



ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	Цифрові технології в енергетиці
Статус дисципліни	Обов'язкова
Форма навчання	Очна (денна)
Рік підготовки, семестр	2 курс, осінній семестр
Обсяг дисципліни	4,5 кредитів ЕКТС (лекції – 36 год., лабораторні заняття – 18 год., самостійна робота – 81 год.)
Семестровий контроль/ контрольні заходи	м.к.р.,екзамен
Розклад занять	http://rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: доцент,к.т.н., доцент Шалденко Олексій Вікторович, <i>email: o.shaldenko@gmail.com</i> Лабораторні заняття: асистент, Здор Костянтин Андрійович, <i>email: Kostya9919moonlight@gmail.com</i> , старший викладач Круш Ольга Євгенівна, <i>email: jwolgakrush@gmail.com</i>
Розміщення курсу	https://campus.kpi.ua

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання
Метою вивчення дисципліни є формування у студентів здатностей:

- опанувати об'єктно-орієнтований підхід для розробки та розвитку програмних систем;
- розуміти методологію об'єктно-орієнтованого проектування, оволодіти нею і використовувати її впродовж життєвого циклу програмного забезпечення;
- розробляти програмне забезпечення за допомогою сучасних інструментальних засобів створення програмного забезпечення

Основні завдання при вивченні дисципліни

Дисципліна «Об'єктно-орієнтоване програмування» забезпечує наступні програмні компетентності і програмні результати освітньо-професійної програми (ОПП):

ЗК 2	Здатність застосовувати знання у практичних ситуаціях
ЗК 8	Здатність генерувати нові ідеї (креативність)
ЗК 11	Здатність приймати обґрунтовані рішення
ФК 3	Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем
ФК 8	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

В результаті засвоєння кредитного модуля студенти мають продемонструвати такі програмні результати навчання:

ПР5	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.
ПР9	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.

знання:

- знання об'єктно-орієнтованої мови програмування;
- патернів об'єктно-орієнтованого проектування;
- знання інструментальних засобів та інтегрованих середовищ для створення програмного забезпечення

програмного забезпечення

уміння:

- аналізувати вимоги до програмного забезпечення;
- проектувати архітектуру програмного забезпечення;
- використовувати патерни об'єктно-орієнтованого проектування;
- виконувати рефакторинг програмного коду;
- розробляти та налагоджувати програмне забезпечення;
- використовувати потрібні інструментальні засоби та інтегровані середовища для вирішення завдань;

вирішення завдань;

досвід:

- розроблення програмного забезпечення;
- роботи з комп'ютерними інструментальними засобами та інтегрованими середовищами.

середовищами.

2. Пререквізити та постреквізити дисципліни

У структурно-логічній схемі навчання зазначений кредитний модуль розміщується тоді, коли студенти вже прослухали такі дисципліни, як "Програмування алгоритмічних структур", "Проектування та аналіз обчислювальних алгоритмів" та "Алгоритмізація та програмування", і є логічним продовженням навчального процесу набуття знань та навичок сучасних систем програмування.

З іншого боку, викладений матеріал може бути використаний при вивченні дисциплін "Геометричне моделювання та комп'ютерна графіка", "Основи системного аналізу", "Технології розробки програмного забезпечення", які подаються в наступних семестрах.

3. Зміст навчальної дисципліни

В дисципліні вивчаються такі теми:

Розділ 1. Базові поняття мови ООП

Тема 1.1. Парадигми програмування. Основи мови С#. Структури.

Тема 1.2. Класи і об'єкти. Конструктори та деструктори.

Тема 1.3. Вивчення системи контролю версій Git.

Тема 1.4. Перевантаження функцій. Перевантаження операцій. Використання інкапсуляції.

Тема 1.5. Успадкування. Ієрархія класів. Абстрактні класи. Віртуальні та абстрактні методи.

Розділ 2. Властивості класів

Тема 2.1. Використання вказівників з класами.

Тема 2.2. Статичний поліморфізм. Динамічний поліморфізм. Статичний поліморфізм.

Тема 2.3. Використання інтерфейсів.

Розділ 3. Шаблони програмування з використанням ООП.

Тема 3.1. Структурні шаблони.

Тема 3.2. Шаблони створення.

Тема 3.3. Поведінкові шаблони.

4. Навчальні матеріали та ресурси

Базова література:

1. Алещенко О.В. Програмування-2. Об'єктно-орієнтоване програмування: конспект лекцій. [Електронний ресурс]. – Київ : Кафедра ОТ, НТУУ "КПІ", 2021. Режим доступу: https://drive.google.com/drive/folders/1_09ZizH0YBZ1z13HX5j9giRQg42EC5Y?usp=sharing.
2. Карпенко С.Г. Об'єктно-орієнтоване програмування [Електронний ресурс] / НТУУ «КПІ»; уклад. С. Г. Карпенко. – Київ : НТУУ «КПІ», 2012, <https://ela.kpi.ua/handle/123456789/1814>
3. Graham I. Object-Oriented Methods: Principles and Practice. / I. Graham. Addison- Wesley , 2000, 864 p.
4. Schildt H. C++ from the Ground Up 3rd edition. / H. Schildt, McGraw Hill, 2003, 624.
5. Куліков В.М., Іващенко О.В., Успенський О.А. Конспект лекцій з навчальної дисципліни “Об'єктно-орієнтоване програмування”. К.: Вид-во ІСЗЗІ НТУУ “КПІ”, 2011.
6. Гришанович Т.О. Основи об'єктно-орієнтованого програмування / Т.О. Гришанович. Харків, 2020. 102 с.
7. Stroustrup B. The C++ Programming Language / B. Stroustrup. Addison-Wesley Professional, 2013, 1376 p.
8. Lafore R. Object-Oriented Programming in C++ / R. Lafore. Que Publishing, 1997, 1452p.
9. Алхімова С.М. Об'єктно-орієнтоване програмування / С.М. Алхімова. Київ: КПІ ім.Ігоря Сікорського, 2019, 190 с.
10. Rogerson D. Inside COM , Redmont Washington: Microsoft Press, 1996.

Додаткова література:

1. Wiener R., Pinson L. An Introduction to Object-Oriented Programming and C++ / R. Wiener, L. Pinson, 1988, 420p.
2. Wang P. C++ with Object Oriented Programming / P. Wang. Course Technology, 1994, 496p.
3. Schildt H. C++ IT-Tutorial / H. Schildt. mitp-Verlag, 2003, 825 p.
4. Kernighan B, Ritchie D. The C Programming language / B. Kernighan, D. Ritchie. Prentice Hall Software Series, 1988, 257p.
5. Проценко В.С., Чаленко П.Й., Ставровський А.Б. Техніка програмування мовою Сі: Навч. посібник. К.: Либідь, 1993.
6. Lucas P. The C++ Programmer`s Handbook / P. Lucas. Prentice Hall. 2011, 128 p.
7. Schieldt G. Theory and Practice of C++ / G.Schildt. 1995.
8. Budd T. An Introduction to Object-Oriented Programming / T. Budd. Addison-Wesley. 1996, 452 p.

9. Deitel P., Deitel H. C++ How to Program / P. Deitel, H. Deitel. Pearson. 2016, 1080 p.
10. Jouttis N Object-Oriented Programming in C++ / N. Jouttis. 2002, 628 p.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на літературу та завдання на СРС)	К-сть ауд.год
Розділ 1. Базові поняття мови ООП		
Тема 1.1. Парадигми програмування. Основи мови C#. Структури.		
1.	ЛЕКЦІЯ 1. Мови програмування. Поняття про парадигми програмування. Основи програмування на мові C#. Структури, поля структур, ініціалізація структур. Оголошення структур.	2
Тема 1.2. Класи і об'єкти. Конструктори та деструктори.		
2.	ЛЕКЦІЯ 2. Класи. Опис класу. Властивості полів класу. Опис об'єктів. Вказівник this. Конструктори. Типи конструкторів. Деструктори. Особливості створення та ініціалізації об'єктів з динамічними полями.	2
Тема 1.3. Вивчення системи контролю версій Git		
3.	ЛЕКЦІЯ 3. Системи контролю версій. Встановлення Git. Вивчення команд керування Git.	2
Тема 1.4. Перевантаження функцій. Перевантаження операцій. Використання інкапсуляції.		
4.	ЛЕКЦІЯ 4. Перевантаження функцій. Перевантаження операцій. Застосування інкапсуляції для полів та методів класу.	2
Тема 1.5. Успадкування. Ієрархія класів. Абстрактні класи. Віртуальні та абстрактні методи		
5.	ЛЕКЦІЯ 5. Принцип успадкування. Просте успадкування. Ієрархія класів. Конструктори в похідних класах.	2
6.	ЛЕКЦІЯ 6. Доступ до об'єктів в ієрархії класів. Статичне і динамічне зв'язування. Використання абстрактних класів. Використання віртуальних та абстрактних методів.	2
Розділ 2. Властивості класів		
Тема 2.1. Використання вказівників з класами.		
7.	ЛЕКЦІЯ 7. Використання вказівників на класи. Використання вказівників батьківського класу.	2
Тема 2.2. Статичний поліморфізм. Динамічний поліморфізм. Укриття базової реалізації метода.		
8.	ЛЕКЦІЯ 8. Статичний поліморфізм. Використання перевантаження методів та операторів при наслідуванні.	2
9.	ЛЕКЦІЯ 9. Динамічний поліморфізм. Застосування поліморфізму при наслідуванні.	2
10.	ЛЕКЦІЯ 10. Укриття базової реалізації метода.	2
Тема 2.3. Використання інтерфейсів.		
11.	ЛЕКЦІЯ 11. Використання інтерфейсів. Базова реалізація методів та властивостей в інтерфейсах.	2
12.	ЛЕКЦІЯ 12. Порівняння використання інтерфейсів з абстрактними класами.	2
Розділ 3. Шаблони програмування з використанням ООП.		
Тема 3.1. Структурні шаблони.		
13.	ЛЕКЦІЯ 13. Використання структурних шаблонів. Шаблони Adapter, Bridge, Composite.	2
14.	ЛЕКЦІЯ 14. Шаблони Decorator, Façade, Flyweight, Proxy.	2

Тема 3.2. Шаблони створення.		
15.	ЛЕКЦІЯ 15. Використання шаблонів створення. Шаблони Abstract Factory, Builder.	2
16.	ЛЕКЦІЯ 16. Шаблони Prototype, Singleton, Factory Method.	2
Тема 3.3. Поведінкові шаблони.		
17.	ЛЕКЦІЯ 17. Використання поведінкових шаблонів. Шаблони Chain of responsibility, Command, Iterator, Mediator, Memento.	2
18.	ЛЕКЦІЯ 18. Шаблони Observer, State, Strategy, Template method, Visitor	2
	Всього	36

Лабораторні роботи

N	Назва лабораторних робіт	К-сть ауд.год.
1.	Побудова класів для конкретної предметної області з конструкторами, деструктором та полями які відповідають предметні області.	4
2.	Побудова класів для конкретної предметної області. Налаштування взаємодії між об'єктами цих класів. Передача об'єктів класів в інші класи.	4
3.	Створення ієрархії класів з конструкторами, деструктором та перевантаженими операціями останнього похідного класу.	4
4.	Створення класів на основі інтерфейсів. Приведення класів до інтерфейсу та використання їх.	2
5.	Застосування структурних шаблонів та шаблонів створення для налаштування роботи з складними структурами класів.	2
6.	Застосування поведінкових шаблонів для налаштування ефективної взаємодії між класами.	2
	Всього	18

6. Самостійна робота

Самостійна робота студента (81 годин) передбачає підготовку до аудиторних занять та контрольних заходів, проведення розрахунків та підготовка вхідних даних до роботи.

Розподіл годин СРС: передбачає підготовку до аудиторних занять та контрольних заходів, проведення розрахунків та підготовка вхідних даних до роботи.

Розподіл годин СРС: підготовка до екзамену – 30 годин; підготовка до лабораторних робіт – 9 годин; підготовка до МКР – 4 годин; підготовка до лекції – 9 годин, опанування додаткової літератури – 29 годин.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- Відвідування лекцій, а також відсутність на них, не оцінюються. Відвідування лабораторних занять є обов'язковою складовою вивчення матеріалу;

- При захисті лабораторних робіт студент має продемонструвати розроблений програмний код та результати його виконання на тестах, як заздалегідь підготованих, так і запропонованих викладачем. У випадку дистанційної форми навчання захист відбувається на відповідній конференції шляхом демонстрації екрана.

- Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

● Норми етичної поведінки Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

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

Рейтинг студента з освітнього компонента розраховується зі 100 балів. Стартовий рейтинг (протягом семестру) складається з балів, що студент отримує за виконання лабораторних робіт (6 робіт) та МКР;

Критерії нарахування балів за виконання лабораторних робіт.

Ваговий бал за виконання завдань лабораторних робіт складає 9 балів. Максимальна кількість балів за всі завдання дорівнює

$$9 \text{ балів} \times 6 = 54 \text{ балів.}$$

Мінімальна кількість для зарахування лабораторної роботи складає 6 балів (60%)

Максимальна кількість балів за модульну контрольну роботу дорівнює 6 балів. На модульну контрольну роботу вноситься одне теоретичне питання та одне практичне.

Контрольна робота оцінюється наступним чином:

1. правильність написання теоретичного питання – 2 бали;
2. надання прикладу на вказані завдання – 0.3 бали;
3. правильність розв'язання практичного завдання – 3 бали;
4. правильність написання псевдокоду – 0.7 балів.

За активність на лекціях та виконання домашніх робіт нараховується максимум – 4 бали.

Умови допуску до екзамену: зарахування всіх лабораторних робіт. Мінімальна кількість набраних балів – 36 (60%).

На екзамені студенти виконують письмову контрольну роботу. Екзаменаційний білет складається з двох теоретичних питань та одного практичного завдання. Ваговий бал кожного теоретичного питання – 10. Ваговий бал практичного завдання – 20.

Максимальна кількість балів за складання екзамену дорівнює

$$10 \text{ балів} \times 2 + 20 \text{ балів} = 40 \text{ балів.}$$

Теоретична частина оцінюється наступним чином:

- «відмінно», правильна чітко викладена, повна відповідь – (не менше 90% потрібної інформації) – 9-10 балів;

- «добре», достатньо повна відповідь (не менше 70% потрібної інформації) – 7-8 балів;

- «задовільно», неповна відповідь (не менше 50% потрібної інформації) – 5-6 балів;

- «незадовільно», незадовільна відповідь – 0 балів.

Практичне завдання оцінюється наступним чином:

- «відмінно», повне, безпомилкове розв'язування завдання – 18-20 балів;

- «добре», повне, розв'язування завдання із несуттєвими неточностями – 15-17 балів;

- «задовільно», завдання виконане з певними недоліками – 12-14 балів;

- «незадовільно», завдання не виконано.

5. Сума стартових балів і балів за екзаменаційну контрольну роботу переводиться за освітній компонент згідно з таблицею.

Бали: практичні роботи + екзаменаційна контрольна робота	Оцінка
100...95	Відмінно
94...85	Дуже добре
84...75	Добре
74...65	Задовільно

64...60	Достатньо
Менше 60	Незадовільно
Є не зараховані лабораторні роботи	Не допущено

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., доцентом Шалденко Олексієм Вікторовичем

Ухвалено кафедрою ЦТЕ (протокол № 20 від 10.05.23)

Погоджено Методичною комісією НН ІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 26.05.23)