МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Государственное образовательное учреждение высшего профессионального образования
«Новосибирский государственный университет» (НГУ)

Факультет информационных технологий

УТВЕРЖДАЮ

_____

« ___» _____ 20___г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Объектно-ориентированное программирование

(наименование дисциплины)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 230100 «ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА»

Квалификация (степень) выпускника
Бакалавр

Форма обучения очная

Новосибирск
2011

Программа дисциплины «Объектно-ориентированное программирование» составлена в соответствии с требованиями ФГОС ВПО к структуре и результатам освоения основных образовательных программ бакалавриата по Профессиональному циклу по направлению подготовки «Информатика и вычислительная техника», а также задачами, стоящими перед Новосибирским государственным университетом по реализации Программы развития НГУ.

Автор: Рылов Всеволод Юрьевич

Факультет Информационных технологий
Кафедра Общей Информатики

1. Цели освоения дисциплины (курса)

Дисциплина (курс) Объектно-ориентированное программирование имеет своей целью:

- Изучение основ классической теории объектно-ориентированного программирования, в том числе:
    - Пути эволюции технологий программирования от алгоритмического к ООП
    - Основных принципов объектно-ориентированного построения программных систем (Абстракция, Инкапсуляция, Иерархия, Модульность, Типизация, Параллелизм, Сохраняемость)
    - Понятий классов, объектов, взаимоотношений между ними, а также многоуровневой модели OMG
- Изучение средств объектно-ориентированного и обобщенного программирования языка C++, средств стандартной библиотеки STL
- 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

Курс относится к базовой (общепрофессиональной) части профессионального цикла (Б.3.)

Изучение дисциплины опирается на курсы Б.2.Б.2.«Алгебра и геометрия», Б.2.Б.4.«Информатика», Б.3.Б.1.«Программирование», Б.2.В.1.«Математическая логика и теория алгоритмов».

Предварительными требованиями к студентам являются:

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

- Знание основ теории множеств

Данный курс является предшествующим (базовым) для следующих дисциплин/курсов: Б.3.Б.7.«Базы данных», Б.3.Б.9.«Инженерная и компьютерная графика», Б.3.В.4.«Объектно-ориентированный анализ и дизайн»

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

Курс направлен на выработку следующих компетенций:
ОК-1, ОК-6, ОК-8, ОК-12, ОК-18, ОК-22, ПК-2, ПК-3, ПК-4, ПК-5, ПК-6, ПК-12, ПК-13, ПК-25, ПК-30, ПК-32, ПК-38, ПК-50, ПК-53, ПК-85, ИК-6, ИК-7

По окончании курса студенты получают следующие знания и навыки:

- Знание основ технологии объектно-ориентированной декомпозиции программных систем, базовых шаблонов проектирования (Наблюдатель, Итератор, Одиночка, Фабрика, Заместитель), отношений между классами и основ UML (диаграммы классов и последовательностей).
- Знание особенностей построения объектно-ориентированных программных систем на C++.
- Основные инструментальные средства языка C++ и стандартной библиотеки
- Базовые знания платформы Java, особенности построения программных систем Java
- Средства реализации принципов ООП и инструментальные средства языка Java.
- Основы технологий построения простейших распределенных информационных систем и обеспечения безопасности.

4. Структура и содержание дисциплины

В отличие от многих курсов по языкам C++ и Java, данный курс делает акцент не на синтаксисе самих языков и особенностей применения тех или иных конструкций, а на самой технологии объектно-ориентированного подхода и средствах поддержки именно принципов ООП в инструментальных языках. Изложение теории

ООП ведется по схеме: Эволюция и принципы – Объект – Класс – Модуль – Система. Изложение средств языков также подчиняются данной схеме, а именно:

- Для C++: Отличия от C – Объекты (классы, типизация) - Средства управления жизненным циклом объектов – Реализация отношений между объектами средствами классов – Средства построения иерархий классов – Средства построения программных модулей – Обобщенное программирование – Библиотека
- Для Java: Платформа и структура машины – Объекты – Управление жизненным циклом – Реализация отношений между объектами (классами) – Иерархии классов – Модули – Библиотека – Параллельное программирование – Безопасность – Распределенное программирование.

Общая трудоемкость дисциплины составляет 7 зачетных единиц, 252 часа (в том числе аудиторных – 128 часов).

Вид учебной работы	*Всего часов*	*Семестры*	
		3	*4*
Общая трудоемкость дисциплины	252	132	120
Аудиторные занятия, в том числе:	128	68	60
Лекции	64	34	30
Семинары	-	-	-
Лабораторные работы	64	34	30
Самостоятельная работа, в том числе:	124	64	60
Практические задания	64	34	30
Курсовой проект	48	24	24
Реферат			
Расчетные работы			
Другие виды самостоятельной работы			
Письменный коллоквиум	12	6	6
Вид промежуточного контроля		Диф.зачет	Экзамен

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости *(по неделям семестра)* Форма промежуточной аттестации *(по семестрам)*
				Лекции	Семинары	Лаб. работы	Сам. работа	
1	Основные принципы объектно-ориентированного программирования	1	1	2		2	4	
2	Объектно-ориентированная модель	1	2	2		2	4	
3	Классы	1	3	2		2	4	
4	Основные алгоритмические отличия C++ от C	1	4	2		2	4	
5	Средства объектного программирования языка C++	1	5-8	8		8	22	Коллоквиум, контрольная неделя
6	Средства объектно-ориентированного программирования C++	1	9-11	6		6	12	
7	Обобщенное программирование	1	12-13	4		4	8	Контрольная неделя
8	Стандартная библиотека C++	1	14-17	8		8	16	Диф. зачет

ИТОГО:				34		34	74	
9	Виртуальная машина Java	2	1	2		2	4	
10	Алгоритмические средства языка Java	2	2	2		2	4	
11	Средства объектно-ориентированного программирования языка Java	2	3-6	8		8	16	Контрольная неделя
12	Модульность и обобщенное программирование на Java	2	7-8	6		6	12	
13	Стандартная библиотека Java	2	9-12	6		6	12	Коллоквиум, контрольная неделя
14	Модель безопасности Java	2	13	2		2	4	
15	Программирование распределенных приложений	2	14	2		2	4	
16	Технология коллективной разработки Java приложений	2	15	2		2	4	Зачет, экзамен
ИТОГО:				30		30	60	

Детальная программа курса:

Семестр 1 (Объектная модель и инструментальные средства языка C++)

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++ от

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, раздельная 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.

5. Средства объектного программирования языка C++

5.1. Представление объектов и классов.

5.1.1. Реализация поведения объектов на примере добавления функций—членов в структуры. Структура как вырожденный класс.

5.1.2. Структура объявления класса.

5.1.3. Доступ к членам класса.

5.1.4. Поля данных класса как механизм реализации состояния объекта.

5.1.5. Функции члены класса как механизм реализации поведения объекта.

5.1.6. Спецификаторы доступа для обеспечения инкапсуляции.

5.1.7. Средства управления жизнью объекта. Конструкторы и деструкторы.

5.1.8. Конструирование и уничтожение объектов и массивов объектов.

5.1.9. Особенности использования конструктора копии, конструктора по умолчанию, оператора присваивания.

5.1.10. Описание селекторов и модификаторов.

5.1.11. Перегрузка операторов C++ как реализация поведения с предопределенным смыслом.

5.1.12. Дружественность как механизм нарушения инкапсуляции. Достоинства и недостатки механизма дружественности.

5.1.13. Статические поля и методы классов. Инициализация статических полей.

5.2. Реализация отношений между объектами и классами

5.2.1. Ассоциация и агрегация объектов и классов. Зависимость по времени жизни.

5.2.2. Использование и зависимость от интерфейсов.

5.2.3. Объекты при передаче параметров и возврате из методов.

5.2.4. Варианты реализации отношения клиент-сервер.

5.2.5. Внутренние классы.

6. Средства объектно-ориентированного программирования C++

6.1. Наследование как средство организации иерархий классов. Принцип замещения Лисковой.

6.2. Одиночное наследование.

6.2.1. Понятие производного класса.

6.2.2. Управление доступом в производных классах.

6.2.3. Конструкторы и деструкторы, совмещение имен методов при наследовании, иерархии.

6.2.4. Абстрактные классы и виртуальные функции.

6.2.5. Виртуальный полиморфизм.

6.2.6. Информация о типе на этапе выполнения. RTTI.

6.3. Множественное наследование

6.3.1. Проблема множественного наследования. Виртуальное наследование как средство разрешения коллизий.

6.3.2. Порядок вызовов конструкторов и деструкторов при множественном наследовании.

6.3.3. Чистые виртуальные классы, понятие интерфейса.

6.3.4. Принципы дизайна иерархий классов. OCP, DIP, ISP.

6.4. Пространства имен.

6.4.1. Пространства имен как средство реализации модульности.

6.4.2. Поиск имен и разрешение конфликтов.

6.4.3. Объединение пространств имен.

6.4.4. Принципы дизайна пакетов.

6.5. Обработка исключений.

6.5.1. Обработка ошибок.

6.5.2. Группировка и перехват исключений.

6.5.3. Управление ресурсами.

6.5.4. Исключения и эффективность.

6.5.5. Альтернативные методы обработки ошибок.

6.5.6. Стандартные исключения.

7. Обобщенное программирование.

7.1. Шаблоны классов.

7.1.1. Определение шаблона.

7.1.2. Инстанцирование.

7.1.3. Параметры шаблонов и проверка типов.

7.2. Шаблоны функций.

7.3. Специализация.

7.4. Наследование и шаблоны.

8. Стандартная библиотека C++

8.1. Библиотека стандартных шаблонов

8.1.1. Общие сведения (понятия контейнеров, итераторов и объектов-функций)

8.1.2. Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры)

8.1.3. Итераторы (итератор как обобщение указателя, классы итераторов)

8.1.4. Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера)

8.2. Библиотека ввода-вывода

8.2.1. Потоки вывода. Вывод типов определяемых пользователем.

8.2.2. Потоки ввода. Ввод типов определяемых пользователем.

8.2.3. Форматирование в потоках ввода-вывода.

8.2.4. Буферизация.

Семестр 2 (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 приложение 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 Throwble 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 C++ и 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 система контроля версий SVN установленная на 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)
- Выполнение и сдача курсовой работы (задания)

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)

На письменном коллоквиуме студент может набрать от 0 до 21 баллов.

Темы практических заданий и их балловая стоимость:

№ задания	Тема задания	Контрольный срок	Максимальный балл
	1 семестр - C++		
1	Раздельная компиляция и пространства имен	3 неделя	10 баллов
2	Перегрузка функций, указатели на функции, перечисления	5 неделя	10 баллов
3	Классы. Реализация упрощенного григорианского календаря	8 неделя	10+5 = 15 баллов
4	Классы. Перегрузка операций. Реализация матрицы и вектора	10 неделя	8+7 = 15 баллов
5	Иерархии классов, наследование. Реализация командного процессора	13 неделя	15+5 = 20 баллов
6	Шаблоны. Реализация шаблонов вектора и матрицы, реализация шаблона «умного» указателя	15 неделя	10 + 10 = 20 баллов
7	Курсовая работа: Реализация эмулятора АЛУ	17 неделя	30 баллов
	2 семестр - Java		
1	Контейнеры, потоки. Object, String. Реализация программы подсчета частоты встречаемости слов в файле	3 неделя	10 баллов
2	Шаблон проектирования «фабричный метод», журналирование, модульное тестирование. Реализация стекового калькулятора	7 неделя	15+10 = 25 баллов
3	Шаблон проектирования MVC. Графический интерфейс пользователя. Реализация игры сапер или тетрис (по выбору)	11 неделя	25 баллов

4	Многопоточность и параллелизм. Реализация фабрики-конвейера по сбору «изделий» из «деталей»	13 неделя	30 баллов
5	Курсовая: Распределенное программирование. Сетевое взаимодействие. Реализация многопользовательского чата	15 неделя	20+10 баллов

Распределение оценок (максимально за семестр с учетом коллоквиума можно набрать 141 балл):

- 65+ баллов – оценка «удовлетворительно»
- 85+ баллов – оценка «хорошо»
- 105+ баллов – оценка «отлично»

В случае наличия большого количества пропусков (более 3-х) лекций, студент должен подготовить развернутый ответ на один из вопросов при сдаче дифференцированного зачета (в первом семестре) или зачета (во втором семестре).

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

Примерный перечень вопросов к зачету (экзамену) по всему курсу.

- 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. Метамодель. Инстанцирование.
12. Классы. Структура класса. Абстрактные классы и интерфейсы.
13. Классы. Отношения между классами. Ассоциация и агрегация.
14. Классы. Иерархии классов. Зависимость.

- Средства C++:

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, раздельная 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

- 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. Учебно-методическое и информационное обеспечение дисциплины

Основным учебно-образовательным ресурсом курса является WWW сайт http://sites.google.com/site/nguoop

На данном сайте представлены:

- Правила учета успеваемости
- Посещаемость лекций в текущем учебном году
- Демонстрационные презентации лекций курса в формате Microsoft Power Point
- Демонстрационные примеры программ, представленные на лекциях

- Условия практических заданий и курсовых работ для текущего учебного года
- Список основной и дополнительной литературы
- Список вопросов для самоподготовки к экзамену

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 C++, 2-е изд./Пер. с англ. — СПб.; М.: «Невский Диалект» — «Издательство БИНОМ», 1999 г.
3. Б. Страуструп Язык программирования C++, 3-е изд./Пер. с англ. — СПб.; М.: «Невский Диалект» — «Издательство БИНОМ», 1999г.
4. Б. Страуструп Дизайн и эволюция языка C++, Пер. с англ. - ДМК Пресс, Питер, 2006г.
5. Скотт Мейерс, Эффективное использование C++. 50 рекомендаций по улучшению ваших программ и проектов, Пер. с англ. - ДМК, 2006
6. Скотт Мейерс, Эффективное использование C++. 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 6^{th} 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.

5. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования, СпБ.: Питер, 2001

в) программное обеспечение и Интернет-ресурсы:

1. http://www.omg.org – Object Management Group (Теория объектного подхода, UML)
2. http://cplusplus.com – Информация по языку и стандартной библиотеке C++
3. http://msdm.microsoft.com – Microsoft Developer Network
4. http://www.oracle.com/technetwork/java/index.html - Технология Java
5. Visual Studio 2010 Express – бесплатная среда разработки для C++ под Windows
6. Microsoft Visual Studio 2010 – коммерческая среда разработки для C++ под Windows
7. GNU C++, GNU Make, Eclipse IDE for C++ Developers - свободные средства разработки C++ под Unix/Linux
8. Java SE 6 Development Kit (JDK) – свободно распространяемые средства разработки Java
9. Eclipse IDE for Java Developers, NetBeans IDE – свободные кросс-платформенные среды разработки для Java

8. Материально-техническое обеспечение дисциплины

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 C++ (1 семестр) необходим терминальный класс, оснащенный персональными компьютерами под управлением операционных систем Microsoft Windows (в случае использования Visual Studio Express), либо Unix-подобными ОС (Open Solaris, либо Linux) в случае использования GNU C++.

Для проведения лабораторных занятий по Java (2 семестр) 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)

Рецензент (ы) _____

Программа одобрена на заседании

_____

(Наименование уполномоченного органа вуза (УМК, НМС, Ученый совет)

от _____ года.