

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Одеська національна академія зв'язку ім. О.С. Попова**

**Кафедра інформаційних технологій**

**“ ЗАТВЕРДЖУЮ ”**

Ректор ОНАЗ ім. О. С. Попова

**П .П. Воробієнко**

“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 р.

**Об'єктно-орієнтоване програмування**

(назва навчальної дисципліни)

**ПРОГРАМА**

**навчальної дисципліни**

**підготовки бакалаврів**

(назва освітньо-кваліфікаційного рівня)

**спеціальності 121 Інженерія програмного забезпечення**

Одеса  
2020 рік

РОЗРОБЛЕНО ТА ВНЕСЕНО: Одеська національна академія зв'язку ім.О.С.Попова  
(повне найменування вищого навчального закладу)

РОЗРОБНИК ПРОГРАМИ: ст. викл. Косирева Л.А.

\_\_\_\_\_

Програму розглянуто і схвалено на засіданні кафедри ІТ

Протокол № від “\_\_” \_\_\_\_\_ 2020 року Зав. Кафедрою ІТ \_\_\_\_\_

Програму погоджено з кафедрами:

\_\_\_\_\_ / \_\_\_\_\_ /

\_\_\_\_\_ / \_\_\_\_\_ /

Програму розглянуто і схвалено Радою навчально-наукового інституту інфокомунікацій та програмної інженерії

Протокол № від “\_\_” \_\_\_\_\_ 2020 р. Директор ННІ ІПІ \_\_\_\_\_ проф. І.В. Стрелковська

Програму розглянуто і схвалено методичною радою Одеської національної академії зв'язку ім. О.С. Попова

Протокол № від “\_\_” \_\_\_\_\_ 2020 року Голова ради \_\_\_\_\_ / \_\_\_\_\_ /

## ВСТУП

Програма вивчення нормативної навчальної дисципліни “Об’єктно-орієнтоване програмування” складена відповідно до освітньо-професійної програми підготовки бакалаврів спеціальності 121 Інженерія програмного забезпечення.

Курс «Об’єктно-орієнтоване програмування» призначений для вивчення сучасних технологій програмування з використанням об’єктно-орієнтованої парадигми.

Курс надає майбутнім фахівцям теоретичні знання та практичні навички з об’єктно-орієнтованого програмування та дозволяє сформувати найважливіші практичні вміння з використанням актуальних підходів до розробки програмних засобів в різних галузях сучасного суспільства.

Вивчається одна з найпоширеніших у світі мов об’єктно-орієнтованого програмування – мова C#. На практичних та лабораторних заняттях створюються консольні додатки мовою C# за допомогою об’єктно-орієнтованого підходу.

### 1. Мета та завдання навчальної дисципліни

*Метою* викладання навчальної дисципліни є формування базових знань про сучасні технології об’єктно-орієнтованого програмування, практичне засвоєння основних понять і концепцій об’єктно-орієнтованого програмування.

*Основними завданнями* дисципліни є набуття студентами теоретичних знань та практичних навичок принципів об’єктно-орієнтованої декомпозиції складних об’єктів, розробки проектів та написання на їх основі ефективних програм з використанням всіх переваг об’єктно-орієнтованої парадигми проектування та програмування.

Цілі курсу:

- набути теоретичні та практичні знання, вміння та навички програмування мовою C#;
- вивчити фундаментальні концепції об’єктно-орієнтованого програмування;
- оволодіти методами та технологіями об’єктно-орієнтованого аналізу і проектування;
- здобути вміння створювати класи та об’єкти із застосуванням об’єктно-орієнтованих технологій програмування;
- оволодіти методами оброблення виняткових ситуацій;
- набути практичні навички створення програмних продуктів з використанням об’єктно-орієнтованого програмування мовою C#;
- ознайомитись зі сучасними технологіями об’єктно-орієнтованого програмування.

В результаті успішного засвоєння навчальної дисципліни студент повинен продемонструвати такі результати навчання:

*знання:*

- основних принципів об’єктно-орієнтованого програмування;
- основних понять об’єктно-орієнтованого програмування, таких як клас, об’єкт, поле, метод, інкапсуляція, успадкування, поліморфізм;
- елементів алгоритмічної мови програмування C# (алфавіт мови, типи даних, правила записування виразів);

- інструментів об'єктно-орієнтованого програмування в мові програмування C#;
- підходів об'єктно-орієнтованого аналізу та дизайну;
- засобів обробки виняткових ситуацій мови C#.

*уміння:*

- створювати консольні додатки мовою C#;
- створювати ієрархічні моделі мовою програмування C#;
- створювати бібліотеки класів C#;
- проектувати предметну область за допомогою класів та зв'язків між класами;
- опрацьовувати виняткові ситуації, використовуючи відповідні оператори і класи C#.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів **компетентностей:**

*загальних:*

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність застосовувати знання у практичних ситуаціях;
- знання та розуміння предметної області та розуміння професійної діяльності;
- здатність вчитися й оволодівати сучасними знаннями;
- здатність до пошуку, оброблення та аналізу інформації з різних джерел;
- здатність генерувати нові ідеї (креативність);
- здатність працювати в команді;
- здатність приймати обґрунтовані рішення;
- здатність оцінювати та забезпечувати якість виконуваних робіт.

*фахових:*

- здатність до системного мислення, застосування методології системного аналізу для дослідження складних проблем різної природи, методів формалізації та розв'язування системних задач, що мають суперечливі цілі, невизначеності та ризики;
- здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління;
- здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника;
- здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення;
- здатність до аналізу та функціонального моделювання бізнес-процесів, побудови та практичного застосування функціональних моделей організаційно-економічних і виробничо-технічних систем, методів оцінювання ризиків їх проектування;
- здатність використовувати технології та патерни програмування для вирішення найбільш розповсюджених задач; модифікувати існуючі патерни для вирішення конкретної задачі при створенні програмної системи;
- здатність забезпечувати якість комп'ютерних систем та оцінювати їх показники якості з використанням відповідних моделей та засобів на всіх етапах розробки;

- здатність проектувати та реалізовувати програмне забезпечення інформаційних комплексів для різноманітних операційних систем та програмних фреймворків із застосуванням сучасних бібліотек крос-платформного програмування, проводити тестування на декількох апаратних платформах, впроваджувати і підтримувати роботу інформаційних систем на сучасних платформах.

Результати навчання даної дисципліни деталізують такі **програмні результати навчання:**

- Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.
- Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.
- Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.
- Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.
- Розробляти програмне забезпечення систем різних архітектур з використанням відповідних сучасних технологій, патернів та сервісів, забезпечувати розподілені обчислення та розробляти документи та форми в web, описувати та розробляти web-сервіси.

Дисципліна «Об'єктно-орієнтоване програмування» викладається в семестрах 1.2, 2.1 для студентів 1 та 2 курсів, які навчаються за напрямом бакалаврської підготовки за спеціальністю 121 Інженерія програмного забезпечення.

На вивчення навчальної дисципліни відводиться **240** годин / **8** кредитів ECTS.

## 2. Інформаційний обсяг навчальної дисципліни

### Змістовий модуль 1. Консольні прикладні проекти C#.

*Тема 1.* Співвідношення між мовами програмування C, C++ і C#. Вікно консолі. Запис в консоль та читання даних з консолі. Типи даних та оператори в C#. Тестування та налагодження програм мовою C#.

Масиви. Одновимірні і багатовимірні масиви. Масиви масивів. Цикли `foreach` у масивах.

*Тема 2.* Функції у мові C#. Передача аргументів за значенням і за посиланням у функції. Використання модифікатору `out`. Передача і повернення структур. Оголошення змінної типу рядка. Ініціалізація. Клас `string` та його методи.

### Змістовий модуль 2. Класи та об'єкти у мові C#.

*Тема 3.* Класи, об'єкти та властивості (`properties`) у мові C#. Доступ до членів класу. Методи, їх параметри. Ініціалізація екземплярів класів. Специфікатори доступу. Конструктори та деструктори. Ключове слово `this`. Перевантаження операцій. Формат перевантаження для унарного та бінарного операторів.

### Змістовий модуль 3. Успадковування та інкапсуляція, поліморфізм.

*Тема 4.* Успадковування від класу `Object`. Ієрархії нащадків. Області видимості. Управління механізмом доступу до членів базового класу. Особливості використання конструкторів і деструкторів при реалізації механізму успадкування. Виклик затінених методів і базового конструктора.

*Тема 5.* Перекриття методів. Віртуальні методи класів. Використання модифікатора типу `static`. Поняття поліморфного коду. Поліморфізм та пізні зв'язування. Використання вказівників на похідні типи як основа динамічного поліморфізму. Абстрактні класи. Використання ключового слова `sealed` для відвертання спадкоємства.

### Змістовий модуль 4. Об'єктно-орієнтоване програмування.

*Тема 6.* Візуальний інтерфейс середовища Visual Studio .NET. Палітра компонент. Вікно властивостей. Створення віконного додатку. Властивості та методи форми. Події, пов'язані з формою. Основні компоненти.

*Тема 7.* Додатки з кількома формами, додатки MDI. Компоненти-контейнери. Створення і використання динамічних компонент. Підключення подієвих функцій до динамічних компонент. Події клавіатури.

### Змістовий модуль 5. Робота з файлами.

*Тема 8.* Класи для введення і виведення даних. Класи `File` і `Directory`. Клас `FileInfo`. Клас `DirectoryInfo`. Імена шляху і відносні шляхи. Клас `StreamWriter`. Об'єкти цього класу. Клас `StreamReader`. Об'єкти цього класу. Клас `FileStream`. Об'єкти цього класу. Позиція усередині файлу. Читання і запис у файл. Робота з `BinaryWriter` і `BinaryReader`. Компоненти `dataGridView`.

### Змістовий модуль 6. Списки.

*Тема 9.* Списки. Черга, стік. Клас `List`. Клас `Stack`. Клас універсальної колекції `Collection`. Клас `DateTime` для роботи з датами та часом.

### Змістовий модуль 7. Обробка виняткових ситуацій.

*Тема 10.* Виняткові ситуації і відповідні їм класи у C#. Оператори `try` та `finally`. Використання декількох `catch`-блоків. Ієрархія класів виняткових ситуацій.

### Змістовий модуль 8. Побудова графічних зображень у C#.

*Тема 11.* Графічний інструментарій C#. Простір імен `System.Drawing`. Клас `Graphics`.

*Тема 12.* Ієрархії класів. Побудова графічних зображень з використанням ієрархії класів. Приклад ієрархії класів для геометричних зображень. Візуалізація руху.

### Змістовий модуль 9. Делегати. Події. Інтерфейси.

*Тема 13.* Статичні та нестатичні методи. Делегати. Події. Додавання обробників подій. Обробка черги повідомлень.

*Тема 14.* Оголошення та реалізація інтерфейсів. Використання інтерфейсних посилань. Інтерфейсні властивості. Перевантаження інтерфейсних методів. Успадковування інтерфейсів. Явна реалізація членів інтерфейсу. Вибір між інтерфейсом і абстрактним класом. Додаток на програмування подій.

**Змістовий модуль 10.** Створення бібліотек класів.

*Тема 15.* Створення і використання бібліотек класів. Приклади побудови бібліотек. Клас *Component*. Побудова нащадків візуальних компонент.

**Змістовий модуль 11.** Керування поведінкою компонент.

*Тема 16.* Серіалізація. Сервіси режиму розробки проекту. Обробка подій, пов'язаних із змінами компонентів та їх властивостей. Доступ до компонентів через їх імена та типи.

*Тема 17.* Управління подіями компонента. Керування процесами зображення, розміщення та оновлення компонента.

### 3. Рекомендована література

1. Фаронов В.В. Программирование на языке С#. – СПб.: Питер, 2007. – 240 с.
2. Герберт Шилдт Полный справочник по С# - И.Д. Вильямс, 2004.
3. Агуров П.В. С#. Разработка компонентов в Ms Visual Studio 2005/2008. – СПб.: БХВ-Петербург, 2008. – 480 с.
4. Лабор В.В. Си Шарп. Создание приложений для Windows. – Минск, Харвест, 2003.
5. Павловская Т. А. С#. Программирование на языке высокого уровня. Учебник для вузов. – СПб.: Питер, 2009. – 432 с.
6. Сборник задач по программированию / Глазунова Л.В. и др. – Одесса: ОНАС, 2011.
7. Фленов М.Е. Библия С#. – СПб.: БХВ-Петербург, 2009. – 560 с.
8. Троелсен Э. Язык программирования С# 2008 и платформа .NET 3.5, 4-е изд. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2010. — 1344 с.
9. Бишоп Дж., Хорспул Н. С# в кратком изложении. – М.: БИНОМ, 2005. – 472 с .
10. Абрамян М.Э. Visual С# на примерах. – СПб.: БХВ-Петербург, 2008. – 496 с.
11. Н.Б. Культин. С# в задачах и примерах. – СПб.: БХВ-Петербург, 2007. – 240 с.
12. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / Г. Буч ; пер. с англ. - 2-е изд. - М. : Бином; СПб. : Невский диалект, 1998.
13. Усов Т. М. Введение в объектно-ориентированное программирование с примерами на С#, 2019. — Режим доступа: <https://newobj.ru/oop>
14. Нейгел К., Ивьен Б., Глинн Д., Уотсон К., Скиннер М. С# 2008 и платформа .NET 3.5 для профессионалов. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2009. - 1392 с.

### 4. Інформаційні ресурси

1. <http://www.it.onat.edu.ua/> – Сайт кафедри інформаційних технологій, на якому розміщено робочі матеріали з курсу.
2. <http://msdn.microsoft.com/> – Бібліотека класов платформи .NET Framework.
3. <http://www.visualstudio.com/> – Ms Visual Studio.

4. <https://metanit.com/sharp/tutorial>
5. Електронний варіант лекцій
6. Методичні вказівки до виконання лабораторних робіт та практичних занять в електронному вигляді

#### **4. Форма підсумкового контролю успішності навчання:**

Залік, іспит, курсова робота.

#### **5. Засоби діагностики успішності навчання**

Діагностика знань студентів здійснюється з допомогою:

- усного опитування;
- тестування;
- лабораторних робіт;
- екзаменаційних завдань.