



# ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ.

## Курсова робота

### 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	1 кредитів ЕКТС (самостійна робота – 30 год.)
Семестровий контроль/ контрольні заходи	залік
Розклад занять	<a href="http://rozklad.kpi.ua">http://rozklad.kpi.ua</a>
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: доцент, к.т.н., доцент Шалденко Олексій Вікторович, <i>email: o.shaldenko@gmail.com</i>
Розміщення курсу	<a href="https://campus.kpi.ua">https://campus.kpi.ua</a>

#### Програма навчальної дисципліни

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

Метою кредитного модуля є формування у студентів загальних та фахових компетентностей у відповідності до ОПП

ЗК 2	Здатність застосовувати знання у практичних ситуаціях
ЗК 8	Здатність генерувати нові ідеї (креативність)
ЗК 11	Здатність приймати обґрунтовані рішення
ЗК 12	Здатність оцінювати та забезпечувати якість виконуваних робіт
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.

## 2. Пререквізити та постреквізити дисципліни

**Пререквізити дисципліни.** Структура викладання дисципліни побудована таким чином, щоб вивчення дисципліни мало теоретичне та практичне спрямування. Вивчення дисципліни спирається на знання, отримані студентами в попередні роки навчання за освітньою програмою бакалавра в галузі 12 Інформаційні технології.

Студенти повинні мати знання з дисциплін "Програмування алгоритмічних структур", "Проектування та аналіз обчислювальних алгоритмів" та "Алгоритмізація та програмування».

**Постреквізити дисципліни.** Виконувати моделювання і проектування структур та елементів баз даних із застосуванням різноманітних методик та технологій.

## 3. Зміст навчальної дисципліни

В дисципліні вивчаються такі теми:

Розділ 1. Базові поняття мови ООП

Тема 1.1. Парадигми програмування. Структури.

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

Тема 1.3. Перевантаження функцій. Перевантаження операцій.

Тема 1.4. Шаблони функцій.

Тема 1.5. Успадкування. Ієрархія класів. Абстрактні класи.

Розділ 2. Властивості класів

Тема 2.1. Множинне успадкування.

Тема 2.2. Поточкові класи. Функції керування потоками.

Тема 2.3. Шаблони класів.

Розділ 3. Файлові потоки. Винятки. Стандартні шаблони

Тема 3.1. Класи файлових потоків.

Тема 3.2. Обробка виняткових ситуацій.

Тема 3.3. Контейнерні класи.

Аудиторні заняття передбачені в освітньому компоненті "Об'єктно-орієнтоване програмування". Курсова робота є індивідуальним завданням з дисципліни. При виконанні курсової роботи вивчаються та досліджуються основні принципи ООП та підходи до моделювання інформаційної системи.

1. Титульна сторінка
2. Аркуш завдання
3. Зміст
4. Вступ
5. Основна частина; включає в себе розділи:
  1. Аналіз вимог:
    - 1.1. Постановка задачі
    - 1.2. Огляд існуючих рішень
  2. Розробка програмного забезпечення:
    - 2.1. Опис мови програмування.
    - 2.2. Опис використаних принципів ООП.

- 2.3. Опис розроблених алгоритмів.
- 2.4. Структура програми.
- 3. Керівництво користувача
  - 3.1. Робота користувача з системою
- 6. Висновки
- 7. Перелік посилань

#### **Основна 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](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. Алхімова С. М. А54 Об'єктно-орієнтоване програмування : підручник. У 2-х ч. Ч. 2. Об'єктно-орієнтований підхід до розробки програмного забезпечення / С. М. Алхімова. - Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019.-192 с
4. Куліков В.М., Іващенко О.В., Успенський О.А. Конспект лекцій з навчальної дисципліни “Об'єктно-орієнтоване програмування”. К.: Вид-во ІСЗІ НТУУ “КПІ”, 2011.
5. Гришанович Т.О. Основи об'єктно-орієнтованого програмування / Т.О. Гришанович. Харків, 2020, 102 с.

#### **Додаткова:**

1. Schildt Н. С++ IT-Tutorial / Н. Schildt. mitp-Verlag, 2003, 825 p.
2. Lucas P. The C++ Programmer`s Handbook / P. Lucas. Prentice Hall. 2011, 128 p.
3. Deitel P., Deitel Н. С++ How to Program / P. Deitel, Н. Deitel. Pearson. 2016, 1080 p.

### **Навчальний контент**

#### **4. Методика опанування навчальної дисципліни (освітнього компонента)**

Методи проектування спрямовані на виховання критичного мислення, з акцентом на те, що в реальних задачах не існує єдиного правильного рішення. Курсова робота готується до захисту в завершальний період теоретичного навчання. Курсова робота повинна бути підготовлена до захисту в термін, встановлюваний викладачем. До захисту курсової роботи представляється пояснювальна записка.

#### **5. Самостійна робота студента**

Самостійна робота студента (30 годин) передбачає проведення досліджень, формалізацію вхідних даних, розробку програмного забезпечення, оформлення пояснювальної записки до курсової роботи та підготовку до захисту.

Розподіл годин самостійної роботи: підготовка до захисту – 1 година; опанування і програмна реалізація відповідної предметної області – 20 годин; оформлення пояснювальної записки до курсової роботи – 9 годин.

### **Політика та контроль**

#### **Політика навчальної дисципліни (освітнього компонента)**

Перед захистом курсової роботи 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.

Захист курсової роботи відбувається за таким порядком: доповідь з презентацією курсової роботи, відповіді на запитання.

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 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)**

Система рейтингових (вагових) балів та критерії оцінювання

Рейтинг студента з виконання курсової роботи складається з двох складових:

оцінка виконання поставленого завдання (обґрунтування обраних рішень; якість розробленого алгоритмічного та програмного забезпечення; якість пояснювальної записки та дотримання графіку роботи) - характеризує роботу студента протягом підготовки курсової роботи та її результат;

оцінка захисту роботи (презентація виконаної роботи, відповіді на запитання).

Оцінка складається з балів за:

- захист розробленого програмного забезпечення;
- доповідь та відповіді на захисті дипломної роботи.

Захист програмного забезпечення оцінюється із 60 балів і зараховується за наявності досягнення тільки коректних результатів виконання, які відповідають умовам та обмеженням індивідуального завдання. Бали нараховуються таким чином:

обґрунтованість вибору інформаційних технологій та засобів реалізації поставленої задачі (20 балів);

наявність новітніх інформаційних технологій (10 балів);

обґрунтованість обраних методів та моделей (20 балів);

наявність удосконалення існуючих методів та моделей (10 балів).

Штрафні бали призначаються за:

- вибір неоптимальних методів, алгоритмів реалізації – 10 балів;
- не надану або невірну відповідь на запитання – 5 балів.

Доповідь на захисті курсової роботи бакалавра оцінюється із 40 балів.

Бали нараховуються таким чином:

якість доповіді; ступінь володіння матеріалом; вміння захищати свою думку тощо (20 балів);

якість оформлення пояснювальної записки (20 балів).

Таблиця 1. Переведення рейтингових балів до оцінок за університетською шкалою

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

## **Додаткова інформація з дисципліни**

В рамках навчальної дисципліни студенти можуть скористатися можливостями дуальної освіти в рамках договору про співробітництво КПІ ім. Ігоря Сікорського та компанії ТОВ “Українські інформаційні технології” (SoftServe). За погодженням між викладачем освітнього компоненту та ментором компанії SoftServe студенти можуть обрати завдання проекту, наданого компанією за умови виконання ролі розробника системи. В такому випадку захист відбувається на комісії представників кафедри ЦТЕ та компанії SoftServe, в яку обов’язково входять викладачі призначені в комісію з захисту курсової роботи.

### **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)