

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

09.03.02 Информационные системы и технологии

Профиль программы: общий

Уровень высшего образования: бакалавриат

Форма обучения очная, заочная

Статус дисциплины:

входит в часть ОПОП, формируемую участниками образовательных
отношений

Махачкала, 2021

Рабочая программа дисциплины «Объектно-ориентированное программирование» составлена в 2021 году в соответствии с требованиями ФГОС ВО - бакалавриат по направлению подготовки 09.03.02 от 19.09.2017 № 926.

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

Рабочая программа дисциплины одобрена:
на заседании кафедры ИСиТП от «29» 2021 г., протокол №11
Зав.кафедрой Исмиханов З.Н.
(подпись)

на заседании Методической комиссии факультета ИСиТП
от «29» июня 2021 г., протокол №11
Председатель Бакмаев А.Ш.
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением «9» 09 2021 г. _____

(подпись)
Начальник УМУ Гасангаджиева А.З.

Аннотация рабочей программы дисциплины

Дисциплина «Объектно-ориентированное программирование» входит в часть, формируемую участниками образовательных отношений ОПОП 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 объектно-ориентированных 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: общепрофессиональных ОПК-6, ПК-4.

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

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

Объем дисциплины 4 зачетных единиц, в том числе в 144 академических часах по видам учебных занятий.

Объем дисциплины в очной форме

Семестр	Учебные занятия							Форма промежуточной аттестации
	в том числе							
	Всего	Контактная работа обучающихся с преподавателем					СРС, в том числе экзамен	
		Всего	Из них					
			Лекции	Лабораторные занятия	Практические занятия			
4	144	54	36	18			90	зачет

Объем дисциплины в заочной форме

Семестр	Учебные занятия							Форма промежуточной аттестации
	в том числе							
	Всего	Контактная работа обучающихся с преподавателем					СРС, в том числе экзамен	
		Всего						
Лекции			Лабораторные занятия	Практические занятия	Консультация			
4	144	12	4	8			132	зачет

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 «Основы объектно-ориентированного программирования» входит в часть, формируемую участниками образовательных отношений ОПОП бакалавриата по направлению подготовки 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.

Для успешного освоения данного курса студент должен иметь знания по школьному по дисциплине «Информатика и программирование», «Программирование на языке высокого уровня». Список дисциплин, для изучения которых необходимы знания данного курса: «Базы данных», «Разработка и сопровождение программных приложений», «Интернет программирование».

3. Компетенции обучающего, формируемые в результате освоения дисциплины

Код компетенции из ОПОП	Код и наименование индикатора достижения компетенций (в соответствии с ОПОП)	Планируемые результаты обучения	Процедура освоения
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ИД1.ОПК-6.1.Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий ИД2.ОПК-6.2.Умеет применять языки программирования и работы с базами данных, современные	Знает язык объектно-ориентированный язык программирования С++, принципы объектно-ориентированного программирования. Умеет применять методы и технологии языка программирования С++ для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз	Устный опрос

	программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ИДЗ.ОПК-6.3.Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач.	данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	
ПК-4. Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных	ПК-4.1. Знает методы формальных спецификаций и системы управления базами данных ПК-4.2. Умеет применять современные средства и языки программирования ПК-4.3. Имеет навыки использования операционных систем	Знает системы управления базами данных для решения задач объектно-ориентированного программирования. Умеет применять навыки работы с системой управления базами данных для решения задач объектно-ориентированного программирования. Владеет навыками применения средств объектно-ориентированного программирования для создания windows приложений	Устный опрос

4. Объем, структура и содержание дисциплины

4.1. Объем дисциплины составляет 4 зачетных единиц, 144 академических часа.

4.2. Структура дисциплины.

4.2.1. Структура дисциплины в очной форме

№ п/ п	Названия разделов	Семестр	Неделя	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Самостоятельная работа	Формы текущего контроля успеваемости (по неделям семестра)
				Лекции	Практические занятия	Лабораторные занятия	Контроль		
Модуль 1.									
1	Методология объектно-ориентированного программирования	4		2		2		8	Устный опрос
2	Классы и объекты. Понятие. Средства описания данных.	4		4		2		8	
3	Конструкторы и инициализация объектов	4		2				8	
	Итого за модуль:			8		4		24	
Модуль 2.									
1	Управление доступом. Инкапсуляция	4		4		2		6	Устный опрос
2	Наследование и полиморфизм	4		4		2		6	
3	Множественной наследование	4		2		2		8	
	Итого за модуль:			10		6		20	
Модуль 3.									
1	Виртуальные функции. Дружественные функции. Дружественные классы.	4		4		2		12	Устный опрос
2	Интерфейсы	4		4		2		12	
	Итого за модуль:			8		4		24	
Модуль 4.									
1	Шаблоны	4		6		2		8	
2	Перегрузка операторов			4		2		14	
	Итого за модуль:			10		4		22	
	Всего часов	144		36		18		90	

4.2.2. Структура дисциплины в заочной форме

№ п/ п	Названия разделов	Семестр	Неделя	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Самостоятельная работа	Формы текущего контроля успеваемости (по неделям семестра)
				Лекции	Практические занятия	Лабораторные занятия	Контроль		
Модуль 1.									
1	Методология объектно-ориентированного программирования	2		2		2		10	Устный опрос
2	Классы и объекты. Понятие. Средства описания данных.	2						10	
3	Конструкторы и инициализация объектов	2						12	
	Итого за модуль:			2		2		32	
Модуль 2.									
1	Управление доступом. Инкапсуляция	2		2		2		8	Устный опрос
2	Наследование и полиморфизм	2						14	
3	Множественной наследование	2						12	
	Итого за модуль:			2		2		32	
Модуль 3.									
1	Виртуальные функции. Дружественные функции. Дружественные классы.	2				2		16	Устный опрос
2	Интерфейсы	2						18	
	Итого за модуль:					2		34	
Модуль 4.									
1	Шаблоны	2						18	
2	Перегрузка операторов					2		16	
	Итого за модуль:					2		34	
	Всего часов	144		4		8		132	

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.	ОПК-6	Знает язык объектно-ориентированный язык программирования C++, принципы объектно-ориентированного программирования. Умеет применять методы и технологии языка программирования C++ для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	Письменный опрос
2	Классы и объекты. Понятие. Средства описания данных.	4	Понятие класса. Создание объектов. Методы. Конструкторы. Доступ к членам класса. Модификаторы параметров. Необязательные и именованные аргументы. Рекурсия. Ключевое слово static. Модификаторы	ОПК-6	Знает язык объектно-ориентированный язык программирования C++, принципы объектно-ориентированного программирования. Умеет применять методы и технологии языка программирования C++ для автоматизации бизнес-процессов,	Устный опрос

			доступа в аксессуарах. Иерархии классов.		решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	
3	Конструкторы и инициализация объектов	2	Понятие конструктора. Конструктор по умолчанию. Конструктор копии Инициализация констант и ссылок	ПК-4	Знает системы управления базами данных для решения задач объектно-ориентированного программирования Умеет применять навыки работы с системой управления базами данных для решения задач объектно-ориентированного программирования Владеет навыками применения средств объектно-ориентированного программирования для создания windows приложений	
4	Управление доступом. Инкапсуляция	4	спецификаторы доступа. Инкапсуляция	ПК-4	Знает системы управления базами данных для решения задач объектно-ориентированного программирования Умеет применять навыки работы с системой управления базами данных для решения задач объектно-	

					ориентированного программирования Владеет навыками применения средств объектно-ориентированного программирования для создания windows приложений	
5	Наследование и полиморфизм	4	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.	ОПК-6	Знает язык объектно-ориентированный язык программирования C++, принципы объектно-ориентированного программирования. Умеет применять методы и технологии языка программирования C++ для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	Устный опрос
6	Множественное наследование	2	Перекрытие имен функций. Перекрытие виртуальных функций. Представление объекта в памяти.	ОПК-6	Знает язык объектно-ориентированный язык программирования C++, принципы объектно-ориентированного программирования. Умеет применять методы и технологии языка программирования C++ для	

					автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	
7	Виртуальные функции. Дружественные функции. Дружественные классы.	4	Виртуальные функции. Определение дружественных функций в классе. Дружественные классы. Дружественные методы	ОПК-6	Знает язык объектно-ориентированный язык программирования C++, принципы объектно-ориентированного программирования. Умеет применять методы и технологии языка программирования C++ для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	
8	Интерфейсы	4	Интерфейсные ссылки. Интерфейсные свойства и индексы. Наследование интерфейсов. Явная реализация интерфейса.	ОПК-6	Знает язык объектно-ориентированный язык программирования C++, принципы объектно-ориентированного программирования.	Устный опрос

			Структуры. Перечисления.		Умеет применять методы и технологии языка программирования C++ для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	
9	Шаблоны	4	Шаблоны классов. Исключения. Стандартная библиотека шаблонов.	ОПК-6	Знает язык объектно-ориентированный язык программирования C++, принципы объектно-ориентированного программирования. Умеет применять методы и технологии языка программирования C++ для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	

10	Перегрузка операторов		Способы и правила перегрузки. Перегрузка операторов через методы классов	ПК-4	Знает системы управления базами данных для решения задач объектно-ориентированного программирования Умеет применять навыки работы с системой управления базами данных для решения задач объектно-ориентированного программирования Владет навыками применения средств объектно-ориентированного программирования для создания windows приложений.	Устный опрос
----	-----------------------	--	--	------	--	--------------

Лабораторные работы

№ п/п	Наименование темы	Трудоемкость	Содержание	Формируемые компетенции	Результаты освоения (знает, Умеет, Владеет)	Технологии обучения
1	Простейшие классы и объекты	2	Разработка простейших классов, выполнение индивидуальных заданий	ОПК-6	Знает язык объектно-ориентированный язык программирования C++, принципы объектно-ориентированного программирования. Умеет применять методы и технологии языка программирования C++ для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	Устный опрос
2	Разработка классов	2	Разработка простейших классов, выполнение индивидуальных заданий	ПК-4	Знает системы управления базами данных для решения задач объектно-ориентированного	Устный опрос

					программирования Умеет применять навыки работы с системой управления базами данных для решения задач объектно-ориентированного программирования Владеет навыками применения средств объектно-ориентированного программирования для создания windows приложений	
3	Классы для работы с динамическими структурами	4	Описать класс, который используется для представления элементов динамической структуры данных.	ОПК-6	Знает язык объектно-ориентированный язык программирования C++, принципы объектно-ориентированного программирования. Умеет применять методы и технологии языка программирования C++ для автоматизации бизнес-процессов, решения прикладных задач	Устный опрос

					различных классов, ведения баз данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	
4	Наследование	4	Определить базовый класс и производные от него классы. Предусмотреть передачу аргументов конструкторам базового класса; использование виртуальных и перегруженных функций; обработку исключительных ситуаций.	ОПК-6	Знает язык объектно-ориентированный язык программирования C++, принципы объектно-ориентированного программирования. Умеет применять методы и технологии языка программирования C++ для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. Владеет навыками применения современных технологий и программирования.	Устный опрос

5	Потоки, обработка исключительных ситуаций в C++	4	Для соответствующих классов перегрузить операции вставки в поток и извлечения из потока. При динамическом выделении памяти предусмотреть обработку исключения, возникающего при нехватке памяти.	ПК-4	Знает системы управления базами данных для решения задач объектно-ориентированного программирования Умеет применять навыки работы с системой управления базами данных для решения задач объектно-ориентированного программирования Владеет навыками применения средств объектно-ориентированного программирования для создания windows приложений	Устный опрос
---	---	---	--	------	--	--------------

4.3.2. Содержание лабораторных занятий по дисциплине.

Лабораторная работа № 1 Простейшие классы и объекты

Теоретическое введение. Классы представляют абстрактные типы данных с открытым интерфейсом и скрытой внутренней реализацией. В классах реализованы базовые принципы *объектно-ориентированного программирования (ООП)*:

- 1) абстракция данных;
- 2) инкапсуляция – в классах объединяются данные и методы (функции) для работы с ними, так как лишь через методы возможен доступ к сокрытым данным класса;
- 3) наследование – в производных классах наследуются члены базового класса;
- 4) полиморфизм – возможность использования одних и тех же методов для работы с различными объектами базового и порожденных им классов.

Определение простейшего класса без наследования имеет вид: **class имя_класса {**

//по умолчанию раздел **private** – частные члены класса **public**: // открытые функции и переменные класса
};

Имя класса является новым типом данных. Понятию переменной данного типа соответствует понятие объекта класса. Список членов класса включает описание данных и функций. Функции, описания которых находятся в определении класса, называются функциями-членами класса.

Переменные и функции, объявленные в разделе класса по умолчанию или явно как **private**, имеют область видимости в пределах класса. Их можно сделать видимыми вне класса, если объявить в разделе **public**. Обычно переменные объявляются в разделе **private**, а функции в разделе **public**.

Классами в C++ являются также структуры (**struct**) и объединения (**union**). В отличие от класса члены структуры по умолчанию являются открытыми, а не закрытыми. Кроме того, объединения не могут наследоваться и наследовать.

При реализации функциональной части класса могут быть использованы функции-члены класса, конструкторы, деструкторы, функции-операторы. Функции класса всегда объявляются внутри класса. Определение функции может находиться и внутри класса. Такие функции называются **inline**-функциями. Обычно определения функций-членов класса помещаются вне класса. При этом перед именем функции указывается **имя_класса::**.

тип имя_класса:: имя_функции (описание аргументов) { /*тело функции*/ }

Вызов функций осуществляется одним из двух способов:

имя_объекта.имя_функции(аргументы); указатель_на_объект ->

имя_функции(аргументы);

Обращение к данным объекта класса осуществляется с помощью функций, вызываемых из объектов. При этом функции-члену класса передается скрытый указатель **this** на объект, вызывающий функцию.

Функции-«друзья» класса, объявляемые в классе со спецификатором **friend**, указатель **this** не содержат. Объекты, с которыми работают такие функции, должны передаваться в качестве их аргументов. Это обычные функции языка C++, которым разрешен доступ к закрытым членам класса.

Пример.

/* Создается класс Student. Формируется динамический массив объектов. При тестировании выводится: сформированный список студентов, список студентов заданного факультета, список студентов для заданных факультета и курса*/

```

#include <conio.h>
#include <string.h>
#include <iostream.h> struct
date // дата рождения {char
daymon[6];
int year; };
//===== class Student =====
class Student{
char name[30]; //private
date t;
char adr[30],
fac[20]; int kurs;
public:
Student();
char *getfac();
int getkurs();
void show();
};

Student::Student()
{ cout<<"Input name:"; cin>>name;
cout<<"Input date of born\n";
cout<<"Day.mon:"; cin>>t.daymon;
cout<<"Year:"; cin>>t.year;
cout<<"Input adr:"; cin>>adr;
cout<<"Input fac:"; cin>>fac;

cout<<"Input kurs:"; cin>>kurs;
}

void Student::show()
{
cout<<"Name : "<<name<<endl;
cout<<"Was born : "<<t.daymon<<','<<t.year<<endl;
cout<<"Address : "<<adr<<endl;
cout<<"Fac : "<<fac<<endl;
cout<<"Kurs : "<<kurs<<endl;
}

char *Student::getfac() { return fac; }
int Student::getkurs() { return kurs; }
void spisfac(Student spis[],int n)//список студентов заданного факультета
{char fac[20];
cout<<"Input faculty:"; cin>>fac;
for(int i=0;i<n;i++)
if(strcmp(spis[i].getfac(),fac)==0)spis[i].show();
}

void spisfackurs(Student spis[],int n)
//список студентов заданных факультета и курса

```

```

    {int i,k; char
    fac[20];
    cout<<"Input faculty:"; cin>>fac;
    cout<<"Input the course:"; cin>>k;
    for(i=0;i<n;i++)
    if ((strcmp(spis[i].getfac(),fac)==0)&&
        (spis[i].getkurs()==k)) spis[i].show();
    }
//===== main =====
void main()
{ Student *spis;
  int n;
  cout<<"Input a number of students: "; cin>>n;
  spis=new Student [n];
  for(int i=0;i<n;i++) {
    cout<<"=====
    <<endl; spis[i].show();
  }
  spisfac(spis,n);
  spisfackurs(spis,n);
  delete [] spis;
  cout<<"press any
  key!" while(!kbhit());
}

```

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 Visual Studio. Во время лабораторных занятий студенты активно взаимодействуют с преподавателем, задают вопросы по курсу и практическим заданиям, сдают практические задания.

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
	очная	заочная	
Текущая СРС			
Подготовка к лекции, работа с учебной литературой и электронными источниками	10	20	ОПК-6
Подготовка к практическим, лабораторным занятиям	10	20	ПК-4, ОПК-6
подготовка к контрольным работам	10	20	ПК-4, ОПК-6
выполнение домашних заданий в виде решения отдельных задач, расчетно - компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин	10	20	ПК-4, ОПК-6
самостоятельное изучение разделов дисциплины	10	20	ОПК-6
Творческая проблемно-ориентированная самостоятельная работа	10		ПК-4
поиск, изучение и презентация информации по заданной проблеме	10	10	ПК-4
Подготовка письменных работ (рефератов)	10	10	ПК-4, ОПК-6
исследовательская работа, участие в конференциях, семинарах, олимпиадах	10	12	ПК-4, ОПК-6
Итого СРС	90	132	

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

7.1. Типовые контрольные задания

Примерный перечень вопросов 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. Всех свойства и указателей 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 вопросов к зачету

1. В чем различие между видами и методами (способами) абстракции?
2. Дайте характеристику парадигме ООП и специфике интерфейса ОО-программ.
3. Дайте понятие класса в ООП. Опишите отношение "объект - класс".
4. Основные принципы объектного подхода. Абстрагирование.
5. Основные принципы объектного подхода. Инкапсуляция.
6. Основные принципы объектного подхода. Модульность.
7. Основные принципы объектного подхода. Иерархия.
8. Основные принципы объектного подхода. Типизация.
9. Объект с точки зрения ООП. Состояние. Поведение.
10. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов.
11. Объект с точки зрения ООП. Взаимоотношения между объектами.
12. Классы. Природа классов. Мета модель. Инстанцирование.
13. Модель памяти и структура программы. Классы памяти. Ссылки.
14. Средства абстракции C++. Структура класса. Статические члены и их инициализация
15. Средства инкапсуляции C++. Инкапсуляция и наследование. Друзья.
16. Модульность, раздельная компиляция, пространства имен, using директива.
17. Представление иерархических отношений. Наследование.
18. Представление иерархических отношений. Агрегация. Зависимость по времени жизни.
19. Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм.
20. C++: средства реализации состояния объектов; реализация поведения.
21. Перегрузка операторов.
22. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы.
Порядок вызова конструкторов и деструкторов при наследовании.

23. Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов.

24. Исключения в C++. Обработка исключений. Умные указатели.

25. Шаблоны классов и шаблоны функций. Специализация.

7.2. 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 баллов,
- 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

Итоговой формой контроля знаний, умений и навыков по дисциплине является зачет.

Зачет проводится в форме устного опроса.

Критерии оценки зачета по 100-балльной системе:

- 100 баллов - дан полный, развёрнутый ответ на поставленный вопрос, показана совокупность осознанных знаний об объекте, проявляющаяся в свободном ориентировании понятиями, умении выделять существенные и несущественные его признаки, причинно-следственные связи. Ответ формулируется в терминах науки, логичен, доказателен, демонстрирует авторскую позицию студента.

- 90 баллов - дан полный, развёрнутый ответ на поставленный вопрос, показана совокупность осознанных знаний об объекте, доказательно раскрыты основные положения темы; в ответе прослеживается чёткая структура, логическая последовательность, отражающая сущность раскрываемых понятий, теорий, явлений. Ответ изложен литературным языком в терминах науки. Могут быть допущены недочёты в определении понятий, исправленные студентом самостоятельно в процессе ответа.

- 80 баллов - дан полный, развёрнутый ответ на поставленный вопрос, доказательно раскрыты основные положения темы; в ответе прослеживается чёткая структура, логическая последовательность, отражающая сущность раскрываемых понятий, теорий, явлений. Ответ изложен литературным языком в терминах науки. Могут быть допущены недочёты, исправленные студентом с помощью преподавателя.

- 70 баллов - дан полный, но недостаточно последовательный ответ на поставленный вопрос, но при этом показано умение выделить существенные и несущественные признаки и причинно-следственные связи. Ответ логичен и изложен в терминах науки. Могут быть допущены 1-2 ошибки в определении основных понятий, которые студент затрудняется исправить самостоятельно.

- 60 баллов - дан неполный ответ, логика и последовательность изложения имеют существенные нарушения. Допущены грубые ошибки при определении сущности раскрываемых понятий, теорий, явлений, вследствие непонимания студентом их существенных и несущественных признаков и связей. В ответе отсутствуют выводы. Умение раскрыть конкретные проявления обобщённых знаний не показано. Речевое оформление требует поправок, коррекции.

- 50 баллов - дан неполный ответ, представляющий собой разрозненные знания по теме вопроса с существенными ошибками в определениях. Присутствует фрагментарность, нелогичность изложения. Не понимает связь данного понятия, теории, явления с другими объектами дисциплины. Отсутствуют выводы. Конкретизация и доказательность изложения. Речь неграмотная. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента не только на поставленный вопрос, но и на другие вопросы дисциплины.

- 40 баллов - ответ студента правилен лишь частично, при разъяснении материала допускаются серьезные ошибки.

- 20-30 баллов - студент имеет общее представление о теме, но не умеет логически обосновать свои мысли.

- 10 баллов - студент имеет лишь частичное представление о теме.

- 0 баллов – нет ответа.

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 10.03.2021). – 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 10.09.2021)
3. Национальный Открытый Университет «ИНТУИТ» [Электронный ресурс]: www.intuit.ru (дата обращения 10.09.2021)

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 электронного учебного курса «Объектно-ориентированное программирование», размещенного на платформе edu.dg.u.ru - <http://edu.dgu.ru/course/view.php?id=3285> (автор-разработчик Магомедова С.Р.).

При проведении занятий по данной дисциплине используется программное обеспечение - Visual Studio.

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.