

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ 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

подпись

Т.А. Хагуров

«28» мая 2021 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.01 «ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ МОДЕЛИ»

Направление подготовки 01.04.02 Прикладная математика и информатика

Профиль Математическое моделирование в естествознании и технологиях

Форма обучения Очная

Квалификация выпускника Магистр

Краснодар 2021

Рабочая программа дисциплины «Объектно-ориентированные модели» составлена в соответствии с Федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 01.04.02 Прикладная математика и информатика профиль Математическое моделирование в естествознании и технологиях.

Программу составил(и):
Г.А. Кесиян, доцент

подпись

Рабочая программа дисциплины утверждена на заседании кафедры анализа данных и искусственного интеллекта протокол № 8 от «21» мая 2021г.

Заведующий кафедрой Коваленко А.В.

подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 1 «21» мая 2021г.

Председатель УМК факультета Коваленко А.В.

подпись

Рецензенты:

Шапошникова Татьяна Леонидовна.

Доктор педагогических наук, кандидат физико-математических наук, профессор. Почетный работник высшего профессионального образования РФ. Директор института фундаментальных наук (ИФН) ФГБОУ ВО «КубГТУ».

Марков Виталий Николаевич.

Доктор технических наук. Профессор кафедры информационных систем и программирования института компьютерных систем и информационной безопасности (ИКСиИБ) ФГБОУ ВО «КубГТУ».

1 Цели и задачи изучения дисциплины (модуля).

1.1 Цель освоения дисциплины.

Целью освоения учебной дисциплины «Объектно-ориентированные модели» является развитие профессиональных компетентностей приобретения практических навыков проектирования и реализации программных комплексов в рамках объектно-ориентированной парадигмы.

1.2 Задачи дисциплины.

- углубленное изучение объектно-ориентированного программирования;
- приобретение навыков объектно-ориентированного проектирования и методов повторного использования кода;
- овладение технологиями для реализации и сопровождения программ.

1.3 Место дисциплины (модуля) в структуре образовательной программы.

Дисциплина «Объектно-ориентированные модели» относится к вариативной части Блока 1 "Дисциплины" учебного плана.

Данная дисциплина (Объектно-ориентированные модели) тесно связана со следующими дисциплинами: технологии проектирования и сопровождения программных систем.

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

Изучение данной учебной дисциплины направлено на формирование у обучающихся общекультурных и общепрофессиональных компетенций

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
1.	ПК-3	Способен эффективно применять алгоритмические и программные решения в области информационно-коммуникационных технологий, а также участвовать в их проектировании и разработке	подробную структуру классов в ООП, механизмы повторного использования и приемов объектно-ориентированного проектирования, Венгерскую нотацию, нотации диаграмм классов, объектов и взаимодействия	применять современные среды разработки, обосновано выбирать механизмы повторного использования	визуальными средствами разработки, навыками анализа различных архитектурных решений для гибкого проектирования интерфейсов

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
			й		
2.	ПК-6	Способен эффективно определять компонентный состав и архитектуру программного обеспечения или программно-аппаратного комплекса в соответствии с его назначением, осуществлять выбор современных оптимальных технологий и средств разработки и сопровождения	методологию применения паттернов проектирования, такие технологии как рефлексии типов, динамическая загрузка библиотек, позднее связывание, бинарная сериализация и сериализация в XML, LinqToObjects	осуществлять отладку разрабатываемых приложений, разрабатывать пользовательские элементы управления, конфигурировать приложения с помощью внешних бинарных файлов и XML	встроенными библиотеками средств разработки, элементами управления и технологиями компоновки интерфейса, дополнительными технологиями удобного сопровождения программ

2. Структура и содержание дисциплины.

2.1 Распределение трудоёмкости дисциплины по видам работ.

Общая трудоёмкость дисциплины составляет 2 зач.ед. (72 часа), их распределение по видам работ представлено в таблице (для студентов ОФО).

Вид учебной работы	Всего часов	Семестры (часы)			
		2			
Контактная работа, в том числе:	20,3	20,3			
Аудиторные занятия (всего):	20	20			
Занятия лекционного типа	10	10	-	-	-
Лабораторные занятия	10	10	-	-	-
Занятия семинарского типа (семинары, практические занятия)	-	-	-	-	-
	-	-	-	-	-
Иная контактная работа:	0,3	0,3			
Контроль самостоятельной работы (КСР)					
Промежуточная аттестация (ИКР)	0,3	0,3			
Самостоятельная работа, в том числе:	16	16			
Курсовая работа	-	-	-	-	-
Проработка учебного (теоретического) материала	6	6	-	-	-
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)	10	10	-	-	-
Реферат	-	-	-	-	-

Подготовка к текущему контролю		-	-	-	-	-
Контроль:						
Подготовка к экзамену		35,7	35,7	-	-	-
Общая трудоемкость	час.	72	72	-	-	-
	в том числе контактная работа	20,3	20,3			
	зач. ед	2	2			

2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы дисциплины, изучаемые в семестре 2 (очная форма).

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
	Модуль 1 Основные языковые структуры для ООП					
1.	Основные понятия в ООП. Подробная структура классов.	3	1		1	1
2.	Системы обозначений: Венгерская нотация, нотации диаграмм классов, объектов и взаимодействий.	3	1		1	1
3.	Разработка пользовательских элементов управления.	4	1		1	2
	Модуль 2 Механизмы объектно-ориентированного проектирования					
4.	Общие сведения о паттернах проектирования: описание и классификация. Паттерн Одиночка (Singleton).	4	1		1	2
5.	Механизмы повторного использования: наследование, композиция, делегирование, параметризованные типы.	4	1		1	2
6.	Паттерны проектирования: Стратегия (Strategy), Шаблонный метод (Template Method), Адаптер объектов (Adapter), Итератор (Iterator), Компоновщик (Composite).	4	1		1	2
	Модуль 3 Дополнительные технологии разработки и сопровождения приложений					
7.	Рефлексии типов, динамическая загрузка библиотек, позднее связывание.	4	1		1	2
8.	Конфигурирование приложений без перекомпиляции проекта. Технологии бинарной (де)сериализации и (де)сериализации в XML.	4	1		1	2
9.	Основы LINQ. Работа с коллекциями с помощью LINQ to Objects.	3	1		1	1
10.	Основы Front-end разработки с применением технологий Window Forms и WPF (Windows Presentation Foundation)	3	1		1	1
	Всего по разделам дисциплины:					

	Промежуточная аттестация (ИКР)	0,3				
	Контроль самостоятельной работы (КСР)					
	Подготовка к текущему контролю	35,7				
	<i>Итого по дисциплине:</i>	72	10		10	16

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 языковые структуры для ООП	<i>Тема 1. Основные понятия в ООП. Подробная структура классов.</i> Классы, объекты, экземпляры классов, поля, свойства, методы, конструкторы. <i>Тема 2. Системы обозначений: Венгерская нотация, нотации диаграмм классов, объектов и взаимодействий.</i> Обозначения идентификаторов с применением Венгерской нотации, представление программной архитектуры с применением диаграмм классов, объектов и взаимодействий. <i>Тема 3. Разработка пользовательских элементов управления.</i> Базовые классы для разработки пользовательских элементов управления, переопределение требуемых методов отрисовки, взаимодействия с мышью и клавиатурой, создание нужных свойств, методов и событий.	Резюме, аналитический обзор
2.	Механизмы объектно-ориентированного проектирования	<i>Тема 4. Общие сведения о паттернах проектирования: описание и классификация. Паттерн Одиночка (Singleton).</i> Представление удачных архитектурных решений в виде шаблонов проектирования, классификация и обозначения паттернов, пример реализации паттерна Одиночка. <i>Тема 5. Механизмы повторного использования: наследование, композиция, делегирование, параметризованные типы.</i> Принципы объектно-ориентированного проектирования, преимущества использования ссылок на интерфейсы и абстрактные классы, получение дополнительного функционала с помощью механизмов наследования, композиции, делегирования и параметризованных типов. <i>Тема 6. Паттерны проектирования: Стратегия (Strategy), Шаблонный метод (Template Method), Адаптер объектов (Adapter), Итератор (Iterator), Компоновщик (Composite).</i> Изучение и	Опрос по результатам индивидуального задания

		реализация некоторых паттернов проектирования (порождающих, структурирующих и поведения) по основным пунктам их описания.	
3.	Дополнительные технологии разработки и сопровождения приложений	<i>Тема 7. Рефлексии типов, динамическая загрузка библиотек, позднее связывание.</i> Получение сведений о типах, полях, свойствах и методах посредством рефлексий, получение сведений о типах внутри внешних библиотек с помощью динамической загрузки, создание экземпляров определенного типа и вызов его членов во время выполнения с применением позднего связывания. <i>Тема 8. Конфигурирование приложений без перекомпиляции проекта. Технологии бинарной (де)сериализации и (де)сериализации в XML.</i> Разработка учебной архитектуры классов для отрисовки геометрических примитивов и конфигурирование их свойств посредством классов пространства имен System.Xml. Сериализация в XML с помощью XmlSerializer, сериализация коллекций в XML. Запись и чтение бинарных файлов с помощью классов BinaryWriter и BinaryReader. Сериализация бинарных файлов посредством класса BinaryFormatter, сериализация коллекций в бинарные файлы. <i>Тема 9. Основы LINQ. Работа с коллекциями с помощью LINQ to Objects.</i> Создание учебной архитектуры классов Клиенты-Поставщики. Изучение трех этапов для применения LINQ to Objects: инициализация и наполнение объектов, создание запроса, выполнение запроса и получение результатов. Фильтрация данных с помощью команд Contains(), StartsWith(), логических операторов сравнения. Создание запросов с группировкой данных. <i>Тема 10. Основы Front-end разработки с применением технологий Window Forms и WPF (Windows Presentation Foundation).</i> Примеры реализации настольных приложений типа Windows Forms Application. Примеры реализации настольных приложений типа WPF Application.	Резюме, аналитический обзор

2.3.2 Занятия семинарского типа.

Занятия семинарского типа - не предусмотрены.

2.3.3 Лабораторные занятия.

№	Наименование лабораторных работ	Форма текущего контроля
---	---------------------------------	-------------------------

1	3	4
1.	Классы и объекты.	Проверка выполнения лабораторной работы № 1
2.	Написание программного кода по заданной архитектуре классов.	Проверка выполнения лабораторной работы № 2
3.	Разработка пользовательского элемента управления.	Проверка выполнения лабораторной работы № 3
4.	Модификация паттерна Одиночка – пул одиночек.	Проверка выполнения лабораторной работы № 4
5.	Создание диаграммы классов по программному коду.	Проверка выполнения лабораторной работы № 5
6.	Реализация паттерна Компоновщик с красивым выводом иерархической структуры.	Проверка выполнения лабораторной работы № 6
7.	Разработка оконного приложения для получения сведений о типах во внешней библиотеке и вызова методов.	Проверка выполнения лабораторной работы № 7
8.	Расширение функционала приложения с геометрическими примитивами.	Проверка выполнения лабораторной работы № 8
9.	Разработка архитектуры классов для хранения заданной структуры данных с применением нескольких паттернов проектирования.	Проверка выполнения лабораторной работы № 9
10.	Создание оконного приложения на платформе WPF с применением механизма привязок.	Проверка выполнения лабораторной работы № 10

2.3.4 Примерная тематика курсовых работ (проектов)

Курсовые работы - не предусмотрены.

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

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Разработка пользовательских элементов управления.	Microsoft – Официальная страница / Элементы управления Windows Forms [Электронный ресурс]. - Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/controls/ Сайт о программировании / Элементы управления в Windows Forms Visual C# [Электронный ресурс]. - Режим доступа: https://metanit.com/sharp/windowsforms/4.1.php
2	Общие сведения о паттернах проектирования: описание и классификация. Паттерн Одиночка (Singleton).	Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Электронный ресурс] : справочник / Гамма Э., Хелм Р., Джонсон Р. [и др.]. — Электрон. дан. — М. : ДМК Пресс, 2007. — 376 с. — Режим доступа:

		http://e.lanbook.com/books/element.php?pl1_id=1220 — Загл. с экрана. Справочник «Паттерны проектирования». [Электронный ресурс]. - Режим доступа: http://design-pattern.ru .
3	Механизмы повторного использования: наследование, композиция, делегирование, параметризованные типы.	Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования. [Электронный ресурс] : справочник / Гамма Э., Хел Р., Джонсон Р. [и др.]. — Электрон. дан. — М. : ДМ Пресс, 2007. — 376 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=1220 - Загл. с экрана.
4	Паттерны проектирования: Стратегия (Strategy), Шаблонный метод (Template Method), Адаптер объектов (Adapter), Итератор (Iterator), Компоновщик (Composite).	Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования. [Электронный ресурс] : справочник / Гамма Э., Хел Р., Джонсон Р. [и др.]. — Электрон. дан. — М. : ДМ Пресс, 2007. — 376 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=1220 - Загл. с экрана. Рефакторинг и Паттерны проектирования [Электронный ресурс]. - Режим доступа: https://refactoring.guru/ru
5	Рефлексии типов, динамическая загрузка библиотек, позднее связывание.	Павловская Т.А. С#. Программирование на языке высокого уровня : учебник для вузов / Павловская, Татьяна Александровна ; Т. А. Павловская. - Санкт-Петербург [и др.] : Питер, 2014. - 432 с. : ил. - (Учебник для вузов). - Библиогр.: с. 425-426. - ISBN 9785496008617.
6	Конфигурирование приложений без перекомпиляции проекта. Технологии бинарной (де)сериализации и (де)сериализации в XML.	Павловская Т.А. С#. Программирование на языке высокого уровня : учебник для вузов / Павловская Татьяна Александровна ; Т. А. Павловская. - Санкт-Петербург [и др.] : Питер, 2014. - 432 с. : ил. - (Учебник для вузов). - Библиогр.: с. 425-426. - ISBN 9785496008617.

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,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

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,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии.

Используются как традиционные информационно-объяснительные лекции, так и интерактивная подача материала с мультимедийной системой и проведение лекции-визуализации.

Также применяется регламентированная дискуссия и мозговой штурм.

Все материалы сопровождаются разбором практических 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.

Примерные задания на лабораторные работы

1 Классы и объекты.

Разработать консольное приложение в Visual Studio (C#) со следующими классами.

- Создать класс Vehicle (Транспортное средство) со свойствами:
 - Модель – строка;
 - Год выпуска – целое число.
- Создать класс Car (автомобиль) – наследник от Vehicle. С дополнительным свойством:
 - Цвет – строка.
- Создать класс Plane (самолет) – наследник от Vehicle. С дополнительным свойством:
 - Максимальная высота полета – вещественное число.
- Создать коллекцию класса Vehicle, добавить в нее экземпляры классов Car и Plane.
- Вывести все свойства объектов в коллекции.

2 Написание программного кода по заданной архитектуре классов.

Напишите код на C# согласно следующим диаграммам классов:

Диаграмма 1:

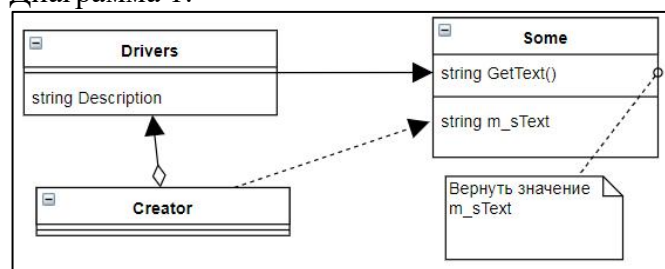
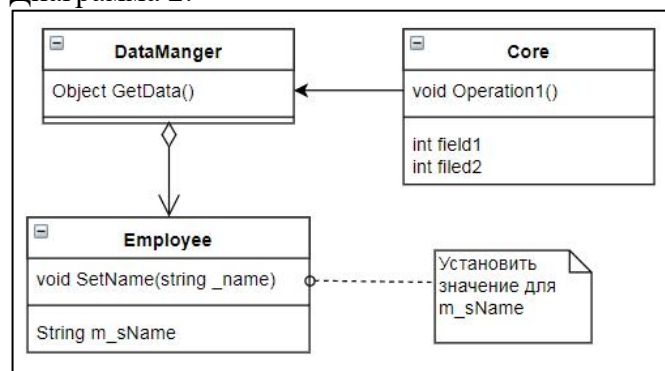


Диаграмма 2:



3 Разработка пользовательского элемента управления.

Задание 1.

Реализовать визуальный компонент «Кнопка» («Button»). Обязательные свойства компонента:

- изображение фона;
- изображение фона, когда кнопка зажата;
- текст;
- событие Click.

Задание 2.

Реализовать визуальный компонент «Выпадающий список» («ComboBox»).

Обязательные свойства компонента:

- количество элементов;
- номер текущего элемента.

Обязательные методы:

- добавить элемент;
- удалить элемент по номеру;
- отчистить список элементов.

Задание 3.

Реализовать визуальный компонент «Будильник» («Clock»). Обязательные свойства компонента:

- текущее время;
- будильник.

Обязательные методы:

- включить;
- выключить;
- звонок.

Задание 4.

Реализовать визуальный компонент «Прогресс» («ProgressBar»). Обязательные свойства компонента:

- минимальное значение;
- максимальное значение;
- текущее значение;
- ширина области отображения прогресса.

4 Модификация паттерна Одиночка – пул одиночек.

Реализовать модификацию паттерна Singleton, которая:

1. Допускает создавать не более заданного количества экземпляров данного класса (например, не более 3-х).
2. Предоставляет точку доступа к каждому из вышеуказанных экземпляров.

5 Создание диаграммы классов по программному коду.

Задание 1. Нарисовать диаграмму классов по следующему коду.

```
public class Department
{
    private List<Employee> m_Employees = null;
    public Department()
    { Employees = new List<Employee>(); }
    public List<Employee> Employees
    {
        get { return m_Employees; };
        set { m_Employees = value; }
    }
}
```

```

    }
    public class Employee : People
    {
        public float Salary { get; set; }}
    public class People
    {
        public String Name { get; set; }
        public int Age { get; set; }
    }

```

Задание 2. Нарисовать диаграмму классов по следующему коду.

```

public class Fabrica
{
    public Product CreateProductA(string _name)
    { return new ProductA (_name);}
}
public class ProductA : Product
{
    public ProductA (string _name)
    { Name = _name;}
}
public abstract class Product
{
    private string m_sName = "";
    public string Name
    {
        get { return m_sName;}
        set {m_sName = value;}
    }
}

```

6 Реализация паттерна Компоновщик с красивым выводом иерархической структуры.

Создать консольное приложение, в котором реализовать паттерн Компоновщик со следующей иерархической структурой:

- имеются абстрактные директории Directory;
- класс File – наследник Directory, является листом иерархии;
- класс Folder - наследник Directory, агрегирует коллекцию Directory.

Создать и наполнить произвольную иерархию вышеуказанных объектов и организовать красивый вывод иерархии в консоль.

7 Разработка оконного приложения для получения сведений о типах во внешней библиотеке и вызова методов.

Разработать приложение Windows-Forms, которое:

1. Загружает динамически сборку (библиотеку классов).
2. Выводит информацию о ее типах (классах) в следующем формате:

```

***** Имя_Типа *****

```

```

----- Methods -----

```

```

-->Возвр_Тип Имя_метода1(Тип_арг arg1, Тип_арг arg2,... Тип_арг argN1)

```

```

-->Возвр_Тип Имя_метода2(Тип_арг arg1, Тип_арг arg2,... Тип_арг argN2)

```

```

...

```

```

-->Возвр_Тип Имя_методаK(Тип_арг arg1, Тип_арг arg2,... Тип_арг argNL)

```

```

----- Properties -----

```

```

-->Тип Имя_свойства1 {[get;] [set;]}

```

```

-->Тип Имя_свойства2 {[get;] [set;]}

```

```

...

```

-->Тип Имя_свойстваM {[get;] [set;]}

3. Позволяет ввести команду в формате:

Имя_Тип.Имя_Метода(Знач_арг1, Знач_арг2, ..., Знач_аргZ)

- Если метод имеет тип возврата Void, то выводить «--> Results: ok.»
- Если метод имеет тип возврата отличный от Void, то выводить этот результат в формате: «--> Results: Знач_Результата»
- Информировать о неверности формата введенной команды.

4. Оформление пользовательского интерфейса осуществляется учащимся самостоятельно в индивидуальной форме.

8 Расширение функционала приложения с геометрическими примитивами.

В созданном на занятиях приложении, в котором разработана архитектура классов для отрисовки геометрических примитивов и конфигурирование их свойств посредством классов пространства имен System.Xml:

- добавить примитивы: линия и треугольник;
- реализовать в добавленных примитивах необходимые методы DetectCollision, Init;
- скорректировать методы OnPaint и Init класса Canvas;
- скорректировать методы для импорта и экспорта XML главной формы;
- реализовать перетаскивание примитивов мышью.

9 Разработка архитектуры классов для хранения заданной структуры данных с применением нескольких паттернов проектирования.

Создать консольное приложение с архитектурой классов для хранения конструкций:

- базовый класс Конструкция (Шифр, Название, Начало эксплуатации);
- класс Свая: Материал сваи, способ погружения, диаметр;
- класс Блок-Бокс: параметры подполья числа VRz, VTin, VR0;
- класс СОУ: площадь конденсатора, термическое сопротивление.

Учесть возможность инициализации общих полей в базовом классе (Метод Init() – применить паттерн Шаблонный метод).

Создать класс Creator, который и будет инстанцировать классы конструкций. Применить к нему паттерн Singleton.

Создать параметризованный класс для работы с набором конструкций. Применить паттерн Iterator для него.

10 Создание оконного приложения на платформе WPF с применением механизма привязок.

Создать оконное WPF приложение с применением паттерна MVVM и механизма привязок, которое отображает каталог товаров и детальную информацию по выбранному товару. Для этого:

1. Выделить пространства имен View, Model и ViewModel.
2. Поместить главную форму в пространство имен View.
3. Создать в пространстве имен Model класс Product со свойствами шифр, группа и наименование.
4. Создать в пространстве имен ViewModel класс VM_Catalog, который содержит коллекцию Product, номер выбранного товара, объект выбранного товара и реализует интерфейс INotifyPropertyChanged.
5. Установить для главной формы VM_Catalog в качестве DataContext.
6. Реализовать на главной форме отображение каталога товаров в таблице DataGrid.
7. Реализовать на главной форме отображение деталей одного товара в виде соответствующих полей.
8. На главной форме при выборе строки в таблице отобразить детали выбранного товара.

9. Компоновка пользовательского интерфейса осуществляется учащимся самостоятельно в индивидуальной форме.

4.2 Фонд оценочных средств для проведения промежуточной аттестации.

Примерный перечень вопросов к экзамену:

- 1 Класс, объект, структура класса (поля, свойства, методы, конструкторы).
- 2 Виртуальные, абстрактные и статические методы. Статические поля и свойства.
- 3 Перегрузка методов. Модификаторы параметров.
- 4 Венгерская нотация.
- 5 Нотация диаграмм классов.
- 6 Нотация диаграмм объектов.
- 7 Нотация диаграмм взаимодействий.
- 8 Принципы наследования, полиморфизма и абстрагирования.
- 9 Механизмы повторного использования: наследование, композиция, делегирование, параметризованные типы.
- 10 Основные принципы разработки собственных компонент. Класс Control.
- 11 Описание и классификация паттернов проектирования.
- 12 Паттерн Одиночка (Singleton).
- 13 Паттерн Стратегия (Strategy).
- 14 Паттерн Шаблонный метод (Template Method).
- 15 Паттерн Адаптер объектов (Adapter).
- 16 Паттерн Итератор (Iterator).
- 17 Паттерн Компоновщик (Composite).
- 18 Рефлексии типов.
- 19 Динамическая загрузка библиотек.
- 20 Позднее связывание.
- 21 Бинарная сериализация и десериализация.
- 22 XML сериализация и десериализация.
- 23 Основы LINQ. Работа с коллекциями с помощью LINQ to Objects.
- 24 Структура проекта Window Forms.
- 25 Основные элементы управления в Window Forms приложениях.
- 26 Основные контейнеры в Window Forms приложениях.
- 27 Структура проекта WPF (Windows Presentation Foundation).
- 28 Принципы создания приложения WPF с помощью паттерна MVVM.

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 (модулю) 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 (модуля).

5.1 Основная литература:

1. Григорьев, М. В. Проектирование информационных систем : учебное пособие для вузов / М. В. Григорьев, И. И. Григорьева. — М. : Издательство Юрайт, 2017.

2. Леоненков, А. Нотация и семантика языка UML / А. Леоненков. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016.

3. Павлова, Е.А. Технологии разработки современных информационных систем на платформе Microsoft.NET [Электронный ресурс] : учеб. пособие — Электрон. дан. — Москва : , 2016. — 128 с. — Режим доступа: <https://e.lanbook.com/book/100519>. — Загл. с экрана.

4. Снетков, В.М. Практикум прикладного программирования на C# в среде VS.NET 2005 [Электронный ресурс] : учеб. пособие — Электрон. дан. — Москва : , 2016. — 2140 с. — Режим доступа: <https://e.lanbook.com/book/100381>. — Загл. с экрана.

5. Снетков, В.М. Практикум прикладного программирования на C# в среде VS.NET 2008 [Электронный ресурс] : учеб. пособие — Электрон. дан. — Москва : , 2016. — 1659 с. — Режим доступа: <https://e.lanbook.com/book/100382>. — Загл. с экрана.

6. Флегонтов А.В., Матюшичев И.Ю. Моделирование информационных систем. Unified Modeling Language: Учебное пособие. – СПб.: Издательство «Лань», 2018. – 112 с.: ил. – (Учебники для вузов. Специальная литература).

7. Чеповский, А.М. Common Intermediate Language и системное программирование в Microsoft .NET [Электронный ресурс] : учеб. пособие / А.М. Чеповский, А.В. Макаров, С.Ю. Скоробогатов. — Электрон. дан. — Москва : , 2016. — 398 с. — Режим доступа: <https://e.lanbook.com/book/100589>. — Загл. с экрана.

Для освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья имеются издания в электронном виде в электронно-библиотечных системах «Лань» и «Юрайт».

5.2 Дополнительная литература:

8. UML. Классика CS. Буч Г., Якобсон А., Рамбо Дж., Орлов С.А. 2-е изд. 2005 год. ISBN 5-469-00599-2.

9. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Электронный ресурс] : справочник / Гамма Э., Хелм Р., Джонсон Р. [и др.]. — Электрон. дан. — М. : ДМК Пресс, 2007. — 376 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=1220 — Загл. с экрана.

10. Иан Грэхем Объектно-ориентированные методы. Принципы и практика. — 3-е изд. — М.: Издательский дом «Вильямс», 2004. — 880 с.

11. Основные концепции и механизмы объектно-ориентированного программирования/ Е.В. Пышкин. - СПб. : БХВ-Петербург, 2005 г.

12. Павловская Т.А. С#. Программирование на языке высокого уровня : учебник для вузов / Павловская, Татьяна Александровна ; Т. А. Павловская. - Санкт-Петербург [и др.] : Питер, 2014. - 432 с. : ил. - (Учебник для вузов). - Библиогр.: с. 425-426. - ISBN 9785496008617.

13. Юркин, А. Г. Задачник по программированию А. Г. Юркин; гл. ред. Е. Строганова - СПб.: Питер, 2002.

5.3. Периодические издания:

Прикладная математика XXI века: материалы объединенной науч. конф. студентов и аспирантов под ред. Ю.В. Кольцова. Краснодар: Кубанский гос. ун-т.

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

1. Википедия, свободная энциклопедия. [Электронный ресурс]. – Wikipedia <http://ru.wikipedia.org>
2. Сервер Информационных Технологий. [Электронный ресурс]. - Режим доступа: <http://citforum.ru>.
3. Справочник «Паттерны проектирования». [Электронный ресурс]. - Режим доступа: <http://design-pattern.ru>.
4. Microsoft – Официальная страница / Элементы управления Windows Forms [Электронный ресурс]. - Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/controls/>.
5. Сайт о программировании / Элементы управления в Windows Forms | Visual C# [Электронный ресурс]. - Режим доступа: <https://metanit.com/sharp/windowsforms/4.1.php>.
6. Рефакторинг и Паттерны проектирования [Электронный ресурс]. - Режим доступа: <https://refactoring.guru/ru>

7. Методические указания для обучающихся по освоению дисциплины (модуля).

По курсу предусмотрено написание самостоятельной реферативной работы студентов. Каждый студент выполняет работу по одной теме.

Для написания реферата необходимо подобрать литературу. Общее количество литературных источников, включая тексты из Интернета, (публикации в журналах), должно составлять не менее 10 наименований. Учебники, как правило, в литературные источники не входят.

Рефераты выполняют на листах формата А4. Страницы текста, рисунки, формулы нумеруют, рисунки снабжают порисуночными надписями. Текст следует печатать шрифтом №14 с интервалом между строками в 1,5 интервала, без недопустимых сокращений. В конце реферата должны быть сделаны выводы.

В конце работы приводят список использованных источников.

Реферат должен быть подписан студентом с указанием даты ее оформления.

Работы, выполненные без соблюдения перечисленных требований, возвращаются на доработку.

Выполненная студентом работа определяется на проверку преподавателю в установленные сроки. Если у преподавателя есть замечания, работа возвращается и после исправлений либо вновь отправляется на проверку, если исправления существенные, либо предъявляется на экзамене, где происходит ее защита.

Примерные темы презентаций и докладов:

– Презентация «Удачные архитектурные решения».

- Презентация «Приемы оживления пользовательского интерфейса».
- Презентация «Технологии доступа к данным».
- Презентация «Дополнительные технологии: WWF».
- Презентация «LINQ to Entities».
- Презентация «LINQ to Sql».
- Презентация «LINQ to XML».
- Презентация «LINQ to DataSet».

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 (модулю).

8.1 Перечень информационных технологий.

- Проверка домашних заданий и консультирование посредством электронной почты.
- Использование интерактивной подачи материала с мультимедийной системой.
- Проведение лекции-визуализации.

8.2 Перечень необходимого программного обеспечения.

- среда разработки Microsoft Visual Studio
- программы для демонстрации и создания презентаций («Microsoft Power Point»).

8.3 Перечень информационных справочных систем:

1. Википедия, свободная энциклопедия. [Электронный ресурс]. – Wikipedia <http://ru.wikipedia.org>.
2. Электронная библиотека КубГУ.
3. Электронная библиотечная система eLIBRARY.RU (<http://www.elibrary.ru>).

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

№	Вид работ	Материально-техническое обеспечение дисциплины (модуля) и оснащенность
1.	Лекционные занятия	Лекционная аудитория, оснащенная проекционным оборудованием (цифровой проектор, экран, ноутбук) и соответствующим программным обеспечением (ПО)
2.	Семинарские занятия	Не предусмотрены
3.	Лабораторные занятия	Компьютерный класс с ПЭВМ, современное лицензионное программное обеспечение Windows 7 или выше, Microsoft Visual Studio 2010 или выше, Microsoft SQL Server 2012 или выше
4.	Курсовое	Не предусмотрено

	проектирование	
5.	Групповые (индивидуальные) консультации	аудитория с учебной мебелью (столы, стулья), соответствующей количеству студентов
6.	Текущий контроль, промежуточная аттестация	аудитория с учебной мебелью (столы, стулья), соответствующей количеству студентов
7.	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.