

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ
СПРАВ
Кафедра кібербезпеки та DATA-технологій, факультет №6**

ПРОГРАМА

**навчальної дисципліни «Об'єктно-орієнтоване програмування»
обов'язкових компонент освітньої програми
першого (бакалаврського) рівня вищої освіти**

**Спеціальність: 125 «Кібербезпека»
(«Безпека інформаційних та комунікаційних систем»)**

Харків 2023

ЗАТВЕРДЖЕНО

Науково-методичною радою
Харківського національного
університету внутрішніх справ
Протокол від 30.01.2023 № 1

СХВАЛЕНО

Вченою радою факультету № 6
Протокол від 18.01.2023 № 1

ПОГОДЖЕНО

Секцією Науково-методичної ради
ХНУВС з технічних дисциплін
Протокол від 27.01.2023 № 1

Розглянуто на засіданні кафедри кібербезпеки та DATA-технологій
факультету № 6 (протокол від 13.01.2023 № 1)

Розробники:

Професор кафедри, к.т.н., доцент; Струков В. М.;

Старший викладач, Цуранов М.В.;

Рецензенти:

- 1. Певнєв В.Я., д.т.н., доцент, професор кафедри комп'ютерних систем, мереж та кібербезпеки факультету радіоелектроніки, комп'ютерних систем та інфокомунікацій НАУ «ХАІ» ім. М.Є. Жуковського;*
- 2. Світличний В.А., к.т.н., доцент, доцент кафедри протидії кіберзлочинності факультету №4 ХНУВС.*

ПОЯСНЮВАЛЬНА ЗАПИСКА

Програма обов'язкової навчальної дисципліни «Об'єктно-орієнтоване програмування» складена відповідно до освітньої програми першого рівня вищої освіти 125 «Кібербезпека» («Безпека інформаційних та комунікаційних систем»)

Предметом вивчення навчальної дисципліни «Об'єктно-орієнтоване програмування» є принципи об'єктно-орієнтованого моделювання і програмування, а також методи їх використання при розробленні додатків.

Міждисциплінарні зв'язки : науковий фундамент дисципліни пов'язаний з такими дисциплінами як «Інформаційні технології», «Архітектура та структурно-логічні основи ЕОМ», «Алгоритмізація та програмування» та ін.

Програма навчальної дисципліни складається з таких тем:

ТЕМА № 1. Основні поняття об'єктно-орієнтованого програмування.

ТЕМА № 2. Основи уніфікованої мови моделювання UML.

ТЕМА № 3. Основи моделювання поведінки системи в UML.

ТЕМА № 4. Моделювання класів в UML.

ТЕМА №5. Архітектура платформи .NET та програмування для .NET Framework

ТЕМА № 6. Синтаксис мови C#. Основні операції, оператори та типи даних C#

ТЕМА №7. Класи та об'єкти. Перетворення типів даних

ТЕМА №8. Спадкування. Інтерфейси та їх використання.

1. Мета та завдання навчальної дисципліни

1.1. Метою викладання навчальної дисципліни "Об'єктно-орієнтоване програмування" є засвоєння необхідних знань з основ об'єктно-орієнтованого моделювання і програмування, а також формування практичних навичок щодо розроблення моделей програмних систем і програмних додатків з використанням об'єктно-орієнтованого підходу.

1.2. надання здобувачам вищої освіти знань з технології об'єктно-орієнтованого моделювання з використанням уніфікованої мови UML і об'єктно-орієнтованого програмування з використанням мови C# і набуття практичних навичок її застосування.

1.3. Згідно з освітньою програмою здобувачі вищої освіти повинні:

знати:

- проблематику курсу «Об'єктно-орієнтоване програмування» і його основних розділів;
- базові поняття і визначення; принципи ООП (інкапсуляція, спадкування, поліморфізм); як оголошуються класи, створюються, ініціалізуються і руйнуються об'єкти класів; типи відношень між класами; порядок проектування класів; життєвий цикл об'єктів;
- принципи моделювання поведінки програмної системи, моделювання класів, взаємодії та поведінки об'єктів в UML;
- види і призначення діаграм UML; відмінність між абстрактними і конкретними класами;

- ієрархію класів потоків введення/виведення; поняття об'єктно-орієнтованого аналізу,
- моделювання, проектування та програмування; способи розроблення моделей програмної системи в середовище Rational Rose.

вміти:

- організовувати власну професійну діяльність, обирати оптимальні методи та способи розв'язування складних спеціалізованих задач та практичних проблем у професійній діяльності, оцінювати їхню ефективність;
- використовувати результати самостійного пошуку, аналізу та синтезу інформації з різних джерел для ефективного рішення спеціалізованих задач професійної діяльності;
- аналізувати, аргументувати, приймати рішення при розв'язанні складних спеціалізованих задач та практичних проблем у професійній діяльності, які характеризуються комплексністю та неповною визначеністю умов, відповідати за прийняті рішення;
- адаптуватися в умовах часткої зміни технологій професійної діяльності, прогнозувати кінцевий результат;
- виконувати аналіз та декомпозицію інформаційно-телекомунікаційних систем;
- виконувати аналіз зв'язків між інформаційними процесами на віддалених обчислювальних системах;
- вирішувати завдання захисту програм та інформації, що обробляється в інформаційно-телекомунікаційних системах програмно-апаратними засобами та давати оцінку результативності якості прийнятих рішень;
- вирішувати задачі аналізу програмного коду на наявність можливих загроз.

1.4. Форма підсумкового контролю – залік

На вивчення навчальної дисципліни відводиться 150 годин /5 кредитів ECTS.

1.5. Програмні компетентності:

Програмні компетентності, які формуються при вивченні навчальної дисципліни:		
Інтегральна компетентність	Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі забезпечення інформаційної безпеки та/або кібербезпеки, що характеризується комплексністю та неповною визначеністю умов.	
Загальні компетентності (ЗК)	ЗК 1	Здатність застосовувати знання у практичних ситуаціях.
	ЗК 5	Здатність до пошуку, оброблення та аналізу інформації.

Спеціальні (фахові, предметні) компетентності (СК)	Ф К 2	Здатність до використання інформаційно-комунікаційних технологій, сучасних методів і моделей інформаційної безпеки та/або кібербезпеки.
	Ф К 3	Здатність до використання програмних та програмно-апаратних комплексів засобів захисту інформації в інформаційно-телекомунікаційних (автоматизованих) системах.
Програмні результати навчання	П РН 2.	організовувати власну професійну діяльність, обирати оптимальні методи та способи розв'язування складних спеціалізованих задач та практичних проблем у професійній діяльності, оцінювати їхню ефективність
	П РН 4.	аналізувати, аргументувати, приймати рішення при розв'язанні складних спеціалізованих задач та практичних проблем у професійній діяльності, які характеризуються комплексністю та неповною визначеністю умов, відповідати за прийняті рішення
	П РН 10.	виконувати аналіз та декомпозицію інформаційнотелекомунікаційних систем.
	П РН 11.	виконувати аналіз зв'язків між інформаційними процесами на віддалених обчислювальних системах.
	П РН 14.	вирішувати завдання захисту програм та інформації, що обробляється в інформаційно-телекомунікаційних системах програмно-апаратними засобами та давати оцінку результативності якості прийнятих рішень
	П РН 17.	забезпечувати процеси захисту та функціонування інформаційнотелекомунікаційних (автоматизованих) систем на основі практик, навичок та знань, щодо структурних (структурно-логічних) схем, топології мережі, сучасних архітектур та моделей захисту електронних інформаційних ресурсів із відображенням взаємозв'язків та інформаційних потоків,

	процесів для внутрішніх і віддалених компонент
П РН 18.	використовувати програмні та програмно-апаратні комплекси захисту інформаційних ресурсів
П РН 20.	забезпечувати функціонування спеціального програмного забезпечення щодо захисту інформації від руйнуючих програмних впливів, руйнуючих кодів у інформаційно-телекомунікаційних системах
П РН 23.	реалізовувати заходи з протидії отриманню несанкціонованого доступу до інформаційних ресурсів і процесів в інформаційних та інформаційно-телекомунікаційних (автоматизованих) системах
П РН 24.	вирішувати задачі управління доступом до інформаційних ресурсів та процесів в інформаційних та інформаційно-телекомунікаційних (автоматизованих) системах на основі моделей управління доступом (мандатних, дискреційних, рольових
П РН 26.	впроваджувати заходи та забезпечувати реалізацію процесів попередження отриманню несанкціонованого доступу і захисту інформаційних, інформаційно-телекомунікаційних (автоматизованих) систем на основі еталонної моделі взаємодії відкритих систем
П РН 27.	вирішувати задачі захисту потоків даних в інформаційних, інформаційно-телекомунікаційних (автоматизованих) системах.
П РН 47.	вирішувати задачі захисту інформації, що обробляється в інформаційно-телекомунікаційних системах з використанням сучасних методів та засобів криптографічного захисту інформації.
П РН 48.	виконувати впровадження та підтримку систем виявлення вторгнень та

	використовувати компоненти криптографічного захисту для забезпечення необхідного рівня захищеності інформації в інформаційно-телекомунікаційних системах.
П РН 49.	забезпечувати належне функціонування системи моніторингу інформаційних ресурсів і процесів в інформаційно-телекомунікаційних системах.
П РН 51.	підтримувати працездатність та забезпечувати конфігурування систем виявлення вторгнень в інформаційно-телекомунікаційних системах.
П РН 53.	вирішувати задачі аналізу програмного коду на наявність можливих загроз
П РН 57.	проводити кримінальний аналіз при вирішенні завдань поліцейської діяльності

2. Короткий опис змісту навчальної дисципліни.

Тема № 1. Основні поняття об'єктно-орієнтованого програмування.

Вступ до ООП. Поняття класу. Атрибути. Об'єкти. фундаментальні властивості об'єктно-орієнтованого програмування. Прості та складні програмні системи. Декомпозиція програмних систем. Способи декомпозиції. Об'єктно-орієнтована декомпозиція. Поняття об'єкта. Характеристики об'єкта. Поняття класу. Співвідношення між класом та його об'єктом. Об'єктно-орієнтований аналіз та його мета. Поняття предметної області. Головні види вимог до програмної системи. Об'єктно-орієнтоване проектування. Елементи архітектури додатку. Визначення класів предметної області. Принципи проекту класів. Об'єктно-орієнтоване програмування. Принципи об'єктно-орієнтованого підходу: абстракція, інкапсуляція, ієрархія, поліморфізм.

Тема № 2. Основи уніфікованої мови моделювання UML.

Загальна характеристика UML. Можливості специфікації, візуалізації, конструювання і документування програмних систем. Концептуальна модель UML. Архітектурний базис UML. Базові поняття, якими оперує мова. Основні сутності UML. Відношення. Відношення залежності, асоціації, узагальнення та реалізації. Основні типи діаграм UML, їх призначення. Представлення моделі. Представлення прецедентів, логічне представлення, представлення процесів, представлення компонентів та представлення розміщення.

Тема № 3. Основи моделювання поведінки системи в UML.

Прецеденти використання системи. Актори. Межі системи. Діаграма прецедентів (Use Case Diagrams). Організація прецедентів. Типи відношень між прецедентами і акторами. Створення прецедентів Case-засобом Rational Rose. Елементи інтерфейсу Rational Rose. Специфікації прецедентів. Структура шаблону специфікації за вимогами Rational Unified Process. Діаграми видів діяльності. Діяльність, переходи, елементи синхронізації, рішення, доріжки. Попередній архітектурний аналіз системи. Набір угод моделювання. Реалізації. Ідентифікація головних абстракцій системи. Класи аналізу.

Тема № 4. Моделювання класів в UML.

Поняття класу і об'єкту. Зображення класу. Специфікація класу, атрибути, операції. Кратність класу. Асоціації між класами. Назва асоціації, роль полюса асоціації. Кратність полюса. Самоасоціація. Асоційований клас. Агрегація та композиція між класами. Приклади. Узагальнення та залежності між класами. Відношення успадкування. Суперкласи. Абстрактний клас. Залежності між класами. Розширення UML для моделей класів програмування і бізнесу. Стереотипи. Класи керування, класи-сутності та межові класи. Діаграма класів.

Тема № 5. Архітектура платформи .NET та програмування для .NET Framework

Архітектура платформи .NET та мови програмування для .NET Framework. Основи програмування мовою C#: змінні та типи даних (величини та посилання), структура програми та точка входу (метод Main). Клас як тип даних. Базовий клас System.Object.

Організація введення та виведення даних у консольних програмах та засоби управління консоллю (базові можливості зміни властивостей). Приклад організації введення числових даних з контролем помилок.

Загальна характеристика стандартної бібліотеки класів та просторів імен.

Тема № 6. Синтаксис мови C#. Основні операції, оператори та типи даних C#

Реалізація класів в програмі: опис класів, полів даних, методів (функційних членів класу). Ініціалізація полів даних та помилки, що виникають при використанні неініційованих даних.

Зони видимості та специфікатори доступу public та private. Огляд принципу інкапсуляції даних.

Передача параметрів (даних та посилань) в методи та дані, що повертаються. Перевантаження методів для розширення функціональності класів. Управляючі оператори, оператори та вирази в мові C#: синтаксис та приклади використання.

Тема № 7. Класи та об'єкти. Перетворення типів даних

Перерахування, структури та клас string. Рядкові константи та їх представлення в програмі. Коди символів та спрощення запису рядків-констант за допомогою символу @.

Створення об'єктів (екземплярів класу) оператором new. Статичні члени класу. Константні члени класу.

Поняття базового класу та спадкування характеристик між класами. Приведення типів: оператори as та is. Боксування (упаковка).

Тема № 8. Спадкування. Інтерфейси та їх використання.

Спадкування реалізації та спадкування інтерфейсу. Віртуальні методи. Закриття методів. Виклик базових версій методів. Абстрактні за закриті класи та методи (ключові слова abstract та sealed). Конструктори в похідних класах.

Огляд модифікаторів для класів, методів та полів.

Визначення та реалізація інтерфейсів. Похідні інтерфейси. Приклади використання інтерфейсів (IClonable, IComparable, IDisposable). Зауваження стосовно множинного спадкування.

Спадкування, поліморфізм та інкапсуляція як базові аспекти ООП – підведення підсумків стосовно концепцій об'єктно-орієнтованого підходу.

3. Рекомендована література (основна, допоміжна), інформаційні ресурси в Інтернеті

Основна література

1. Бублик В.В., Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТ-книга, 2015. – 624 с.: іл.
2. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник / В.С. Авраменко, А.С. Авраменко. – Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. – 434 с.: іл.
3. Коноваленко І.В. Програмування мовою C# 6.0. Тернопіль, ТНТУ, 2016. 227 с.
4. Н. Schildt, "C# 4.0 The Complete Reference McGraw Hill Education; 1st edition" – 984 p., 2017.
5. J. Sharp, "Microsoft Visual C# Step by Step", Pearson Education – 878p., 2018.

Допоміжна література

1. Booch B. Object-Oriented Analysis and Design with Applications. 3rd edition. Addison-Wesley. 720 p.
2. Kimmel P. UML demystified. A self-teaching guide. McGraw-Hill Education. 235 p.
3. A. Troelsen, "Pro C# 7 With .NET and .NET Core" - Apress, — 1372 p., 2017.
4. J. Albahari, B. Albahari, "C# 7.0 in a Nutshell", O'Reilly Media— 1090 p., 2017.
5. Інструментальні програмні засоби розробки ІУС. Методичні вказівки до виконання лабораторних робіт. / уклад.: К.І. Київська–Київ: КНУБА, 2018. – 40 с

Інформаційні ресурси в Інтернеті

1. C# Programming. Yellow Book [Ел. ресурс]. URL: <https://www.robmiles.com/c-yellow-book>
2. Secure Coding Guidelines for .NET [Ел. ресурс]. URL: <https://docs.microsoft.com/en-us/dotnet/standard/security/secure-coding-guidelines>
3. C# Reference, сайт розробників MSDN. [Електронний ресурс] – Режим доступу: <https://docs.microsoft.com/uk-ua/dotnet/csharp/language/reference/index>

4. Об'єктно-орієнтоване програмування. Частина 1. Основи об'єктно-орієнтованого програмування на мові C#: Навчальний посібник. / Д.В. Настенко, А. Б. Нестерко. – К.: НТУУ «КПІ», 2016. - 76с. [Електронний ресурс]. – Режим доступу https://ela.kpi.ua/bitstream/123456789/16671/1/OOP_manual.pdf

4. Засоби оцінювання здобувачів вищої освіти.

1. Перелічіть основні властивості ООП.
2. Розкрийте зміст поняття – інкапсуляція класу.
3. Дайте визначення актора.
4. Дайте визначення класу й об'єкта.
5. Дайте визначення прецеденту
6. Опишіть шаблон специфікації прецеденту.
7. Для чого використовують діаграму видів діяльності?
8. Як утворити кооперацію реалізації прецеденту?
9. Коротко охарактеризуйте специфікацію атрибута класу.
10. Коротко охарактеризуйте специфікацію операції класу.
11. Коротко охарактеризуйте можливі застосування відношення асоціації на діаграмах класів.
12. Що таке асоційований клас і багатополюсна асоціація?
13. Коротко охарактеризуйте можливі застосування відношення агрегації на діаграмах класів.
14. Коротко охарактеризуйте можливі застосування відношення композиції на діаграмах класів.
15. Що відображає діаграма кооперацій?
16. Що відображає діаграма послідовностей?
17. Що відображає діаграма станів?
18. Що таке життєва лінія об'єкта?
19. Опишіть специфікацію найпростішого переходу між об'єктами.
20. Що відображає діаграма компонентів?
21. Що відображає діаграма розміщення?
22. Для чого використовують представлення прецедентів?
23. Для чого використовують логічне представлення?
24. Яку роль в ООП відіграє успадкування?
25. Що таке поліморфізм?
26. Яким ключовим словом розпочинається визначення класу?
27. Що розуміється під елементами класу?
28. За допомогою якої операції здійснюється доступ до елементів класу в сполученні з об'єктом класу?
29. За допомогою якої операції здійснюється доступ до елементів класу в сполученні з вказівником на об'єкт класу?
30. Яка функція використовується для присвоювання значень закритим даним – елементам класу?
31. Яка різниця між відкритими (public) і закритими (private) даними членами?

32. Чи можуть функції-члени бути закритими?
33. Чи можуть змінні-члени бути закритими?
34. Чи потрібно оголошення класу завершувати крапкою з комою?
35. Яка функція викликається для виконання ініціалізації класу?
36. Який конструктор використовується за умовчужанням?
37. Яке призначення деструктора?
38. Коли викликається конструктор копіювання?
39. Які можливості надає успадкування при розробці програмного забезпечення?
40. Дайте означення базового класу.
41. Чи може довільний клас мати доступ до закритих елементів базового класу?
42. Як здійснюється доступ до елементів базових класів?
43. Як визначається просте успадкування?