

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
ФАКУЛЬТЕТ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**Рабочая программа дисциплины  
«Объектно-ориентированное программирование»**

Кафедра **Информатики и Информационных технологий**

**Образовательная программа**

информационные системы и технологии 09.03.02

**Профиль подготовки:** информационные системы и технологии


**Уровень высшего образования:** бакалавриат

**Форма обучения:** очная

**Статус дисциплины:** вариативная

**Махачкала 2017**

Рабочая программа дисциплины «Объектно-ориентированное программирование» составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки **09.03.02 – Информационные системы и технологии**, уровень подготовки: бакалавриат, утвержденного приказом Минобрнауки РФ от 12 марта 2015 г. №219.

Составитель: Ахмедова Написат Мурадовна, старший преподаватель кафедры информатики и информационных технологий. 

Рабочая программа одобрена на заседании кафедры информационных систем и технологий  
17.08.2017 г., протокол № 8.

Заведующий кафедрой  Ахмедов С.А.

Рабочая программа одобрена на заседании Методической комиссии факультета информатики и информационных технологий  
31.03.2017 г., протокол № 7.

Председатель методсовета факультета 

Рабочая программа согласована с учебно-методическим управлением  
31.03.2017 г., 

## Аннотация рабочей программы дисциплины

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

Дисциплина реализуется на факультете ИиИТ кафедрой ИиИТ.

Содержание дисциплины охватывает круг вопросов связанных с базовыми понятиями: объект, класс, парадигмы программирования.

Дисциплина нацелена на формирование следующих компетенций выпускника: профессиональных – ПК-33.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, лабораторные работы и самостоятельную работу.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме контрольных работ и промежуточный контроль в форме экзамена.

Объем дисциплины 4 зачетные единицы, в том числе в академических часах по видам учебных занятий:

Се мес тр	Учебные занятия						Форма промежуточн ой аттестации  (зачет, дифференцир ованный зачет, экзамен)	
	в том числе							
	Контактная работа обучающихся с преподавателем					СРС, в том числ е экза мен		
	Всег о	из них						
Лек ции		Лаборат орные занятия	Практи ческие заняти я	КСР	консул ьтации			
7	144	32	72	0	6	4	30	экзамен

## **1. Цели освоения дисциплины**

Цели дисциплины заключаются в следующем:

- 1) Изучение основ классической теории объектно-ориентированного программирования, в том числе:
- 2) Пути эволюции технологий программирования от алгоритмического к ООП
- 3) Основных принципов объектно-ориентированного построения программных систем (Абстракция, Инкапсуляция, Иерархия, Модульность, Типизация, Параллелизм, Сохраняемость)
- 4) Понятий классов, объектов, взаимоотношений между ними, а также многоуровневой модели OMG
- 5) Изучение средств объектно-ориентированного и обобщенного программирования языка C++, средств стандартной библиотеки STL
- 6) Изучение средств объектно-ориентированного программирования языка Java, платформы Java, стандартной библиотеки классов, основ многопоточного и распределенного программирования, безопасности программных систем использующих технологию Java.

## **2. Место дисциплины в структуре образовательной программы**

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

Изучение дисциплины опирается на курсы «Алгебра и геометрия», «Информатика», «Технологии программирование», «Методы и средства проектирования информационных систем и технологий».

Предварительными требованиями к студентам являются:

- 1) Знание одного из классических процедурно-ориентированных языков, предпочтительно языка C;
- 2) Знания в области алгоритмической декомпозиции, основных структур данных и технологий работы с ним;

3) Знание основ теории множеств.

Данный курс является предшествующим для следующих дисциплин/курсов:  
«Надежность информационных систем», «Геоинформационные системы».

### 3. Компетенции обучающегося, формируемые в результате освоения дисциплины

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО по данному направлению:

Компетенции	Формулировка компетенции из ФГОС ВО	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)
ПК-33	способностью составлять инструкции по эксплуатации информационных систем	<b>Знать:</b> 1) основы технологии объектно-ориентированной декомпозиции программных систем, базовых шаблонов проектирования (Наблюдатель, Итератор, Одиночка, Фабрика, Заместитель), отношений между классами и основ UML (диаграммы классов и последовательностей). 2) особенности построения объектно-ориентированных программных систем на C++ 3) платформы Java, особенности построения программных систем Java. <b>Уметь:</b> 1) пользоваться основными инструментальные средства языка C++ и стандартной библиотеки 2) пользоваться средствами реализации принципов ООП и инструментальными средства языка Java <b>Владеть:</b> 1) Основами технологий построения простейших распределенных информационных систем и обеспечения безопасности.

#### 4. Объем, структура и содержание дисциплины

4.1. Объем дисциплины составляет 4 зачетные единицы, 144 академических часа.

4.2. Структура дисциплины.

№ раз- дела	Наименование разделов и тем	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)					Формы текуще го контро ля успевае мости (по неделя м семест ра)
		семестр	Аудиторна я Работа		КС Р	Вне- ауд. рабо та СР	
			Л	ЛР			
<b>Модуль 1</b>							
1	<b>Раздел 1.</b> Введение. Система. Информация.	7	12	12		12	Устны й и письме нный опросы , контр. работа, доклад ы по темам,
<b>Модуль 2</b>							
2	<b>Раздел 2.</b> Основные задачи теории информационных систем и составные части системы.	7	10	12		14	Устны й и письме нный опросы , контр. работа, доклад ы по темам
<b>Модуль 3</b>							

3	<b>Раздел 3.</b> Информационное обеспечение информационных систем. Методы и модели описания систем.	7	10	12		14	Устный и письменный опросы, контр. работа, доклады по темам
<b>Модуль 4</b>							
	<b>Подготовка к экзамену</b>	7			6	3	
	<b>Экзамен</b>				27		
	<b>Итого за 4 модуль:</b>				33	3	
	<b>Итого</b>	<b>144</b>	32	36	33	43	

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

Данный курс делает акцент на технологии объектно-ориентированного подхода и средствах поддержки принципов ООП в инструментальных языках.

##### Детальная программа курса:

##### *Объектная модель*

#### **1. Основные принципы объектно-ориентированного программирования**

##### 1.1. Эволюция методологий программирования

1.1.1. Начало начал, или первое поколение языков программирования.

1.1.2. Развитие алгоритмических абстракций, или второе поколение языков программирования.

1.1.3. Модуль как единица построения программных систем, третье поколение языков программирования.

1.1.4. Зарождение объектной модели, четвертое поколение языков программирования.

1.1.5. Объектные языки программирования, объектно-ориентированные языки

программирования, объектно-ориентированный анализ, дизайн и проектирование.

1.1.6. Парадигмы программирования.

##### 1.2. Составные части объектного подхода

1.2.1. Абстрагирование

1.2.2. Инкапсуляция

- 1.2.3. Модульность
- 1.2.4. Иерархия
- 1.2.5. Типизация
- 1.2.6. Параллелизм
- 1.2.7. Сохраняемость

## **2. Объектно-ориентированная модель**

- 2.1. Понятие объекта
- 2.2. Свойства, присущие объектам
  - 2.2.1. Состояние
  - 2.2.2. Поведение
  - 2.2.3. Идентичность
- 2.3. Отношения между объектами
  - 2.3.1. Типы отношений
  - 2.3.2. Связь (ассоциация)
  - 2.3.3. Агрегация

## **3. Классы**

- 3.1. Природа классов.
- 3.2. UML – унифицированный язык моделирования. Четырехуровневая метамодель MOF
- 3.3. Отношения между классами.
  - 3.3.1. Типы отношений
  - 3.3.2. Ассоциация
  - 3.3.3. Агрегация
  - 3.3.4. Использование
  - 3.3.5. Наследование
  - 3.3.6. Инстанцирование
- 3.4. Отношения между классами и объектами

## **4. Основные алгоритмические отличия C++**

- 4.1. Использование ссылок. Передача аргументов функции по ссылке.
- 4.2. Использование констант.
- 4.3. Логические тип и перечисления.
- 4.4. Операторы управления динамической памятью, инициализация массивов.
- 4.5. Структура программы, отдельная компиляция и особенности использования статической памяти.
- 4.6. Пространства имен и исключения (краткий обзор)
- 4.7. Библиотека ввода вывода (краткий обзор iostream)
- 4.8. Функциональный полиморфизм.

## ***Инструментальные средства языка Java***

### **1. Виртуальная машина Java**

- 1.1. История и предпосылки появления Java.
- 1.2. Понятие виртуальной машины. Среда исполнения и байт-код. Взаимодействие виртуальной машины с операционной системой.
- 1.3. Пространства классов. Структура приложений на Java. Загрузка классов и



инициализация объектов.

1.4.Сферы применения Java в современном информационном мире.

1.5.Версии Java машины и их эволюция.

1.6.Средства ООП, непосредственно поддерживаемые в Java.

1.7.Простейшие приложения на Java.

## **2. Алгоритмические средства языка Java.**

2.1.Строгая типизация Java.

2.2.Базовые типы языка. Строки и литералы.

2.3.Преобразования типов в выражениях.

2.4.Особенности инициализации массивов. Операторы управления памятью.

2.5.Краткий обзор операторов. Использование break.

## **3. Средства объектного программирования языка Java.**

3.1.Представление объектов и классов.

3.1.1. Структура объявления класса.

3.1.2. Доступ к членам класса.

3.1.3. Спецификаторы доступа для обеспечения инкапсуляции.

3.1.4. Знакомство с final.

3.1.5. Средства управления жизнью объекта. Конструкторы и метод finalize().

3.1.6. Принципы работы сборщика мусора.

3.1.7. Работа с массивами объектов.

3.1.8. Статические поля и методы классов. Классы – утилиты.

3.1.9. Блок статической инициализации.

3.1.10. Принцип работы ClassLoader.

3.2.Реализация отношений между объектами и классами.

3.2.1. Ассоциация и агрегация объектов и классов.

3.2.2. Использование и зависимость от интерфейсов.

3.2.3. Объекты при передаче параметров и возврате из методов.

3.2.4. Реализация отношения Клиент – Сервер.

3.2.5. Внутренние классы.

## **4. Средства объектно-ориентированного программирования Java.**

4.1.Наследование в Java.

4.1.1. Производные классы.

4.1.2. Класс Object. Метод toString().

4.1.3. Управление доступом в производных классах.

4.1.4. Абстрактные классы и интерфейсы.

4.1.5. Реализация интерфейсов как альтернатива множественному наследованию.

4.1.6. Информация о типе на этапе исполнения. Оператор instanceof. Приведение типов.

4.1.7. Использование класса Class.

4.2.Перечисления Java (java.lang.Enum)

4.2.1. Особенности классов перечислений

4.2.2. Члены перечисления, поведение

4.2.3. Использование перечислений

4.3.Пакеты.

- 4.3.1. Определение пакета.
- 4.3.2. Ограничение доступа.
- 4.3.3. Импортирование пакетов. Разрешение конфликтов имен.
- 4.4. Обработка исключений.
- 4.4.1. Основные принципы и типы исключительных ситуаций.
- 4.4.2. Перехват исключительных ситуаций. Операторы try, throw, throws, catch, finally.
- 4.4.3. Использование нескольких блоков catch и вложенный оператор try.
- 4.4.4. Не перехваченные исключительные ситуации.
- 4.4.5. Встроенные исключительные исключения Java. Классы Throwable и Exception.
- 4.4.6. Принципы создания и использования исключительных ситуаций.
- 4.5. Родовые компоненты и обобщенное программирование
- 4.5.1. Java Generics
- 4.5.2. Отличие от шаблонов C++
- 4.5.3. Ограничения на параметры
- 4.5.4. Совместимость на уровне байт-кода
- 4.6. Многопоточное программирование на Java. Параллелизм.
- 4.6.1. Модель потока в Java. Зависимость от реализации потока в операционной системе.
- 4.6.2. Свойства потока. Синхронизация. Передача сообщений.
- 4.6.3. Класс Thread и интерфейс Runnable.
- 4.6.4. Главный поток и способы создания потоков.
- 4.6.5. Управление потоками и приоритеты потоков.
- 4.6.6. Группы потоков.
- 4.6.7. Особенности написания многопоточных программ.
- 4.6.8. Использование синхронизирующих блоков и мониторов объектов. Синхронизированные методы объектов.
- 4.6.9. Взаимная блокировка.
- 4.6.10. Использование пула потоков.
- 5. Стандартная библиотека Java.**
- 5.1. Организация пакетов стандартной библиотеки Java. Пакеты java и javax.
- 5.2. Обработка строк.
- 5.2.1. Использование и методы класса String. Класс String – краеугольный камень производительности в Java приложениях.
- 5.2.2. Использование StringBuffer.
- 5.3. Пакет java.lang.
- 5.3.1. Структура и назначение.
- 5.3.2. Использование класса System. Управление средой исполнения.
- 5.3.3. Использование классов Number, Double, Integer, Character и др.
- 5.3.4. Класс Math.
- 5.3.5. Класс Compiler и класс ClassLoader.
- 5.4. Пакет java.util
- 5.4.1. Общие принципы организации контейнеров и коллекций в Java.
- 5.4.2. Использование множеств и списков.

- 5.4.3. Использование отображений и ассоциативных контейнеров.
- 5.4.4. Итераторы и исключительные ситуации при работе с классами утилит.
- 5.4.5. Класс Properties.
- 5.4.6. Классы Date и Calendar.
- 5.5. Подсистема ввода вывода java.io
  - 5.5.1. Общие концепции организации ввода – вывода в библиотеке Java.
  - 5.5.2. Проблема платформенной независимости и локализации.
  - 5.5.3. Основные классы потоков ввода-вывода в Java и методы работы с ними.
  - 5.5.4. Использование потоков ввода вывода при работе с файлами. Эффективность.
  - 5.5.5. Концепция Reader и Writer. Управление локализацией.
  - 5.5.6. Использование Tokenizer.
- 5.6. Пакет java.net.
  - 5.6.1. Основы работы с сетью в Internet. Адресация.
  - 5.6.2. Сокеты. Жизненный цикл сокета.
  - 5.6.3. Работа с протоколом HTTP и класс URL.
- 6. Графическая подсистема Java. JFC**
  - 6.1. Классы AWT.
    - 6.1.1. Основы работы с окнами. Component, Container, Panel, Window, Frame, Canvas.
    - 6.1.2. Доставка и обработка событий в графической подсистеме. Механизм Listeners.
    - 6.1.3. Создание программы с оконным интерфейсом. Рисование графических примитивов.
    - 6.1.4. Использование управляющих элементов, диспетчеров компоновки и меню.
    - 6.1.5. Связь классов AWT с оконным интерфейсом операционной системы.
  - 6.2. Классы Swing.
    - 6.2.1. Основные принципы графической системы Swing. Платформенная независимость, понятие Look&Feel
    - 6.2.2. Написание графического интерфейса с использованием Swing компонентов.
    - 6.2.3. Работа с таблицами, текстом, диалогами и HTML.
  - 6.3. Апплеты.
    - 6.3.1. Основы работы с апплетом.
    - 6.3.2. Жизненный цикл апплета.
    - 6.3.3. Дескриптор APPLET.
    - 6.3.4. Передача параметров и загрузка апплета.
- 7. Модель безопасности Java.**
  - 7.1. Принципы организации и эволюция модели безопасности в Java.
  - 7.2. SecurityManager. Инициализация и функции.
  - 7.3. Права доступа. Управление и проверка прав доступа.
  - 7.4. Исключительные ситуации.
  - 7.5. Java Cryptography Extension

7.6. Алгоритмы шифрования. Ключи и цифровые подписи.

## **8. Программирование распределенных приложений.**

8.1. Принципы построения распределенных приложений.

8.2. Проблемы передачи объектов и синхронизации в распределенных приложениях.

8.3. Реализация сохраняемости.

8.4. Three-tier технология.

8.4.1. Уровень интерфейса.

8.4.2. Уровень бизнес логики.

8.4.3. Уровень сохранения.

8.5. Remote Method Invocation

8.5.1. Основные принципы и протокол взаимодействия.

8.5.2. Интерфейс Remote и класс UnicastRemoteObject.

8.5.3. Класс Naming и rmiregistry сервис.

8.5.4. RMI сервер.

8.5.5. RMI клиент.

8.5.6. Модель безопасности, синхронизация и сборка мусора в распределенных RMI приложениях.

8.5.7. Механизм Activation.

8.6. Основы Java Enterprise технологии.

8.6.1. Интеграция с Web.

8.6.2. Java сервлеты.

8.6.3. Java Server Pages. Web контейнеры.

8.6.4. Механизм SessionBeans.

8.6.5. EJB технология. Bean контейнеры.

8.7. JDBC технология.

8.8. Java Micro Edition.

8.8.1. Java для мобильных телефонов. K-virtual machine.

8.8.2. CLDC, MIDP и iMode расширения Java 2 ME.

8.8.3. Технология JavaCard.

8.8.4. Примеры реальных приложений. Сложность разработки.

8.9. Java media framework, технологии обработки звука и голоса.

## **9. Технология коллективной разработки Java приложений.**

9.1. Использование документирующих комментариев.

9.2. Соглашения при написании кода и именовании классов и объектов.

9.3. Build система Ant. Автоматизация сборки и размещения Java приложений.

9.4. Система log4j.

## **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.

В процессе лекции студентам предлагаются вопросы для коллективного обсуждения и анализа, студенты имеют возможность активно задавать вопросы.

Лабораторные занятия проходят в терминальных классах, оснащенных персональными компьютерами с установленными средами разработки для Java.

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

Дополнительно преподаватели осуществляют прием и проверку заданий по электронной почте.

Для хранения исходного кода проектов выполняемых студентами используется директория расположенная на ftp сервере факультета.

#### **6. Оценочные средства для текущего контроля успеваемости, промежуточной**

#### **аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов**

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 практических заданий (баллово-рейтинговая система)

Итоговая оценка за работу в семестре выставляется по количеству баллов, набранных студентом.

В случае если студент сдает задание после контрольного срока или сдает задания с недочетами, за задачу ставится балл меньше максимального, но не менее 50% от балловой стоимости задачи. Задача, как правило, состоит из нескольких частей, за выполнение каждой из которой начисляются отдельные баллы. Таким образом студент может выполнить только часть практического задания (по желанию)

Итоговая оценка за курс выставляется на основе оценок полученных за работу в семестрах и устного ответа на вопросы билета на экзамене.

#### **Перечень тем для самостоятельной работы**

1. Модель памяти и структура программы. Классы памяти. Ссылки.
2. Средства абстракции C++. Структура класса. Статические члены.
3. Средства инкапсуляции C++. Инкапсуляция и наследование. Друзья.
4. Модульность, отдельная компиляция, пространства имен, using директива.
5. Представление иерархических отношений. Наследование.
6. Представление иерархических отношений. Агрегация. Зависимость по

времени жизни.

7. Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм.

8. C++: средства реализации состояния объектов; реализация поведения.

9. Перегрузка операторов.

10. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании.

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

12. Исключения в C++. Обработка исключений.

13. Шаблоны классов и шаблоны функций. Специализация.

14. Основы STL. Структура и назначение. Контейнеры. Алгоритмы

15. Стандартная библиотека, ввод-вывод

Отчет принимается в виде реферата с презентацией.

№ п/п	Семестр	Виды и содержание контрольных мероприятий
1	2	3
Модуль 1	7	Защита рефератов по темам № 1,2,3,4,5
Модуль 2	7	Защита презентаций по темам № 6,7,8,9,10
Модуль 3	7	Защита рефератов по темам № 11,12,13,14,15
Модуль 4	7	Экзамен

Форма контроля и критерий оценок

В соответствии с учебным планом предусмотрен экзамен в 7 семестр.

Формы контроля: текущий контроль, промежуточный контроль по модулю, итоговый контроль.

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

**7.1. Перечень компетенций с указанием этапов их формирования в**

**процессе освоения образовательной программы.**

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

Компетенция	Знания, умения, навыки	Процедура освоения
ПК-33	способностью составлять инструкции по эксплуатации информационных систем	Выполнение практических работ

**7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания.**

ПК-33

Схема оценки уровня формирования компетенции «способность составлять инструкции по эксплуатации информационных систем»

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Положительный	способность составлять инструкции по эксплуатации информационных систем.	<b>Знает</b> основы технологии объектно-ориентированной декомпозиции программных систем, базовых шаблонов проектирования (Наблюдатель, Итератор, Одиночка, Фабрика, Заместитель), отношений между классами и основ UML (диаграммы классов и последовательностей).	<b>Умеет</b> пользоваться основными инструментальными средствами языка C++ и стандартной библиотеки, способен пользоваться средствами реализации принципов ООП и инструментальными средствами языка Java. Знает платформы Java, особенности построения программных систем Java.	<b>Владеет</b> Основами технологий построения простейших распределенных информационных систем и обеспечения безопасности.

### 7.3. Типовые контрольные задания

#### Перечень контрольных вопросов для подготовки к итоговой аттестации по дисциплине

1. Эволюция методологий программирования. Парадигмы программирования.
2. Основные принципы объектного подхода. Инкапсуляция.
3. Основные принципы объектного подхода. Абстрагирование.
4. Основные принципы объектного подхода. Модульность.
5. Основные принципы объектного подхода. Иерархия.
6. Основные принципы объектного подхода. Типизация.
7. Основные принципы объектного подхода. Параллелизм. Сохраняемость.
8. Объект с точки зрения ООП. Состояние. Поведение.
9. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов.
10. Объект с точки зрения ООП. Взаимоотношения между объектами.4
11. Классы. Природа классов. Мета модель. Инстанцирование.
12. Классы. Структура класса. Абстрактные классы и интерфейсы.
13. Классы. Отношения между классами. Ассоциация и агрегация.
14. Классы. Иерархии классов. Зависимость.
15. Виртуальная машина. Структура программ. Типы переменных в Java.
16. Основные операторы Java
17. Простые типы данных, литералы. Приведение типов в Java.
18. Управляющие инструкции Java.
19. Массивы в Java.
20. Методы в Java.
21. Наследование и переопределение методов.
22. Методы в Java.
23. Пакеты в Java.
24. Интерфейсы в Java.
25. Обработка исключительных ситуаций.
26. Система ввода/вывода.
27. Создание программ с графическим интерфейсом.

#### Перечень тем рефератов

Общая теория ООП.

1. Эволюция методологий программирования. Парадигмы программирования.
2. Основные принципы объектного подхода. Абстрагирование.
3. Основные принципы объектного подхода. Инкапсуляция.
4. Основные принципы объектного подхода. Модульность.
5. Основные принципы объектного подхода. Иерархия.
6. Основные принципы объектного подхода. Типизация.



7. Основные принципы объектного подхода. Параллелизм. Сохраняемость.
8. Объект с точки зрения ООП. Состояние. Поведение.
9. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов.
10. Объект с точки зрения ООП. Взаимоотношения между объектами.
11. Классы. Природа классов. Мета модель. Инстанцирование.
12. Классы. Структура класса. Абстрактные классы и интерфейсы.
13. Классы. Отношения между классами. Ассоциация и агрегация.
14. Классы. Иерархии классов. Зависимость.

□ Средства Java:

1. Виртуальная машина. Структура программ. Типы переменных в Java. Принципы работы ClassLoader.
2. Средства абстракции Java. Структура класса. Статические члены.
3. Внутренние и вложенные классы. Статический и динамический контекст. Локальные и анонимные классы. Перечисления
4. Средства инкапсуляции Java. Поддержка модульности. Пакеты.
5. Представление иерархических отношений. Наследование. Интерфейсы и абстрактные классы.
6. Родовые компоненты (Generics)
7. Агрегация и зависимость от времени жизни. Реализация отношений клиент-сервер. Стандартные контейнеры.
8. Типизация. Правила преобразования типов. instanceof и ClassCastException. Класс Class.
9. Средства поддержки параллелизма. Активные и пассивные объекты. Класс Object.
10. Использование Thread и Runnable. Пул потоков, назначение и принципы реализации.
11. Исключения. Обработка исключительных ситуаций.
12. Сохраняемость. Serializable и Externalizable. Программирование распределенных приложений.
13. Модель безопасности Java. Policy, Permissions, AccessController.
14. Графическая подсистема. Основы AWT, Applet, Swing components. Событийная модель.
15. Средства поддержки Java машины. System, Runtime, сборка мусора.

#### **7.4. 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 - 3 баллов,
- выполнение лабораторных заданий – 10 баллов,
- выполнение домашних (аудиторных) контрольных работ - 5 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 3 баллов,
- письменная контрольная работа - 5 баллов,
- тестирование - 10 баллов.

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

а) основная литература:

1. Мухортов В.В., Рылов В.Ю. Объектно-ориентированное программирование, анализ и дизайн. Методическое пособие. ИМ СО РАН, 2002 г.
2. Г. Буч Объектно ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд./Пер. с англ. — СПб.; М.: «Невский Диалект» — «Издательство БИНОМ», 1999 г.
3. Б. Страуструп Язык программирования С++, 3-е изд./Пер. с англ. — СПб.; М.: «Невский Диалект» — «Издательство БИНОМ», 1999г.
4. Б. Страуструп Дизайн и эволюция языка С++, Пер. с англ. - ДМК Пресс, Питер, 2006г.
5. Скотт Мейерс, Эффективное использование С++. 50 рекомендаций по улучшению ваших программ и проектов, Пер. с англ. - ДМК, 2006
6. Скотт Мейерс, Эффективное использование С++. 35 новых способов улучшить стиль программирования, Пер. с англ. - ДМК, 2006

б) дополнительная литература:

1. Брюс, Эккель, Философия Java, Пер. с англ. - Питер, 2003, 2009
2. Герберт Шилдт, Полный справочник по Java, Java SE 6th edition, 7-е издание, Пер. с англ. - Вильямс, 2007
3. OMG Unified Modeling Language Specification version 1.3, Object Management Group, 1999, <http://www.omg.org>
4. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. — М. ДМК, 2000. 25
5. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования, СПб.: Питер, 2001

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

1. <http://www.omg.org> – Object Management Group (Теория объектного подхода, UML)
2. <http://cplusplus.com> – Информация по языку и стандартной библиотеке С++

3. <http://msdm.microsoft.com> – Microsoft Developer Network
4. <http://www.oracle.com/technetwork/java/index.html> - Технология Java

#### **10. Методические указания для обучающихся по освоению дисциплины.**

Для более полного понимания целей, задач и практических результатов ООП следует:

- 1) Ознакомиться с дополнительной литературой, особенно с трудами основоположников.
- 2) Ознакомиться, хотя бы поверхностно, с другими подходами языками ООП.
- 3) Попытаться в рамках лабораторных занятий самостоятельно и полностью выполнить все задания.

Также для самостоятельной работы рекомендуется использовать перечень тем домашних работ и список основной литературы настоящей рабочей программы.

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

1. Visual Studio 2010 Express – бесплатная среда разработки для C++ под Windows
2. Microsoft Visual Studio 2010 – коммерческая среда разработки для C++ под Windows
3. GNU C++, GNU Make, Eclipse IDE for C++ Developers - свободные средства разработки C++ под Unix/Linux
4. Java SE 6 Development Kit (JDK) – свободно распространяемые средства разработки Java
5. Eclipse IDE for Java Developers, NetBeans IDE – свободные кросс-платформенные среды разработки для Java

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

Для проведения лекций необходима аудитория оснащенная экраном и проектором, подключенным к персональному компьютеру/ноутбуку оснащенным средством просмотра демонстраций в формате Microsoft Power Point.

Для проведения лабораторных занятий по Java необходим терминальный класс, оснащенный персональными компьютерами под управлением любой операционной системы, на которую может быть установлен Java SE 6 Development Kit (Microsoft Windows, Linux, Open Solaris, Mac OS X).