Studio XE2 для разработки программы в C++ Builder XE осуществляется из главного меню системы Windows, которое вызывается по кнопке ПУСК на панели задач:

Пуск > Программы > Программирование > Embarcadero RAD Studio XE2 > C++ Builder XE

После старта программы, система Embarcadero RAD Studio XE2 предоставляет пользователю рабочий стол, на котором расположена группа из нескольких окон. Рабочий стол имеет вид, как это показано на рис. 1.

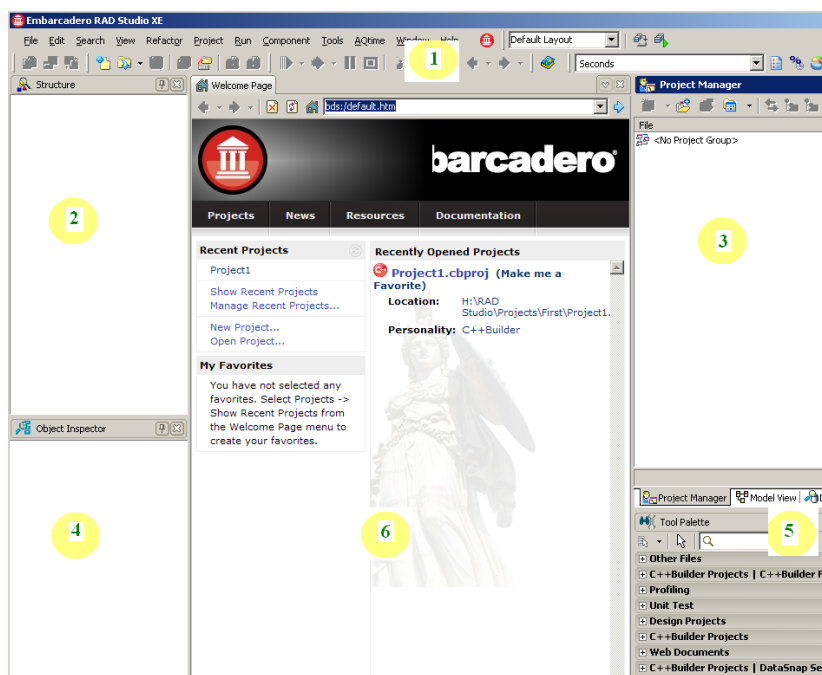


Рис.1. Рабочий стол системы Embarcadero RAD Studio XE2

Основными окнами (номера окон вписаны в кружочки желтого цвета) Embarcadero RAD Studio XE2 являются:

- окно 1 – представляет главное меню среды разработки и панель инструментов *Embarcadero RAD Studio XE2*;
- окно 2 – окно Structure, в котором изображается дерево объектов проекта;
- окно 3 – содержит 3 страницы/закладки:
 1. менеджер проектов Project Manager;
 2. Data Explore;

3. Model View;

- окно 4 – инспектор объектов Object Inspector;
- окно 5 – панель инструментов Tool Palette;
- окно 6 – окно Embarcadero (страница приглашения – Welcome Page).

Здесь можно задать команды создания нового приложения или открытия ранее созданного. А через кнопки News, Resource и Documentation попасть в Интернет за новостями, ресурсами или документацией.

После старта проекта центральное окно (окно 6) превращается в окно с закладками. И при работе с несколькими модулями (файлами и формами) программист открывает необходимые дизайнеры для каждой из форм и редакторы для каждого текстового модуля. То есть центральное окно содержит закладки для одной страницы приглашения, для одного или нескольких файлов проекта. Оно является многооконным, но переходить от окна к окну (в том числе и к странице приглашения) можно по закладкам.

Главное меню располагается, как и в любом окне Windows, под строкой заголовка окна. Назначение многих команд стандартно для приложений Windows.

Задание

Создать приложение, в котором при щелчке пользователя на кнопке появится надпись на форме. Выполните для этого последовательно следующие шаги.

1. Запустите C++Builder.
2. Поместите на пустую форму кнопку Button со страницы Standard палитры компонентов, для этого нужно выделить пиктограмму кнопки и затем щелкнуть курсором мыши в нужном месте формы. На форме появится кнопка, которой C++Builder присвоит имя по умолчанию - Button1.
3. Перенесите на форму со страницы Standard палитры компонентов метку Label. В этой метке в процессе выполнения приложения будет

появляться текст при нажатии пользователем кнопки. C++Builder присвоит метке имя Label1.

4. Разместите компоненты на форме примерно так, как показано на рисунке.

Выделите на форме компонент Button1. В Инспекторе Объектов измените ее свойство Caption, которое по умолчанию равно Button1, на «Пуск».

6. Укажите метке Label1, что надписи на ней надо делать жирным шрифтом. Для этого выделите метку, в окне Инспектора Объектов раскройте свойство Font (шрифт), затем раскройте подсвойство Style (стиль) и установите в true свойство fsBold (жирный).

7. Сотрите текст в свойстве Caption метки Label1, чтобы он не высвечивался, пока пользователь не нажмет кнопку приложения.

Теперь осталось только написать оператор, который заносил бы в свойство Caption метки Label1 нужный текст в нужный момент. Этот момент определяется щелчком пользователя на кнопке. При щелчке в кнопке генерируется событие OnClick. Следовательно, обработчик этого события и нужно написать.

8. Выделите кнопку Button1 на форме, перейдите в Инспектор Объектов, откройте в нем страницу событий (Events), найдите событие кнопки OnClick и сделайте двойной щелчок в окне справа от имени этого события. Это стандартный способ задания обработчиков любых событий. Но перейти в обработчик события OnClick можно и иначе: достаточно сделать двойной щелчок на компоненте Button1 на форме. В обоих случаях откроется окно Редактора Кода и увидите там текст:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{

}
```

Заголовок этой функции складывается из имени класса формы (TForm1), имени компонента (Button1) и имени события без префикса On (Click).

9. Напишите в обработчике оператор задания надписи метки Label1.

```
Label1->Caption = "Это мое первое приложение!";
```

Таким образом, обработчик события должен иметь вид:

```
void _fastcall TForm1: :Button1Click (TObject *Sender)
{
    Label1->Caption = "Это мое первое приложение!";
}
```

Этот оператор означает следующее. Символ "=" обозначает операцию присваивания. Запись Label1->Caption означает, что присваиваете значение свойству Caption компонента Label1.

Все указания свойств и методов производятся аналогичным образом: пишется имя компонента, затем ставятся символы операции стрелка "->": символ минус "-" и символ больше ">", записанные без пробела. После этих символов пишется имя свойства или метода. В данном случае свойству Caption присваиваете строку текста «Это мое первое приложение!».

Итак, приложение готово. Можно откомпилировать и выполнить его. Для этого выполните команду Run | Run, или нажмите соответствующую быструю кнопку, или нажмите клавишу F9. Появится окно приложения. При нажатии на кнопке «Пуск» в метке появится строка текста. Можете попробовать различные манипуляции с окном: перемещение его, изменение размеров его рамки курсором мыши, сворачивание и разворачивание.

Закройте приложение, щелкнув на кнопке в его правом верхнем углу.

5. Образовательные технологии

Лекционные занятия на курсе проводятся с использованием мультимедийного проектора и в сопровождении с презентациями в формате Power Point. Дополнительно на лекциях проводятся демонстрации работы

основных средств языков/платформ с использованием среды разработки и отладчика.

Лабораторные занятия проходят в компьютерных классах, оснащенных персональными компьютерами с установленной средой разработки Embarcadero Rad Studio 2010. Во время лабораторных занятий студенты активно взаимодействуют с преподавателем, задают вопросы по курсу и практическим заданиям, сдают практические задания. Для хранения исходного кода проектов выполняемых студентами используется раздел «Задания» в созданной команде в среде Microsoft Teams.

6. Учебно-методическое обеспечение самостоятельной работы студентов

Форма контроля и критерий оценок

В процессе обучения студентов применяются следующие формы контроля успеваемости:

- посещаемость лекций
- результат письменного теста на коллоквиуме (баллово-рейтинговая система) посещаемость лабораторных занятий (работ)
- выполнение и сдача лабораторных заданий (баллово-рейтинговая система)

Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоёмкость	Формируемые компетенции
Текущая СРС		
Подготовка к лекции, работа с учебной литературой и электронными источниками	10	ПК-2
Подготовка к практическим, лабораторным занятиям	10	ПК-6, ПК-2
подготовка к контрольным работам, тестированию	10	ПК-6, ПК-2

выполнение домашних заданий в виде решения отдельных задач, расчетно - компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин	10	ПК-6, ПК-2
самостоятельное изучение разделов дисциплины	10	ПК-6, ПК-2
подготовка к экзамену	10	
Творческая проблемно-ориентированная самостоятельная работа		
поиск, изучение и презентация информации по заданной проблеме	10	ПК-2
анализ научных публикаций по заданной теме	10	ПК-2
Подготовка письменных работ (рефератов)	10	ПК-6, ПК-2
исследовательская работа, участие в конференциях, семинарах, олимпиадах	10	ПК-6, ПК-2
Разработка глоссария по разделам дисциплины	8	ПК-6, ПК-2
Итого СРС	108	

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.

Код компетенции из	Код и наименование	Планируемые результаты обучения	Процедура освоения
--------------------	--------------------	---------------------------------	--------------------

ФГОС ВО	индикатора достижения компетенций		
ПК-2 Способность разрабатывать и адаптировать прикладное программное обеспечение	ПК-2.1. Демонстрирует знание основных принципов и технологий промышленной разработки программного обеспечения ПК-2.2. Демонстрирует умения разрабатывать программное обеспечение с применением инструментов автоматизации сборки, интеграции, тестирования	<i>Знает:</i> принципы разработки программного обеспечения, концепции и понятия объектно- ориентированного подхода к программированию, механизмы его реализации в языке программирования <i>Умеет:</i> 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. <i>Владеет:</i> навыками анализа поставленных задач, проектирования и разработки приложений, приемами разработки программных комплексов для решения прикладных задач, методами использования современных технологий	Устный опрос, тестирование

		программирования, тестирования и документирования программных комплексов	
ПК-6 Способность программировать приложения и создавать программные прототипы решения прикладных задач.	ПК-6.1. Демонстрирует знание принципов и шаблонов проектирования программного обеспечения, баз данных, программных интерфейсов, основ моделирования предметной области ПК-6.2. Демонстрирует умение выполнять проектирование компонентов программного обеспечения по заданным требованиям в рамках определенной предметной области	<i>Знает:</i> проблемы и процессы анализа предметной области программных решений, современные подходы анализа предметной области программных решений <i>Умеет:</i> разрабатывать программные приложения для предметной области <i>Владеет:</i> практическими навыками использования языков программирования для создания программных прототипов решения прикладных задач	Устный опрос, тестирование

7.2. Типовые контрольные задания

Примерный перечень тестов к промежуточному контролю по всему изучаемому курсу:

1. Прародителями всех языков ООП является
 1. Симула
 2. C#
 3. Delphi

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

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

7. 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
8. 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
9. Класс может содержать члены-свойства _____ типа.
1. базового
 2. производного
 3. пользовательского
 4. произвольного
10. _____ - это специальная компонентная функция, которую можно явно включать в определение класса для инициализации объектов.
11. 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.
12. Если язык обеспечивает возможность создания пользовательских типов данных, то говорят, что язык называется:
1. наследуемым;
 2. инкапсулируемым;
 3. перегруженным;
 4. расширяемым.

13. Операция, выполняющая заданные действия над пользовательским типом данных, называется:

1. полиморфической;
2. инкапсулированной;
3. классифицированной;
4. перегруженной.

14. Все члены класса - данные и методы - являются по умолчанию

1. открытыми
2. закрытыми
3. защищёнными
4. абстрактными
5. виртуальными

15. 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

16. 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

17. Состояние объекта определяется:

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

18. 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
19. 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
20. Конструктор вызывается автоматически в момент _____ объекта.
21. Для инициализации переменных-членов класса применяется
1. деструктор
 2. параметр
 3. конструктор
 4. оператор присваивания
 5. начало объявления класса
22. _____ - другое имя уже существующего объекта.
1. идентификатор;
 2. ключевое слово;
 3. массив;
 4. указатель;
 5. ссылка.
23. 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. Все свойства и указатели на методы класса

24. 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;

25. Атрибуты класса могут быть

1. только целыми числами
2. любыми встроенными типами
3. любого определенного в программе типа

26. 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

Примерный перечень тем рефератов

1. История развития языков объектно-ориентированного программирования.
2. Виртуальная машина Java. Версии Java машины и их эволюция.
3. Сравнение языков ООП: Java, C#, Object Pascal.
4. Первые языки объектно-ориентированного программирования.
5. Особенности объектно-ориентированных языков.
6. Методология структурного программирования
7. Методология модульного программирования
8. Методологии объектно-ориентированного и компонентного программирования
9. Особенности алгоритма в структурном программировании.
Нисходящее и восходящее программирование

Примерный перечень вопросов к экзамену

1. Дайте характеристику парадигме ООП и специфике интерфейса ОО-программ.

2. Дайте понятие класса в ООП. Опишите отношение "объект - класс".
3. Основные принципы объектного подхода. Абстрагирование.
4. Основные принципы объектного подхода. Инкапсуляция.
5. Основные принципы объектного подхода. Модульность.
6. Основные принципы объектного подхода. Иерархия.
7. Основные принципы объектного подхода. Типизация.
8. Объект с точки зрения ООП. Состояние. Поведение.
9. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов.
10. Объект с точки зрения ООП. Взаимоотношения между объектами.
11. Классы. Природа классов. Мета модель. Инстанцирование.
12. Модель памяти и структура программы. Классы памяти. Ссылки.
13. Средства абстракции C++. Структура класса. Статические члены и их инициализация
14. Средства инкапсуляции C++. Инкапсуляция и наследование. Друзья.
15. Модульность, раздельная компиляция, пространства имен, using директива.
16. Представление иерархических отношений. Наследование.
17. Представление иерархических отношений. Агрегация. Зависимость по времени жизни.
18. Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм.
19. C++: средства реализации состояния объектов; реализация поведения.
20. Перегрузка операторов.
21. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании.
22. Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов.
23. Исключения в C++. Обработка исключений. Умные указатели.

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

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 50% и промежуточного контроля - 50%.

Текущий контроль по дисциплине включает:

- посещение занятий – 5 балла,
- участие на практических занятиях - 20 баллов,
- выполнение лабораторных заданий – 60 баллов,
- выполнение домашних (аудиторных) контрольных работ – 15 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 30 баллов,
- письменная контрольная работа - 30 баллов,
- тестирование - 40 баллов

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

а) основная литература:

1. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс] / Б. Мейер. - Электрон. текстовые данные. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 285 с. - 2227-8397. - Режим доступа: <http://www.iprbookshop.ru/39552.html>

2. Николаев Е.И. Объектно-ориентированное программирование [Электронный ресурс]: учебное пособие / Е.И. Николаев. - Электрон. текстовые данные. - Ставрополь: Северо-Кавказский федеральный университет, 2015. - 225 с. - 2227-8397. - Режим доступа: <http://www.iprbookshop.ru/62967.html>

3.Павловская Т.А. С/С++. Программирование на языке высокого уровня: для магистров и бакалавров / Павловская, Татьяна Александровна. - СПб. [и др.] : Питер, 2012. - 460 с. - (Учебник для вузов). - Допущено МО РФ. - ISBN 978-5- 94723-568-5 : 357-00.

б) дополнительная литература:

1. Лисицин Д.В. Объектно-ориентированное программирование [Электронный ресурс]: конспект лекций / Д.В. Лисицин. - Электрон. текстовые данные. - Новосибирск: Новосибирский государственный технический университет, 2010. - 88 с. - 978-5-7782-1454-5. - Режим доступа: <http://www.iprbookshop.ru/44970.html>

2.Сорокин А.А. Объектно-ориентированное программирование [Электронный ресурс]: учебное пособие. Курс лекций / А.А. Сорокин. - Электрон. текстовые данные. - Ставрополь: Северо-Кавказский федеральный университет, 2014. - 174 с. - 2227-8397. - Режим доступа: <http://www.iprbookshop.ru/63110.html>

3. Иванова Г.С. Объектно-ориентированное программирование: Учеб. для вузов / Иванова, Галина Сергеевна ; Т.Н.Ничушкина, Е.К.Пугачев; Под ред. Г.С.Ивановой. - М.: Изд-во МГТУ им. Н.Э.Баумана, 2001. - 317 с. - (Информатика в техническом университете). - ISBN 5-7038-1525-8: 0-0

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

1. eLIBRARY.Ru [Электронный ресурс]: электронная библиотека / Науч. электр. б-ка.- МОСКВА.1999. – Режим доступа: <http://elibrary.ru> (дата обращения 10.03.2020). – Яз. рус., англ.

2. Электронный каталог НБ ДГУ [Электронный ресурс]: база данных содержит сведения о всех видах лит., поступающих в фонд НБ ДГУ / Дагестанский гос.унив. – Махачкала. – 2010. – Режим доступа: <http://elib.dgu.ru>. свободный (дата обращения 10.09.2020)

3. Национальный Открытый Университет «ИНТУИТ» [Электронный ресурс]: www.intuit.ru (дата обращения 10.09.2020)

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

В процессе преподавания дисциплины предполагается использование современных технологий визуализации учебной информации (создание и демонстрация презентаций), использование ресурсов электронной информационно-образовательной среды университета, в том числе учебного курса «Объектно-ориентированное программирование», размещенного на платформе Microsoft Teams ДГУ <https://teams.microsoft.com/> (автор-разработчик Магомедова С.Р.).

При проведении занятий по данной дисциплине используется программное обеспечение - Embarcadero Rad Studio 2010.

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

Лекции и практические занятия по данной дисциплине проводятся в мультимедийном лекционном зале, где установлен проектор и экран. Практические занятия проводятся в двух компьютерных классах где установлены по 15 компьютеров, все они подключены локальной сети университета т.е. имеют доступ к локальным ресурсам ДГУ и глобальной сети Интернет. На компьютерах установлена операционная система Microsoft Windows 7, пакет прикладных программ Microsoft Office.