

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



УТВЕРЖДАЮ

Проректор
по образовательной деятельности

А.А.Панфилов

« 31 » 08 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ И ПРОГРАММИРОВАНИЕ

(наименование дисциплины)

Направление подготовки 38.03.05 «Бизнес-информатика»

Профиль/программа подготовки – «Информационно-аналитическое обеспечение предпринимательской деятельности»

Уровень высшего образования - бакалавриат

Форма обучения – заочная

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточной аттестации (экзамен/зачет/зачет с оценкой)
4	4/144	4	4	4	132	Зачет с оценкой
Итого	4/144	4	4	4	132	Зачет с оценкой

Владимир 2020

ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель освоения дисциплины:

- получение студентами знаний основ объектно-ориентированного анализа;
- формирование у студентов приобретение теоретических знаний в области идеологии объектно-ориентированного программирования практической работы в среде объектно-ориентированного программирования;
- привитие навыков работы с научно-технической документацией по разработке программных продуктов, овладение технологическими ориентированного анализа и программирования.

Задачи - сформировать у студентов: представление о парадигме объектно-ориентированного подхода (инкапсуляция, полиморфизм, наследование), а также языках программирования его использующих; представление о принципах построения объектно-ориентированных программных систем для современных операционных систем; создании объектно-ориентированных приложений; обучить студентов практическим навыкам разработки объектно-ориентированных систем (ООС).

1. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Объектно-ориентированный анализ и проектирование» относится к базовой части. Пререквизиты дисциплины: «Информатика», «Исследование операций».

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП

Код формируемых компетенций	Уровень освоения компетенции	Планируемые результаты обучения по дисциплине характеризующие этапы формирования компетенций (показатели освоения компетенции)
1	2	3
<p>ОПК–1 - способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</p>	<p>Частичный</p>	<p>Знать: требования ИБ в области профессиональной деятельности.</p> <p>Уметь: определять набор требований по защите информации.</p> <p>Владеть: навыками работы с Российской и зарубежной научно-исследовательской литературой.</p>
<p>ПК–13 умение проектировать и внедрять компоненты ИТ-инфраструктуры предприятия, обеспечивающие достижение стратегических целей и поддержку бизнес-процессов</p>	<p>Частичный</p>	<p>Знать: методы технического проектирования; -принципы проектирования и внедрения компонентов ИТ-инфраструктуры предприятия, обеспечивающих достижение стратегических целей и поддержку бизнес-процессов; сущность методологии имитационного моделирования бизнес-процессов сложных систем; ключевые современные подходы к проектированию и внедрению компонентов ИТ-инфраструктуры предприятия.</p> <p>Уметь: разрабатывать и анализировать технические решения для различных инфокоммуникационных объектов; проектировать и внедрять компоненты ИТ-инфраструктуры предприятия, обеспечивающие достижение стратегических целей и поддержку бизнес-процессов; проектировать и внедрять имитационные модели с целью достижения стратегических целей и поддержки бизнес-процессов</p> <p>- осуществлять выбор проектных решений и разрабатывать программу внедрения бухгалтерского ПО в соответствии со стратегическими целями и бизнес-процессами организации.</p> <p>Владеть: методами создания технических решений вычислительных сетей; навыками проектирования и внедрения компонентов ИТ-инфраструктуры предприятия, обеспечивающих достижение стратегических целей и поддержку бизнес-процессов; навыками проектирования и внедрения имитационных моделей с целью достижения стратегических целей и поддержки бизнес-процессов..</p>

3. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 3 зачетных единицы, 108 часов

№ п/п	Наименование тем и/или разделов/тем 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, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	СРС		
1	Объектно-ориентированное программирование	4	1-2	2	2	2-	6	3/50	
2	Объектно-ориентированный анализ	4	3	2	2	2-	6	3/50	
3	Объектно-ориентированная модель предметной области: объекты	4	4-5	2	2	2-	6	3/50	Рейтинг-контроль №1
4	Объектно-ориентированная модель предметной области: классы	4	6-7	2	2	2	6	3/50	
5	Программирование классов на С#: поля, константы и методы	4	8-10	2	2	2	6	3/50	
6	Программирование классов на С#: конструкторы и деструкторы	4	11-12	2	2	2	6	3/50	Рейтинг-контроль №2
7	Программирование классов на С#: свойства и индексы. Реализация принципа инкапсуляции.	4	13-14	2	2	2	6	3/50	
8	Реализация принципа наследования и полиморфизма на С#.	4	15-16	2	2	2	6	3/50	
9	Делегаты и события на С#. Многопоточные приложения	4	17-18	2	2	2	6	3/50	Рейтинг-контроль №3
Всего за 4 семестр:				18	18	18	54	27(50%)	Зачет
Наличие в дисциплине КП/КР									
Итого по дисциплине				18	18	18	54	27(50%)	Зачет

Содержание лекционных занятий по дисциплине

Тема 1. Объектно-ориентированное программирование.

Из истории языков программирования. Основные принципы структурной методологии. Понятие объектно-ориентированного программирования. Абстрактные типы данных. Объекты и классы. Базовые принципы ООП. Пример объектной декомпозиции. Опи-

сание объектного типа. Иерархия типов (наследование). Полиморфизм и виртуальные методы.

Тема 2. Объектно-ориентированный анализ.

Объектно-ориентированные модели. Классические подходы к анализу объектно-ориентированных систем. Анализ поведения. Анализ предметной области. Анализ вариантов. CRC-карточки. Неформальное описание. Структурный анализ.

Тема 3. Объектно-ориентированная модель предметной области: объекты.

Стандарт ODMG. Объект типа БИБЛИОТЕКА. Объекты типа КНИГА.

Тема 4. Объектно-ориентированная модель предметной области: классы.

Общая форма определения класса. Данные-члены. Значение свойства типа class. Класс АБОНЕНТ. Класс КАТАЛОГ. Класс ВЫДАЧА.

Тема 5. Программирование классов на С#: поля, константы и методы.

Данные: поля и константы. Спецификатор Описание. Спецификаторы полей и констант класса. Обращение к полю класса. Функции-члены. Пометки констант (public, private, protected, internal, protected internal или private protected.). Методы (method).

Тема 6. Программирование классов на С#: конструкторы и деструкторы.

Конструктор. Конструктор с атрибутом static. Деструктор. Деструктор класса. Деструктора класса Person.

Тема 7. Программирование классов на С#: свойства и индексы. Реализация принципа инкапсуляции.

Свойства класса. Синтаксис описания свойства. Индексы. Параметры индекса. Модификаторы тип this. Инкапсуляция. Модификаторы доступа. Реализация принципа инкапсуляции.

Тема 8. Реализация принципа наследования и полиморфизма на С#.

Принципы объектно-ориентированного программирования. Наследование. Понятия «предок - потомок». Полиморфизм. Реализация принципов наследования и полиморфизма.

Тема 9. Делегаты и события на С#. Многопоточные приложения

Типы делегатов DelegateArticle.FirstDelegate. DelegateArticle.Sample.SecondDelegate. Экземпляры делегатов. Создание экземпляров делегатов. Вызов экземпляров делегатов. Комбинирование делегатов. События. Field-like события. Потокбезопасные события. Многопоточность. Класс Thread. Получение информации о потоке. Статус потока. Приоритеты потоков.

Тема 1. Объектно-ориентированное программирование.

1. Из истории языков программирования.
2. Основные принципы структурной методологии.
3. Понятие объектно-ориентированного программирования.
4. Абстрактные типы данных.
5. Объекты и классы.
6. Базовые принципы ООП.
7. Пример объектной декомпозиции.
8. Описание объектного типа.
9. Иерархия типов (наследование). Полиморфизм и виртуальные методы.

Тема 2. Объектно-ориентированный анализ.

1. Объектно-ориентированные модели.
2. Классические подходы к анализу объектно-ориентированных систем.
3. Анализ поведения.
4. Анализ предметной области.
5. Анализ вариантов.
6. CRC-карточки.
7. Неформальное описание.
8. Структурный анализ.

Тема 3. Объектно-ориентированная модель предметной области: объекты.

1. Стандарт ODMG.
2. Объект типа БИБЛИОТЕКА.
3. Объекты типа КНИГА.

Тема 4. Объектно-ориентированная модель предметной области: классы.

1. Общая форма определения класса.
2. Данные-члены.
3. Значение свойства типа class.
4. Класс АБОНЕНТ.
5. Класс КАТАЛОГ.
6. Класс ВЫДАЧА.

Тема 5. Программирование классов на C#: поля, константы и методы.

1. Данные: поля и константы.
2. Спецификатор Описание.
3. Спецификаторы полей и констант класса.
4. Обращение к полю класса.
5. Функции-члены.

6. Пометки констант (public, private, protected, internal, protected internal или private protected.).

7. Методы (method).

Тема 6. Программирование классов на C#: конструкторы и деструкторы.

1. Конструктор.

2. Конструктор с атрибутом static.

3. Деструктор.

4. Деструктор класса.

5. Деструктора класса Person.

Тема 7. Программирование классов на C#: свойства и индексы. Реализация принципа инкапсуляции.

1. Свойства класса.

2. Синтаксис описания свойства.

3. Индексы.

4. Параметры индекса.

5. Модификаторы типа this.

6. Инкапсуляция.

7. Модификаторы доступа.

8. Реализация принципа инкапсуляции.

Тема 8. Реализация принципа наследования и полиморфизма на C#.

1. Принципы объектно-ориентированного программирования.

2. Наследование.

3. Понятия «предок - потомок».

4. Полиморфизм.

5. Реализация принципов наследования и полиморфизма.

Тема 9. Делегаты и события на C#. Многопоточные приложения.

1. Типы делегатов

2. DelegateArticle.FirstDelegate.

3. DelegateArticle.Sample.SecondDelegate.

4. Экземпляры делегатов.

5. Создание экземпляров делегатов.

6. Вызов экземпляров делегатов.

7. Комбинирование делегатов.

8. События. Field-like события.

9. Потокбезопасные события.

10. Многопоточность.

11. Класс Thread.
12. Получение информации о потоке.
13. Статус потока.
14. Приоритеты потоков.

Содержание практических/лабораторных занятий по дисциплине

Тема 1. Объектно – ориентированное программирование.

1. Из истории языков программирования.
2. Основные принципы структурной методологии.
3. Понятие объектно-ориентированного программирования.
4. Абстрактные типы данных.
5. Объекты и классы.
6. Базовые принципы ООП.
7. Пример объектной декомпозиции.
8. Описание объектного типа.
9. Иерархия типов (наследование). Полиморфизм и виртуальные методы.

Тема 2. Объектно- ориентированный анализ.

1. Объектно-ориентированные модели.
2. Классические подходы к анализу объектно-ориентированных систем.
3. Анализ поведения.
4. Анализ предметной области.
5. Анализ вариантов.
6. CRC-карточки.
7. Неформальное описание.
8. Структурный анализ.

Тема 3. Объектно-ориентированная модель предметной области: объекты.

1. Стандарт ODMG.
2. Объект типа БИБЛИОТЕКА.
3. Объекты типа КНИГА.

Тема 4. Объектно-ориентированная модель предметной области: классы.

1. Общая форма определения класса.
2. Данные-члены.
3. Значение свойства типа class.
4. Класс АБОНЕНТ.
5. Класс КАТАЛОГ.

6. Класс ВЫДАЧА.

Тема 5. Программирование классов на С#: поля, константы и методы.

1. Данные: поля и константы.
2. Спецификатор Описание.
3. Спецификаторы полей и констант класса.
4. Обращение к полю класса.
5. Функции-члены.
6. Пометки констант (public, private, protected, internal, protected internal или private protected.).
7. Методы (method).

Тема 6. Программирование классов на С#: конструкторы и деструкторы.

1. Конструктор.
2. Конструктор с атрибутом static.
3. Деструктор.
4. Деструктор класса.
5. Деструктора класса Person.

Тема 7. Программирование классов на С#: свойства и индексаторы. Реализация принципа инкапсуляции.

1. Свойства класса.
2. Синтаксис описания свойства.
3. Индексаторы.
4. Параметры индексатора.
5. Модификаторы тип this.
6. Инкапсуляция.
7. Модификаторы доступа.
8. Реализация принципа инкапсуляции.

Тема 8. Реализация принципа наследования и полиморфизма на С#.

1. Принципы объектно-ориентированного программирования.
2. Наследование.
3. Понятия «предок - потомок».
4. Полиморфизм.
5. Реализация принципов наследования и полиморфизма.

Тема 9. Делегаты и события на С#. Многопоточные приложения.

1. Типы делегатов
2. DelegateArticle.FirstDelegate.
3. DelegateArticle.Sample.SecondDelegate.

4. Экземпляры делегатов.
5. Создание экземпляров делегатов.
6. Вызов экземпляров делегатов.
7. Комбинирование делегатов.
8. События. Field-like события.
9. Потокбезопасные события.
10. Многопоточность.
11. Класс Thread.
12. Получение информации о потоке.
13. Статус потока.
14. Приоритеты потоков.

4. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В преподавании дисциплины «Объектно-ориентированный анализ и проектирование» используются разнообразные образовательные технологии как традиционные, так и с применением активных и интерактивных методов обучения.

Активные и интерактивные методы обучения:

- Интерактивная лекция (темы № 1);
- Групповая дискуссия (темы № 2,3,4);
- Ролевые игры (темы № 5)
- Тренинг (темы № 7);
- Анализ ситуаций (темы № 6);
- Применение имитационных моделей (темы № 8);
- Разбор конкретных ситуаций (темы № 9).

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

Текущий контроль знаний студентов производится в дискретные временные интервалы преподавателем, ведущим практические работы по дисциплине, в следующих формах:

- тестирование;
- выполнение разного рода практических заданий;
- дискуссии;
- рейтинг-контроль.

Промежуточная аттестация знаний студентов производится по результатам работы в 4 семестре, в форме зачета, который включает в себя ответы на теоретические вопросы.

Фонды оценочных средств, включающие типовые задания и методы контроля позволяющие оценить знания по данной дисциплине, включены в состав УМКД.

Типовые тестовые задания для проведения текущего контроля приведены ниже.

Рейтинг-контроль №1

1. Из истории языков программирования.
2. Основные принципы структурной методологии.
3. Понятие объектно-ориентированного программирования.
4. Абстрактные типы данных.
5. Объекты и классы.
6. Базовые принципы ООП.
7. Пример объектной декомпозиции.
8. Описание объектного типа.
9. Иерархия типов (наследование). Полиморфизм и виртуальные методы.
10. Объектно-ориентированные модели.
11. Классические подходы к анализу объектно-ориентированных систем.
12. Анализ поведения.
13. Анализ предметной области.
14. Анализ вариантов.
15. CRC-карточки.
16. Неформальное описание.
17. Структурный анализ.
18. Стандарт ODMG.
19. Объект типа БИБЛИОТЕКА.
20. Объекты типа КНИГА.
21. Общая форма определения класса.
22. Данные-члены.
23. Значение свойства типа class.
24. Класс АБОНЕНТ.
25. Класс КАТАЛОГ.
26. Класс ВЫДАЧА.

Тестовые задания

1 Методика разработки программ, в основе которой лежит понятие объекта как не-какой структуры, описывающей объект реального мира, его поведение, - это ...

- 1) Объектно-ориентированное программирование
- 2) Объект
- 3) Инкапсуляция
- 4) Наследование
- 5) Полиморфизм

2 Некоторая часть окружающего нас мира, которая может быть рассмотрена как единое целое, - это ...

- 1) Объектно-ориентированное программирование.
- 2) Объект.
- 3) Инкапсуляция.
- 4) Наследование.
- 5) Полиморфизм.

3. Возможность скрыть внутреннее устройство объекта от его пользователей, предоставив через интерфейс доступ только к тем членам объекта, с которыми клиенту разрешается работать напрямую, - это ...

- 1) Объектно-ориентированное программирование.
- 2) Объект.
- 3) Инкапсуляция.
- 4) Наследование.
- 5) Полиморфизм.

4. Возможность при описании класса указывать на его происхождение от другого класса, - это ...

- 1) Объектно-ориентированное программирование.
- 2) Объект.
- 3) Инкапсуляция.
- 4) Наследование.
- 5) Полиморфизм.

5. Возможность объектов с одинаковой спецификацией иметь различную реализацию, - это ...

- 1) Объектно-ориентированное программирование.
- 2) Объект.
- 3) Инкапсуляция.
- 4) Наследование.
- 5) Полиморфизм

6. Полиморфизм – это ...

- 1) Возможность объектов с одинаковой спецификацией иметь различную реализацию.

- 2) Возможность при описании класса указывать на его происхождение от другого класса.
- 3) Возможность скрыть внутреннее устройство объекта от его пользователей, предоставив через интерфейс доступ только к тем членам объекта, с которыми клиенту разрешается работать напрямую.
- 4) Некоторая часть окружающего нас мира, которая может быть рассмотрена как единое целое.
- 5) Методика разработки программ, в основе которой лежит понятие объекта как некоторой структуры, описывающей объект реального мира, его поведение.

7. Наследование – это ...

- 1) Возможность объектов с одинаковой спецификацией иметь различную реализацию.
- 2) Возможность при описании класса указывать на его происхождение от другого класса.
- 3) Возможность скрыть внутреннее устройство объекта от его пользователей, предоставив через интерфейс доступ только к тем членам объекта, с которыми клиенту разрешается работать напрямую.
- 4) Некоторая часть окружающего нас мира, которая может быть рассмотрена как единое целое.
- 5) Методика разработки программ, в основе которой лежит понятие объекта как некоторой структуры, описывающей объект реального мира, его поведение

Рейтинг-контроль №2

1. Данные: поля и константы.
2. Спецификатор Описание.
3. Спецификаторы полей и констант класса.
4. Обращение к полю класса.
5. Функции-члены.
6. Пометки констант (public, private, protected, internal, protected internal или private protected.).
7. Методы (method).
8. Конструктор.
9. Конструктор с атрибутом static.
10. Деструктор.
11. Деструктор класса.
12. Деструктора класса Person.
13. Свойства класса.
14. Синтаксис описания свойства.

15. Индексаторы.
16. Параметры индексатора.
17. Модификаторы тип `this`.
18. Инкапсуляция.
19. Модификаторы доступа.
20. Реализация принципа инкапсуляции.

Тестовые задания

1. Инкапсуляция – это ...

- 1) Возможность объектов с одинаковой спецификацией иметь различную реализацию.
- 2) Возможность при описании класса указывать на его происхождение от другого класса.
- 3) Возможность скрыть внутреннее устройство объекта от его пользователей, предоставив через интерфейс доступ только к тем членам объекта, с которыми клиенту разрешается работать напрямую.
- 4) Некоторая часть окружающего нас мира, которая может быть рассмотрена как единое целое
- 5) Методика разработки программ, в основе которой лежит понятие объекта как некоторой структуры, описывающей объект реального мира, его поведение.

2. Объект – это ...

- 1) Возможность объектов с одинаковой спецификацией иметь различную реализацию.
- 2) Возможность при описании класса указывать на его происхождение от другого класса.
- 3) Возможность скрыть внутреннее устройство объекта от его пользователей, предоставив через интерфейс доступ только к тем членам объекта, с которыми клиенту разрешается работать напрямую.
- 4) Некоторая часть окружающего нас мира, которая может быть рассмотрена как единое целое.
- 5) Методика разработки программ, в основе которой лежит понятие объекта как некоторой структуры, описывающей объект реального мира, его поведение.

3. Объектно-ориентированное программирование – это ...

- 1) Присваивание определенному действию одного имени, которое затем совместно используется по всей иерархии объектов сверху донизу, причем каждый объект иерархии выполняет это действие характерным именно для него способом.
- 2) Определение объекта и дальнейшее использование всех его свойств для построения иерархии порожденных объектов с возможностью для каждого порожденного объекта, относящегося к иерархии, доступа к коду и данным всех порождающих объектов.

- 3) Объединение записей с процедурами и функциями, работающими с полями этих записей, которое формирует новый тип данных – объект.
- 4) Некоторая часть окружающего нас мира, которая может быть рассмотрена как единое целое.
- 5) Методика разработки программ, в основе которой лежит понятие объекта как некоторой структуры, описывающей объект реального мира, его поведение.

4. Именованные категории, позволяющие группировать сходные объекты, - это ...

- 1) Классы.
- 2) Объекты.
- 3) События.
- 4) Свойства.
- 5) Методы класса.

5. Отдельные, четко обозначенные экземпляры некоторого класса, - это ...

- 1). Классы.
- 2) Объекты.
- 3) События.
- 4) Свойства.
- 5) Методы класса.

Рейтинг-контроль №3

1. Принципы объектно-ориентированного программирования.
2. Наследование.
3. Понятия «предок - потомок».
4. Полиморфизм.
5. Реализация принципов наследования и полиморфизма.
6. Типы делегатов
7. DelegateArticle.FirstDelegate.
8. DelegateArticle.Sample.SecondDelegate.
9. Экземпляры делегатов.
10. Создание экземпляров делегатов.
11. Вызов экземпляров делегатов.
12. Комбинирование делегатов.
13. События. Field-like события.
14. Потокбезопасные события.

15. Многопоточность.
16. Класс Thread.
17. Получение информации о потоке.
18. Статус потока.
19. Приоритеты потоков.

Тестовые задания (образец)

1. Некоторые особые состояния, в которые может попадать объект, - это ...

- 1) Классы.
- 2) Объекты.
- 3) События.
- 4) Свойства.
- 5) Методы класса.

2. Характеристики объекта - это ...

- 1) Классы.
- 2) Объекты.
- 3) События.
- 4) Свойства.
- 5) Методы класса.

3. Процедуры и функции, объявление которых включено в описание класса, выполняющие действия над объектами класса, - это ...

- 1) Классы.
- 2) Объекты.
- 3) События.
- 4) Свойства.
- 5) Методы класса.

4. Методы класса - это ...

- 1) Процедуры и функции, объявление которых включено в описание класса, выполняющие действия над объектами класса.
- 2) Характеристики объекта.
- 3) Некоторые особые состояния, в которые может попадать объект.
- 4) Отдельные, четко обозначенные экземпляры некоторого класса.
- 5) Именованные категории, позволяющие группировать сходные объекты.

5. Свойства - это ...

- 1) Процедуры и функции, объявление которых включено в описание класса, выполняющие действия над объектами класса.

- 2) Характеристики объекта.
- 3) Некоторые особые состояния, в которые может попадать объект.
- 4) Отдельные, четко обозначенные экземпляры некоторого класса.
- 5) Именованные категории, позволяющие группировать сходные объекты.

6. События - это ...

- 1) Процедуры и функции, объявление которых включено в описание класса, выполняющие действия над объектами класса.
- 2) Характеристики объекта.
- 3) Некоторые особые состояния, в которые может попадать объект.
- 4) Отдельные, четко обозначенные экземпляры некоторого класса.
- 5) Именованные категории, позволяющие группировать сходные объекты.

7. Объекты – это ...

- 1) Процедуры и функции, объявление которых включено в описание класса, выполняющие действия над объектами класса.
- 2) Характеристики объекта.
- 3) Некоторые особые состояния, в которые может попадать объект.
- 4) Отдельные, четко обозначенные экземпляры некоторого класса.
- 5) Именованные категории, позволяющие группировать сходные объекты.

8. Классы – это ...

- 1) Процедуры и функции, объявление которых включено в описание класса, выполняющие действия над объектами класса.
- 2) Характеристики объекта.
- 3) Некоторые особые состояния, в которые может попадать объект.
- 4) Отдельные, четко обозначенные экземпляры некоторого класса.
- 5) Именованные категории, позволяющие группировать сходные объекты

Самостоятельная работа студентов

Самостоятельная работа студентов является неотъемлемой частью процесса изучения дисциплины. Она направлена на усвоение системы научных и профессиональных знаний, формирования умений и навыков, приобретение опыта самостоятельной творческой деятельности. СРС помогает формировать культуру мышления студентов, расширять познавательную деятельность.

Виды самостоятельной работы по курсу:

- а) по целям: подготовка к лекциям, к практическим занятиям, рейтингам.

б) по характеру работы: изучение конспекта лекций, выполнение практических/лабораторных заданий и тестов, организация круглых столов, подготовка докладов, презентаций.

Примерная тематика самостоятельной работы

Тема 1. Объектно – ориентированное программирование.

1. Из истории языков программирования.
2. Основные принципы структурной методологии.
3. Понятие объектно-ориентированного программирования.
4. Абстрактные типы данных.
5. Объекты и классы.
6. Базовые принципы ООП.

Тема 2. Объектно- ориентированный анализ.

1. Объектно-ориентированные модели.
2. Классические подходы к анализу объектно-ориентированных систем.
3. Анализ поведения.
4. Анализ предметной области.
5. Анализ вариантов.

Тема 3. Объектно-ориентированная модель предметной области: объекты.

1. Стандарт ODMG.

Тема 4. Объектно-ориентированная модель предметной области: классы.

1. Общая форма определения класса.
2. Данные-члены.

Тема 5. Программирование классов на С#: поля, константы и методы.

1. Данные: поля и константы.
2. Спецификатор Описание.
3. Спецификаторы полей и констант класса.

Тема 6. Программирование классов на С#: конструкторы и деструкторы.

1. Конструктор.
2. Деструктор.
4. Деструктор класса.

Тема 7. Программирование классов на С#: свойства и индексаторы. Реализация принципа инкапсуляции.

1. Свойства класса.
2. Индексаторы.
3. Модификаторы.

Тема 8. Реализация принципа наследования и полиморфизма на C#.

1. Принципы объектно-ориентированного программирования.

Тема 9. Делегаты и события на C#. Многопоточные приложения.

1. Типы делегатов

2. Потокобезопасные события.

3. Многопоточность.

4. Приоритеты потоков.

Примерная тематика заданий для самостоятельной работы

1. Информационная система обслуживания библиотеки. Разработать информационную систему обслуживания библиотеки, которая содержит информацию о книгах на бумажном носителе и книгах на электронном носителе, а именно: название, ФИО авторов, жанр, наименование издательства, телефон издательства, год издания, количество страниц, количество иллюстраций, стоимость, книжный формат, формат файла, объем файла, название и адрес книгохранилища, в котором находится книга, количество имеющихся в библиотеке экземпляров конкретной книги, количество студентов, которым выдавалась конкретная книга (востребованность), названия и телефоны деканатов факультетов, в учебном процессе которых используется указанная книга. Одна и та же книга одновременно в библиотеки может быть представлена как на бумажном, так и электронном носителе.

2. Информационная система музыкального магазина. Разработать информационную систему музыкального магазина, которая содержит информацию о музыкантах, музыкальных произведениях и обстоятельствах их исполнения. Несколько музыкантов, образующих единый коллектив, называются ансамблем. Это может быть классический оркестр, джазовая группа, квартет, квинтет и т.д. К музыкантам причисляют исполнителей (играющих на одном или нескольких инструментах), композиторов, дирижеров и руководителей ансамблей. Кроме того, в базе данных хранится информация о пластинках, которыми магазин торгует. Каждая пластинка, а точнее, ее наклейка, идентифицируется отдельным номером, так что всем копиям, отпечатанным с матрицы в разное время, присвоены одинаковые номера. На пластинке может быть записано несколько исполнений одного и того же произведения для каждого из них в базе заведена отдельная запись. Когда выходит новая пластинка, регистрируется название выпускавшей ее компании (например, EMI), а также адрес оптовой фирмы, у которой магазин может приобрести эту пластинку. Не исключено, что компания производитель занимается и оптовой продажей своих пластинок. Магазин фиксирует текущие оптовые и розничные цены на каждую пластинку,

дату ее выпуска, количество экземпляров, проданных за прошлый год и в нынешнем году, а также число еще не распроданных пластинок.

3. Информационная система обслуживания работы склада. Разработать информационную систему обслуживания работы склада (автоматизация складского учета), которая содержит: 1) информацию о единицах хранения (номер ордера, дата, код поставщика, балансный счет, код сопроводительного документа по справочнику документов, номер сопроводительного документа, код материала по справочнику материалов, количество пришедшего материала, цена единицы измерения); 2) информация о хранящихся на складе материалах (справочник материалов – код класса материала, код группы материала, наименование материала, код единицы измерения); 3) информация о единицах измерения конкретных видов материалов – код материала, единица измерения (метры, килограммы, литры и т.д.); 4) информация о поставщиках материалов – код поставщика, его наименование, ИНН, юридический адрес (индекс, город, улица, дом), адрес банка (индекс, город, улица, дом), номер банковского счета.

4. Информационная система обслуживания работы конференции. Разработать информационную систему обслуживания работы конференции, которая содержит: 1) справочник персоналий участников конференции (фамилия, имя, отчество, ученая степень, ученое звание, научное направление, место работы, кафедра (отдел), должность, страна, город, почтовый индекс, адрес, рабочий телефон, домашний телефон, e-mail); 2) информацию, связанную с участием в конференции (докладчик или участник, дата рассылки 1-го приглашения, дата поступления заявки, тема доклада, отметка о поступлении тезисов, дата рассылки 2-го приглашения, дата поступления оргвзноса, размер поступившего оргвзноса, дата приезда, дата отъезда, потребность в гостинице).

5. Информационная система регистрации происшествий. Разработать информационную систему регистрации происшествий, которая содержит: 1) данные для регистрации сообщений о происшествиях (регистрационный номер сообщения, дата регистрации, краткая фабула (тип происшествия), пострадавшее лицо, подозреваемое в совершении преступления лицо; 2) информацию о принятом по происшествию решении (отказано в возбуждении дел, удовлетворено ходатайство о возбуждении уголовного дела с указанием регистрационного номера заведенного дела, отправлено по территориальному признаку); 3) информацию о лицах, виновных или подозреваемых в совершении происшествия (регистрационный номер лица, фамилия, имя, отчество, адрес, количество судимостей), отношение конкретных лиц к конкретным происшествиям (виновник, потерпевший, подозреваемый, свидетель); 4) информацию о лицах, пострадавших при совершении происшествия (фамилия, имя, отчество, адрес)).

6. Информационная система учета успеваемости студентов. Разработать информационную систему учета успеваемости студентов, которая содержит данные: 1) о контингенте студентов (фамилия, имя, отчество, год поступления, форма обучения (дневная/вечерняя/заочная), номер или название группы); 2) об учебном плане (название специальности, дисциплина, семестр, количество отводимых на дисциплину часов, форма отчетности (экзамен/зачет)); 3) о журнале успеваемости студентов (семестр, дата сдачи, студент, дисциплина, оценка).

7. Информационная система учета аудиторного фонда университета. Разработать информационную систему учета аудиторного фонда университета, которая содержит данные: 1) об аудиторном фонде университета (наименование корпуса, в котором расположено помещение, номер комнаты, расположение комнаты в корпусе, ширина и длина комнаты в метрах, назначение и вид помещения, подразделение университета, за которым закреплено помещение.); 2) о подразделении университета (институт, факультет, кафедра - следует также учитывать, что структура подразделений университета имеет иерархический вид, когда одни подразделения входят в состав других).

8. Информационная система медицинской клиники. Разработать информационную систему медицинской клиники, которая содержит данные: 1) о пациентах (имя, пол, дата рождения, домашний адрес, номер медицинской страховки); 2) о врачах (фамилия, имя, отчество, кабинет, медицинская специализация); 3) о приёмах пациентов (дата и место, где проводится осмотр, симптомы, диагноз и предписания больному, пациент, врач); 4) о выписанных рецептах (дата, пациент, врач, название лекарства, способ его приема, словесное описание предполагаемого действия и возможных побочных эффектов).

9. Информационная система Городской Думы. Разработать информационную систему Государственной Думы, которая содержит данные: 1) о депутатах (имена, адреса, домашние и служебные телефоны, номер округа избрания, политическая партия); 2) о депутатских комиссиях (наименование, профиль (например, вопросы образования, проблемы, связанные с жильем и так далее), дата создания); 3) о составе депутатских комиссий (депутат, дата включения и выхода из состава комиссии); 4) о законопроектах (номер, дата поступления, наименование, авторы (депутаты), раздел права (например, семейное право, трудовое право и т.д.), статус (новый, подготовлен 1 чтению, вынесен на 1 чтение, принят в 1 чтении и т.д.)).

10. Информационная система учета аренды площадей торговых помещений. Разработать информационную систему учета аренды площадей торговых помещений в торговом центре, которая содержит данные: 1) о торговых помещениях (этаж, номер, площадь, стоимость аренды за месяц); 2) об арендаторах (наименование, ИНН, КПП, юридический адрес, почтовый адрес, номер банковского счета, банк арендатора); 3) о сделках

по сдаче в аренду помещений (период аренды - год, месяц, арендуемая площадь торгового помещения, арендатор, отметка об оплате).

Вопросы к зачету.

1. Абстрагирование как составная часть объектного подхода.
2. Инкапсуляция как составная часть объектного подхода.
3. Модульность как составная часть объектного подхода.
4. Иерархия как составная часть объектного подхода.
5. Типизация как составная часть объектного подхода.
6. Параллелизм как составная часть объектного подхода.
7. Сохраняемость как составная часть объектного подхода.
8. Сущность объектно-ориентированной модели предметной области.
9. Объектно-ориентированный анализ.
10. Язык UML.
11. Основные элементы UML языка UML.
12. Общие и специальные диаграммы.
13. Понятие объекта.
14. Жизненный цикл объекта.
15. Состояние как свойство, присущее объектам.
16. Поведение как свойство, присущее объектам.
17. Идентичность как свойство, присущее объектам.
18. Типы отношений между объектами: ассоциация и агрегация.
19. Природа классов.
20. Ассоциация как тип отношений между классами.
21. Агрегация как тип отношений между классами.
22. Использование как тип отношений между классами.
23. Наследование как тип отношений между классами.
24. Инстанцирование как тип отношений между классами.
25. Отношения между классами и объектами
26. Структура объявления класса.
27. Доступ к членам класса.
28. Поля данных класса как механизм реализации состояния объекта.
29. Методы члены класса как механизм реализации поведения объекта.
30. Спецификаторы доступа для обеспечения инкапсуляции.
31. Статические поля и методы классов. Инициализация статических полей.
32. Средства управления жизнью объекта.
33. Конструкторы и деструкторы.

34. Особенности использования конструктора по умолчанию.
35. Конструкторы и деструкторы: совмещение имен методов при наследовании, иерархии.
36. Реализация отношений между объектами и классами.
37. Свойства: понятие, методы записи и чтения.
38. Индексаторы.
39. Инкапсуляция: понятие, способы реализации принципа инкапсуляции.
40. Наследование как средство организации иерархий классов замещения.
41. Понятие производного класса.
42. Управление доступом в производных классах.
43. Абстрактные классы и виртуальные функции
44. Вложенные типы (классы)
45. Полиморфизм.
46. Проблема множественного наследования. Понятие интерфейса.
47. Делегаты: понятие, описание, использование, операции с делегатами
48. Использование делегатов для получения возможности определять вызываемый метод динамически во время выполнения программы.
49. Использование делегатов для обеспечения связи между объектами по типу «источник - наблюдатель».
50. Использование делегатов для создания универсальных методов.
51. Использование делегатов для поддержки механизма обратных вызовов.
52. События: понятие, описание, обработчики событий, стандартный класс EventArgs.
53. Многопоточные приложения: понятие, назначение, организация многопоточных приложений.
54. Потoki: создание и использование, методы управления, методы синхронизации.
55. Потoki: методы синхронизации.
56. Асинхронные делегаты: понятие, назначение, порядок применения.

Примерный перечень практических заданий на зачете

1. Описать функциональные возможности существующих онлайн-сервисов (по вариантам) и создать на основе данного описания диаграмму прецедентов.
2. Описать реализацию одной функциональных возможностей существующих онлайн-сервисов (по вариантам).
3. Составить диаграмму классов и диаграммы внутренней структуры по предложенному тексту программы.
4. Создать класс, имеющий два поля и три метода:
 - 1) конструктор для инициализации объекта;

2) функция формирования строки с информацией об объекте;

3) функция обработки значений полей (по варианту).

5. Создать класс с полями (по вариантам). Реализовать в классе методы:

1) конструктор по умолчанию;

2) конструктор перезагрузки с параметрами;

3) деструктор для освобождения памяти (с сообщением об уничтожении объекта);

4) функции-методы обработки данных;

5) функцию формирования строки информации об объекте.

6. Создать класс, имеющий два поля и три свойства:

1) два свойства для реализации принципа инкапсуляции при чтении и записей значений полей;

2) свойство для чтения результата обработки значений полей (по варианту).

7. Для класса создать класс-потомок с дополнительным полем (по вариантам). Реализовать в классе-потомке методы: конструктор; функцию обработки данных, указанную в индивидуальном задании. Продемонстрировать ввод и вывод информации об объектах: классе-родителе и классе-потомке.

8. Создать родительский класс с определенными полями и методами (по вариантам):

1) конструктор;

2) функция, которая определяет «качество» объекта по заданной формуле;

3) вывод информации об объекте.

9. Построить класс-потомок, который содержит:

1) дополнительное поле (P);

2) функцию, которая определяет «качество» объекта класса-потомка, которая перекрывает функцию качества родительского класса, выполняя вычисление по новой формуле. Продемонстрировать ввод и вывод информации об объектах класса-родителя и класса-потомка.

10. Написать программу согласно заданию (по вариантам). Во всех классах описать необходимые конструкторы, при помощи которых будут создаваться объекты классов. Параметры создаваемых объектов задавать константами или вводить с клавиатуры/формы (по желанию студента) и передавать в конструкторы объектов в виде параметров. Вывод информации должен осуществляться на консоль или в многострочное текстовое поле.

11. Составить диаграмму автомата по предложенному тексту многопоточного приложения.

12. Составить диаграмму деятельности по предложенному тексту многопоточного приложения.

13. Составить диаграмму взаимодействия по предложенному тексту многопоточно-го приложения

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1. Книгообеспеченность

Наименование литературы: автор, название, вид издания, издательство	Год изда- ния	КНИГООБЕСПЕЧЕННОСТЬ	
		Количество экземпляров изданий в библиотеке ВлГУ в соответствии с ФГОС ВО	Наличие в электронной биб- лиотеке ВлГУ
1	2	3	4
Основная литература*			
1 Объектно-ориентированное программирование с примерами на C# : Учебное пособие / Хорев П.Б. - М.: Форум, НИЦ ИНФРА-М, 2016. - 200 с.: 70x100 1/16. - (Высшее образование: Бакалавриат) (Обложка) ISBN 978-5-00091-144-0	2016		http://znanium.com/catalog/product/529350
2. Гуськова, О.И. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. - Москва : МПГУ, 2018. - 240 с. - ISBN 978-5-4263-0648-6..	2018		http://znanium.com/catalog/product/1020593
3 Ашарина, И.В. Объектно-ориентированное программирование в C++ : лекции и упражнения : учеб. пособие для вузов / И.В. Ашарина. - 2-е изд., перераб. и доп. - Москва : Горячая линия - Телеком, 2017. - 336 с. - ISBN 978-5-9912-0423-1.	2017		https://new.znanium.com/catalog/product/1040247
Дополнительная литература			
1 Дудецкий, В.Н. Объектно-ориентированные языки программирования : учеб. пособие : в 3 ч. Ч. I / В.Н. Дудецкий. — 2-е изд., стер. — Москва : ФЛИНТА, 2016. - 48 с. - ISBN 978-5-9765-2252-7.	2016		http://znanium.com/catalog/product/1037567
2 Дудецкий, В.Н. Объектно-ориентированные языки программирования : учеб. пособие : в 3 ч. Ч. II / В.Н. Дудецкий. — 2-е изд., стер. — Москва : ФЛИНТА, 2016. — 40 с. - ISBN 978-5-9765-2253-4.	2016		http://znanium.com/catalog/product/1037569
3 Программирование на языке высокого уровня. Программирование на языке C++ : учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагаиной. — М. : ИД «ФОРУМ» : ИНФРА-М, 2018. — 512 с.	2018		http://znanium.com/catalog/product/918098

7.2. Периодические издания

Журнал БИЗНЕС-ИНФОРМАТИКА. ISSN печатной версии 1998-0663

7.3. Интернет-ресурсы


1. <http://www.spssbase.com/> Иллюстрированный самоучитель по SPSS
2. <http://www.spss.ru> Официальный сайт российского офиса компании SPSS

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации данной дисциплины имеются специальные помещения для проведения занятий лекционного типа, занятий практического/лабораторного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы: аудитории, оснащенные мульти-медиа оборудованием, компьютерные классы с доступом в интернет, аудитории без спец. оборудования.

Компьютерная техника, используемая в учебном процессе, имеет лицензионное программное обеспечение:

- Операционная система семейства MicrosoftWindows.
- Пакет офисных программ MicrosoftOffice.
- Консультант+.

Рабочую программу составил  к.ф.-м.н., доцент Крылов В.Е.

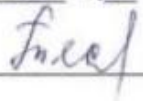
Рецензент:

Генеральный директор

ООО «Хрустальное небо»  Козырев В.Н.

Программа рассмотрена и одобрена на заседании кафедры БИЭ

протокол № 1 от «31» августа 2020 года.

Заведующий кафедрой  д.э.н., профессор Тесленко И.Б.

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 38.03.05 Бизнес-информатика

протокол № 1 от «31» августа 2020 года.

Председатель комиссии  д.э.н., профессор Тесленко И.Б.

**ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ
РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ (МОДУЛЯ)**

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 _____

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

в рабочую программу дисциплины

ЦИФРОВАЯ ЭКОНОМИКА

образовательной программы направления подготовки 38.03.05 «Бизнес-информатика»,

программа подготовки «Информационно-аналитическое обеспечение

предпринимательской деятельности»

Номер изменения	Внесены изменения в части/разделы рабочей программы	Исполнитель ФИО	Основание (номер и дата протокола заседания кафедры)
1			
2			

Заведующий кафедрой БИЭ _____ д.э.н., профессор Тесленко И.Б.