

2015
2016

Министерство образования и науки Российской Федерации
 Федеральное государственное бюджетное образовательное учреждение
 высшего профессионального образования
 «Владимирский государственный университет
 имени Александра Григорьевича и Николая Григорьевича Столетовых»
 (ВлГУ)



УТВЕРЖДАЮ

Проректор по УМР
А.А.Панфилов

« 29 » _____ 2015 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ
 (наименование дисциплины)

Направление подготовки: 02.03.01 Математика и компьютерные науки

Профиль/программа подготовки: Математические методы в экономике и финансах

Уровень высшего образования: бакалавриат

Форма обучения: очная

Семестр	Трудоёмкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточного контроля (экз./зачет)
4	4 / 144	36	–	36	72	зачёт с оценкой
Итого	4 / 144	36	–	36	72	зачёт с оценкой

Владимир 2015

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Важным фактором подготовки специалистов в области компьютерных наук является умение программировать, используя современные языки, включающие объектные возможности, знакомство с основными методами и современными технологиями программирования, в том числе с использованием объектных библиотек конкретных языков.

Цель курса: на базе одного из самых распространенных языков программирования высокого уровня C++, позволяющего использовать объектно-ориентированное программирование, познакомиться с общепринятыми методиками создания приложений в современных объектно-ориентированных программных средах. Помимо этого, целью является и знакомство с основанным на C++ языком C#.

Цель обусловила и задачи курса: приобретение теоретических знаний и практических навыков объективно-ориентированного программирования прикладных задач на языках C++ и C# в интегрированной среде программирования MS Visual Studio, в том числе с использованием стандартных библиотек классов. При практической работе на ЭВМ рекомендуются в качестве прикладных задач типовые вычислительные задачи, решение которых на предыдущем курсе выполнялось с использованием других подходов.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Объектно-ориентированное программирование» относится к обязательным дисциплинам вариативной части блока Б1 ОПОП подготовки бакалавров по направлению «Математика и компьютерные науки».

Дисциплина изучается в четвёртом семестре и требует освоения следующих курсов:

- Алгоритмы и алгоритмические языки;
- Иностранный язык.

В рамках перечисленных дисциплин студенты получают следующие знания и навыки, необходимые для освоения курса «Объектно-ориентированное программирование»:

- Умение применять методики алгоритмизации задач, выбирать наиболее эффективные алгоритмы;
- Знание основных современных средств разработки программных продуктов;
- Умение получать информацию из источников на иностранном языке.

Дисциплина формирует знания и навыки, необходимые в ходе дальнейшего обучения и в практической деятельности квалифицированного специалиста. В рамках учебного процесса может быть использована в ходе изучения дисциплин:

- Компьютерная геометрия и геометрическое моделирование;
- Базы данных;
- Численные методы;
- Безопасность информационных систем / Теория кодирования;
- Распределённая обработка информации / Параллельное программирование;
- Веб-программирование и основы веб-дизайна.

Кроме того, полученные навыки будут необходимы при прохождении практик и подготовке выпускной квалификационной работы.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате изучения дисциплины студент должен частично овладеть следующими компетенциями:

- ОПК-2, способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
- ПК-6, способностью передавать результат проведенных физико-математических и прикладных исследований в виде конкретных рекомендаций, выраженных в терминах предметной области изучавшегося явления.

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

1) Знать:

- Основные концептуальные положения объектно-ориентированного направления программирования, методы, способы и средства разработки программ в рамках этого направления (ОПК-2);
- Основные проблемы объектно-ориентированного подхода и тенденции развития объектно-ориентированных языков программирования (ОПК-2);

2) Уметь:

- Проектировать систему классов для заданной предметной области (ПК-6);
- Программно реализовывать разработанную систему классов для заданной предметной области (ОПК-2);
- Анализировать эффективность объектно-ориентированного программного кода (ОПК-2);

2) Владеть

- Средствами разработки объектно-ориентированного программного кода (ОПК-2, ПК-6).

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часов.

№ п/п	Раздел (тема) дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем учебной работы, с применением интерактивных методов (в часах / %)	Формы текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	Контрольные работы	СРС	КП / КР		
1	Раздел 1. Парадигма объектно-ориентированного программирования	4	1	2	-	-	-	4	-		Рейтинг-контроль №1
2	Раздел 2. Абстракция и инкапсуляция	4	1-4	6	-	6	-	10	-	6 / 50%	

3	Раздел 3. Одиночное наследование и полиморфизм	4	4-9	8	-	12	-	14	-	12 / 60%	Рейтинг-контроль №2
4	Раздел 4. Множественное наследование	4	9-14	8	-	10	-	14	-	10 / 55%	Рейтинг-контроль №3
5	Раздел 5. Дополнительные возможности C++.	4	13-14	4	-	-	-	18	-		
6	Раздел 5. Основы языка C#	4	15-18	8	-	8	-	12	-	8 / 50%	
Всего		4	18	36	-	36	-	72	-	36 / 50%	зачёт с оценкой

СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Темы лекций

Раздел 1. Парадигма объектно-ориентированного программирования.

1. Общая качественная характеристика ООП, основные свойства и возможности. Составной тип класс: объявление, структура, возможные составляющие компоненты; объект типа класс. Примеры.

Раздел 2. Абстракция и инкапсуляция.

1. Метки видимости. Доступ к открытым и закрытым членам. Указатель this. Примеры.
2. Конструкторы и деструкторы: назначение, варианты объявления и использования (в том числе создание и разрушение объектов в динамической памяти).
3. Статистические члены класса. Примеры. Друзья класса: назначение и примеры использования.
4. Шаблоны классов. "Вмещающие" классы. Примеры.

Раздел 3. Одиночное наследование и полиморфизм.

1. Иерархия классов: базовый класс, производный класс. Модификаторы доступа. Доступ к членам базовых и производных классов "изнутри" производственного класса и "извне".
2. Указатели на объекты классов. Преобразование типов для объектов и указателей по иерархии наследования, возможные операции присваивания.
3. Общепринятые соглашения об именах произвольных типов (ссылок и указателей). Доступ к одноименным членам классов по иерархии наследования, механизм раннего связывания.
4. Программные возможности распознавания типа, к которому принадлежит указуемый объект (при работе по указателям). Механизм виртуальных функций как вариант позднего связывания объекта с методами: особенности компиляции типа (таблицы виртуальных методов), объектов, вызов методов в этой ситуации (через объекты, через указатели разных типов).
5. Виртуальные деструкторы. Абстрактные классы.
6. Указатели на члены класса (поля и методы): объявление типов, переменных, примеры и рекомендации для использования.
7. Перегрузка операторов для классов (++ , — в префиксном и постфиксном исполнении; +, -, =; « и »; последние две операции на стандартных потоках ввода/вывода).
8. Ввод/вывод с логических устройств в C++ через объекты стандартных классов; потоки с дисковых файлов.

Раздел 4. Множественное наследование.

1. Множественное наследование и виртуальные базовые классы.

Раздел 5. Дополнительные возможности C++.

1. Обработка исключений

2. Рекомендации к разработке многомодульных программ с использованием классов, к созданию пользовательских объектных библиотек.
3. Разработка оконных приложений

Раздел 6. Основы языка C#.

1. Базовые конструкции языка C#
2. Объектная модель языка C#
3. Особенности и перспективы использования и развития языка C#

Лабораторный практикум

Состоит из семи лабораторных работ. Список тем лабораторных занятий:

- 1) Абстракция и инкапсуляция (2 ч).
- 2) Перегрузка операций (4 ч).
- 3) Одиночное наследование и полиморфизм (6 ч).
- 4) Шаблонные классы (6 ч).
- 5) Множественное наследование (10 ч).
- 6) Абстракция и инкапсуляция в языке C# (4 ч).
- 7) Наследование и полиморфизм в языке C# (4 ч).

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В рамках лекционного курса:

- Технология проблемного обучения (case study). При рассмотрении вопросов практического применения рассмотренного теоретического материала, используется диалог со студентами на предмет возможных способов решения поставленной задачи. Особенно активно данная технология применяется в рамках четвертого раздела курса.
- Встречи с представителями фирм-разработчиков программного обеспечения при изучении заключительного раздела дисциплины.

В рамках лабораторного практикума:

- Технология уровневой дифференциации. Прежде всего, при постановке заданий на лабораторные работы, а также при проведении контрольных мероприятий и при проведении экзамена.

В рамках курсового проектирования и самостоятельной работы:

- Метод проектов. В группах по 2-3 человека студенты на практике осваивают принципы объектно-ориентированного программирования.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Рейтинг-контроль №1 (контроль освоения разделов 1, 2)

Проводится в письменной форме и призван вывить знания студентами принципов инкапсуляции и абстракции в объектно-ориентированном программировании.

Список заданий:

- 1) *Составной тип класс: объявление, структура, возможные составляющие компоненты; объект типа класс. Примеры.*
- 2) *Метки видимости. Доступ к открытым и закрытым членам. Указатель this. Примеры.*
- 3) *Конструкторы и деструкторы: назначение, варианты объявления и использования (в том числе создание и разрушение объектов в динамической памяти).*
- 4) *Статистические члены класса. Примеры. Друзья класса: назначение и примеры использования.*
- 5) *Перегрузка операторов для классов (++ , — в префиксном и постфиксном исполнении; +, -, =; « и »; последние две операции на стандартных потоках ввода/вывода).*
- 6) *Ввод/вывод с логических устройств в C++ через объекты стандартных классов; потоки с дисковых файлов.*

7) *Шаблоны классов. "Вмещающие" классы. Примеры.*

Рейтинг-контроль №2 (контроль освоения раздела 3)

Проводится в письменной форме и призван вывить знания студентами принципов одиночного наследования и полиморфизма в объектно-ориентированном программировании.

Список заданий:

- 1) *Иерархия классов: базовый класс, производный класс.*
- 2) *Модификаторы доступа при наследовании. Доступ к членам базовых и производных классов "изнутри" производственного класса и "извне".*
- 3) *Указатели на объекты классов. Преобразование типов для объектов и указателей по иерархии наследования, возможные операции присваивания.*
- 4) *Общепринятые соглашения об именах произвольных типов (ссылок и указателей). Доступ к одноименным членам классов по иерархии наследования, механизм раннего связывания.*
- 5) *Механизм виртуальных функций как вариант позднего связывания объекта с методами: особенности компиляции типа (таблицы виртуальных методов), объектов, вызов методов в этой ситуации (через объекты, через указатели разных типов).*
- 6) *Виртуальные деструкторы. Абстрактные классы.*
- 7) *Указатели на члены класса (поля и методы): объявление типов, переменных, примеры и рекомендации для использования.*

Рейтинг-контроль №3 (контроль освоения разделов 4-6)

Проводится в письменной форме и призван вывить знания студентами принципов множественного наследования и языка C#.

Список заданий:

- 1) *Множественное наследование. Правила доступа к полям и методам.*
- 2) *Множественное наследование и виртуальные базовые классы.*
- 3) *Полиморфизм при множественном наследовании.*
- 4) *Обработка исключительных ситуаций.*
- 5) *Особенности C# как Си-подобного языка*
- 6) *Ввод/вывод в C#*
- 7) *Особенности наследования в C#.*
- 8) *Работа с графическими библиотеками в C#.*

Самостоятельная работа студента (контроль освоения всех разделов курса)

Следует отметить, что в связи с существенной практической направленностью курса требуется развитие и закрепление практических навыков по объектно-ориентированному программированию в рамках самостоятельной работы студентов. С этой целью выдаётся практическое задание, ход выполнения которого контролируется в течение семестра и учитывается при проставлении дифференцированного зачёта.

В ходе выполнения самостоятельной работы группе студентов из 2-3 человек необходимо разработать оконное приложение, тестирующее систему классов, спроектированную и реализованную студентами для решения конкретной задачи из некоторой предметной области.

Примерная тематика работ:

- 1) *Библиотека классов, реализующих графические примитивы на плоскости с возможностью аффинных преобразований.*
- 2) *Библиотека классов, реализующих графические примитивы на плоскости с реализацией операций над множествами.*
- 3) *Библиотека классов, реализующих графические примитивы в трёхмерном пространстве с возможностью аффинных преобразований.*
- 4) *Классы для описания оконного графического интерфейса, аналогичного интерфейсу MS Windows. Должны быть реализованы обработчики событий с использованием виртуальных функций.*
- 5) *Библиотека классов, реализующих комплексную арифметику.*
- 6) *Библиотека классов, реализующих векторы в n-мерном пространстве.*
- 7) *Реализация строк и операций над ними, включая работу с регулярными выражениями.*
- 8) *«Обобщённый массив» (позволяющий хранить данные произвольных типов).*
- 9) *Ассоциативный массив (хэш-массив) (с возможностью хранения данных произвольных типов).*

- 10) Реализация различных типов графов и операций над ними.
- 11) Система классов для обеспечения работы с абонентами телефонной компании.
- 12) Система классов для обеспечения работы деканата.
- 13) Система классов, описывающих сотрудников предприятия/организации с их функциями (сотрудник, менеджер, ...).
- 14) Система классов, описывающих различные транспортные средства.
- 15) Моделирование замкнутой биологической системы (корм, травоядное, хищник).
- 16) Моделирование муравейника (несколько типов муравьёв, источники питания, внешние раздражители, ...).
- 17) Моделирование дорожного движения на заданной карте дорог.
- 18) Моделирование компьютерной сети (стационарной).
- 19) Моделирование компьютерной сети с изменяющейся топологией.
- 20) Моделирование планетарной системы.
- 21) Система классов для реализации матричной арифметики.
- 22) Библиотека классов для реализации игрового приложения.

Зачёт с оценкой (4-й семестр, контроль освоения всех разделов курса)

Предполагает письменное решение студентом практической задачи: написание программ на языках C++ и C#. Конкретная задача выбирается случайным образом из базы задач, имеющейся у преподавателя. Решение задачи должно быть написано на обоих языках. Во время решения задачи студент может пользоваться справочной литературой и конспектами. Уровень решения определяет выставяемую оценку. При этом студентом предварительно должны быть выполнены все задания лабораторного практикума.

Примеры задач:

1. Описать класс *TDisplay*, инкапсулирующий следующую информацию о компьютерном мониторе: производитель, модель, размер диагонали, масса, потребляемая мощность. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для одного из свойств описать пару методов, предназначенных для изменения значения свойства и для получения этого значения. Описать класс *TTouchpad*, описывающий сенсорный дисплей и являющийся наследником класса *TDisplay*. Набор дополнительных свойств этого класса определить самостоятельно. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор.
2. Описать класс *TPlane*, инкапсулирующий следующую информацию о самолёте: длина, размах крыльев, масса, максимальная дальность перелёта. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для одного из свойств описать пару методов, предназначенных для изменения значения свойства и для получения этого значения. Описать класс *T Bomber*, описывающий самолёт-бомбардировщик и являющийся наследником класса *TPlane*. Набор дополнительных свойств этого класса определить самостоятельно. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор.
3. Описать класс *TDog*, инкапсулирующий следующую информацию о собаке: порода, кличка, высота в холке, вес, длина хвоста. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для одного из свойств описать пару методов, предназначенных для изменения значения свойства и для получения этого значения. Описать класс *TSheepdog*, описывающий пастушью собаку и являющийся наследником класса *TDog*. Набор дополнительных свойств этого класса определить самостоятельно. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор.
4. Описать класс *TCat*, инкапсулирующий следующую информацию о кошке: кличка, длительность сна в день, возраст, вес. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для одного из свойств описать пару методов, предназначенных для изменения значения свойства и для получения этого значения. Описать класс *TTiger*, описывающий дрессированного тигра и являющийся наследником класса *TCat*. Набор дополнительных свойств этого класса определить самостоятельно. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература:

1. Язык программирования С++. / Страуструп Б. - М.: Бином, 2011. -1136 с.: ил. ISBN 978-5-7989-0425-9
2. Программирование на языке С++: Учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 512 с.: ил.; 60x90 1/16 + CD-ROM. - (Проф. обр.). (п, cd rom) ISBN 978-5-8199-0492-3
3. Введение в программирование на языке Visual С#: Учебное пособие / С.Р. Гуриков. - М.: Форум: НИЦ ИНФРА-М, 2013. - 448 с.: 70x100 1/16. - (Высшее образование: Бакалавриат). (переплет) ISBN 978-5-91134-738-3

б) дополнительная литература:

1. Практикум по объектно-ориентированному программированию [Электронный ресурс] / И. А. Бабушкина, С.М. Окулов.-4-е изд. (эл.).-Электрон. текстовые дан. (1 файл pdf : 369 с.). - М. : БИНОМ. Лаборатория знаний, 2015. - ISBN 978-5-9963-2542-9
2. Эффективное использование С++. 55 верных способов улучшить структуру и код ваших программ - М.: ДМК Пресс, 2008. - 300 с.: ил. - ISBN 5-94074-304-8
3. Программирование на языке высокого уровня. Программирование на языке Object Pascal: Учебное пособие / Т.И. Немцова; Под ред. Л.Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2015. - 496 с.: ил.; 60x90 1/16. - (ПО). (cd rom) ISBN 978-5-8199-0372-8

в) периодические издания:

1. Computerworld Россия, ISSN: 1560-5213.
2. Мир ПК, ISSN: 0235-3520.

г) интернет-ресурсы:

1. MSDN. Возвращение к С++ (современный С++). // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/hh279654.aspx>
2. MSDN. Справочник по языку С++. // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/3bstk3k5.aspx>
3. MSDN. Руководство по программированию на С#. // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>
4. MSDN. Справочник по С#. // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/618ayhy6.aspx>
5. MSDN. Visual Studio. // Режим доступа: <https://www.visualstudio.com/>

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Компьютерные классы для проведения лабораторных занятий. Компьютеры должны иметь выход в Интернет для получения доступа к справочной информации Microsoft Developer Network.

Среда разработки Microsoft Visual Studio.

ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Рабочая программа одобрена на 2018/2019 учебный год.

Протокол заседания кафедры № 1 от 04.09.18 года.

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год.

Протокол заседания кафедры № __ от _____ года.

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год.

Протокол заседания кафедры № __ от _____ года.

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год.

Протокол заседания кафедры № __ от _____ года.

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год.

Протокол заседания кафедры № __ от _____ года.

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год.

Протокол заседания кафедры № __ от _____ года.

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год.

Протокол заседания кафедры № __ от _____ года.

Заведующий кафедрой _____