

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кубанский государственный университет»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ:

Проректор по учебной работе,
качеству образования – первый
проректор

Иванов А.Г.



2015 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.Б.26 ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ РЕШЕНИЕ ЗАДАЧ

Направление подготовки/специальность 02.03.02 «Фундаментальная информатика и информационные технологии»

Направленность (профиль) / специализация «Вычислительные технологии»

Программа подготовки академическая

Форма обучения очная

Квалификация (степень) выпускника бакалавр

Краснодар 2015

Рабочая программа дисциплины Б1.Б.26 "Объектно-ориентированное решение задач" составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии»

Программу составил(и):

А.И.Миков, заведующий кафедрой вычислительных технологий, д.ф.-м.н., профессор



подпись

Ермоленко С.С. преподаватель кафедры ВТ ФКТ и ПМ



Рабочая программа дисциплины «Объектно-ориентированное решение задач» утверждена на заседании кафедры вычислительных технологий, протокол № 9 «20» апреля 2015 г.

Заведующий кафедрой (разработчика) Миков А.И. _____

фамилия, инициалы



подпись

Рабочая программа обсуждена на заседании кафедры вычислительных технологий протокол № 9 «20» апреля 2015 г.

Заведующий кафедрой (выпускающей) Миков А.И. _____

фамилия, инициалы



подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 5 « 29» апреля 2015 г.

Председатель УМК факультета



К.В. Малыхин

Рецензенты:

Гаркуша О.В., доцент кафедры информационных технологий
ФБГОУ ВО «Кубанский государственный университет»,
кандидат физико-математических наук.

Зайков В.П. Ректор НЧОУ ВО «Кубанский институт информзащиты»
д.экон. наук, к.т.н., доцент

1 Цели и задачи изучения дисциплины.

1.1 Цель освоения дисциплины.

Целью преподавания и изучения дисциплины «Объектно-ориентированное решение задач» является формирование у бакалавров знаний и умений в области разработки программ на основе объектов, понимания концепций объектно-ориентированного программирования (ООП).

1.2 Задачи дисциплины.

Основные задачи освоения дисциплины:

изучить основные принципы объектно-ориентированного программирования;

- изучить реализацию этих принципов на языках C++ и Java;
- научиться писать программы на языках C++ и Java;
- научиться проектировать и разрабатывать объектно-ориентированные программы.

1.3 Место дисциплины в структуре образовательной программы.

Дисциплина «Объектно-ориентированное решение задач» относится к базовой части блока Б1 учебного плана. Для изучения дисциплины необходимо знание дисциплин «Дискретная математика», «Основы программирования». Знания, получаемые при изучении объектно-ориентированного программирования, используются при изучении таких дисциплин учебного плана бакалавра как «Компьютерные сети», «Распределенные задачи и алгоритмы», «Алгоритмы цифровой обработки изображений», «Оценка сложности алгоритмов», «Гиперграфовые модели».

1.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих **профессиональных компетенций**:

№ п.п.	Индекс компетенции	Содержание компетенции (или ее части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
1.	ОК-6	способностью работать в команде, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия	Правила работы в команде при решении задач проектирования программных проектов	Распределять функции разработчиков в составе командного проекта	способностью работать в команде, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия
	ПК-2	способностью понимать, совершенствовать и применять современный	четыре принципа объектно-ориентированного программирования; основные формы	создавать классы на C++ и Java и их использовать; осуществлять перегрузку	навыками программирования на C++ и Java; объектно-

		математический аппарат, фундаментальные концепции и системные методологии, международные и профессиональные стандарты в области информационных технологий	наследования; способы реализации полиморфизма в языках C++ и Java; преимущества и недостатки наследования и композиции. способы реализации множественного наследования в C++ и Java.	операторов в C++; создавать иерархию классов на C++ и Java; использовать полиморфизм; проектировать с учетом множественного наследования; создавать шаблоны функции и классов; использовать механизм обработки исключений; использовать библиотеку потоков; использовать стандартную библиотеку шаблонов STL.	ориентированно го проектирования и анализа; составления отчета с описанием логической и физической модели системы с точки зрения объектно-ориентированно го проектирования .
--	--	---	--	---	--

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 3 зач.ед. (108 часов), их распределение по видам работ представлено в таблице
(для студентов ОФО).

Вид учебной работы	Всего часов	Семестры (часы)			
		5			
Контактная работа, в том числе:					
Аудиторные занятия (всего):	76,2	76,2			
Занятия лекционного типа	38	38			
Лабораторные занятия	38	38			
Занятия семинарского типа (семинары, практические занятия)	—	—			
	—	—			
Иная контактная работа:					
Контроль самостоятельной работы (КСР)					
Промежуточная аттестация (ИКР)	0,2	0,2			
Самостоятельная работа, в том числе:	31,8	31,8			
Курсовая работа	—	—			
Проработка учебного (теоретического) материала	20	20			
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)	9	9			
Реферат	—	—			

Подготовка к текущему контролю		2,8	2,8			
Контроль:		зачет	зачет			
Подготовка к экзамену		–	–			
Общая трудоёмкость	час.	108	108			
	в том числе контактная работа	76,2	76,2			
	зач. ед.	3	3			

2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы дисциплины, изучаемые в 3 семестре (очная форма)

№ раздела	Наименование разделов	Количество часов				
		Все го	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
1	Основные понятия объектно-ориентированного анализа. Язык C++. Язык Java.	18	8	–	6	4
2	Классы и объекты.	20	8	–	4	8
3	Объектно-ориентированная методология программирования. Перегрузка операций.	18	6	–	8	4
4	Наследование, полиморфизм.	15	4	–	4	7
5	Виртуальные и дружественные функции. Исключения. Стандартная библиотека шаблонов.	16	6	–	6	4
6	Потоки и файлы.	18	6	–	8	4
7	Обзор изученного материала и приём зачёта	2,8		–	2	0,8
8	ИКР	0,2				
Итого по дисциплине:		108	38	–	38	31,8

Примечание: Л – лекции, ПЗ – практические занятия / семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов дисциплины:

2.3.1 Занятия лекционного типа.

№	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
1	Основные понятия объектно-ориентированного анализа. Язык C++. Язык Java.	Сущность объектно-ориентированного подхода в программировании. Цикл разработки программного обеспечения (ПО), назначение и содержание этапов. Роль анализа в процессе разработки программного обеспечения. Основные понятия объектно-ориентированного	ЛР

		анализа. Язык C++. Язык Java.	
2	Классы и объекты.	Классы и объекты. Отношения, основные типы отношений. Язык UML. Основные средства анализа и моделирования предметной области в языке UML. Статические данные. Конструктор, деструктор. Операции new и delete.	ЛР
3	Объектно-ориентированная методология программирования. Перегрузка операций.	Объектно-ориентированная методология программирования. Технология применения объектно-ориентированных языков, их классификация и архитектура. Перегрузка операций. Преобразование типов.	ЛР
4	Наследование, полиморфизм.	Наследование, базовый и производный классы. Простое и сложное наследование. Абстракция данных, наследование и полиморфизм.	ЛР
5	Виртуальные и дружественные функции. Исключения. Стандартная библиотека шаблонов.	Виртуальные функции. Дружественные функции. Дружественные классы. Шаблоны функций. Шаблоны классов. Исключения. Стандартная библиотека шаблонов.	ЛР
6	Потоки и файлы.	Потоки и файлы. Стандартная библиотека классов для управления потоками. Методы и средства организации и программирования интерфейса.	ЛР

Содержание дисциплины

1. Тема 1 Сущность объектно-ориентированного подхода в программировании. Цикл разработки программного обеспечения (ПО), назначение и содержание этапов. Роль анализа в процессе разработки программного обеспечения. Основные понятия объектно-ориентированного анализа. Язык C++. Язык Java.

Формы и методы проведения занятий по теме: лекция.

Форма текущего контроля: текущий опрос.

Виды самостоятельной подготовки студентов по теме: подготовка к текущему и промежуточному тестированию.

2. Тема 2 Классы и объекты. Отношения, основные типы отношений. Язык UML. Основные средства анализа и моделирования предметной области в языке UML. Статические данные. Конструктор, деструктор. Операции new и delete.

Формы и методы проведения занятий по теме: лекция, лабораторная работа.

Форма текущего контроля: текущий опрос, отчет о выполнении лабораторной работы.

Виды самостоятельной подготовки студентов по теме: подготовка к текущему и промежуточному тестированию, лабораторным работам.

3. Тема 3 Объектно-ориентированная методология программирования. Технология применения объектно-ориентированных языков, их классификация и архитектура. Перегрузка операций. Преобразование типов.

Формы и методы проведения занятий по теме: лекция, лабораторная работа.

Форма текущего контроля: текущий опрос, отчет о выполнении лабораторной работы.

Виды самостоятельной подготовки студентов по теме: подготовка к текущему и промежуточному тестированию, лабораторным работам.

4. Тема 4 Наследование, базовый и производный классы. Простое и сложное наследование. Абстракция данных, наследование и полиморфизм..

Формы и методы проведения занятий по теме: лекция, лабораторная работа.

Форма текущего контроля: текущий опрос, отчет о выполнении лабораторной работы.

Виды самостоятельной подготовки студентов по теме: подготовка к текущему и промежуточному тестированию, лабораторным работам.

5. Тема 5 Виртуальные функции. Дружественные функции. Дружественные классы. Шаблоны функций. Шаблоны классов. Исключения. Стандартная библиотека шаблонов.

Формы и методы проведения занятий по теме: лекция, лабораторная работа.

Форма текущего контроля: текущий опрос, отчет о выполнении лабораторной работы.

Виды самостоятельной подготовки студентов по теме: подготовка к текущему и промежуточному тестированию, лабораторным работам.

6. Тема 6 Потоки и файлы. Стандартная библиотека классов для управления потоками. Методы и средства организации и программирования интерфейса.

Формы и методы проведения занятий по теме: лекция.

Форма текущего контроля: текущий опрос.

Виды самостоятельной подготовки студентов по теме: подготовка к текущему и промежуточному тестированию.

2.3.2 Занятия семинарского типа.

Занятия семинарского типа не предусмотрены.

2.3.3 Лабораторные занятия.

№	Наименование лабораторных работ	Форма текущего контроля
1	2	3
1.	Классы и объекты в C++	Отчет по лабораторной работе
2.	Наследование и виртуальные функции	Отчет по лабораторной работе
3.	Иерархия объектов и группа. Итераторы	Отчет по лабораторной работе
4.	Обработка событий	Отчет по лабораторной работе
5.	Перегрузка операций	Отчет по лабораторной работе
6.	Шаблоны функций и классов	Отчет по лабораторной работе
7.	Потоковые классы	Отчет по лабораторной работе
8.	Стандартная библиотека шаблонов	Отчет по лабораторной работе
9.	Динамические списки	Решение задач, отчет по ЛР
10.	Деревья двоичного поиска	Решение задач, отчет по ЛР
11.	Неориентированные графы	Решение задач, отчет по ЛР
12.	Ориентированные графы	Решение задач, отчет по ЛР

2.3.4 Примерная тематика курсовых работ (проектов)

Курсовые работы не предусмотрены.

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Решение задач	Стандарты оформления исходного кода программ и современные интегрированные среды разработки программного обеспечения: учеб.-метод.пособие/ Ю.В.Кольцов [и др.]. – Краснодар:Кубанский гос.ун-т, 2015.-111с., утвержденные кафедрой информационных технологий, протокол № 7 от 09.04.2015 г.
2	Отчет по лабораторной работе	Стандарты оформления исходного кода программ и современные интегрированные среды разработки программного обеспечения: учеб.-метод.пособие/ Ю.В.Кольцов [и др.]. – Краснодар:Кубанский гос.ун-т, 2015.-111с., утвержденные кафедрой информационных технологий, протокол № 7 от 09.04.2015 г.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии.

Применяются следующие образовательные технологии.

Проблемные лекции «Рекурсивные алгоритмы в ООП», «Моделирование стека, списка, очереди с помощью ООП».

На лабораторных занятиях используется метод малых групп, разбор практических задач и кейсов, технология фасетного построения учебных задач.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации.

4.1 Фонд оценочных средств для проведения текущего контроля.

Варианты типовой контрольной работы (длительность написания 45 мин).

Вариант1. Разработайте UML – диаграмму и программу:

Написать иерархию классов, описывающих имущество налогоплательщиков. Она должна состоять из абстрактного базового класса **Property** и производных от него классов **Appartment**, **Car** и **Country House**. Базовый класс должен иметь поле **worth** (стоимость), конструктор с одним параметром, заполняющий это поле, и чисто виртуальный метод расчета налога, переопределенный в каждом из производных классов. Налог на квартиру вычисляется как 1/1000 ее стоимости, на машину — 1/200, на дачу — 1/500. Также, каждый производный класс должен иметь конструктор с одним параметром, передающий свой параметр конструктору базового класса. В методе **main** создать массив из 7 указателей на **Property** и заполнить его объектами производных классов (первые 3 — **Appartment**, следующие 2 — **Car** и последние 2 — **Country House**). Вывести на экран величину налога для всех 7 объектов. **В массив вносить только уникальные объекты.** Выполнить переопределение **Equals** и **ToString** в каждом классе.

Вариант 2: Разработайте UML – диаграмму и программу:

Написать иерархию классов "научный сотрудник", описывающих ставку и принцип вычисления надбавок. Она должна состоять из абстрактного базового класса **Scientist** и производных от него классов "Старший научный сотрудник", "Младший научный сотрудник" и "ИТР". Базовый класс должен иметь поле **salary** (ставка) и **seniority** (стаж), конструктор с двумя параметрами, заполняющий эти поля, и чисто виртуальный метод расчета надбавок, переопределенный в каждом из производных классов. Надбавки вычисляются так: $salary \cdot (1 + seniority \cdot x)$, где $x=0,03$, $0,02$ и $0,01$ для СтаршегоНС, МладшегоНС и ИТР соответственно.

Также, каждый производный класс должен иметь конструктор с двумя параметрами, передающий свои параметры конструктору базового класса. В методе **main** создать массив из 9 указателей на **Scientist** и заполнить его объектами производных классов. Вывести на экран величину надбавок для всех объектов. **В массив вносить только уникальные объекты.** Выполнить переопределение **Equals** и **ToString** в каждом классе.

Вариант 3: Разработайте UML – диаграмму и программу:

базовый класс Машина (**Car**) с полями "Марка", "год", "объем двигателя", "тип коробки передач", "базовое значение страхового сбора" – **base**, и функцией расчета страхового сбора для дочерних классов "Иномарка" – **InBrand** – $base + 0,05 \cdot year \cdot volume$, "Отечественный бренд" – **HomeBrand**: $base + 0,03 \cdot year \cdot volume$. Каждый производный класс (Иномарка и Отечественный бренд) должен иметь конструктор с несколькими параметрами, передающий свои параметры конструктору базового класса. В методе **main** создать массив из 7 указателей на **Car** и заполнить его объектами производных классов. Вывести на экран величину страхового сбора для всех объектов. **В массив вносить только уникальные объекты.** Выполнить переопределение **Equals** и **ToString** в каждом классе.

Вариант 4: Разработать иерархию геометрических фигур (не менее 3-х дочерних классов) с базовым классом **Shape**. Базовый класс содержит одну координату для вывода фигур на экран и виртуальные методы **Square()** и **Perimetr()**, которые подсчитают площадь и периметр фигуры соответственно. Дочерние классы (например, линия, прямоугольник и окружность) должны содержать параметры фигуры, по которым их можно нарисовать и рассчитать площадь и периметр. В методе **main** создать массив из 8 указателей на **Shape** и заполнить его объектами производных классов. Вывести на экран величину **S** и **P** для всех объектов. **В массив вносить только уникальные объекты.** Выполнить переопределение **Equals** и **ToString** в каждом классе.

4.2 Фонд оценочных средств для проведения промежуточной аттестации.

Вариант контрольной работы 2

1. Дайте понятие полиморфизма. Назовите виды и способы реализации полиморфизма, приведите пример.

2. Что случится, когда вы попытаетесь исполнить следующий код? Каков будет ответ, если раскомментировать две строчки?

```
class Base{
    private void amethod(int iBase){
        System.out.println("Base.amethod");
    }
}
class Over extends Base{
    public static void main(String argv[]){
        Over o = new Over();
        int iBase=0; o.amethod(iBase);

        // Base b=new Base();
        // b.amethod(iBase);
    }
    public void amethod(int iOver){
        System.out.println("Over.amethod");
    }
}
```

A Компилятор выдаст ошибку о том, что Base.amethod is private

B Ошибка времени выполнения: Base.amethod is private

C Код скомпилируется и выведется "Base.amethod"

D Код скомпилируется и выведется "Over.amethod"

E Код скомпилируется и выведется "Over.amethod" , "Base.amethod".

3. Разработать класс "программа ТВ на неделю", содержащий список программ на 7 дней. Создайте класс на протяжении дня и недели. Если в программе нет новостей – их нужно добавить, а передача, назначенная на 20-00 должна быть сдвинута на час позже. Вначале напечатайте свою программу ТВ, затем добавьте некоторые передачи и распечатайте снова. Затем добавьте "прерывающие" новости и напечатайте целую программу снова.

Критерии оценки:

Критерий	Оценка
На один из двух вопросов (теория и практикоориентированный тестовый вопрос) дан правильный ответ (1,2). Верна концепция решения практического задания, верны разработаны основные классы программы. Возможные небольшие неточности.	Зачет
Не даны ответы на теоретические вопросы (1, 2). Классы программы не представлены даже концептуально либо разработаны со значительными ошибками.	Не зачет

Варианты практической задачи к зачету:

1. Написать шаблон класса для работы с очередью FIFO. Определить функции включения и исключения элементов. Добавить механизм обработки исключений при превышении размера очереди и при попытке удалить данные из пустой очереди. Это можно сделать, добавив элемент данных – счетчик текущего числа элементов. Исключения генерируются, если счетчик превысил размер массива или если он стал меньше 0.

2. Опишите классы PointXY и PointPolar, объекты которых задают декартовы и полярные координаты точки на плоскости. Перегрузите для этих классов операции сложения, вычитания и умножения как скалярного произведения, так, чтобы в них могли участвовать объекты как одного, так и обоих классов. Кроме того, задайте функцию преобразования одного класса в другой (для обоих классов).

3. Определить класс stack, который позволяет реализовать структуру данных типа стек для хранения целых чисел. Конструктор класса должен содержать параметр, определяющий размер стека. Определить для класса функции pop() (достать из стека), push() (положить в стек) и операцию определения текущего размера стека. Функции должны осуществлять проверку на выход за пределы стека. Определить класс fifo, реализующий структуру данных типа очередь для хранения целых чисел, породив его от класса stack, добавив нужные поля и переопределив функции pop(), push() и определение текущего размера очереди. Продемонстрировать работу.

4. В любой визуальной среде создать класс Figure с виртуальным методом draw(), осуществляющим прорисовку объекта на визуальном компоненте. Создать производные классы: Rectangle (прямоугольник), Circle (круг), Triangle (треугольник). Описать в производных классах функции draw() для каждой из фигур, продемонстрировать работу механизма виртуальных функций.

<i>Критерий</i>	<i>Оценка</i>
В предлагаемом решении верно оформлен ввод и вывод данных, верно определены классы, их поля и методы. Допускаются неточности в определениях при ответах на вопросы по программе.	Зачет
Не верно определена структура программы, отсутствует понятие основных концепций ООП.	Не зачет

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

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

5.1 Основная литература:

1. Николаев, Е.И. Объектно-ориентированное программирование : учебное пособие / Е.И. Николаев ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». – Ставрополь : СКФУ, 2015. – 225 с. : ил. – Библиогр. В кн. ; То же [Электронный ресурс]. – URL: <http://biblioclub.ru/index.php?page=book&id=458133>
2. Объектно-ориентированное программирование : лабораторный практикум : в 2 ч. / авт.-сост. Е.И. Николаев ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». – Ставрополь : СКФУ, 2015. – Ч. 2. – 156 с. : ил. – Библиогр. В кн. ; То же [Электронный ресурс]. – URL: <http://biblioclub.ru/index.php?page=book&id=458135>
3. Объектно-ориентированное программирование : лабораторный практикум : в 2 ч. / авт.-сост. Е.И. Николаев ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». – Ставрополь : СКФУ, 2015. – Ч. 1. – 183 с. : ил. – Библиогр.: с. 179. ; То же [Электронный ресурс]. – URL: <http://biblioclub.ru/index.php?page=book&id=458134>

12.2 Дополнительная литература:

1. Варфоломеева, Т.Н. Лабораторный практикум по **объектно-ориентированному программированию** / Т.Н. Варфоломеева, И.Ю. Ефимова. - 2-е изд., стер. - Москва : Издательство «Флинта», 2014. - 75 с. : ил. - Библиогр. в кн. - ISBN 978-5-9765-2042-4 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=482219>
2. Сорокин, А.А. **Объектно-ориентированное программирование** : учебное пособие (курс лекций) / А.А. Сорокин ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь : СКФУ, 2014. - 174 с. : ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=457696>
3. Корчуганова, М.Р. **Объектно-ориентированное программирование** на C++ : электронное учебное пособие / М.Р. Корчуганова, К.С. Иванов, Л.В. Бондарева ; Министерство образования и науки РФ, Кемеровский государственный университет, Кафедра вычислительной математики. - Кемерово : Кемеровский государственный университет, 2015. - 196 с. : ил. - Библиогр. в кн. - ISBN 978-5-8353-1832-2 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=481559>

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

1. Основы программирования на языке C++: Учебное пособие/ Сост. С.М. Наместников http://sernam.ru/lect_c.php
2. Основы C++. Е. Линский. <https://www.lektorium.tv/course/22825>

7. Методические указания для обучающихся по освоению дисциплины.

По курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал; лабораторных занятий, на которых приводятся примеры решений задач по основным учебным темам, выполняются на компьютере с использованием компилятора C++ или Java JVM, задачи и упражнения, соответствующие разделам лекционного курса.

Важнейшим этапом курса является самостоятельная работа по дисциплине «Объектно-ориентированное решение задач».

При самостоятельной работе студентов необходимо изучить литературу, приведенную в перечне литературных источников, для осмысления вводимых понятий, анализа предложенных подходов и методов разработки программ. Разрабатывая решение новой задачи, студент должен уметь выбрать эффективные и надежные структуры данных для представления информации, подобрать соответствующие алгоритмы для их обработки, учесть специфику языка программирования, на котором будет выполнена реализация. Студент должен уметь выполнять тестирование и отладку алгоритмов решения задач с целью обнаружения и устранения в них ошибок.

В качестве систем программирования для решения задач и изучения методов и алгоритмов, приведенных в лекциях, рекомендуется использовать на практических занятиях и при самостоятельной работе компилятор C++ или Java JVM. Для эффективного программирования рекомендуется использовать встроенные отладчики.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.

8.1 Перечень информационных технологий.

- Проверка домашних заданий и консультирование посредством электронной почты.
- Использование электронных презентаций при проведении лекционных занятий.

8.2 Перечень необходимого программного обеспечения.

- Компилятор Microsoft Visual Studio C++, для разработки программ.
- Свободное ПО NetBeans.

8.3 Перечень информационных справочных систем:

Электронные библиотечные источники:

1. ЭБС Издательства «Лань» <http://e.lanbook.com> ,
2. ЭБС «Университетская библиотека онлайн» www.biblioclub.ru ,
3. ЭБС «Юрайт» <http://www.biblio-online.ru> ,
4. ЭБС «ZNANIUM.COM» www.znanium.com ,
5. ЭБС «BOOK.ru» <https://www.book.ru>.

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.

№	Вид работ	Материально-техническое обеспечение дисциплины и
---	-----------	--

		оснащенность
1.	Лекционные занятия	Лекционная аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) и соответствующим программным обеспечением (ПО), доска
2.	Лабораторные занятия	Лаборатория, укомплектованная техническими средствами обучения – компьютерами с соответствующим программным обеспечением, маркерная доска.
3.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная маркерной доской и оснащенная компьютером.
4.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная техническими средствами обучения – компьютерами с соответствующим программным обеспечением
5.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.