



ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>124 Системний аналіз</i>
Освітня програма	<i>Системний аналіз і управління</i>
Статус дисципліни	<i>Обов'язкова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>II курс, осінній семестр</i>
Обсяг дисципліни	<i>4 кредити ECTS / 120 годин (лекції – 36 год., лабораторні роботи – 36 год., СРС – 48 год.)</i>
Семестровий контроль/ контрольні заходи	<i>Залік/МКР</i>
Розклад занять	<i>http://rozklad.kpi.ua</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., доцент, Куєвда Ю.В., https://do.ipu.kpi.ua/message/index.php?id=62692 Лабораторні заняття: к.т.н., доцент, Куєвда Ю.В.</i>
Розміщення курсу	<i>https://do.ipu.kpi.ua/course/view.php?id=7515</i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Силабус навчальної дисципліни «Об'єктно-орієнтоване програмування» складено відповідно до освітньої програми «Системний аналіз і управління» підготовки бакалаврів спеціальності 124 – Системний аналіз.

***Метою навчальної дисципліни** є формування та закріплення у здобувачів наступних компетентностей: ЗК01 Здатність застосовувати знання в практичних ситуаціях, ЗК02 Здатність планувати і управляти часом, ЗК05 Здатність спілкуватися державною мовою усно і письмово, ЗК15 Здатність реалізовувати свої права та обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні, дотримуватися академічної доброчесності, ФК07 Здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем, а саме: об'єктно-орієнтований підхід при проектуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань, ФК08 Здатність організувати роботу з аналізу та проектування складних систем, створення відповідних інформаційних технологій та програмного забезпечення*

***Предмет навчальної дисципліни** – концепції та фундаментальні принципи програмування в рамках об'єктно-орієнтованої парадигми, застосування об'єктно-орієнтованих технологій в аналізі та проектуванні складних систем*

***Програмні результати навчання, на формування та покращення яких спрямована дисципліна:** ПРН08 Володіти сучасними методами розробки програм і програмних комплексів та прийняття*

оптимальних рішень щодо складу програмного забезпечення, алгоритмів процедур і операцій, ПРН13 Проектувати, реалізовувати, тестувати, впроваджувати, супроводжувати, експлуатувати програмні засоби роботи з даними і знаннями в комп'ютерних системах і мережах.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Для успішного засвоєння дисципліни здобувач повинен засвоїти дисципліну «Програмування та алгоритмічні мови». Компетентності, отримані в процесі вивчення дисципліни «Об'єктно-орієнтоване програмування» є необхідними для реалізації освітнього компоненту Дипломне проектування.

3. Зміст навчальної дисципліни

Розділ 1. Об'єктно-орієнтований підхід до розробки програмного забезпечення. Уніфікована мова моделювання UML

Тема 1.1. Концепції об'єктно-орієнтованого підходу

Тема 1.2. Уніфікована мова моделювання UML. Уніфікований процес моделювання UP

Тема 1.3. Структурні UML діаграми. Діаграми класів

Тема 1.4. Поведінкові діаграми UML

Тема 1.5. UML діаграми рівня реалізації

Розділ 2. Основні принципи об'єктно-орієнтованого програмування та їх реалізація в мовах програмування

Тема 2.1. ООП інструментарій мов програмування: абстракція та інкапсуляція

Тема 2.2. ООП інструментарій мов програмування: узагальнення, наслідування та ієрархії

Тема 2.3. ООП інструментарій мов програмування: поліморфізм

Тема 2.4. Реалізація відношень асоціації, агрегації та композиції в програмному коді

Розділ 3. Технології об'єктно-орієнтованого програмування

Тема 3.1. Обробка виключних ситуацій в ООП

Тема 3.2. Тестування класів

Тема 3.3. Абстрактні класи та інтерфейси

Розділ 4. Системи принципів та патерни проектування об'єктно-орієнтованого програмного забезпечення

Тема 4.1. Принципи об'єктно-орієнтованого проектування для створення надійного та підтримуваного коду

Тема 4.2. Патерни та антипатерни об'єктно-орієнтованого проектування. Поведінкові патерни

Тема 4.3. Структурні патерни проектування

Тема 4.4. Породжувальні патерни проектування

Тема 4.5. Композиція патернів в кодi. Патерн MVC

Тема 4.6. Техніки рефакторингу програмного коду в ООП

4. Навчальні матеріали та ресурси

Основна література:

1. Куєвда, Ю. В. Об'єктно-орієнтоване програмування. Курс лекцій [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Системний аналіз і управління» спец. 124 Системний аналіз / Ю. В. Куєвда ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2024. – 164 с. <https://ela.kpi.ua/handle/123456789/72205>

2. Куєвда, Ю. В. *Об'єктно-орієнтоване програмування. Лабораторний практикум [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Системний аналіз і управління» спец. 124 Системний аналіз / Ю. В. Куєвда ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2024. – 68 с. <https://ela.kpi.ua/handle/123456789/72204>*
3. *Програмування та алгоритмічні мови. Частина 2. Програмування. Конспект лекцій. [Електронний ресурс]: навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Системний аналіз і управління» за спец. 124 Системний аналіз. Укладач: Назарчук І.В. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 143 с. <https://ela.kpi.ua/handle/123456789/49907>*
4. Бублик В.В. *Об'єктно-орієнтоване програмування: підручник для студентів, які навч. за напрямками «Комп'ютерні науки», «Комп'ютерна інженерія», «Програмна інженерія», «Системна інженерія», «Інформатика», «Прикладна математика». Київ : «ІТ-Книга», 2015. 637 с.*
5. *Об'єктно-орієнтоване програмування [Електронний ресурс] : методичні вказівки до комп'ютерного практикуму для студентів напрямів підготовки «Системний аналіз», «Інформатика», «Комп'ютерні науки» / НТУУ «КПІ» ; уклад. І. В. Назарчук. - Київ : НТУУ «КПІ», 2011. <https://ela.kpi.ua/handle/123456789/1477>*
6. Назарчук І.В. *Об'єктно-орієнтоване програмування : навч. посіб. Київ : «Політехніка», 2004. 114 с.*

Додаткова література:

7. Порєв В.М. *Об'єктно-орієнтоване програмування : конспект лекцій : навч. посібник для студ. освітньої програми «Інженерія програмного забезпечення комп'ютерних систем» спеціальності 121 «Інженерія програмного забезпечення». Київ : КПІ ім. Ігоря Сікорського, 2022. 271 с.*
8. Алхімова С.М. *Об'єктно-орієнтоване програмування: підручник. у 2-х ч. Ч. 2. Об'єктно-орієнтований підхід до розробки програмного забезпечення. Київ : «Політехніка», 2019. 192 с.*
9. *Object-oriented analysis and design with applications / Grady Booch et al. 3rd ed. Addison-Wesley Professional, 2007.*
10. Booch G., Rumbaugh J., Jacobson I. *The unified modeling language user. 2nd ed. Addison-Wesley Professional, 2005.*
11. Швець О. *Рефакторинг і патерни проектування. Refactoring Guru. URL: <https://refactoring.guru/uk>*
12. Fakhroutdinov K. *The Unified Modeling Language. UML-Diagrams. URL: <https://www.uml-diagrams.org/>*
13. *C++ Language Reference. URL: <https://learn.microsoft.com/en-us/cpp/cpp/cpp-language-reference>*
14. *C++ reference URL: <https://en.cppreference.com/>*
15. *Python 3 documentation. URL: <https://docs.python.org/3/>*
16. *Draw.io – free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams. URL: <https://app.diagrams.net/>*
17. *Lucidchart | Diagramming Powered By Intelligence. URL: <https://www.lucidchart.com>*
18. *Online Compiler, Visual Debugger, and AI Tutor for Python, Java, C, C++, and JavaScript. URL: <https://pythontutor.com/>*

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань
1	Лекція 1. Тема 1.1. Концепції об'єктно-орієнтованого підходу. Методи аналізу, проектування та моделювання складних систем; парадигми програмування та місце ООП серед них; основні принципи об'єктно-орієнтованого підходу; нотації, процеси, інструменти.
2	Лекція 2. Тема 1.2. Уніфікована мова моделювання UML. Уніфікований процес моделювання UP. UML в історичній перспективі; концептуальна модель UML; структурні та поведінкові діаграми UML; стандартизований процес для методології моделювання; інструменти автоматизації для моделювання
3	Лекція 3. Тема 1.3. Структурні UML діаграми. Діаграми класів Діаграми класів (Class Diagram); елементи діаграми класів; види відношень між класами; рівні реалізації UML діаграм.
4	Лекція 4. Тема 1.4. Поведінкові діаграми UML Діаграма варіантів використання (Use Case Diagram); діаграма послідовності (Sequence Diagram); діаграма діяльності (Activity Diagram); діаграма комунікації (Communication Diagram).
5	Лекція 5. Тема 1.5. UML діаграми рівня реалізації Діаграма компонентів (Component Diagram); діаграма розгортання (Deployment diagram); бібліотеки та фреймворки.
6	Лекція 6. Тема 2.1. ООП інструментарій мов програмування: абстракція та інкапсуляція Класи; модифікатори доступу до членів класу; спеціальні методи класів; застосування специфікаторів до атрибутів та методів; функції-селектори та функції модифікатори, властивості; маніпулювання об'єктами та особливості роботи з пам'яттю.
7	Лекція 7. Тема 2.2. ООП інструментарій мов програмування: узагальнення, наслідування та ієрархії Наслідування; конструктори та деструктори при наслідуванні; множинне наслідування, віртуальні класи.
8	Лекція 8. Тема 2.3. ООП інструментарій мов програмування: поліморфізм Види поліморфізму; примусовий поліморфізм; перевантаження операцій та функцій; параметричний поліморфізм; поліморфізм підтипів; віртуальні деструктори в C++; приховування функцій-членів базових класів (function hiding).
9	Лекція 9. Тема 2.4. Реалізація відношень асоціації, агрегації та композиції в програмному коді Реалізація композиції; вкладені класи; методи реалізації агрегації; реалізація асоціації; приклади реалізації відношень на C++ та Python. МКР 1
10	Лекція 10. Тема 3.1. Обробка виключних ситуацій в ООП Аварійні варіанти завершення та варіанти обробки помилок; обробка виключних ситуацій; виняткові ситуації та розмотування стека (stack unwinding); загальні рекомендації щодо обробки виняткових ситуацій.

11	Лекція 11. Тема 3.2. Тестування класів Канонічна форма класу; драйвери для тестування класів; тестування пов'язаних класів; тестування механізмів обробки виняткових ситуацій.
12	Лекція 12. Тема 3.3. Абстрактні класи та інтерфейси Абстрактні класи; інтерфейси та класи-інтерфейси; приклади використання абстрактних класів та класів-інтерфейсів на C++ та Python.
13	Лекція 13. Тема 4.1. Принципи об'єктно-орієнтованого проектування для створення надійного та підтримуваного коду Огляд принципів об'єктно-орієнтованого проектування; Принципи/патерни GRASP; принципи SOLID; закон Деметри (Law of Demeter).
14	Лекція 14. Тема 4.2. Патерни та антипатерни об'єктно-орієнтованого проектування. Поведінкові патерни Огляд та класифікація патернів проектування; поведінкові патерни: шаблонний метод, інтерпретатор, посередник, ланцюжок обов'язків, спостерігач, стратегія, команда, стан, відвідувач, ітератор, знімок.
15	Лекція 15. Тема 4.3. Структурні патерни проектування Огляд структурних патернів, види структурних патернів: адаптер, міст, компонувальник, декоратор, фасад, легковаговик, замісник.
16	Лекція 16. Тема 4.4. Породжувальні патерни проектування Огляд породжувальних патернів; види породжувальних патернів: фабричний метод, абстрактна фабрика, будівельник, прототип, одинак.
17	Лекція 17. Тема 4.5. Композиція патернів в коді. Патерн MVC Композиція патернів: які патерни зазвичай компонуються між собою; патерн Model–View–Controller (MVC). МКР 2
18	Лекція 18. Тема 4.6. Техніки рефакторингу програмного коду в ООП Коли потрібен рефакторинг; технології рефакторингу; приклади рефакторингу: заміна циклу конвеєром; заміна умовних операторів поліморфізмом; заміна класифікаційного атрибута дочірніми класами; заміна підкласу/суперкласу делегатом.

Лабораторні роботи

1. Порівняння процедурної та об'єктно-орієнтованої парадигми програмування (6 год.)
2. Об'єктно-орієнтоване проектування з використанням UML (6 год.)
3. Створення об'єктно-орієнтованого програмного коду на основі UML-діаграм (6 год.)
4. Обробка виключних ситуацій (6 год.)
5. Unit-тестування ООП коду (6 год.)
6. Застосування патернів проектування в об'єктно-орієнтованій парадигмі (6 год.)

Самостійна робота здобувача вищої освіти

№ з/п	Вид самостійної роботи	Кількість годин СРС
1	Опрацювання навчального матеріалу, викладеному на лекціях 1-9	4
2	Опрацювання навчального матеріалу лекцій 1-9, винесеного на самостійне вивчення	4
3	Підготовка до лабораторних занять 1-3	4
	Оформлення звітів до лабораторних робіт 1-3	4

4	Підготовка до МКР1	5
5	Опрацювання навчального матеріалу, викладеному на лекціях 10-18	4
6	Опрацювання навчального матеріалу лекцій 10-18, винесеного на самостійне вивчення	4
7	Підготовка до лабораторних занять 4-6	4
8	Оформлення звітів до лабораторних робіт 4-6	4
9	Підготовка до МКР2	5
10	Підготовка до заліку	6

6. Контрольні роботи

Метою контрольних робіт є закріплення та перевірка теоретичних знань з дисципліни. Модульні контрольні роботи (МКР) 1 та 2 проводяться у вигляді тесту за лекціями 1-9 на платформі дистанційного навчання «Сікорський». Тести можуть містити відкриті питання, які перевіряються викладачем. МКР1 включає теми розділів 1 та 2, а МКР2 включає теми розділів усього курсу.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог, які викладач ставить перед студентом:

- правила відвідування занять: відповідно до Наказу 1-273 від 14.09.2020 р. заборонено оцінювати присутність або відсутність здобувача на аудиторному занятті, в тому числі нараховувати заохочувальні або штрафні бали. Бали нараховують за вказані види навчальної активності на лекційних та лабораторних заняттях відповідно до РСО даної дисципліни.

- політика щодо академічної доброчесності: Кодекс честі Національного технічного університету України «Київський політехнічний інститут» <https://kpi.ua/files/honorcode.pdf> встановлює загальні моральні принципи, правила етичної поведінки осіб та передбачає політику академічної доброчесності для осіб, що працюють і навчаються в університеті, якими вони мають керуватись у своїй діяльності, в тому числі при вивченні та складанні контрольних заходів з дисципліни;

- при використанні цифрових засобів зв'язку з викладачем (мобільний зв'язок, електронна пошта, переписка на форумах та у соцмережах тощо) необхідно дотримуватись загальноприйнятих етичних норм, зокрема бути ввічливим та обмежувати спілкування робочим часом викладача;

- захист лабораторних робіт відбувається при наявності звіту в усній формі з демонстрацією результатів лабораторної роботи; при груповому виконанні лабораторної роботи здобувачами обов'язково вказується зона їх відповідальності;

- дедлайни захисту лабораторних робіт та графік виконання контрольних робіт вказуються на дистанційній платформі дисципліни.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: МКР, захист лабораторних робіт.

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Семестровий контроль: залік.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
95-100	Відмінно
85-94	Дуже добре
75-84	Добре
65-74	Задовільно
60-64	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

Таблиця балів за окремі види роботи здобувача

Вид поточного контролю	Максимальний бал*
Лабораторна робота 1	10
Лабораторна робота 2	10
Лабораторна робота 3	10
Модульна контрольна робота 1	20
Лабораторна робота 4	10
Лабораторна робота 5	10
Лабораторна робота 6	10
Модульна контрольна робота 2	20

* у разі несвоєчасної здачі звіту та захисту, лабораторна робота може бути оцінена не більше 90% від максимального балу в межах першого тижня після встановленого кінцевого строку, далі – не більше 80%.

Зарахування результатів неформальної/інформальної освіти. Модульна контрольна робота 1 може бути оцінена на максимальний бал за результатами, отриманими у неформальній/інформальній освіті. Зарахування сертифікатів проходження курсів неформальної/інформальної освіти можливо за умови узгодження з викладачем відповідності програми цього курсу тематиці дисципліни та усної оглядової демонстрації курсу викладачу.

Заохочувальні бали: участь у модернізації лабораторних робіт, контрольних робіт, активність на аудиторних заняттях, призові місця на олімпіадах з програмування, підготовка оглядів наукових праць тощо, усього не більше 10 балів.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Умови допуску до семестрового контролю: зарахування лабораторних робіт №1-5 не менше, ніж на 50% від максимального балу, спроба складання усіх модульних контрольних робіт, поточний рейтинг не менше 30 балів.

Семестровий контроль: залік

Оцінювання результатів навчання здійснюється за 100-бальною шкалою (див. <https://osvita.kpi.ua/node/37>). Здобувачі, які виконали всі умови допуску до заліку та мають рейтингову оцінку 60 і більше балів, отримують відповідну до набраного рейтингу оцінку без додаткових випробувань. Зі здобувачами, які виконали всі умови допуску до заліку та мають рейтингову оцінку менше 60 балів, а також з тими здобувачами, хто бажає підвищити свою рейтингову оцінку, на заліку викладач проводить семестровий контроль у вигляді залікової контрольної роботи або співбесіди.

Залікова робота оцінюється до 40 балів. При цьому набрані бали протягом семестру за МКР 1 та 2 анулюються, а за лабораторні роботи зберігаються, оцінка за залікову контрольну роботу є остаточною. Здобувач проходить тестування у середовищі Moodle. На тестування пропонується 40 тестових питань, кожне з яких оцінюється в 1 бал, з частиною відкритих питань, які оцінюються викладачем та захищаються студентом усно. Для отримання позитивної оцінки необхідно набрати у сумі з лабораторними роботами 60 балів і вище. Час тестування зазвичай складає 40 хвилин, але може бути скоригований лектором та (або) викладачам, що приймає залік.

Критерії нарахування балів за контрольні заходи:

-«Відмінно» (95-100%): здобувач продемонстрував глибокі, систематичні знання дисципліни, вільно виконував всі завдання програми; засвоїв основну та додаткову літературу, виявив творчий підхід у трактуванні матеріалу, чітко, логічно та стисло висвітлюючи ключові ідеї; зрозумів взаємозв'язки основних понять дисципліни та їх значення для професійної діяльності.

-«Дуже добре» (85-94%): здобувач показав систематичні знання дисципліни на рівні, що перевищує середній; виконав усі завдання програми, засвоїв основну та додаткову літературу, відповів на більшість запитань за темами курсу; показав розуміння взаємозв'язку основних понять дисципліни та їх значення для професійної діяльності.

-«Добре» (75-84%): здобувач виявив загалом добрі знання матеріалу, хоча при виконанні завдань допустив деякі помилки; виконав основні завдання програми, засвоїв основну літературу, відповів на більшість запитань за темами курсу; має систематичні знання та здатний до самостійного використання та поповнення знань у подальшому навчанні та професійній діяльності.

-«Задовільно» (65-74%): здобувач засвоїв основний матеріал дисципліни в обсязі, достатньому для подальшого навчання та професійної діяльності; виконав основні завдання програми, ознайомився з основною літературою, але припустив значну кількість помилок або недоліків, які може виправити самостійно.

-«Достатньо» (60-64%): здобувач виявив мінімальні знання основного матеріалу, необхідні для подальшого навчання та професійної діяльності; в цілому виконав завдання програми, ознайомився з основною літературою, але припустив помилки, які він може виправити тільки за допомогою викладача.

-«Незадовільно» (30-54%): здобувач мав значні прогалини в знаннях, допускав серйозні помилки при виконанні завдань програми, проте здатен самостійно доопрацювати матеріал і підготуватися до повторної задачі дисципліни.

-«Не допущено» (0-29%): здобувач не володів значною частиною навчального матеріалу, допускав принципові помилки у більшості завдань програми або не виконував їх зовсім.

Робочу програму навчальної дисципліни (силабус):

Складено доцент кафедри ММСА, к.т.н, доц., Куєвда Юлія Валеріївна

Ухвалено кафедрою ММСА (протокол №3 від 08.10.2025 р.)

Погоджено Методичною комісією НН ІПСА (протокол № 2 від 09.10.2025 р.)