



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДФУ)

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

«СОГЛАСОВАНО»

Руководитель ОП

Добржинский Ю.В.

«01» сентября 2017 г.



«УТВЕРЖДАЮ»

Заведующий кафедрой

«Информационные системы управления»

А.И. Сухомлинов

«01» сентября 2017 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Объектно-ориентированное программирование

**Направление подготовки 09.03.01 Информатика и вычислительная техника**  
профиль «Автоматизированные системы обработки информации и управления»

**Форма подготовки очная**

курс 2 семестр 3

лекции 36 час.

практические занятия 18 час.

лабораторные работы 36 час.

в том числе с использованием МАО лек. \_\_\_/пр. 18 /лаб. \_\_\_ час.

в том числе в электронной форме лек. \_\_\_/пр. \_\_\_/лаб. \_\_\_ час.

всего часов аудиторной нагрузки 90 час.

в том числе с использованием МАО 18 час.

в том числе в электронной форме \_\_\_ час.

самостоятельная работа 54 час.

в том числе на подготовку к экзамену 27 час.

курсовая работа / курсовой проект \_\_\_ семестр

зачет \_\_\_ семестр

экзамен 3 семестр

Рабочая программа составлена в соответствии с образовательным стандартом, самостоятельно установленный ДФУ по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденный приказом ректора ДФУ от 04.04.2016 № 12-13-593  
Рабочая программа обсуждена на заседании кафедры «Информационные системы управления», протокол № 1 от «1» сентября 2017 г.

Заведующий кафедрой ИСУ к.т.н., доцент А.И. Сухомлинов

Составитель: ст. преподаватель Березкина Г.Л.

**I. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**II. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

## **ABSTRACT**

**Bachelor's degree in** 09.03.01 Informatics and Computer Technology

**Study profile:** «Computer Aided Systems of Information Processing and Management»

**Course title:** Object-Oriented Programming

**Basic part of Block, 4 credits**

**Instructor:** Galina Berezkina

**At the beginning of the course a student should be able to:**

computer science; information technology, modeling of economic and technical systems.

**Learning outcomes:**

Contents discipline covers the following issues since using programming language constructs simple and ending with object-oriented elements. During the course of study is considered tools for developing applications and various programming environments.

**Course description:**

The study area of the course includes basic concept, object-oriented programming, programming languages, objects, classes, polymorphism, inheritance, abstraction, concurrency, data retention, limiting access, visibility, storage class, class methods.

**Main course literature:**

1. [http://e.lanbook.com/books/element.php?pl1\\_cid=25&pl1\\_id=5115](http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=5115) Asarina I. V. Object-oriented programming in C++: lectures and exercises. - "Hot line - Telecom" Publisher: 978-5-9912-7001-4 ISBN: 2012: 2nd ed. stereotype. Edition: 320 pages.

2. [http://e.lanbook.com/books/element.php?pl1\\_cid=25&pl1\\_id=8781](http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=8781) Babushkin, I. A., Okulov S. M. Workshop on object - oriented programming.- "Binom. Laboratory of knowledge" Publishing house: 978-5-9963-0954-2 ISBN: 2012:3rd ed. (e)Edition:366 p.

3. Grady Booch, Robert A Maksimchuk, Object-oriented analysis and design with examples of applications (3rd edition) Moscow, Publishing house: Williams, 2008 – p. 721

4. Stroustrup B. The C++ programming Language. 3rd edition of St-Petersburg, Publishing house BinomNevsky Dialect, 2008(3rd) - , 1054стр.

**Form of final knowledge control:** exam

## **Аннотация к рабочей программе дисциплины «Объектно-ориентированное программирование»**

Дисциплина «Объектно-ориентированное программирование» предназначена для обучения бакалаврами направления 09.03.01 – «Информатика и вычислительная техника» и входит в состав блока учебных дисциплин выбора по профилю подготовки «Автоматизированные системы обработки информации и управления». Дисциплина изучается в третьем семестре 2 курса бакалавриата. Трудоемкость дисциплины составляет 4 зачетных единиц/ 144 часа. Из них 36 часов составляют лекции, 36 часов лабораторные работы, 18 часов практических занятий и 54 часов самостоятельной работы, включая подготовку к экзамену 27 часов. Изучению данной дисциплины предшествует освоение предмета «Введение в программирование».

**Цель** дисциплины - обучение бакалавров теоретическим основам и практическим навыкам разработки и реализации программного обеспечения на основе использования объектно-ориентированного подхода.

**Задачи** дисциплины:

- освоение студентами фундаментальных теоретических положений из области объектно-ориентированного программирования;
- формирование у студентов интегрированного восприятия существующих подходов программирования;
- овладение обучаемыми технологиями и коммерчески доступными инструментальными средствами объектно-ориентированного программирования;
- приобретение студентами умений составления, документирования, тестирования, отладки, верификации и валидации разрабатываемых программных компонент.

Дисциплина изучается в третьем семестре 2 курса бакалавриата.

Для успешного изучения дисциплины «Объектно-ориентированное программирование» у обучающихся должны быть предварительно сформированы следующие компетенции:

- ОПК-1 способностью инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем,
- ОПК-5 способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.

В результате изучения данной дисциплины у студентов формируются следующие компетенции:

В результате изучения данной дисциплины у студентов формируются следующие компетенции:

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК-2 - способностью осваивать методики использования программных средств для решения практических задач	Знает	принципы, методы формализации, алгоритмизации и реализации программного обеспечения с помощью языков программирования;
	Умеет	проводить анализ существующих средств разработки программного обеспечения систем, их выбор, внедрение и применение для решения поставленных задач;
	Владеет	методами и средствами реализации программного обеспечения;
ОПК-4 - способностью участвовать в настройке и наладке программно-аппаратных комплексов	Знает	современные технологии разработки программных комплексов с использованием CASE-средств
	Умеет	использовать на практике CASE-средства
	Владеет	инструментальными средами для различных языков программирования
ПК-3 способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования	Знает	Принципы системного подхода, и связанные с ним технологии программирования
	Умеет	Основополагающие теоретические положения, определяющие процесс разработки программного обеспечения информационных систем;
	Владеет	Различными методологиями и технологиями разработки программного обеспечения информационных систем

Изучение дисциплины включает в себя освоение теоретического материала на лекциях и выполнение лабораторных работ.

Для формирования вышеуказанных компетенций в рамках дисциплины «Объектно-ориентированное программирование» применяются следующие методы активного/интерактивного обучения: лекция беседа, лекция консультация, лекция пресс-конференция.

## I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 5 зачетных единиц, т.е. 180 академических часов, в т.ч. аудиторные занятия - 90 часов, самостоятельная работа студента (СРС) – 90 часов.

## **Раздел 1. Язык СИ++ как элемент объектно-ориентированной технологии (12 часов)**

### **Тема 1. Введение в язык программирования С++. Основы языка программирования (2 часа)**

Введение в язык СИ++. Основные типы данных и их внутреннее представление. Элементарный ввод-вывод.

### **Тема 2. Операции и выражения (2 часа)**

Операции и выражения. Преобразование типа в выражениях. Условный оператор. Переключатель. Операторы цикла.

### **Тема 3. Приоритет операций и порядок их выполнения. Операторы управления вычислительным процессом (2 часа)**

Порядок приоритета выполнения операций. Операторы управления потоком вычислений.

### **Тема 4. Указатели и массивы (2 часа)**

Указатели и массивы. Строковые литералы. Массивы указателей. Указатели на указатели.

### **Тема 5. Указатели и функции (2 часа)**

Функции. Описание функций. Указатель на функцию.

### **Тема 6. Классы хранения и видимость переменных (2 часа)**

Классы хранения. Локальные переменные. Глобальные переменные.

## **Раздел 2. Объектно-ориентированная составляющая языка СИ++(8 часов)**

### **Тема 1. Агрегатные типы данных (2 часа)**

Структуры, объединения, перечисления. Указатели на структурную переменную. Использование структур в функциях.

### **Тема 2. Препроцессор (2 часа)**

Операторы препроцессора. Зарезервированные имена переменных.

### **Тема 3. Классы (2 часа)**

Классы. Конструкторы и деструкторы. Наследование. Полиморфизм.  
Указатели на класс.

### **Тема 4. Потоки ввода - вывода. Шаблоны и исключения (2 часа)**

Классы и потоки ввода-вывода. Связные списки, очереди и стеки.  
Шаблоны классов.

## **Раздел 3. Объектно-ориентированное проектирование программ как новая информационная технология (16 часов)**

### **Тема 1. Сложность программного обеспечения (2 часа)**

Категории методов проектирования сложных систем. Методы проектирования сложных систем

### **Тема 2. Объектные модели (4 часа)**

Абстрагирование. Инкапсуляция. Модульность. Иерархия. Типизация. Параллелизм. Сохраняемость.

### **Тема 3. Классы и объекты (4 часа)**

Классы и их свойства, связи. Отношения между классами. Объекты и их составляющие. Отношения между объектами

### **Тема 4. Классификация (2 часа)**

Основные подходы к классификации. Методы классификации

### **Тема 5. Процесс проектирования (4 часа)**

Цели, применение этапа ОО проектирования «Выявление классов и объектов». Цели, применение этапа ОО проектирования «Выяснение семантики классов и объектов». Цели, применение этапа ОО проектирования «Выявление связей между классами и объектами». Цели, применение этапа ОО проектирования «Реализация классов и объектов».

Цель, действия и результаты этапа концептуализация. Цель, действия и результаты этапа анализ. Цель, действия и результаты этапа проектирование. Цель действия и результаты этапа эволюция. Цель, действия и результаты этапа сопровождение.

Для данного курса часть лекций проводятся в форме лекции беседы, часть как лекция консультация и часть лекций проводится с использованием мультимедийных средств, заключительная лекция проводится в форме лекции - пресс-конференция. Более 60 процентов лекционных занятий проводятся с использованием активных форм обучения

Во время лекции у бакалавров должен быть раздаточный материал, который они должны активно использовать.

## **II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ ДИСЦИПЛИНЫ**

Учебным планом предусмотрены практические занятия (18 час.) и лабораторные работы (36 часов). Цель занятий – закрепить знания, полученные при изучении теоретической части дисциплины и получить практические навыки управления проектами.

Содержание лабораторных работ

**Лабораторная работа №1. Основные конструкции языка C++ (4 часа)**

Разработка и отладка алгоритмов с использованием основных конструкций языка C++.

**Лабораторная работа №2. Функции, определенные пользователем (4 часа)**

Разработка и отладка алгоритмов с функциями, определенными пользователями.

**Лабораторная работа №3. Строковые переменные на языке C++ (4 часа)**

Разработка и отладка алгоритмов строковых переменных.

**Лабораторная работа №4. Структуры и сортировка на языке C++ (4 часа)**

Разработка и отладка программ с использованием структур, сортировка

**Лабораторная работа №5. Файлы на языке C++ (4 часа)**

Разработка и отладка программ с использованием файлов.

**Лабораторная работа №6. Графика (4 часа)**

Разработка и отладка программ с использованием графики.

**Лабораторная работа №7. Динамические структуры на языке C++ (4 часа)**

Разработка и отладка программ с динамическим использованием памяти. **Лабораторная работа №8. Элементы объектно-ориентированного программирования C++ (8 часа)**

Разработка программ с элементами объектно-ориентированного программирования

Содержание практических занятий

**Практическое занятие № 1. Простейшие конструкции языка (2 часа)**

Составление и примеры программ с использованием простейших конструкций языка.

**Практическое занятие № 2. Функции, определяемые пользователем (2 часа)**

Составление и примеры программ с использованием функций, определяемых пользователем.

**Практическое занятие № 3. Обработка символьных данных(2 часа)**

Составление и примеры программ с использованием обработки символьных данных.

**Практическое занятие № 4. Структуры, сортировка и поиск(2 часа)**

Составление и примеры программ с использованием структур, сортировки и поиска.

**Практическое занятие № 5. Организация работы с файлами(2 часа)**

Составление и примеры программ с использованием организации работы с файлами.

**Практическое занятие № 6. Графика(2 часа)**

Составление и примеры программ с использованием графики.

**Практическое занятие № 7. Динамические структуры(2 часа)**

Составление и примеры программ с использованием динамических структур.

### **Практическое занятие № 8. Элементы С++ и объектно-ориентированное программирование (4 часа)**

Составление и примеры программ с использованием объектно-ориентированного программирования.

Форма проведения практических занятий – коллективное занятие с постановкой и решением проблемного задания, закрепляющего знания, полученные на лекции, и навыки, полученные на практических занятиях.

В начале занятия один из обучаемых выполняет задание у доски совместно с преподавателем и другими обучаемыми, в дальнейшем все обучаемыми получают индивидуальные задания.

### **III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине представлено в Приложении 1 и включает в себя:

план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;

характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

### **IV .КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА**

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежуточная аттестация
1	Разделы 1-3	ОПК-2 - способностью осваивать методики использования программных	Методологии, методы и средства управления разработкой информационных систем	Контрольная работа 1, собеседование (УО-1)	Вопросы к экзамену 1 - 67

		средств для решения практических задач	Разрабатывать и реализовывать планы управления разработкой информационной системы предприятия, ведущие к созданию и внедрению современной информационной системы предприятия		
			методами управления проектами		
2	Разделы 1-2	ОПК-4 - способностью участвовать в настройке и наладке программно-аппаратных комплексов	<p>Основополагающие теоретические положения, определяющие управлением процессом разработки информационных систем;</p> <p>Проводить анализ существующих методологий/средств управления проектами создания и реализации информационных систем.</p> <p>Способностью применять, контролировать качество разрабатываемых программных продуктов</p>	Контрольная работа 2 собеседование (УО-1)	Вопросы к экзамену 1 - 67
3	Раздел 3.	ПК-3 способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструменталь-	<p>Принципы планирования процесса разработки информационных систем</p> <p>проводить анализ существующих методологий/средств разработки</p>	Контрольная работа 3 собеседование (УО-1), разноуровневые задания (ПР-11)	Вопросы к экзамену 1 - 67

		ные средства и технологии программирования	систем, выбирать последовательность работ соответствующую выбранной методологии		
			методами и средствами контроля за ходом выполнения планов реализации информационной систем		

### Этапы, формы и методы для текущего контроля успеваемости

В процессе обучения проводится текущий, промежуточный и итоговый контроль достижений

Текущий контроль проводится в течение всего периода изучения курса.

В семестре текущий контроль осуществляется на лекционных и практических занятиях. На лекционных занятиях проводятся контрольные работы по основным разделам курса. Цель проведения контрольных работ - определить уровень усвоения студентами и оценить качество их теоретических знаний по данным темам.

На практических занятиях контроль осуществляется при сдаче практических заданий. Бакалавр должен свободно ориентироваться в представленном материале (отчет по практическому заданию и объяснить полученные результаты).

В соответствии с учебным планом дисциплины предусмотрены формы итоговой аттестации:

3 семестр – экзамен.

3 семестр – курсовая работа

Перечень типовых вопросов для промежуточного контроля:

По первому разделу (пример теста)

1. Представлена часть кода:

```
#definesum3( x, y, z ) x+y+z
```

...

```
int s; s=2*sum(1,2,3);
```

Каково значение s ?

2. Объявлен массив: `floatarray[ 5 ]`; Каковы значения элементов массива?

3. Объявлены массивы: `intarray_1[ 7 ]`, `array_2[ 7 ]`;

Возможна ли следующая операция? `array_1=array_2`;

4. Задана строка-литерал: `charmessage[ 30 ]="Goodbye!"`;

Каков размер данной строки в байтах?

5. Какие из следующих определений являются неверными?

а) Указатель - это адрес памяти.

б) Символ \* используется для задания указателя на объект.

в) Имя массива является константой-указателем на начальный адрес данных массива.

6. Представлена часть кода:

```
inti=1, j=2, k=3, s;
```

```
if (i<k) if (i>j) s=100; else s=55;
```

Каково значение s ?

7. Дано следующее представление:

```
typedef int (*fun_ptr) ();
```

Идентификатор `fun_ptr` задаёт:

а) указатель на функцию без параметров, возвращающую значение целого типа

б) функцию, возвращающую указатель на тип `int`

8. Автоматическими переменными являются:

а) переменные, описанные спецификатором `auto`

б) переменные, описанные внутри функции или какого-либо блока

в) переменные, имеющие только целые значения

9. Какие определения верны:

а) внешние переменные всегда отрицательны

б) класс памяти для функций external, если перед её описанием не стоит static

с) внешние переменные задаются спецификатором extern

10. Задан шаблон структуры:

```
struct DATA
{
    int day;
    int month;
    int year;    };
struct DATA *ptr;
struct DATA str1, str2;
```

Какое обращение к полю структуры верно?

11. Является ли следующий оператор ошибочным? `charch='f'-`  
б;

12. Какая функция является обязательной в программе на Си?

13. В каком порядке следуют стадии прохождения программы на Си для трансляции и перевода в исполняемый файл ( exe-файл ) ?

а) препроцессирование

б) загрузка ( сборка )

с) компиляция

14. `inti=10;float r=2.5;`

Какие операторы необходимы для вывода переменных `i` и `r` ?

15. Какие понятия являются базовыми в объектно-ориентированном программировании ?

16. Можно ли перегрузить функции (переводит строковую переменную в число) так, чтобы у них было одно имя

`double atof (const char *s)`

`int atoi (const char *s)`

`long atol (const char *s)`

1. Какая сделана ошибка при записи следующего класса:

```
class sample{
```

```
double a,b,c;
public:
double sample();
}
```

18. Есть ли ошибка в заданном фрагменте программы:

```
#include <iostream.h>
class myclass {
inti;
public:
...};
void main ()
{ myclass ob;
...
ob.i=10;
...}
```

19. Какое значение a будет выведено

```
void function (inta, intb)
{ for (inti=0;i<4;i++) a+=b;
}
void main ()
{ int a=1,b=2;
function (a,b);
```

cout<< a; 20. Может ли адрес объекта передаваться в функцию в качестве аргумента?

```
}
```

По второму разделу (основные вопросы):

1. Характерные черты сложных программных объектов.
2. Виды стиля программирования.
3. Основные составляющие объектного стиля программирования.

4. Классы и объекты.
5. Основные виды отношений между объектами.
6. Основные виды отношений между классами.
7. Примеры одиночного и множественного наследования.
8. Основные подходы к классификации объектов.
9. Характерные черты удачных проектов ПО.
10. Элементы микро процесса проектирования.

**Тематика курсовых работ:**

1. Телефонная станция. Определить перечень услуг предоставляемой телефонной станцией с точки зрения пользователя телефона и с точки зрения обслуживающего персонала.

2. Почта. Определить перечень услуг предоставляемых почтовым отделением с точки зрения клиента, и сточки зрения работника почты.

3. Абитуриент. Разработать информационную систему, которая позволяла бы осуществлять ввод информации об абитуриенте, отслеживать сдачу им экзаменов и в результате выдавать необходимую информацию. Предусмотреть выдачу информации необходимой для абитуриента.

4. Отдел кадров. Разработать информационную систему, связанную с автоматизацией работы отдела кадров, предусмотреть наличие информации о трех видов сотрудников: студентов, преподавателей и учебно-вспомогательного персонала.

5. Магазин. Разработать информационную систему, связанную с обслуживанием покупателей в магазине, предусмотреть заказ товаров и обслуживание покупателей.

6. Больница. Разработать информационную систему, связанную с регистрацией больных, предусмотреть занесение информации о лечащем враче диагнозе, номере палаты, временем болезни и т.д.

7. Клуб собаководства. Разработать информационную систему по учету собаководов. Предусмотреть ведение личной карточки владельца и питомцев, деление на породы собак, возможность выбора и просмотра по

различным критериям: порода, проживание владельца, возраст, награды и т.п.

8. Коллекция цветов. Разработать информационную систему для цветоводов-любителей с возможностью ведения каталога видов цветов (вид, название, шифр, описание, условия выращивания), адресов других коллекционеров и данных о их коллекциях.

9. Видео прокат. Разработать информационную систему для салона видео проката с возможностью ведения записей о кассетах (тип записи, когда снят, выпущен в прокат, стоимость проката, стоимость утери/покупки, время нахождения в салоне) и учета проката (список всех, бравших кассету, дата возврата, взнос, дата аренды).

10. Оптовая торговля. Разработать информационную систему учета продаж/поступлений на склад/наличия товара на складе. Предусмотреть возможность предварительного заказа товара, ведение статистики заказов/покупок/продаж, ведение базы поставщиков/заказчиков.

11. Аптека. Разработать информационную систему учета продаж/закупок/наличия препаратов в аптеке. Предусмотреть возможность продажи препаратов по рецептам, ведение баланса покупок/продаж.

12. Продажа ЖД билетов. Разработать информационную систему кассового зала ЖД вокзала. Предусмотреть три вида обращений: бронирование, покупка в кассе, отказ. Вести статистику покупок/отказов по направлениям и общей загруженности направления.

13. Продажа авиа билетов. Разработать информационную систему кассового зала аэропорта. Предусмотреть три вида обращений: бронирование, покупка в кассе, отказ. Вести статистику покупок/отказов по направлениям и общей загруженности направления для различных авиакомпаний.

14. Детский клуб (кружки, секции ...). Разработать информационную систему статистического анализа работы детского клуба. Предусмотреть возможность добавления/удаления/переименования кружков,

ведение расписания занятий и посещаемости кружка; получение информации о составе занимающихся групп, преподавателях и т.д.

15. Рекламное агентство. Разработать информационную систему для ведения бухгалтерии рекламного агентства. Предусмотреть учет заказчиков, посредников, исполнителей.

16. Продажа недвижимости. Разработать информационную систему для продажи/покупки недвижимости, ведение базы продавцов и покупателей с регистрацией покупки/продажи, регистрацией номера свидетельства о праве собственности. Предусмотреть возможность выдачи статистики о изменении стоимости квадратного метра по районам города за заданный период времени.

17. Валютные операции. Разработать информационную систему для продажи/покупки различных валют, ведение базы продавцов и покупателей валют с регистрацией номера счета-квитанции о покупке (продаже). Предусмотреть возможность выдачи статистики о изменении курса валют за заданный период времени.

18. Домашняя бухгалтерия. Разработать информационную систему для ведения домашней бухгалтерии одной семьи. Предусмотреть различные типы доходов и расходов, ведение баланса, заполнение налоговых документов, счетов.

19. Регистрация машин в ГАИ. Разработать информационную систему для выдачи информации о государственном регистрационном знаке, марка, модель, тип ТС, его идентификационном номере машины, года выпуска, номере двигателя, номере кузова владельце, предыдущем владельце, номере техпаспорта, серии и номере свидетельства о регистрации ТС и дате регистрации.

20. Коммунальные платежи. Разработать информационную систему для расчета и выдачи квитанций об оплате коммунальных платежей, регистрации оплаты.

## **V. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **Основная литература**

1. [http://e.lanbook.com/books/element.php?pl1\\_cid=25&pl1\\_id=5115](http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=5115)А шарина И.В. Объектно-ориентированное программирование вC++: лекции и упражнения. - "Горячая линия - Телеком"Издательство: 978-5-9912-7001-4ISBN: 2012: 2-е изд., стереотип. Издание: 320 стр.
2. [http://e.lanbook.com/books/element.php?pl1\\_cid=25&pl1\\_id=8781](http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=8781) Бабушкина И.А. Окулов С.М. Практикум по объектно - ориентированному программированию.- "Бином. Лаборатория знаний" Издательство: 978-5-9963-0954-2ISBN: 2012:3-е изд. (эл.)Издание:366 стр.
3. Гради Буч, Роберт АМаксимчук, Объектно-ориентированный анализ и проектирование с примерами приложений (3-е издание) Москва, Издательство: Вильямс, 2008 – 721 с.
4. Страуструп Б. Язык программирования C++. 3-е издание СПб, Издательство Бином, Невский Диалект, 2008(3-е) - , 1054стр.

### **Дополнительная литература**

1. Березкина Г.Л. Сборник заданий по языку C++ (1 занятие). Простейшие алгоритмы.,2010г.
2. Березкина Г.Л. Сборник заданий по языку C++ (2 занятие). Функции, определяемые пользователем. 2010г.
3. Березкина Г.Л. Сборник заданий по языку C++ (3 занятие). Обработка символьных данных,2010г.
4. Березкина Г.Л. Сборник заданий по языку C++ (4 занятие). Структуры. Сортировка и поиск.,2010г.
5. Березкина Г.Л. Сборник заданий по языку C++ (5 занятие). Организация работы с файлами,2010г.
6. Березкина Г.Л. Сборник заданий по языку C++ (6 занятие). Графика,2010г.

7. Березкина Г.Л. Сборник заданий по языку C++ (7 занятие). Динамические структуры, 2010г.

8. Березкина Г.Л. Сборник заданий по языку C++ (8 занятие). Элементы C++ и элементы. 2010г.

## **VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Результаты освоения дисциплины достигаются за счет использования в процессе обучения: лекций с применением мультимедийных технологий, активных методов обучения с использованием LMSBlackboard; лабораторных занятий на базе компьютерной сети на платформах Linux и Windows.

Все необходимые примеры выполнения практических заданий приведены в LMSBlackboard.

## **VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Для обеспечения учебного процесса по дисциплине «Объектно-ориентированное программирование» используется следующее материально-техническое обеспечение: компьютеры, операционная система Windows, Интернет, текстовый редактор MSWord, табличный процессор MSExcel, компьютерный класс, LMSBlackboard, LMSBlackboardCollaborate, VisualStudio 2013, персональные компьютеры студентов, а также программное обеспечение, разработанное преподавателем.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ  
РАБОТЫ ОБУЧАЮЩИХСЯ**

**по дисциплине «Объектно-ориентированное программирование»**  
**Направление – 09.03.01 «Информатика и вычислительная техника»**  
**Профиль – «Автоматизированные системы**  
**обработки информации и управления»**  
**Форма подготовки – очная**

**Владивосток**  
**2017**

## План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	В течение семестра	Проработка курса лекций и подготовка к семинарам	5	Работа на лекциях, семинарах
2	В течение семестра	Подготовка к занятиям	5	Работа на практических занятиях
3	В течение семестра	Изучение тем:  Методологии программирования  Технологии программирования	5	Выступление на семинарах/ Публикация статьи/ Выступление с докладом на студенческой научной конференции
4	В течении семестра	Выполнение курсовой работы	10	Текст пояснительной записки
5	В течении семестра	Подготовка к экзамену	27	

Количество часов на самостоятельную работу – 54 часов.

### Методические указания по самостоятельной работе студентов

1. Работа над конспектом лекции: лекции – основной источник информации по многим предметам, позволяющий не только изучить материал, но и получить представление о наличии других источников, сопоставить разные взгляды на основные проблемы данного курса. Лекции предоставляют возможность «интерактивного» обучения, когда есть возможность задавать преподавателю вопросы и получать на них ответы. Поэтому имеет смысл находить время для хотя бы беглого просмотра информации по материалу лекций (учебники, справочники и пр.) и непонятные, а также дискуссионные моменты обсуждать с преподавателем, другими студентами;

2. Подготовка к практическому/лабораторному занятию: производится, как правило, с использованием методических пособий, состоит в теоретической подготовке (особенно для семинаров) и выполнении практических заданий (решение задач, ответы на вопросы и т.д.). В данном курсе используются следующие формы практических занятий:

3. - лабораторные занятия с использованием вычислительной техники;

1. Доработка конспекта лекции с применением учебника, методической литературы, дополнительной литературы: этот вид самостоятельной работы студентов особенно важен в том случае, когда изучаемый предмет содержит много неоднозначно трактуемых вопросов, проблем. Тогда преподаватель заведомо не может успеть изложить различные точки зрения, и студент должен самостоятельно ознакомиться с ними по имеющейся литературе. Кроме того, рабочая программа предметов предполагает рассмотрение некоторых относительно несложных тем только во время самостоятельных занятий, без чтения лектором;

2. Подбор, изучение, анализ и конспектирование рекомендованной литературы;

3. Самостоятельное изучение отдельных тем, параграфов;

4. Консультации по сложным, непонятным вопросам лекций, семинаров, зачетов;

5. Подготовка к экзамену: один из самых ответственных видов самостоятельной работы, и в то же время возможность сэкономить большое количество времени в период сессии, если эту подготовку начинать заблаговременно. Одно из главных правил – представлять себе общую логику предмета, что достигается проработкой планов лекций, составлением опорных конспектов, схем, таблиц. Фактически основной вид подготовки к экзамену – «свертывание» большого объема информации в компактный вид, а также тренировка в ее «развертывании» (примеры к теории, выведение одних закономерностей из других и т.д.). Владение этими технологиями обеспечивает,

пожалуй, более половины успеха. Тем более что преподаватель обычно замечает в течение семестра целенаправленную подготовку такого студента и может поощрить его тем или иным способом. Надо также правильно распределить силы, не только готовясь к самому экзамену, но и позаботившись о допуске к нему (часто это хорошее посещение занятий, выполнение в назначенный срок практических заданий, активность на семинарах). Наконец, необходимо выяснить условия проведения самого экзаменационного испытания, используя для этой цели, прежде всего консультацию (хотя преподаватель обычно касается этой темы заранее), - количество и характер вопросов, форму проведения (устно или письменно), возможность использования при подготовке различных материалов и пособий (таблицы, схемы, тетради для практических занятий и т.д.).

б. Используется следующая форма научной самостоятельной работы (долговременная): подготовка доклада к конференции: часто студенты для выступлений на научных и научно-практических конференциях используют материалы курсовых работ. Это вполне оправдано, но тогда возникают два вопроса: как обеспечить этим материалам качество научного доклада, который должен решать определенную проблему, иметь новизну и актуальность: как быть первокурсникам, еще не защитившим ни одну курсовую работу. Видимо, каждый студент должен определиться с первой научной темой уже в первые месяцы учебы, что позволит расширить круг интересов, приобрести важные навыки педагога - исследователя, необходимые в дальнейшем совершенствовании в своей профессии. Отсюда следует полезность раннего начала знакомства с литературой, что является вторым этапом любой научной (и методической) работы (первый этап – определение проблемы, темы и гипотезы исследования). Следующий очень важный шаг – правильно спроектировать и осуществить практическую реализацию. Один из самых ответственных этапов – обобщение результатов реализации, что сопровождается анализом качества проекта и анализом заиратнаегорелизацио. Последнее – формулировка выводов, содержащих данные о решении проблемы предмет-

ной области или исследования, положительном или отрицательном (в чем нет ничего страшного) результате. В заключении часто намечают основные пути расширения работы, ее продолжения. Обычно доклад иллюстрируется наглядными презентациями, которые необходимо заранее подготовить.

Таким образом, все виды самостоятельной работы взаимосвязаны и взаимообусловлены, ведущее место занимает учебная самостоятельная деятельность.

Все они направлены на повышение как личностных, так и компитентностных качеств будущего специалиста.

### ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

Курсовая работа является одним из видов самостоятельной работы студентов. Курсовая работа по дисциплине “Объектно-ориентированное программирование” – курсовая работа, выполняется в 3 (осеннем) семестре по индивидуальным заданиям.

Целью выполнения курсовой работы является получение навыков самостоятельной работы по решению поставленной задачи: поиск и анализ теоретического материала, разработка алгоритма, составление программы, ее отладка и тестирование.

Тематика курсовых работ:

- I. Проектирование и реализация объектно-ориентированных приложений
- II. Проектирование и реализация сетевых обучающие программ по разделам дисциплины.

Темы курсовых работ:

1. Разработка объектно-ориентированного приложения «Телефонная станция»
2. Разработка объектно-ориентированного приложения «Почта»
3. Разработка объектно-ориентированного приложения «Абитуриент»

4. Разработка объектно-ориентированного приложения «Отдел кадров»
5. Разработка объектно-ориентированного приложения «Магазин»
6. Разработка объектно-ориентированного приложения «Больница»
7. Разработка объектно-ориентированного приложения «Клуб собаководства»
8. Разработка объектно-ориентированного приложения «Коллекция цветов»
9. Разработка объектно-ориентированного приложения «Видео прокат»
10. Разработка объектно-ориентированного приложения «Оптовая торговля»
11. Разработка объектно-ориентированного приложения «Аптека»
12. Разработка объектно-ориентированного приложения «Продажа ЖД билетов»
13. Разработка объектно-ориентированного приложения «Продажа авиа билетов»
14. Разработка объектно-ориентированного приложения «Детский клуб»
15. Разработка объектно-ориентированного приложения «Рекламное агентство»
16. Разработка объектно-ориентированного приложения «Продажа недвижимости»
17. Разработка объектно-ориентированного приложения «Валютные операции»
18. Разработка объектно-ориентированного приложения «Домашняя бухгалтерия»
19. Разработка объектно-ориентированного приложения «Регистрация машин в ГАИ»

## 20. Разработка объектно-ориентированного приложения «Коммунальные платежи»

Объектно-ориентированное приложение по любой из перечисленных тем, должно быть организовано как приложение, имеющее общий программный интерфейс, и демонстрировать выполнение всех функций в любой последовательности.

К работе оформляется отчет. Он должен иметь объем не менее 15 страниц текста, выполнен в соответствии с ГОСТ 7.32-91 и документа “Требования к оформлению письменных работ, выполняемых студентами и слушателями ДВФУ”.

Документация выполняется с помощью компьютерного набора с применением печатающих и графических устройств вывода ЭВМ. Поля оставляют по всем четырем сторонам текста. Размер левого поля - 30 мм, правого - 10, верхнего и нижнего - 20 мм. Следует использовать шрифт TimesNewRomanСур., размер шрифта 14, межстрочный интервал 1,5.

Иллюстрации (чертежи, схемы, графики, диаграммы, фотоснимки) и таблицы располагаются непосредственно в тексте, в котором они упоминаются впервые. На все иллюстрации и таблицы в тексте должны быть ссылки. Примечания приводят в документации, если необходимы пояснения или справочные данные к содержанию текста, таблиц или графического материала. Примечания помещают непосредственно после текстового, графического материала или в таблице, к которым они относятся.

Схемы алгоритмов и программ разрабатываются согласно требованиям стандартов ГОСТ 34.xxx и ГОСТ 19.xxx. В случае применения современных методов моделирования и методологий разработки систем допускается использование обозначений, дополняющих указанные стандарты. При этом используемые обозначения должны соответствовать общепринятым, а работа содержать ссылку на источник, определяющий эти обозначения.

Отчет должен содержать:

1. Техническое задание и пояснительную записку

2. CD с разработанным программным обеспечением,
3. Необходимую графическую информацию
4. Структуру информационного обеспечения
5. Структуру программного обеспечения
6. Основные экранные формы

Курсовая работа предоставляется в сроки, установленные графиком, и защищается путем публичной презентации. Студент предьявляет разработанный им отчет и проводит презентацию, изложенных в нем результатов исследований. По результатам презентации выставляется дифференцированная оценка.

### **Этапы создания объектно-ориентированного приложения**

Исходными данными для составления пояснительной записки является задание на курсовую работу, пользовательские требования, разработанные обучаемым, анализ предметной области, литературных источников и т.д. Работа над разделами включает рассмотрение основных теоретических вопросов, необходимых для ясного представления о путях реализации функций приложения, его архитектуре, взаимосвязи и совместимости процессов автоматизированной и не автоматизированной деятельности и т.д. Определяются все функции и действия, которые будет выполнять разрабатываемое приложение, рассматриваются все ограничения (конструктивные, технологические, алгоритмические и т.п.), которые необходимо учесть, формируются требования к структуре приложения, алгоритмам обработки, структурам данных, базе данных, интерфейсу и т.п.

Таблица 1. Содержание работ и их результат

№	Этап	Содержание работ	Результат работ в ПЗ
1.	Определение задачи	Определение цели приложения, выполняемые задачи. Поиск путей решения, анализ альтернатив, имеющихся решений.	Анализ пользовательских требований (реализуемость, тестируемость). Анализ.
2.	Планирование работ	Планирование работ по созданию системы/ПО. Определение и анализ рисков.	План работ по созданию системы/ПО (диаграмма Ганта). Анализ рисков.

3.	Установление границ приложения	<p>Построение модели приложения.</p> <p>Анализ с целью установить, что является, а что не является частью разрабатываемого приложения.</p>	<p>Структурная модель приложения высокого уровня, модель процессов, словарь, бизнес прецеденты, модель данных.</p> <p>Выделение задач решаемых ПО.</p>
4.	Специфицирование требований ПО приложения	<p>Построение моделей ПО с целью детализированного описания перечня сервисов, которые должно выполнять приложение с указанием как приложение реагирует на те или иные входные данные, как ведет себя в определенных ситуациях и т.д.</p> <p>Выявление функциональных требований к ПО приложения, системных требований, проверка их реализуемости и тестируемости. Выявление нефункциональных требований. Определение ограничений накладываемых на действия и функции, выполняемые приложением, временные ограничения, стандарты и т.д. Обеспечение качества ПО.</p>	<p>Модели (состояний, поведения, изменения состояния) и их описания.</p> <p>Методы использованные при выявлении требований (опросные листы и анкеты, протоколы совещаний, ...).</p> <p>Документ спецификации требований.</p> <p>Возможный вид: ТЗ на приложение (ГОСТ 34.602-89. Автоматизированные системы. Техническое задание на создание автоматизированной системы.); спецификация требований к программному обеспечению (SRS) (Стандарт IEEE 830-1998); шаблоны документов описания требований (ISO, IEEE/ANSI 830-1993, RUP).</p> <p>План тестирования.</p>
5.	Определение архитектуры ПО	<p>Определение совокупности частей приложения и взаимодействия между ними.</p> <p>Выбор модели управления взаимоотношениями между частями ПО.</p>	<p>Описание архитектуры ПО.</p> <p>Описание управления ПО.</p>
6.	Проектирование интерфейса пользователя .	<p>Анализ деятельности пользователя. Создание проекта прототипа интерфейса.</p>	<p>Описание требований и критериев создания пользовательского интерфейса.</p> <p>Варианты страниц приложения, представления данных, графических элементов.</p>
7.	Проектирование	<p>Выявление классов. Шаблоны проектирования. Проектирование базы данных. Тестирование.</p> <p>Документирование.</p> <p>Организация сопровождения и развития ПО.</p>	<p>Методы выявления классов. Диаграмма классов, деятельности, последовательности. Проект базы данных. Реализация плана тестирования. Примеры тестов.</p> <p>Обоснование перечня необходимых документов.</p>
8.	Прототипирование	<p>Создание прототипа ПО, демонстрирующего правильность принятых в ходе проектирования решений.</p>	<p>Прототип ПО.</p>



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
по дисциплине **«Объектно-ориентированное программирование»**  
Направление – **09.03.01 «Информатика и вычислительная техника»**  
Профиль – **«Автоматизированные системы  
обработки информации и управления»**  
**Форма подготовки – очная**

**Владивосток**  
**2017**

## Паспорт ФОС

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК-2 - способностью осваивать методики использования программных средств для решения практических задач	Знает	принципы, методы формализации, алгоритмизации и реализации программного обеспечения с помощью языков программирования;
	Умеет	проводить анализ существующих средств разработки программного обеспечения систем, их выбор, внедрение и применение для решения поставленных задач;
	Владеет	методами и средствами реализации программного обеспечения;
ОПК-4 - способностью участвовать в настройке и наладке программно-аппаратных комплексов	Знает	современные технологии разработки программных комплексов с использованием CASE-средств
	Умеет	использовать на практике CASE-средства
	Владеет	инструментальными средами для различных языков программирования
ПК-3 способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования	Знает	Принципы системного подхода, и связанные с ним технологии программирования
	Умеет	Основополагающие теоретические положения, определяющие процесс разработки программного обеспечения информационных систем;
	Владеет	Различными методологиями и технологиями разработки программного обеспечения информационных систем

N п/п	Контролируемые разделы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименования	
				текущий контроль	промежуточная аттестация
1	Язык СИ++ как элемент объектно-ориентированной технологии	ОПК-2, ОПК-4,  ПК-3	Знает	собеседование УО-1	собеседование УО-1
			Умеет	собеседование УО-1	собеседование УО-1
			Владеет	собеседование УО-1	собеседование УО-1
2	Объектно-ориентированная составляющая языка СИ++	ОПК-2, ОПК-4,  ПК-3	Знает	собеседование УО-1	собеседование УО-1
			Умеет	собеседование УО-1	собеседование УО-1

			Владеет	собеседование УО-1	собеседование УО-1
3	Объектно-ориентированное проектирование программ как новая информационная технология	ОК-2, ОК-4, ПК-3	Знает	собеседование УО-1	собеседование УО-1
			Умеет	собеседование УО-1	собеседование УО-1
			Владеет	собеседование УО-1	собеседование УО-1

### Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
	знает (пороговый уровень)	умеет (продвинутый)		
ОПК-2 - способностью осваивать методики использования программных средств для решения практических задач	знает (пороговый уровень)	принципы, методы формализации, алгоритмизации и реализации программного обеспечения;	знает методы формализации, алгоритмизации и реализации программного обеспечения	способность дать определенным методам формализации, алгоритмизации и реализации программного обеспечения
	умеет (продвинутый)	проводить анализ существующих методологий/средств разработки систем, их выбор, внедрение и применение для решения поставленных задач;	умеет проводить анализ существующих методологий/средств разработки систем, их выбор, внедрение и применение для решения поставленных задач	способность проводить анализ существующих методологий/средств разработки систем, их выбор, внедрение и применение для решения поставленных задач
	владеет (высокий)	методами и средствами анализа, моделирования и оптимизации объектов профессиональной деятельности и их компонентов;	владеет методами и средствами анализа, моделирования и оптимизации объектов профессиональной деятельности и их компонентов	способность использовать методы и средства анализа, моделирования и оптимизации объектов профессиональной деятельности и их компонентов
ОПК-4 - способностью участвовать в настройке и наладке программно-аппаратных комплексов	знает (пороговый уровень)	современные технологии разработки программных комплексов с использованием CASE-средств	знает современные технологии разработки программных комплексов с использованием CASE-средств	способность дать определение, определить современные технологии разработки программных комплексов с использованием CASE-средств
	умеет (продвинутый)	использовать на практике навыки организации исследовательских и проектных работ, в управлении коллективом	умеет проводить анализ с помощью современных технологий разработки программных комплексов с использованием CASE-средств	способность проводить анализ с помощью современных технологий разработки программных комплексов с использованием CASE-средств
	владеет (высокий)	формировать технические задания и участвовать в разработке аппаратных и/или программных средств вычислительной тех-	владеет методами анализа с помощью современных технологий разработки программных комплексов с использованием CASE-средств	способность практического применения результатов анализа на основе современных технологий разработки программных комплексов с ис-

		ники		пользованием CASE-средств
ПК-3 способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования	знает (пороговый уровень)	Принципы объектного подхода, и связанные с ним объектно-ориентированные языки программирования	знает принципы объектного подхода, и связанные с ним объектно-ориентированные языки программирования	способность назвать, определить принципы объектного подхода, и связанные с ним объектно-ориентированные языки программирования
	умеет (продвинутый)	Основополагающие теоретические положения, определяющие процесс разработки программного обеспечения информационных систем;	умеет свободно использовать основополагающие теоретические положения, определяющие процесс разработки программного обеспечения информационных систем	способность свободно использовать основополагающие теоретические положения, определяющие процесс разработки программного обеспечения информационных систем
	владеет (высокий)	Объектно-ориентированной методологией разработки программного обеспечения информационных систем	владеет объектно-ориентированной методологией разработки программного обеспечения информационных систем	способность свободно реализовывать объектно-ориентированную методологию разработки программного обеспечения информационных систем

### **Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины**

**Текущая аттестация студентов.** Текущая аттестация студентов по дисциплине проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация по дисциплине «Объектно-ориентированное программирование» проводится в форме контрольных мероприятий (выполнение практических заданий) по оцениванию фактических результатов обучения студентов и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине);
- степень усвоения теоретических знаний;
- уровень овладения практическими умениями и навыками по всем видам учебной работы;
- результаты самостоятельной работы.

Краткая характеристика оценочных средств:

- УО-1 - Собеседование - средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний, обучающегося по определенному разделу, теме, проблеме и т.п.
- УО-3 - Доклад, сообщение - продукт самостоятельной работы обучающегося, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы
- УО-4 - Круглый стол, дискуссия, полемика, диспут, дебаты - оценочные средства, позволяющие включить обучающихся в процесс обсуждения спорного вопроса, проблемы и оценить их умение аргументировать собственную точку зрения.
- ПР-1 – Тест – система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающихся.
- ПР-11 - Разноуровневые задачи - реконструктивного уровня, позволяющие оценивать и диагностировать умения синтезировать, анализировать, обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением причинно-следственных связей; творческого уровня, позволяющие оценивать и диагностировать умения, интегрировать знания различных областей, аргументировать собственную точку зрения.

### **Критерии оценки устных ответов**

- 100-85 баллов - если ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

- 85-76 - баллов - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.

- 75-61 - балл – оценивается ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

- 60-50 баллов – ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области.

### **Критерии оценки выполнения практических занятий**

- 100-86 баллов - если ответ показывает глубокое и систематическое знание всего программного материала и структуры конкретного вопроса, а также основного содержания и новаций лекционного курса по сравнению с учебной литературой. Студент демонстрирует отчетливое и свободное владение концептуально-понятийным аппаратом, научным языком и терминологией соответствующей научной области. Знание основной литературы и зна-

комство с дополнительно рекомендованной литературой. Логически корректное и убедительное изложение ответа.

- 85-76 - баллов - знание узловых проблем программы и основного содержания лекционного курса; умение пользоваться концептуально-понятийным аппаратом в процессе анализа основных проблем в рамках данной темы; знание важнейших работ из списка рекомендованной литературы. В целом логически корректное, но не всегда точное и аргументированное изложение ответа.

- 75-61 - балл – фрагментарные, поверхностные знания важнейших разделов программы и содержания лекционного курса; затруднения с использованием научно-понятийного аппарата и терминологии учебной дисциплины; неполное знакомство с рекомендованной литературой; частичные затруднения с выполнением предусмотренных программой заданий; стремление логически определено и последовательно изложить ответ.

- 60-50 баллов – незнание, либо отрывочное представление о данной проблеме в рамках учебно-программного материала; неумение использовать понятийный аппарат; отсутствие логической связи в ответе.

#### **. Методические указания по проведению аттестации студентов**

Промежуточная аттестация студентов по дисциплине «объектно-ориентированное программирование» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Промежуточная аттестация (зачёт) предусмотрена в устной форме с использованием такого оценочного средства, как устный опрос в форме собеседования.

Устный опрос в форме собеседования (УО-1) по ранее известному кругу вопросов позволяет оценить не только знания, но и кругозор обучающегося, навыки логического построения ответов. В ходе собеседования создаются условия, при которых обучающийся имеет возможность показать владение научной лексикой, продемонстрировать, насколько хорошо он ориентируется в предметной области, связанной с данной дисциплиной.

## Критерии оценивания решения тестовых заданий

По результатам решения тестовых заданий количество правильно решенных заданий переводится в традиционные оценки посредством применения следующей шкалы:

86% правильно решенных заданий – «отлично»,

75% правильно решенных заданий – «хорошо»,

61% правильно решенных заданий – «удовлетворительно»,

менее 61% - «неудовлетворительно».

## Тестовые задания

1. Задан фрагмент программы:

```
#include<iostream>

int main ( )
{... f(); ...
return 0; }

void f()
{... cout<< “программа “; }
```

Будет ли она скомпилирована как программа на языке C++, C?

да, как программа C

да, как программа C++

нет

2. Какое определение правильное:

Дружественная функция - это

а) функция не являющаяся членом класса, но обеспечивающая доступ к закрытой части класса, для которого она дружественная.

б) функция, которая на самом деле не вызывается, а ее тело встраивается в программу на месте ее вызова.

в) функция, которая определяет базовый набор операций, применяющийся к различным типам данных.

3. Что выведет на печать следующий фрагмент кода ?

```
int I;  
for (I=0;I<5;I++)  
cout<<I;  
cout<< "\n"
```

4. Можно ли перегрузить функции (переводит строковую переменную в число) так, чтобы у них было одно имя

doubleatof (const char \*s)

intatoi (const char \*s)

longatol (const char \*s)

да

нет

возможно, при условии ...

5. Объявлены массивы:

```
intarray_1[ 7 ], array_2[ 7 ];
```

Возможна ли следующая операция ?

```
*array_1=*array_2;
```

да

нет

6. Является ли данный код правильным? Если да, то что он выведет?  
(Напишите)

```
cout << (int *) "Home of the jolly bytes"
```

7. Пусть дан следующий класс, каким образом объявить объект ob, что бы значение 100 передать переменной a, значение 'x' переменной c:

```
class sample {  
    int a; char c;  
    public:  
    sample(int x, char ch) { a=x; c=ch;}
```

```
    sample ob;
```

```
    a.ob=100;
```

```
    x.ob='x';
```

```
    sampleob(100, 'x')
```

8. Какая сделана ошибка при записи следующего класса:

```
class sample{  
    doublea,b,c;  
    public:  
    doublesample(); }
```

не описан способ доступа к переменным

конструктор возвращает значение

нет параметра у конструктора

нет дистриктора

9. Представлена часть кода:

```
#define sum3( x, y, z ) x+y+z
```

```
...
```

```
int s;
```

```
s=2*sum3(1,2,3);
```

Каково значение s ?

s=7

s=12

неопределено

10. Может быть дружественная функция дружественна более чем одному классу?

да, двум

да

нет

11. Правильен ли следующий фрагмент:

```
int&f( ); ... ;int *x; x=f( );
```

да

нет

12. Задан шаблон структуры:

```
struct DATA
{
    int day;
    int month;
    int year;    };
struct DATA *ptr;
struct DATA str1, str2;
```

Какое обращение к полю структуры верно?

```
str2.month="april";
```

```
str1.day=60;
```

```
int gg=ptr -> year;
```

```
intdd=str2 -> day;
```

13. Какие определения верны:

- а) внешние переменные всегда отрицательны
- б) класс памяти для функций external, если перед её описанием не стоит static
- с) внешние переменные задаются спецификатором extern

а)

б)

с)

14. Является ли следующий оператор правильным? charch='f' - б;

да

нет

15. Дано следующее представление:

```
typedef int ( *fun_ptr ) ( );
```

Идентификатор fun\_ptr задаёт:

- а) указатель на функцию без параметров, возвращающую значение целого типа
- б) функцию, возвращающую указатель на тип int
- в) нет верного ответа

16. Есть ли ошибка в заданном фрагменте программы:

```
#include <iostream.h>
```

```
class myclass {
```

```
inti;
```

```
public: ... };
```

```
void main ()
```

```
{ myclassob; ...ob.i=10; ...}
```

да

нет

17. Какое значение a будет выведено

```
voidfunction (inta, intb)
```

```
{for (inti=0;i<4;i++) a+=b;
```

```
}
```

```
void main ()
```

```
{ int a=1,b=2;
```

```
function (a,b);
```

```
cout<< a; }
```

7

1

NULL

18. Как называются составные части программы на C++;

процедуры

функции

модули

19. Что будет выведено после выполнения фрагмента программы:

```
int function (int a)
```

```
{ int t=a+g;
```

```
return t;}
```

```
void main ()
```

```
{int g=5; cout<< function ( 4) ;}
```

ничего

5

сообщение об ошибке

20. Какие значения будут выведены

```
int function(int a){
```

```
    static int g;
```

```
    if(a<0)g=3;
```

```
        g+=a;}
```

```
void main(){
```

```
    int p=-1;
```

```
    cout<<function(p);
```

```
        p=4;
```

```
    cout<<function(p); }
```

2 и 6

2 и 7

2 и неизвестно что

неизвестно что и неизвестно что

сообщение об ошибке

### **Вопросы для итогового экзамена**

1. Основными принципами объектно-ориентированного подхода являются.
2. Что такое объектно-ориентированное программирование.
3. Составляющие ООП 3 основных части.
4. Что такое ООП.
5. Что такое ООА (ОО анализ).

6. Определение и виды стиля программирования.
7. Составляющие ООС (ОО стиль).
8. Что такое абстрагирование.
9. Виды абстракций.
10. Какой объект называют клиентом.
11. Из чего складывается поведение объекта.
12. Что такое протокол объекта.
13. Приведите примеры статических и динамических свойств объекта.
14. Что такое ограничение доступа.
15. Что такое модульность.
16. Правила, используемые при разделении на модули.
17. Что такое иерархия.
18. Приведите пример иерархии.
19. Что такое типизация.
20. Что такое параллелизм.
21. Что такое устойчивость.
22. Перечислите объекты, определяющие устойчивость.
23. Составляющие микропроцесса ОО проектирования.
24. Цели этапа ОО проектирования «Выявление классов и объектов».
25. Применение этапа «Выявление классов и объектов» в анализе и проектировании.
26. Цели этапа ОО проектирования «Выяснение семантики классов и объектов».
27. Применение этапа «Выявление классов и объектов» в анализе и проектировании.
28. Цели этапа ОО проектирования «Выявление связей между классами и объектами».
29. Применение этапа «Выявление классов и объектов» в анализе и проектировании.
30. Цели этапа ОО проектирования «Реализация классов и объектов».

31. Применение этапа «Выявление классов и объектов» в анализе и проектировании.
32. Действия макропроцесса ОО проектирования.
33. Цель этапа концептуализация
34. Результаты этапа концептуализация
35. концептуализация
36. Цель этапа анализ
37. Результаты этапа анализ
38. Цель этапа проектирование
39. Результаты этапа проектирование
40. Цель этапа эволюция
41. Результаты этапа эволюция
42. Цель этапа сопровождение
43. Результаты этапа сопровождения
44. Определение объекта.
45. Характеристики объекта
46. Состояние.
47. Поведение
48. Идентичность
49. Виды отношений между классами.
50. Ассоциация
51. Агрегация
52. Инстанирование
53. Наследование
54. Использование
55. Метакласс