

Б1.О.17 Объектно-ориентированное программирование

Цели дисциплины

Целями освоения дисциплины Объектно-ориентированное программирование является формирование общепрофессиональных компетенций будущих специалистов в области информационной безопасности. Опираясь на знания, полученные при изучении курсов высшей математики и основ программирования, рассмотреть новые подходы и технологии программирования, а также развить навыки проектирования и реализации алгоритмов решения прикладных задач.

Задачи дисциплины

Основными задачами изучения дисциплины являются:

- знакомство с методами объектно-ориентированного программирования как наиболее распространенными и эффективными методами разработки программных продуктов;
- формирование навыков разработки приложений на основе объектно-ориентированного подхода.

Формируемые компетенции и индикаторы их достижения по дисциплине

Код компетенции	Содержание компетенции	Код и наименование индикатора достижения компетенции
ОПК-7	Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности;	ОПК-7.1 - Знает основные технологии разработки программных средств для решения задач в области профессиональной деятельности; ОПК-7.2 - Умеет применять языки программирования для решения профессиональных задач; ОПК-7.3 - Владеет навыками выбора и разработки алгоритмов при решении типовых задач программирования, а также навыками разработки и тестирования программ по поставленной спецификации.

Содержание разделов дисциплины

4 семестр

Тема 1. Перегрузка операторов

Перегрузка унарных и бинарных операций. Выполнение операций над значениями встроенных C#-типов. Перегрузка операторов отношения. Перегрузка операторов true и false. Перегрузка логических операторов. Включение операторов, действующих по сокращенной схеме вычислений. Операторы преобразования.

Тема 2. Пространства имен, препроцессор, сборки

Предотвращение конфликтов имен с помощью пространства имен. Директивы using, Аддитивный характер пространств имен. Вложенные пространства имен. Применение описателя псевдонима пространства имен :: Препроцессор. Директивы препроцессора.

Тема 3. Динамическая идентификация типов, рефлексия и атрибуты

Динамическая идентификация типов. Рефлексия. Применение рефлексии. Основы применения атрибутов: создание, присоединение, получение атрибутов. Встроенные атрибуты.

Тема 4. Небезопасный код, указатели, обнуляемые типы

Обнуляемые типы. Частичные типы. Частичные методы. Создание объектов динамического типа.

Тема 5. Организация системы ввода- вывода в C# на потоках

Байтовые и символьные потоки. Классы потоков. Классы-оболочки. Консольный ввод-вывод. Класс FileStream и байтовый ввод-вывод в файл. Символьный ввод-вывод в файл. Класс BinaryReader. Файлы с произвольным доступом. Применение класса MemoryStream. Классы File и FileInfo.

5 семестр

Тема 1. Основы языка интегрированных запросов LINQ

Основы LINQ. Отбор запрашиваемых значений. Группирование результатов запросов. Анонимные типы. Режимы выполнения запросов: отложенный и немедленный. Деревья выражений.

Тема 2. Сериализация и десериализация

Механизмы сериализации и десериализации объектов в C#. Форматы JSON (System.Text.Json), XML (XmlSerializer, XmlDocument) и бинарная сериализация. Атрибуты управления процессом сериализации, обработки циклических ссылок и работе с конфигурационными файлами.

Тема 3. Основы работы с базами данных

Изучение технологий доступа к данным в .NET. Основы ADO.NET (SqlConnection, SqlCommand, SqlDataReader), параметризованные запросы и защита от SQL-инъекций.

Тема 4. Многопоточное программирование

Модель многопоточности в C#. Классы Thread и ThreadPool, механизмы синхронизации (lock, Monitor, Mutex, Semaphore). Анализ типичных проблем параллельного выполнения: гонки данных, взаимные блокировки, race conditions.

Тема 5. Асинхронное программирование

Современные средства асинхронного и параллельного программирования в .NET. Тип Task, ключевые слова async и await, отмена операций через CancellationToken. Библиотека параллельных задач (TPL) и PLINQ.

Тема 6. Паттерны проектирования

Классические паттерны проектирования и их реализации на языке C#: порождающие (Singleton, Factory Method), структурные (Adapter, Decorator) и поведенческие (Observer, Strategy, Command) паттерны. Принцип Dependency Injection. Использование встроенного DI-контейнера .NET