

Федеральное государственное автономное образовательное учреждение высшего образования
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ имени В.И. ВЕРНАДСКОГО»

УТВЕРЖДАЮ
Директор
Физико-технического института
_____ Нудьга А.А.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

"Объектно-ориентированное программирование" *(наименование дисциплины)*

Направление подготовки (специальность)

09.03.04 "Программная инженерия"

Направленность программы
"Программная инженерия"

РАЗРАБОТАНО

Заведующий кафедрой
компьютерной инженерии и моделирования

 Милюков В.В.

СОГЛАСОВАНО

Заведующий кафедрой
компьютерной инженерии и моделирования

 Милюков В.В.

Председатель

учебно-методической комиссии
Физико-технического института

_____ М.А.Бакуменко

1. Место дисциплины в структуре основной профессиональной образовательной программы

Дисциплина Объектно-ориентированное программирование относится к базовой части ОПОП и является обязательной для освоения.

2. Входные требования для освоения дисциплины, предварительные условия

Базовые знания для изучения дисциплины должны быть получены в курсах: Структуры и алгоритмы обработки данных, Программирование, Дискретная математика.

Требования к «входным» знаниям, умениям и навыкам обучающегося, необходимым для освоения дисциплины: структуры данных, алгоритмы обработки информации, знание основ структурного и процедурного программирования, знание синтаксиса языка C++, умение работать с логическими операторами, умение работать с циклами, массивами, строками, файлами, навыки программирования с помощью языка C++.

Дисциплина является базовой для освоения курсов «Системное программирование», «Системное ПО», «Веб - программирование», «Параллельные и распределенные вычисления» и других дисциплин профессиональной подготовки.

3. Планируемые результаты обучения по дисциплине, соотнесенные с требуемыми компетенциями выпускников

В результате освоения учебной дисциплины, у обучающегося должны быть сформированы комплекс знаний, умений и навыков, характеризующих этапы формирования компетенций и обеспечивающих достижение планируемых результатов освоения образовательной программы.

Таблица 1. Перечень компетенций

Компетенции		Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций		
код	формулировка	знать	уметь	владеть
ОПК-6	способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	основные понятия и термины: пространство имен, инкапсуляция, наследование и полиморфизм, типы-значения и ссылочные типы, классы, структуры, перечисления, поля, свойства, индексы, лямбда выражения, инкапсуляция, модификаторы доступа, конструкторы, перегрузка и переопределение методов, перегрузка операторов, интерфейсы, делегаты и события, преобразование типов, метаданные, принцип работы сборщика мусора, классы многопоточности	уметь проектировать программы и писать код для реализации быстродействующих и надежных алгоритмов и программ, пригодных для практического применения, разрабатывать визуальные интерфейсы программ, отлаживать и тестировать код	основными приемами визуального программирования с использованием основных программных конструкций Си - подобных языков, владеть приемами получения и использования информации из электронных источников, включая MSDN, Github и др.
ПК-13	готовность применять основные методы и инструменты разработки программного обеспечения	основы функционирования среды .Net Framework, среду разработки Visual Studio, основы визуального программирования WinForm и WPF, возможности использования различных программных средств в соответствии с решаемой задачей, концепции языков программирования и сферу их применения, принципы ООП,	инсталлировать и настраивать среду разработки Visual Studio, ориентироваться в выборе концепции и языка программирования в соответствии с задачей, использовать основные программные конструкции Си - подобных языков, применять современные концепции и технологии ООП программирования,	навыками проектирования и реализации прикладного ПО, владение приемами оценки достоверности полученных результатов

		основные программные конструкции Си - подобных языков	инсталлировать программное обеспечение	
ПК-14	владение навыками использования различных технологий разработки программного обеспечения	программные конструкции в языках высокого уровня, позволяющие быстро конструировать эффективный код, средства отладки и тестирования кода, отладка и обработка исключительных ситуаций, стратегию инкапсуляции полей, сигнатуру событий, принципы разработки надежного кода, основные паттерны проектирования ПО	создавать Windows Forms и WPF программы с логически верным интерфейсом пользователя, создавать и управлять событиями с помощью делегатов, работать с абстрактными классами и интерфейсами, динамически создавать компоненты, создавать многопоточные приложения, управлять исключениями и защищенными блоками, обрабатывать события мыши и клавиатуры	практическими навыками создания эффективного программного кода с использованием различных концепций и технологий программирования, (ООП, автоматного, функционального), владеть средствами отладки и тестирования кода, принципами разработки надежного ПО

4. Содержание дисциплины

4.1. Перечень тем теоретической подготовки

Темы лекций:
Лекция 1. Введение в C# и платформу .NET. Язык C# и платформа .NET. Общезыковая исполнительная среда CLR. Управляемый код. Основы работы с Visual Studio. Файлы проекта на C#. Особенности платформ WinForms и WPF. Структура программы. Пространства имен.
Лекция 2. Использование конструкций языка C#. Переменные, константы. Литералы. Система типов .Net. Типы значений и ссылочные типы. Простые (примитивные) типы. Явная и неявная типизация. Явное, неявное и пользовательское преобразования базовых типов данных. Проверка на null, операторы ? и ?? Псевдонимы типов и статический импорт.
Лекция 3. Операторы и операции. Методы. Арифметические операции. Поразрядные операции. Операции присваивания. Условные выражения. Конструкция if..else и тернарная операция. Конструкция switch. Циклы. Перечисления enum. Методы. Сигнатура методов. Вызов методов. Перегрузка методов. Статические методы. Массив параметров и ключевое слово params. Передача и возвращение значений по ссылке в метод. Модификаторы ref, out, in. Кортежи.
Лекция 4. Классы, структуры, конструкторы, модификаторы доступа. Классы, синтаксис классов. Структуры. Конструкторы, инициализаторы и деструкторы. Область видимости (контекст) переменных. Модификаторы доступа. Модификатор ReadOnly только для чтения. Создание и использование статических полей, статических методов, статических конструкторов и статических методов расширения.
Лекция 5. Базовые понятия и принципы ООП в C#. Инкапсуляция, наследование и полиморфизм. Особенности наследования в C#. Инкапсуляция полей с помощью свойств. Виртуальные методы и свойства. Переопределение и сокрытие методов и свойств. Перегрузка операторов. Абстрактные классы. Обобщенные типы. Ограничения обобщений.
Лекция 6. Делегаты, анонимные функции и лямбда выражения. Делегаты, развязка методов с помощью делегатов. Делегаты как параметры методов. Анонимные типы и методы. Делегаты и функции обратного вызова. Лямбда выражения как параметры и результат методов.

<p>Лекция 7. Делегаты и события.</p> <p>Встроенные делегаты Action, Predicate и Func. Ковариантность и контравариантность делегатов. Класс событий Event. Явная реализация методов add и remove. Делегат EventHandler и класс EventArgs. Проблемы безопасности с обработчиками событий (игнорирование коллег, переопределение аргументов и утечка памяти из-за событий).</p>
<p>Лекция 8. Интерфейсы.</p> <p>Интерфейсы как система контрактов ("чистые интерфейсы" до C#8). Интерфейсные свойства. Явная реализация интерфейсов. Выбор между интерфейсом и абстрактным классом. Шесть причин использования интерфейсов в промышленном программировании.</p>
<p>Лекция 9. Новые интерфейсы, C# 8, 9, 10, 11</p> <p>Методы по умолчанию в интерфейсах. Статические поля и константы в интерфейсах. Явное указание модификаторов доступа в интерфейсах.</p>
<p>Лекция 10. Классы общего назначения</p> <p>Работа с файловой системой. Классы File, FileInfo и FileStream. Чтение и запись файла. Чтение и запись текстовых файлов. StreamReader и StreamWriter. Бинарные файлы. BinaryWriter и BinaryReader. Сериализация в JSON. Регулярные выражения. Работа с датами и временем. Отложенная инициализация и тип Lazy.</p>
<p>Лекция 11. Классы общего назначения</p> <p>Массивы. Коллекции. Индексаторы. Работа со строками. Класс StringBuilder. Математические вычисления и класс Math. Класс Exception. Обработка исключений. Конструкция try..catch..finally. Генерация исключений и оператор throw.</p>
<p>Лекция **. Практическое программирование в WPF (факультативно)</p> <p>Работа с XML. Создание "резинового интерфейса". Классы элементов управления. Наследование форм. Организация меню. Инструментальные панели. Классы рисования.</p>
<p>Лекция 12. Параллельное программирование и многопоточность.</p> <p>Многопоточность. Класс Thread. Синхронизация потоков. Мониторы. Мьютексы. Семафоры. Параллельное программирование и библиотека TPL. Задачи и класс Task. Класс Parallel. Асинхронное программирование. Асинхронные методы, async и await.</p>
<p>Лекция 13. Рефлексия, класс Reflection. Dynamic Language Runtime.</p> <p>Рефлексия, класс Reflection. Применение рефлексии и исследование типов. Динамическая загрузка сборок и позднее связывание.</p>
<p>Лекция **. Опасный код. (факультативно)</p> <p>Атрибуты в .NET. Dynamic Language Runtime. DLR в C#. Опасный код. Сборка "мусора", класс Garbage Collector. Работа с базами данных и сетями.</p>
<p>Лекция 14. Паттерны проектирования ПО на C#.</p>
<p>Лекция ** Защита ПО, разработанного на C# от копирования и взлома (факультативно)</p> <p>Статические анализаторы кода ReSharper. Дизассемблирование и декомпиляция, IISpy? DnSpy, DotPeek. Очистка отладочной информации. Обфускация кода.</p>
<p>Лекция 15. Особенности программирования на C# в Unity.</p> <p>Основы работы в Unity. Встраивание собственных скриптов в проекты.</p>
<p>Лекция 16. Тестирование знаний по курсу ООП.</p> <p>Компьютерное тестирование.</p>

4.2. Перечень тем практической работы

Практич. работа 1	Использование программных конструкций C# Обязательное предварительное задание - преобразование типов (явное, неявное, пользовательское, Tryparse, Convert) Exercise 1: Calculating Square Roots with Improved Accuracy Exercise 2: Converting Integer Numeric Data to Binary Exercise 3: Multiplying Matrices Exercise 4: Generating Documentation for an Application
Практич. работа 2	Описание и вызов методов Exercise 1: Calculating the Greatest Common Divisor of Two Integers by Using Euclid's Algorithm Exercise 2: Calculating the GCD of Three, Four, or Five Integers [Exercise 3: Comparing the Efficiency of Two Algorithms] Exercise 4: Displaying Results Graphically
Практич. работа 3	Обработка исключительных ситуаций Exercise 1: Making a Method Fail-Safe Exercise 2: Detecting an Exceptional Condition Exercise 3: Checking for Numeric Overflow Задание 4: Подпись отчет к работе №3 хэшем строки
Практич. работа 4	Типы перечислений и структуры Задание 1. Создание структуры для работы с комплексными числами. Задание 2. Создание структуры для работы с трехмерными векторами Задание 3. Использование перечислений. Задание 4. Использование структур
Практич. работа 5	Наследование классов и использование интерфейсов Задание 1. Использование интерфейсов Задание 2. Создание абстрактного класса Задание 3. Создание архитектуры приложения с помощью интерфейсов
Практич. работа 6	Делегаты, события, лямбда выражения, многопоточное и асинхронное программирование Задание 1. Использование событий Задание 2. Использование лямбда-выражений Задание 3. Создание многопоточных приложений и асинхронных методов.
Практич. работа 7	Практическое программирование. Работа с файлами. Паттерны проектирования Задание 1. Реализовать приложение, использующее классы File, FileInfo и FileStream, StreamReader и StreamWriter, BinaryWriter и BinaryReader. Задание 2. Написать авторский реферат о паттернах проектирования Задание 3. Реализовать 6 паттернов проектирования не менее 3 типов на языке программирования C#.
Практич. работа. 8	Тема: Коллекции, индексаторы, словари, LINQ, сериализация JSON Задание 1. Коллекции Задание 2. Словари Задание 3. LINQ Задание 4. Сериализация JSON

5. Контроль результатов обучения по дисциплине

Текущий контроль и промежуточная аттестация осуществляется в соответствии с «Положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по основным профессиональным образовательным программам высшего образования в ФГАОУ ВО «КФУ им. В. И. Вернадского» и «Порядком применения балльно-рейтинговой системы оценивания успеваемости обучающихся по основным профессиональным образовательным программам высшего образования ФГАОУ ВО «КФУ им. В. И. Вернадского».

Вид промежуточной аттестации – экзамен

Форма проведения промежуточной аттестации – устный экзамен

Ввид оценочных средств промежуточной аттестации - экзаменационные билеты, тесты, профессиональные задачи и пр.

Материалы, используемые для контроля результатов обучения по дисциплине, приводятся в Фонде оценочных средств по дисциплине

6. Учебно-методическое обеспечение

6.1. Основная учебная литература:

1. Вафин, Р. Р. Объектно-ориентированное программирование на C#.NET : учебно-методическое пособие / Р. Р. Вафин. — Казань : КНИТУ-КАИ, 2020. — 96 с. — ISBN 978-5-7579-2431-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/264920> (дата обращения: 10.07.2024). — Режим доступа: для авториз. пользователей.
2. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# / Л. А. Залогова. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 192 с. — ISBN 978-5-507-46825-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/321197> (дата обращения: 10.07.2024). — Режим доступа: для авториз. пользователей.
3. Кудрина, Е В. Основы алгоритмизации и программирования на языке C#: учебное пособие для бакалавриата и специалитета / Е. В. Кудрина, М. В. Огнева. Издательство Юрайт, 2019. - 322 с. - (Серия : Бакалавр. Академический курс).

6.2. Дополнительная учебная литература:

1. Корпорация Microsoft. C#. Спецификация языка. (входит в комплект поставки Visual Studio и находится в каталоге Program Files\Microsoft Visual Studio)
2. Шилдт Герберт. Полный справочник по C#. Пер. с англ. - М.: Издательский дом "Вильямс", 2008. - 752с.: ил.
3. Ватсон Б. C# 4.0 на примерах. — СПб.: БХ В-Петербург, 2011. — 608 с.: ил.
4. Зиборов В. В. Visual C# 2012 на примерах. — СПб.: БХВ-Петербург, 2013. — 480 с.: ил.
5. Гриффитс И. Программирование на C# 5.0. (пер. с англ. М. А. Райтмана) . — М. : Эксмо, 2014. — 1136 с.
6. Нэш, Трей. C# 2010: ускоренный курс для профессионалов. : Пер. с англ. — М. : ООО “И.Д.Вильямс”, 2010. — 592 с.
7. Троелсен. Эндрю. Язык программирования C# 2010 и платформа .NET 4.0. 5-е изд. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2011. — 1392 с. : ил. — Парал. тит. англ.
8. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#. 3-е изд. — СПб.: Питер, 2012. — 928 с.: ил.
9. Климов А. П. C#. Советы программистам. — СПб.: БХВ-Петербург, 2008.

6.3. Методические материалы:

1. Лекции в электронном виде (Милюков В.В.)
2. Официальные презентации Microsoft по курсу С# (курс 10266)
3. Методические указания к выполнению практических работ (Милюков В.В.) - 2024г.

6.4. Перечень ресурсов информационно-телекоммуникационной сети "Интернет":

1. Справочник по С#. Корпорация Microsoft.
<http://msdn.microsoft.com/ru-ru/library/618ayhy6.aspx>
2. Биллиг В.А. Основы программирования на С#. Интернет-университет информационных технологий. <http://www.intuit.ru/studies/courses/2247/18/info>
3. Павловская Т. Программирование на языке высокого уровня С#. <http://www.intuit.ru/studies/courses/629/485/info>
5. Руководство по программированию на С#. Корпорация Microsoft.
<http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>
6. Материалы сайта Metanit.com.

Профессиональные базы данных и информационные справочные системы:

1. Министерство науки и высшего образования Российской Федерации
<https://minobrnauki.gov.ru/>
2. Федеральный портал «Российское образование» <http://www.edu.ru/>

7. Перечень информационных технологий, используемых при осуществлении образовательного процесса

Программная среда Visual Studio (2022 или выше) (базовая версия Community не требует лицензирования).

Электронно-образовательный ресурс по дисциплине, размещенный в ЭИОС ФГАОУ ВО «КФУ им В.И Вернадского» – режим доступа: <https://moodle.cfuv.ru/>

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса

Реализация учебной дисциплины требует наличия лекционных кабинетов - 1; вычислительных классов - 1.

Оборудование лекционного кабинета: доска для маркеров, мультимедийный проектор.