

Балаковский инженерно-технологический институт – филиал федерального
государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»

Факультет атомной энергетики и технологий
Кафедра «Информационные системы и технологии»

РАБОЧАЯ ПРОГРАММА

по дисциплине «Современные технологии
объектно-ориентированного программирования»

Направления подготовки/специальность

«09.03.02 Информационные системы и технологии»

Основная профессиональная образовательная программа

«Информационные системы и технологии»

Квалификация выпускника

Бакалавр

Форма обучения

Очная

Цель освоения дисциплины

Цель освоения дисциплины «Современные технологии объектно-ориентированного программирования» в области обучения, воспитания, развития, соотносённые с общими целями ООП ВО и требованиями профессиональных стандартов «Специалист в области информационных технологий на атомных станциях (разработка и сопровождение программного обеспечения)»: формирование культуры исследовательской и инженерной деятельности, формирование, развитие и совершенствование у студентов профессиональных компетенций в области программной инженерии на этапе разработки исполняемых программных решений и документирования требований к ним с применением объектно-ориентированной парадигмы.

Место учебной дисциплины в структуре ООП ВО

Необходимыми условиями для освоения дисциплины являются знания, умения и практические навыки по предшествующим дисциплинам и практикам:

- Языки программирования
- Современные среды визуального программирования
- Технологии программирования
- Объектно-ориентированное программирование
- Учебная практика (по получению первичных профессиональных умений и навыков)

Знания, умения и практические навыки, полученные при освоении дисциплины, необходимы при изучении следующих дисциплин и прохождения практик:

- Инструментальные средства информационных систем
- Архитектура информационных систем
- Современные технологии управления базами данных
- Интеллектуальные системы и технологии
- Современные технологии интернет-программирования
- Стандартизация разработки программного обеспечения
- Производственная практика (преддипломная)
- Государственная итоговая аттестация

При освоении данной дисциплины студент сможет частично продемонстрировать следующие обобщенные трудовые функции: эксплуатация информационной системы представления технологических параметров и коммерческого учета электроэнергии атомной станции.

Компетенции обучающегося, формируемые в результате освоения дисциплины

В процессе освоения данной дисциплины у студента формируются следующие компетенции: общепрофессиональные

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	З-ОПК-6 – методы алгоритмизации, языки и технологии программирования; библиотеки программных модулей; шаблоны, классы объектов, используемые при разработке программного обеспечения У-ОПК-6 – создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов; использовать выбранную среду программирования для написания программного кода В-ОПК-6 – языками и средами программирования для разработки алгоритмов и программ

профессиональные компетенции

Задачи профессиональной деятельности (ЗПД)	Объект или область знания	Код и наименование компетенции	Индикаторы достижения компетенции
--	---------------------------	--------------------------------	-----------------------------------

разработка и внедрение технологий разработки объектов профессиональной деятельности в различных областях и сферах деятельности	информационные процессы, технологии, системы и сети, их инструментальное (программное, техническое, организационное) обеспечение, способы и методы проектирования, отладки, производства и эксплуатации информационных технологий и систем в различных областях и сферах деятельности.	ПК-6 Способен разрабатывать технические спецификации на программные компоненты и осуществлять их реализацию	З-ПК-6 Знать: виды технических спецификаций и требования к ним У-ПК-6 Уметь: разрабатывать технические спецификации на программные компоненты и осуществлять их реализацию В-ПК-6 Владеть: средствами разработки технической документации
--	--	---	---

задачи воспитания, реализуемые в рамках освоения дисциплины

Направление/цели	Создание условий, обеспечивающих	Использование воспитательного потенциала учебной дисциплины	Вовлечение в разноплановую внеучебную деятельность
Профессиональное воспитание	формирование культуры исследовательской и инженерной деятельности (В16)	Использование воспитательного потенциала дисциплин профессионального модуля для формирования формирования культуры исследовательской и инженерной деятельности за счёт практических студенческих исследований современных производственных систем; проектной деятельности студентов по разработке и оптимизации технологических систем, связанной с решением реальных производственных задач; прохождения через разнообразные игровые, творческие, фестивальные формы, требующие анализа сложного объекта, постановки относительно него преобразовательных задач для их оптимального решения.	1. Организация научно-практических конференций и встреч с ведущими специалистами предприятий города и ветеранами атомной отрасли. 2. Организация и проведение предметных олимпиад и участие в конкурсах профессионального мастерства.

Структура и содержание учебной дисциплины

Дисциплина изучается студентами в 5-ом семестре. Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 ак. часов.

Календарный план

№ Раздела	№ Темы	Наименование раздела (темы) дисциплины	Виды учебной деятельности (час.)					Аттестация раздела (форма*)	Максимальный балл за раздел **
			Всего	Лекции	Лабораторные	Практические	СРС/КРС		
1	Инструментарий языков, реализующих парадигму ООП							Т	30
	1	Языковые средства реализации	20/4	4/2		6/2	10		

		концепции ООП		*					
	2	Разработка пользовательских интерфейсов и спецификаций на них	20/4	2/2		8/2	10		
	3	Технологии подключения и работы с базами данных	16/2	2		4/2	10		
2	Техники написания, поддержки и документирования кода								
	4	Паттерны ООП	26/6	4/2		12/4	10		
	5	Рефакторинг и оптимизация кода	12	2			10		
	6	Информационное обеспечение процесса разработки программного кода.	14	2		2	10		
		Зачёт						3	40
Итого			108/16	16/6	-	32/10	60		

* - сокращенное наименование формы контроля

** - сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращенное наименование форм текущего контроля и аттестации разделов:

Обозначение	Полное наименование
Т	Тестирование
З	Зачет

Содержание лекционного курса

Тема лекции. Вопросы, отрабатываемые на лекции	Всего часов	Учебно-методическое обеспечение
1	2	3
Лекция 1. Языковые средства реализации концепции ООП. 1. Делегаты, события, лямбды. 2. Коллекции, индексаторы. 3. Отражения.	4	1-7
Лекция 2. Инструменты построения графических интерфейсов экосистемы платформы .Net. 1. Особенности платформы WPF. 2. Введение в язык XAML, структура и пространства имён XAML, взаимодействие кода C# и XAML, элементы компоновки и управления. 3. Модель событий в WPF. 4. Создание стилей, триггеры, свойства, кисти, привязки. 5. Разработка спецификаций на пользовательские интерфейсы.	2	1-7
Лекция 3. Технологии подключения и работы с базами данных. 1. Работа с данными в WPF. 2. Взаимодействие с базой данных. 3. Реализация паттерна MVVM	2	1-7
Лекция 4. Паттерны ООП. 1. Использование паттернов ООП в качестве готовых шаблонов при разработке программных решений. 2. Порождающие, структурные и поведенческие паттерны. 3. Использование диаграмм классов в качестве спецификаций для представления структуры программного кода, реализующего типовые решения с использованием паттернов.	4	1-7

Лекция 5. Рефакторинг и оптимизация кода. 1. Техники рефакторинга. 2. Оформление и рецензирование кода, юнит тесты.	2	1-7
Информационное обеспечение процесса разработки программного кода. Управление проектами, системы контроля версий.	2	1-7

Перечень практических занятий

Тема практического занятия. Задания, вопросы, обрабатываемые на практическом занятии	Всего часов	Учебно-методическое обеспечение
Тема 1. Языковые средства реализации концепции ООП 1. Использование делегатов для реализации механизма позднего связывания в .NET. 2. Работа с коллекциями, использование индексов для работы с коллекциями 3. Динамический доступ к объектам через механизм Отражения	6	1-7
Тема 2. Разработка пользовательских интерфейсов и спецификаций на них 1. Элементы управления и компоновки в WPF. Взаимодействие между окнами 2. Определение набора характеристик форматирования с помощью стилей 3. Модель событий в WPF. Триггеры. 4. Привязка в WPF. Реализация паттерна MVVM	8	1-7
Тема 3. Технологии подключения и работы с базами данных 1. Работа с данными в WPF. 2. Взаимодействие с базой данных	4	1-7
Тема 4. Паттерны ООП 1. Порождающие паттерны 2. Структурные паттерны 3. Поведенческие паттерны	12	1-7
Тема 6. Информационное обеспечение процесса разработки программного кода. 1. Создания unit-тестов 2. Контроль версий и проектные репозитории	2	1-7

Перечень лабораторных работ

Лабораторные работы не предусмотрены учебным планом

Задания для самостоятельной работы студентов

Задания, вопросы, для самостоятельного изучения (задания)	Всего часов	Учебно-методическое обеспечение
1	2	3
Динамическая загрузка сборок и позднее связывание. Лямбды.	10	1-7
Форматирование значений привязки и конвертеры значений	10	1-7
Работа с Entity Framework	10	1-7
Паттерны и антипаттерны ООП	10	1-7
Техники рефакторинга	10	1-7
Распределённые системы управления версиями	10	1-7

Оценка результатов самостоятельной работы организуется как единство двух форм: самоконтроль и контроль со стороны преподавателя.

С целью самоконтроля каждый раздел в учебно-методических материалах завершается набором тестовых заданий.

Расчетно-графическая работа

Расчетно-графическая работа не предусмотрена учебным планом.

Курсовая работа

Курсовая работа не предусмотрена учебным планом

Курсовой проект

Курсовой проект не предусмотрен учебным планом

Образовательные технологии

При изучении дисциплины «Современные технологии объектно-ориентированного программирования» используются интерактивные формы проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Преподавание дисциплины ведется с применением следующих видов образовательных технологий: использование электронных образовательных ресурсов, обеспечивающих обучение в информационной образовательной среде; лекционные занятия проводятся с использованием мультимедийного лекционного материала, практических занятий, с использованием ПК при разработке программ и спецификаций. Самостоятельная работа студентов проводится под руководством преподавателей, с оказанием консультаций и помощи при подготовке к контрольным работам.

Фонд оценочных средств

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

№ п/п	Наименование контролируемых разделов (темы)	Код и наименование индикатора достижения компетенций	Наименование оценочного средства
Входной контроль			
1	Входной контроль		Вопросы входного контроля (письменно)
Аттестация разделов, текущий контроль успеваемости			
2	Инструментарий языков, реализующих парадигму ООП	З-ОПК-6, У-ОПК-6, В-ОПК-6	Тестирование (письменно)
3	Техники написания, поддержки и документирования кода	З-ПК-6, У-ПК-6, З-ОПК-6, У-ОПК-6, В-ОПК-6	Тестирование (письменно)
Промежуточная аттестация			
4	Зачет	З-ОПК-6, У-ОПК-6, В-ОПК-6, З-ПК-6, У-ПК-6, В-ПК-6	Вопросы к зачету (устно)

Оценочные средства для входного контроля, текущего контроля и промежуточной аттестации (аннотация)

Входной контроль предназначен для выявления пробелов в знаниях студентов и готовности их к получению новых знаний. Оценочные средства для входного контроля представляют собой вопросы, которые задаются студентам в устной форме.

Задания входного контроля

1. Какое сообщение после выполнения данной программы будет выведено на экран?

```
package client;
interface I1 {
    final int p = 10;
    public int get();
}
class B implements I1 {
    private int p = 20;
    public int get() {
        return p;
    }
}
class C extends B implements I1 {
    public int get() {
        return p;
    }
}
public class A {

    public static void main(String...args){
        B obj = new C();
        System.out.println(obj.get());

    }
}
```

- а) 10
- б) 20
- в) Ошибка компиляции
- г) Ошибка времени выполнения

2. Какие из парадигм объектно-ориентированного программирования проиллюстрированы данным листингом? Укажите номера строк.

3. Какое имя имеет супер класс? Какую функцию выполняет интерфейс?

4. Что необходимо изменить в тексте программы, чтобы на экран выводился результат: «Result:20»?

5. Как необходимо определить видимость переменной p в классе B, чтобы исключение из программы реализации в классе C интерфейса I не вызывало ошибки при компиляции?

Оценочные средства текущего контроля включают в себя:

На текущем контроле успеваемости:

ПР - практическая работа: средство, которое может быть использовано для контроля приобретенных студентом профессиональных навыков и умений по управлению конкретным материальным объектом. По результатам выполнения практической работы проводится оценка текущей успеваемости, которая суммируется при аттестации разделов.

На рубежном контроле:

Тест – тестирование, средство контроля, представляющее собой систему стандартизированных заданий, позволяющую автоматизировать процедуру измерения уровня знаний и умений обучающегося.

Тестовые задания для аттестации раздела 1

1. Какой элемент компоновки в WPF располагает все элементы в ряд либо по горизонтали, либо по вертикали в зависимости от ориентации?

- a) Grid
- b) StackPanel
- c) Canvas
- d) GridSplitter

2. Какое значение будет принимать свойство контейнера StackPanel, которое устанавливает выравнивание по вертикали, в следующей записи XAML?

```
<StackPanel x:Name="Stack6" VerticalAlignment="Top" Background="Gainsboro"
HorizontalAlignment="Left" Panel.ZIndex="0" Height="450" Width="1068">
</StackPanel>
```

- a) Left
- b) Center
- c) Vertical
- d) Top

3. Какое значение примет переменная result в результате выполнения программы:

```
class Math
{
    ссылка: 1
    public int X(int x, int y) { return x * y; }
}
Ссылка: 0
class Program
{
    delegate int Operation(int x, int y);

    Ссылка: 0
    static void Main(string[] args)
    {
        int a = 4;
        Math math = new Math();
        Operation del = math.X;
        int result = del(a, 5);
        Console.WriteLine(result);

        Console.Read();
    }
}
```

- a) 4
- b) 5
- c) 9
- d) 20

4. В результате выполнения программы на экран будет выведена запись:

```
class Program
{
    delegate int Operation(ref int x, int y);
    Ссылка: 0
    static void Main(string[] args)
    {
        int n = 10;
        Operation operation = (ref int x, int y) => x + y;
        Console.WriteLine(operation(ref n, 20) + operation(ref n, 2*20));
        Console.Read();
    }
}
```

- a) 10
- b) 20
- c) 50
- d) 80

Тестовые задания для аттестации раздела 2

1. Паттерн проектирования - это:

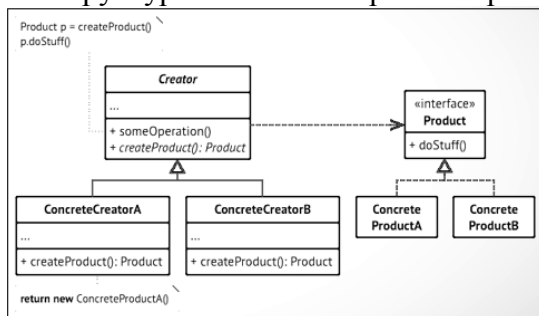
- a) Повторимая архитектурная конструкция, используемая для решения часто встречающихся проблем проектирования
- b) Схематичное представление процесса, системы или компьютерного алгоритма.
- c) Диаграмма, отражающая отношения между пользователями системы и сценариями использования системы и являющаяся составной частью модели прецедентов,

позволяющей описать систему на концептуальном уровне

2. Какие паттерны относятся к группе порождающих паттернов?

- a) Мост
- b) Прототип
- c) Адаптер
- d) Посредник
- e) Одиночка
- f) Шаблонный метод
- g) Строитель

3. Структура какого паттерна изображена на диаграмме классов?



- a) Строитель
- b) Абстрактная фабрика
- c) Фабричный метод
- d) Прототип

4. Какой паттерн определяет интерфейс для создания объектов некоторого класса, при этом непосредственное решение о том, объект какого класса создавать происходит в подклассах-наследниках?

- a) Строитель
- b) Прототип
- c) Фабричный метод
- d) Абстрактная фабрика

5. Какой структурный паттерн, позволяет разделять абстракцию и реализацию так, чтобы они могли изменяться независимо, при этом изменение структуры интерфейса не мешает изменению структуры реализации?

- a) Mediator
- b) State
- c) Strategy
- d) Memento

Система оценки аттестации разделов дисциплины

Каждый контрольный тест оценивается в 10 баллов. Тест считается сданным, если студент правильно ответил на 60 процентов от заданных ему вопросов.

Оценка	Количество верно данных ответов, %
10 баллов	95-100%
9 баллов	85-94%
8 баллов	75-84%
7 баллов	65-74%
6 баллов	55-64%
5 баллов	45-50%
Менее 5 баллов	менее 45%

Итоговые баллы за раздел выставляются в соответствии со шкалой оценки, приведённой ниже

Баллы (рейтинговой оценки)	Оценка (стандартная)	Требования к знаниям
28-30	«отлично»	Оценка «отлично» за раздел дисциплины выставляется студенту, если он получил за тест оценку “отлично”, выполнил на отлично и защитил практические работы, предусмотренные курсом в данном разделе, показал отличные знания в области объектно-ориентированного программирования. Показывает отличные знания языковых средств реализации концепции ООП, владеет навыками разработки пользовательских интерфейсов и спецификаций на них, уверенно демонстрирует приемы работы с базами данных, знает техники написания, поддержки и документирования кода, а также рефакторинга и оптимизации кода. Умеет применять паттерны ООП, владеет средствами информационной поддержки процесса разработки программного кода. При этом не затрудняется с ответом при видеоизменении заданий.
23-27	«хорошо»	Оценка «хорошо» выставляется студенту, если он получил за тест оценку “хорошо”, хорошо выполнил и защитил практические работы, предусмотренные курсом в данном разделе, показывает хорошие знания языковых средств реализации концепции ООП, владеет навыками разработки пользовательских интерфейсов и спецификаций на них, знает техники написания, поддержки и документирования кода, а также рефакторинга и оптимизации кода. Показывает уверенный навык подключения баз данных к программному решению, знает как применить паттерны ООП в процессе разработки программ. При этом не допускает существенных неточностей в ответе на вопрос.
18-22	«удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, если он получил за тест оценку “удовлетворительно”, выполнил основную часть практических работ, предусмотренные курсом в данном разделе, показывает удовлетворительные знания в области объектно-ориентированного программирования, владеет навыками разработки пользовательских интерфейсов, знает виды паттернов ООП. При этом не усвоил всех деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении материала.
менее 18	«неудовлетворительно»	Оценка «неудовлетворительно» выставляется студенту, который получил за тест оценку “неудовлетворительно”, не выполнил основную часть практических работ, предусмотренные курсом в данном разделе, не владеет знаниями по материалам курса.

По итогам обучения выставляется зачет.

Методика проведения зачета

Промежуточная аттестация по дисциплине в соответствии с учебным планом направления проводится в форме зачета. Зачет по дисциплине представляет собой

итоговое испытание по профессионально-ориентированным проблемам, устанавливающее соответствие подготовленности студентов требованиям образовательного стандарта. Зачет проводится с целью проверки уровня и качества форсированности компетенций в рамках соответствующего этапа и позволяет выявить и оценить теоретическую и практическую подготовку студента для решения профессиональных задач.

Зачет проводится в письменной форме по индивидуальному заданию в два этапа:

1 этап – выполнение первого задания, которое включает подготовку ответа на теоретический вопрос в билете;

2 этап - выполнение второго задания, которое включает решение практического задания.

Перечень вопросов к зачету

1. Понятие делегата в C#. Описание и использование. Анонимные методы.
2. Пространство имен System.Collections. Основные элементы класса ArrayList.
3. Индексаторы.
4. Отражения в C#. Применение рефлексии и исследование типов.
5. Расширяемый язык разметки XAML. Загрузка и компиляция XAML
6. Компоновка в WPF. Элементы компоновки, свойства компоновки элементов.
7. Элементы управления в WPF и их свойства.
8. События в WPF приложениях. События клавиатуры и мыши.
9. Привязка данных в WPF приложениях.
10. Использование ресурсов и стилей в WPF приложениях.
11. Триггеры в WPF приложениях.
12. Фигуры и кисти в WPF приложениях.
13. Шаблон проектирования MVVM.
14. Работа с данными в WPF. Взаимодействие с базой данных.
15. Порождающие паттерны ООП.
16. Структурные паттерны ООП.
17. Поведенческие паттерны ООП.
18. Рефакторинг и оптимизация кода. Техники рефакторинга.
19. Оформление и рецензирование кода, юнит тесты.
20. Управление проектами, системы контроля версий.

Примерные практические задания к зачету

1. После выполнения следующего программного кода в окно консоли будет выведена строка:

```
class Program
{
    public static void swap(ref string s1, ref string s2)
    {
        string tmpString = s1;
        s1 = s2;
        s2 = tmpString;
    }

    static void Main(string[] args)
    {
        string s1="123";
        string s2="abc";
        swap(ref s1, ref s2);
        Console.WriteLine("s1: {0}, s2: {1}", s1, s2);
        Console.ReadLine();
    }
}
```

2. Что будет выведено в результате выполнения следующего программного кода?

```

class Program
{
    static void Main(string[] args)
    {
        ArrayList list = new ArrayList();
        list.Add(2.3);
        list.Add(35);
        list.AddRange(new string[] { "Hello", "world" });
        Console.WriteLine(list[3]);
        Console.ReadLine();
    }
}

```

3. Какой элемент класса User с помощью механизма Reflection будет получен?

```

class Program
{
    static void Main(string[] args)
    {
        Type myType = Type.GetType("HelloApp.User", false, true);

        foreach (ConstructorInfo ctor in myType.GetConstructors())
        {
            Console.WriteLine(myType.Name + " (" + ctor.ToString() + ")");
        }

        Console.ReadLine();
    }
}

```

```

public class User
{
    public string Name { get; set; }
    public int Age { get; set; }
    public User(string n, int a)
    {
        Name = n;
        Age = a;
    }
    public void Display()
    {
        Console.WriteLine($"Имя: {Name} Возраст: {Age}");
    }
    public int Payment(int hours, int perhour)
    {
        return hours * perhour;
    }
}

```

Критерии оценки зачета

Шкалы оценки образовательных достижений

Сумма баллов	Оценка по 4-х балльной шкале	Зачет	Оценка (ECTS)
90-100	отлично	Зачтено	A
85-89	хорошо		B
75-84			C
70-74			D
65-69	удовлетворительно		E
60-64			
Ниже 60	неудовлетворительно	не зачтено	F

Баллы (итоговой рейтинговой оценки)	Баллы (зачет)	Требования к знаниям
100-90	Зачтено 24 – 40 баллов	теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному
85 - 89		теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному.
75 - 84		теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с

		освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками.
65 - 74		теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки.
60-64		теоретическое содержание курса освоено частично, некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены, либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному
Ниже 60	не зачтено 0-23 баллов	очень слабые знания, недостаточные для понимания курса, имеется большое количество основных ошибок и недочетов

Учебно-методическое и информационное обеспечение учебной дисциплины

Основная литература

1. Николаев, Е. И. Объектно-ориентированное программирование : учебное пособие / Е. И. Николаев. — Ставрополь : СКФУ, 2015. — 225 с. <https://e.lanbook.com/reader/book/155240/#2>
2. Объектно-ориентированное программирование : учебное пособие : в 2 частях / составитель Е. И. Николаев. — Ставрополь : СКФУ, 2015 — Часть 1 — 2015. — 183 с. <https://e.lanbook.com/reader/book/155241/#3>
3. Подбельский, В. В. Язык декларативного программирования XAML / В. В. Подбельский. — Москва : ДМК Пресс, 2018. — 336 с. — ISBN 978-5-97060-573-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/111428>
4. Романов, Е. Л. Программная инженерия : учебное пособие / Е. Л. Романов. — Новосибирск : НГТУ, 2017. — 395 с. — ISBN 978-5-7782-3455-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/118221>

Дополнительная литература

5. Карпенко, С. Н. Основы объектно-ориентированного программирования на языке C++ : учебно-методическое пособие / С. Н. Карпенко. — Нижний Новгород : ННГУ им. Н. И. Лобачевского, 2018. — 104 с. <https://e.lanbook.com/reader/book/144808/#3>
6. Кудинов, Ю. И. Основы современной информатики: учебное пособие / Ю. И. Кудинов, Ф. Ф. Пашенко. — 5-е изд., стер. — Санкт-Петербург : Лань, 2018. — 256 с. <https://e.lanbook.com/reader/book/107061/#255>

Программное обеспечение и Интернет-ресурсы

1. Электронно-библиотечная система «Лань».
2. Электронно-библиотечная система «Юрайт».
3. Российская национальная библиотека - <http://www.nlr.ru/>
4. Поисковая система - <http://www.rambler.ru>.
5. Поисковая система - <http://www.yandex.ru>.
6. Гарант - <http://base.garant.ru/>.
7. Интернет-Университет Информационных Технологий - <http://www.intuit.ru>

Материально-техническое обеспечение учебной дисциплины

Лекционные занятия проводятся в аудиториях: учебная мебель, учебная доска, комплект мультимедийного оборудования

Практические занятия проводятся в компьютерных классах: учебная мебель, учебная доска, комплект мультимедийного оборудования, персональные компьютеры.

Для самостоятельной работы обучающихся имеется: читальный зал с выходом в сеть Интернет: Учебная мебель, комплект мультимедийного оборудования, персональные компьютеры, МФУ.

Учебно-методические рекомендации для студентов

1) Следует тщательно планировать и организовывать время, необходимое для изучения дисциплины. Недопустимо откладывать ознакомление с теоретической частью, выполнение индивидуальных заданий по темам практических работ на конец семестра, поскольку это неминуемо приведет к снижению качества освоения материала, индивидуальных заданий. Все виды работ по дисциплине рекомендуется выполнять по календарному плану, приведенному в Рабочей программе.

2) «Сценарий изучения дисциплины» предусматривает следующие схемы: по теоретическому курсу: ознакомление с тематикой лекции; изучение литературы по теме; прослушивание лекции; обсуждение вопросов.

По выполнению индивидуальных работ по темам практических занятий: подготовка к выполнению работы по методическим указаниям; работа на практическом занятии выполнение работы и оформление отчета; защита работы.

3) Изучение дисциплины требует непрерывной работы с литературой. Перед прослушиванием каждой лекции студент должен ознакомиться с материалом по списку, приведенному по теме лекции в рабочей программе. Перед выполнением индивидуальных занятий по темам практических работ необходимо изучить теоретические сведения, приведенные в методических указаниях к выполнению практических работ. Отчет, составляемый после выполнения работы, должен соответствовать варианту, выданному преподавателем в начале выполнения практической работы.

Методические рекомендации для преподавателей

1. Указания для проведения лекций

На первой вводной лекции сделать общий обзор содержания курса и отметить новые методы и подходы к решению задач, рассматриваемых в курсе, довести до студентов требования кафедры, ответить на вопросы.

При подготовке к лекционным занятиям необходимо продумать план его проведения, содержание вступительной, основной и заключительной части лекции, ознакомиться с новинками учебной и методической литературы, публикациями периодической печати по теме лекционного занятия. Уточнить план проведения практического занятия по теме лекции. Перед изложением текущего лекционного материала напомнить об основных итогах, достигнутых на предыдущих лекциях. С этой целью задать несколько вопросов аудитории и осуществить выборочный контроль знания студентов.

В ходе лекционного занятия преподаватель должен назвать тему, учебные вопросы, ознакомить студентов с перечнем основной и дополнительной литературы по теме занятия. Раскрывая содержание учебных вопросов, акцентировать внимание студентов на основных категориях, явлениях и процессах, особенностях их протекания. Раскрывать сущность и содержание различных точек зрения и научных подходов к объяснению тех или иных явлений и процессов.

Следует аргументировано обосновать собственную позицию по спорным теоретическим вопросам. Приводить примеры. Задавать по ходу изложения лекционного материала риторические вопросы и самому давать на них ответ. Это способствует активизации мыслительной деятельности студентов, повышению их внимания и интереса

к материалу лекции, ее содержанию. Преподаватель должен руководить работой студентов по конспектированию лекционного материала, подчеркивать необходимость отражения в конспектах основных положений изучаемой темы, особо выделяя, категориальный аппарат. В заключительной части лекции необходимо сформулировать общие выводы по теме, раскрывающие содержание всех вопросов, поставленных в лекции. Объявить план очередного практического занятия, дать краткие рекомендации по подготовке студентов к семинару. Определить место и время консультации студентам, пожелавшим выступить на семинаре с докладами и рефератами.

На последней лекции уделить время для обзора наиболее важных положений, рассмотренных в курсе.

2. Указания для проведения практических занятий

Четко обозначить тему занятия.

Обсудить основные понятия, связанные с темой.

В процессе решения задач вести дискуссию со студентами о правильности применения теоретических знаний.

Отмечать студентов, наиболее активно участвующих в решении задач и дискуссиях.

В конце практики задать аудитории несколько контрольных вопросов.

3. Указания по контролю самостоятельной работы студентов

По усмотрению преподавателя задание на самостоятельную работу может быть индивидуальным или фронтальным.

При использовании индивидуальных заданий требовать от студента письменный отчет о проделанной работе.

При применении фронтальных заданий вести коллективные обсуждения со студентами основных теоретических положений.

С целью контроля качества выполнения самостоятельной работы требовать индивидуальные отчеты (допустимо вместо письменного отчета применять индивидуальные контрольные вопросы).

Программа составлена в соответствии с требованиями ОС НИЯУ МИФИ по направлению 09.03.02 Информационные системы и технологии.

Рабочую программу составил:

Ст.преподаватель



А.Г. Мотков



Рецензент: профессор

О.В. Виштак

Программа одобрена на заседании УМКН «Информационные системы и технологии» от 04.07.2023 года, протокол №5.



Председатель учебно-методической комиссии

О.В. Виштак