



## ОСНОВИ ПРОГРАМУВАННЯ. Частина 2. МЕТОДОЛОГІЇ ПРОГРАМУВАННЯ

### Робоча програма навчальної дисципліни (Силабус)

#### Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці
Статус дисципліни	Обов'язкова (нормативна)
Форма навчання	Заочна
Рік підготовки, семестр	І курс, другий (весінній) семестр
Обсяг дисципліни	180 годин / 6 кредитів ЄКТС
Семестровий контроль / контрольні заходи	Екзамен
Розклад занять	<a href="http://rozklad.kpi.ua/">http://rozklad.kpi.ua/</a>
Мова викладання	Українська
Інформація про керівника курсу / викладачів	доцент, к.т.н. Шпурик Вадим Вадимович
Розміщення курсу	Кампус – <a href="http://campus.kpi.ua">campus.kpi.ua</a> <a href="http://route.ho.ua/kpi/c2/">http://route.ho.ua/kpi/c2/</a>

#### Програма навчальної дисципліни

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

Майбутньому фахівцеві варто вивчати дисципліну «Основи програмування» тому, що вона є базовим курсом для всіх тих, хто хоче вивчити мови програмування, зокрема, C і C++, та стати досвідченим розробником програмного забезпечення. Курс містить також спеціальні розділи, присвячені життєвому циклу програмних засобів сучасної ІТ-індустрії, проблеми помилок в програмах і методах верифікації програмного забезпечення, стилю програмування. Після проходження лабораторного практикуму студент буде здатний писати програми середньої складності на мовах C, C++. Процес побудований таким чином, що слухачі навчаються не тільки писати код і розуміти код, написаний іншими програмістами,

але і вирішувати проблеми, з якими неминуче доведеться зіткнутися при створенні програмного забезпечення.

**Метою** опанування дисципліни є освоєння фундаментальних основ програмування; ознайомлення з усіма основними термінами і поняттями; здобуття практичних навичок розробки алгоритмів; отримання реального досвіду розробки; навчитися писати доволі складні програми; познайомитися з передовими мовами програмування і напрямками програмування; чітко розуміти, який напрямок вам найцікавіше; отримати матеріали та інструкції для подальшого розвитку.

**Предметом** кредитного модуля є вивчення фундаментальних принципів сучасного процедурного та об'єктно-орієнтованого програмування на базі мови системного програмування C.

### **Програмні результати навчання**

Дисципліна сприяє формуванню у студентів певних компетентностей, тобто здатностей до виконання певних завдань самостійно. Компетентності формуються як фахові, так і загальні.

#### **Загальні компетентності:**

- здатність до абстрактного мислення, аналізу та синтезу (ЗК 01);
- здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК 06).

#### **Фахові компетентності:**

- здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення (ФК 01);
- здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування (ФК 02);
- здатність розробляти архітектури, модулі та компоненти програмних систем (ФК 03);
- володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних (ФК 07);
- здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення (ФК 08);
- здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя (ФК 10);
- здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення (ФК 11);
- здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення (ФК 13);

- здатність до алгоритмічного та логічного мислення (ФК 14);
- володіти скриптовими та декларативними мовами програмування (ФК 16).

Згідно з вимогами програми навчальної дисципліни після засвоєння навчальної дисципліни студенти мають продемонструвати певні **програмні результати навчання**, тобто набуті ними знання й уміння описані далі:

- аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки (ПРН 01);
- знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення (ПРН 03);
- уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення (ПРН 06);
- знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення (ПРН 07);
- знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань (ПРН 13);
- мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення (ПРН 15);
- знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних (ПРН 18).

## **2. Пререквізити та постреквізити дисципліни**

Вивчення дисципліни «Основи програмування» проходить паралельно з вивченням студентами основоположних дисциплін за спеціальністю, тому розміщується дисципліна у структурно-логічній схемі паралельно з нормативними освітніми компонентами циклу загальної підготовки та з дисциплінами циклу професійної підготовки, такими як: «Алгоритми та структури даних», «Основи комп'ютерних мереж і систем» та «Комп'ютерна дискретна математика».

Дисципліна «Основи програмування» забезпечує вивчення таких дисциплін, як: «Бази даних», «Компоненти програмної інженерії», «Об'єктно-орієнтований аналіз та конструювання програмних систем», «Курсова робота з основ Веб-програмування» та «Методології розробки інтелектуальних комп'ютерних програм», а також забезпечує написання курсової роботи з циклу професійної підготовки «Основи програмування. Курсова робота».

## **3. Зміст навчальної дисципліни**

Навчальна дисципліна «Основи програмування» складається з двох кредитних модулів, другий з них – «Основи програмування. Частина 2. Методології програмування» має структуру представлену в таблиці далі.

Розділ 1. Додаткові відомості мови С. Структури даних	Тема. Прості абстрактні структури даних
Розділ 2. Введення в мову С++: основні поняття та конструкції	Тема – 1. Вступ до С++: основні поняття
	Тема – 2. Вступ до С++: конструкції мови
	Тема – 3. Масиви об'єктів, використання вказівників і посилання
Розділ 3. Конструкції мови С++	Тема – 1. Використання переважаних конструкцій
	Тема – 2. Використання наслідування (успадкування)
Розділ 4. Введення та виведення даних у С++. Робота з файлами	Тема – 1. Введення та виведення даних
	Тема – 2. Файлове введення та виведення даних
Розділ 5. Віртуальні функції, шаблони та виключення, RTTI	Тема – 1. Поліморфізм і віртуальні функції в програмах
	Тема – 2. Шаблони та обробка виключень
	Тема – 3. Типи даних і дії з ними
Розділ 6. Простори імен і шаблони проектування	Тема – 1. Простори імен і перетворення типів
	Тема – 2. Шаблони проектування

#### **4. Навчальні матеріали та ресурси**

Для кращого засвоєння дисципліни та виконання лабораторних робіт і самостійної роботи студентам надаються список рекомендованої літератури та навчальні матеріали, розміщені в Інтернет і університетських ресурсах.

##### **Базові ресурси та література:**

1. Конспект лекцій. 2 семестр – URL: <http://route.ho.ua/kpi/c2/>
2. Frank M. Carrano, Janet J. Prichard. Data Abstraction & Problem Solving with C++: Walls and Mirrors : Addison-Wesley Professional. 2001. 994 p.
3. Ivor Horton Peter Van Weert. Beginning C++23: From Beginner to Pro. 7th Edition : Apress. 2023. 948 p.
4. Nesteruk D. Design Patterns in Modern C++20. 2nd Edition : Apress. 2022. 386 p.
5. Bjarne Stroustrup. Programming: Principles and Practice Using C++. 2nd Edition : Addison-Wesley Professional. 2014. 1312 p.

##### **Додаткова література:**

1. Bjarne Stroustrup. A Tour of C++ (2nd Edition) (C++ In-Depth Series) 2nd Edition : Addison-Wesley Professional. 2018. 256 p.
2. Scott Meyers. Effective C++: 55 Specific Ways to Improve Your Programs and Designs. 3rd Edition : Addison-Wesley Professional. 2005. 320 p.

### **Навчальний контент**

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

Навчальна дисципліна складається з двох кредитних модулів, які охоплюють 11 кредитів або 330 годин. За порядком слідування та розподілом годин кредитні модулі розміщуються на першому курсі в осінньому та весінньому семестрах.

У весняному семестрі першого курсу передбачено другий кредитний модуль дисципліни «Основи програмування» – освітній компонент «Основи програмування. Частина 2. Методології програмування», на який відводиться 6 кредитів або 180 годин.

Другий кредитний модуль дисципліни включає в себе проведення лекційних занять – у розмірі 6 годин, практичних занять (комп’ютерних практикумів) – 4 години і лабораторних робіт – 2 години протягом семестру.

### Календарний план запланованих заходів: лекційні заняття

№	Види робіт за розділами та темами
1	Розділ 2. Введення в мову C++: основні поняття та конструкції Тема – 1. Вступ до C++: основні поняття Лекція 3. Вступ до C++
	Розділ 3. Конструкції мови C++: перевантаження та наслідування Тема – 1. Використання перевантажених конструкцій Лекція 6. Перевантаження функцій Лекція 7. Перевантаження операторів
2	Тема – 2. Використання наслідування (успадкування) в класах Лекція 8. Наслідування: типи, можливості, обмеження
	Розділ 4. Введення та виведення даних у C++. Робота з файлами Тема – 2. Файлове введення та виведення даних Лекція 11. Розширені можливості вводу/виводу. Файлове введення та виведення даних
3	Розділ 5. Віртуальні функції, шаблони та виключення, RTTI Тема – 1. Поліморфізм і віртуальні функції в програмах Лекція 13. Поліморфізм: поняття та використання. Віртуальні функції
	Тема – 2. Шаблони та обробка виключень Лекція 14. Шаблони та обробка виключень: родові функції та класи, виключення в програмах

### Календарний план запланованих заходів: лабораторні роботи

№	Види робіт за розділами та темами
1	Розділ 4. Введення та виведення даних у C++. Робота з файлами Тема – 2. Файлове введення та виведення даних Лабораторна робота 11. Введення та виведення даних: робота з файлами
	Розділ 5. Віртуальні функції, шаблони та виключення, RTTI Тема – 1. Поліморфізм і віртуальні функції в програмах Лабораторна робота 13. Робота з віртуальними функціями. Поліморфізм

### Календарний план запланованих заходів: практичні заняття

№	Види робіт за розділами та темами
1	Розділ 3. Конструкції мови C++: перевантаження та наслідування Тема – 2. Використання наслідування (успадкування) в класах Практичне заняття 4. Наслідування класів
	Розділ 5. Віртуальні функції, шаблони та виключення, RTTI Тема – 2. Шаблони та обробка виключень Практичне заняття 7. Шаблони та обробка виключень

## 2. Самостійна робота студента

Самостійна робота студента в кредитному модулі «Основи програмування. Частина 2. Методології програмування» розрахована на 168 годин за приблизним розподілом в таблиці.

№	Вид самостійної роботи	Кількість годин СРС
1	Опрацювання тем, які винесені на самостійне опрацювання	45
2	Виконання лабораторних робіт, що підлягають оцінюванню	81
3	Виконання практичних завдань для відпрацювання навичок	30
4	Підготовка до написанні МКР	4
5	Підготовка до екзамену	8

Самостійна робота складається з завдань, які студенти мають виконати самостійно в поза аудиторні години навчання. Завдання містять як теоретичні, так і практичні задачі.

Далі наведений перелік завдань за термінами їх виконання протягом семестру.

### Календарний план запланованої роботи студентів, що винесена на самостійне опрацювання

№	Види робіт за розділами та темами	
Розділ 1. Додаткові відомості мови С. Структури даних		
1	Тема. Прості абстрактні структури даних	Лекція 1. Прості абстрактні структури даних Лабораторна робота 1. Структури даних
2		Лекція 2. Прості абстрактні структури даних. Продовження Лабораторна робота 2. Структури даних. Продовження Практичне заняття 1. Структури даних на прикладах: масиви та списки, робота з деревами, хеш-структури
Розділ 2. Введення в мову С++: основні поняття та конструкції		
3	Тема – 1. Вступ до С++: основні поняття	Лабораторна робота 3. Введення та виведення даних, приведення типів, конструктори та деструктори класів
4	Тема – 2. Вступ до С++: основні конструкції	Лекція 4. Вступ до С++. Продовження Лабораторна робота 4. Основні конструкції: класи, об'єкти та функції Практичне заняття 2. Основні конструкції
5	Тема – 3. Масиви та використання вказівників	Лекція 5. Масиви, вказівники та посилання Лабораторна робота 5. Робота з масивами та їх елементами: виділення пам'яті та контроль границь
Розділ 3. Конструкції мови С++: перевантаження та наслідування		
6	Тема – 1. Використання перевантажених конструкцій	Лабораторна робота 6. Робота з перевантаженими функціями Практичне заняття 3. Конструктори: перевантаження та копіювання
7		Лабораторна робота 7. Робота з перевантаженими операторами
8	Тема – 2. Використання наслідування (успадкування) в класах	Лабораторна робота 8. Робота з наслідуваннями
9		Лекція 9. Наслідування. Продовження Лабораторна робота 9. Робота з наслідуваннями. Продовження

<b>Розділ 4. Введення та виведення даних у C++. Робота з файлами</b>		
10	Тема – 1. Введення та виведення даних	Лекція 10. Введення та виведення даних Лабораторна робота 10. Введення та виведення даних: прапори форматів, маніпулятори вводу/виводу Практичне заняття 5. Введення та виведення даних
11	Тема – 2. Файлове введення та виведення даних	Лекція 12. Файлове введення та виведення даних. Продовження Лабораторна робота 12. Робота з файлами. Продовження Практичне заняття 6. Файлове введення та виведення даних
<b>Розділ 5. Віртуальні функції, шаблони та виключення, RTTI</b>		
12	Тема – 2. Шаблони та обробка виключень	Лабораторна робота 14. Робота з родовими функціями та класами
13	Тема – 3. Типи даних і дії з ними	Лекція 15. Ідентифікація типів даних (RTTI), оператори приведення типів Лабораторна робота 15. Робота з операторами приведення типів
<b>Розділ 6. Простори імен і шаблони проектування</b>		
14	Тема – 1. Простори імен і перетворення типів	Лекція 16. Простори імен, функції перетворення типів і додаткові питання Лабораторна робота 16. Функції перетворення типів Практичне заняття 8. Простори імен
15	Тема – 2. Шаблони проектування	Лекція 17. Шаблони проектування Лабораторна робота 17. Функції перетворення типів. Продовження

## **Політика та контроль**

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

В освітньому компоненті «Основи програмування. Частина 2. Методології програмування» протягом семестру передбачено проведення: лекційних і практичних занять (комп'ютерних практикумів) та лабораторних робіт.

#### **Відвідування занять**

Відвідування усіх видів аудиторних занять, а саме: лекційних, практичних (комп'ютерних практикумів) і лабораторних є обов'язковим.

Виняток становить лише відсутність на заняттях з поважних причин (тобто хвороби, форм-мажорні обставини, участь в олімпіадах, тощо), про що студент має повідомити викладача та надати підтвердження причини. У разі відсутності студента на аудиторному занятті з поважних причин нарахування штрафних балів не здійснюється.

#### **Правила захисту лабораторних робіт**

Протягом семестру студенти мають виконувати і захищати лабораторні роботи. Для кожної з робіт встановлені граничні терміни захисту (дедлайни).

Для отримання максимальних балів за кожну з передбачених програмою робіт студент повинен виконати роботу в повній мірі, відповідно теми, без помилок і вчасно її захистити.

Якщо студент пропустив термін захисту роботи з поважної причини, він має можливість виконати, здати та захистити роботу без втрати балів.

Якщо студент пропустив термін захисту роботи без поважних причин – йому надається можливість відпрацювання пропущеної роботи шляхом написання самостійної роботи за відповідною темою. Але в такому випадку максимальний бал за роботу становитиме 75% від найвищої оцінки за завдання.

Причиною, що унеможлиблює захист пропущених лабораторних робіт є дедлайн на відповідну роботу та закінчення навчального процесу (семестру).

### **Політика дедлайнів та перескладань**

На кожен лабораторну роботу за календарним планом кредитного модуля відводиться по два тижні для виконання та захисту роботи. Всі терміни оголошуються студентам на початку семестру.

Перескладання лабораторних робіт не передбачено. Передбачено відпрацювання у вигляді самостійної роботи протягом двох тижнів після завершення основного дедлайну на захист відповідної роботи.

В кінці кожного семестру передбачено перескладання модульної контрольної роботи.

## **2. Види контролю результатів навчання**

Для кредитного модуля «Основи програмування. Частина 2. Методології програмування» передбачені наступні види контролю:

- проведення поточного контролю;
- проведення семестрового контролю.

### **Поточний контроль**

Поточний контроль протягом семестру проводиться у вигляді захисту лабораторних робіт і модульної контрольної роботи. Поточний контроль має на меті перевірку знань та вмінь студентів, засвоєних і здобутих ними на усіх видах аудиторних занять.

Модульна контрольна робота проводиться за матеріалами лекційних і практичних занять, лабораторних робіт. Завдання складаються з теоретичних і практичних запитань. Контрольна робота проводиться в кінці семестру. Питання та опорні теми для повторення оголошуються студентам заздалегідь.

Для проведення модульної контрольної роботи виділяються дві учбові години за рахунок лекційного заняття.

### **Семестровий контроль**

Семестровий контроль з кредитного модуля здійснюється у вигляді екзамену.

Умовами допуску до здачі екзамену є виконання усіх завдань, передбачених лабораторними роботами. А також, написання самостійних робіт календарного контролю та модульної контрольної роботи на оцінки не нижче оцінки «достатньо» (60% від максимального балу). Загальний рейтинг студента за семестр для допуску до екзамену має становити не менше 36 балів.



Екзамен проводиться в письмовому вигляді, містить теоретичні питання та практичні завдання. Перелік питань та опорні теми для підготовки до екзамену заздалегідь надаються студентам.

### **3. Рейтингова система оцінювання результатів навчання**

#### **Система оцінювання результатів навчання**

Оцінювання результатів навчання здійснюється згідно рейтингової системи оцінювання за 100-бальною шкалою.

Рейтинг студента складається з балів за захист лабораторних робіт, написання модульної контрольної роботи та складання іспиту.

Також під час роботи на лекційних і практичних заняттях студенти можуть отримати заохочувальні додаткові бали. За невиконання умов навчання передбачені штрафні бали. Кількість заохочувальних або штрафних балів рівна 5 балам на семестр.

#### **Оцінювання лабораторних робіт**

Для освітнього компонента «Основи програмування. Частина 2. Методології програмування» передбачено 16 лабораторних робіт на оцінку. Оцінюються роботи по 3 бали кожна за наступними критеріями:

- бездоганна робота оцінюється в 3 бали – завдання виконано в повній мірі, вирішує всі поставлені задачі, виконано без помилок або є лише незначні недоліки;
- є певні недоліки в виконанні роботи – 2-1 бали – завдання виконано неповністю, допущені суттєві помилки;
- робота не виконана, не захищена або не зданий звіт – 0 балів – робота здана лише у вигляді звіту, але не захищена особисто; робота здана особисто, але не має звіту; робота виконана менш, ніж на 60% або здана пізніше дедлайну.

#### **Оцінювання самостійних робіт**

Самостійні роботи, що виконуються студентами з умовою відпрацювання пропущених робіт без поважної причини оцінюються в 2,25 балів за критеріями відповідними лабораторним роботам, без здачі звіту.

#### **Оцінювання модульної контрольної роботи**

Контрольна робота складається з чотирьох завдань: двох теоретичних питань і двох практичних завдань. Оцінювання завдань контрольної роботи виконується за таким розподілом: теоретичні питання оцінюються в 2 бали, а практичні завдання по 4 бали кожне.

Оцінюються питання модульної контрольної роботи за такими критеріями:

- «відмінно», повна відповідь, не менше 90% потрібної інформації, програмний код написаний згідно стандартів кодування, вирішує основне завдання, включає всі необхідні програмні модулі, конструкції та хоча б частину перевірок; відповіді написані точними формулюваннями та розгорнуто;

– «добре», достатньо повна відповідь, не менше 75% потрібної інформації, що виконана згідно з вимогами, є незначні неточності, відсутні перевірки та код неструктурований; відповіді короткі, але точні;

– «задовільно», неповна відповідь, не менше 60% потрібної інформації, завдання виконане з певними недоліками, в коді відсутні перевірки, наявні помилки; відповіді неповні;

– «незадовільно», відповідь не відповідає умовам до «достатньо», наявні суттєві помилки, відповідь відсутня або не вирішує поставлені задачі.

### **Оцінювання екзаменаційної роботи**

Умовою допуску до екзамену є зарахування всіх лабораторних робіт, написання модульної контрольної роботи і стартовий рейтинг не менше 36 балів.

Екзамен студенти виконують письмово. Білети складаються з теоретичних питань і практичних завдань. Теоретичні запитання оцінюються по 8 балів кожне, практичні завдання по 12 балів кожне, за наступними критеріями:

– «відмінно», повна відповідь, не менше 90% потрібної інформації, що виконана згідно з вимогами до рівня вмінь;

– «добре», достатньо повна відповідь, не менше 75% потрібної інформації, що виконана згідно з вимогами до рівня вмінь або є незначні неточності;

– «задовільно», неповна відповідь, не менше 60% потрібної інформації, завдання виконане з певними недоліками;

– «незадовільно», відповідь не відповідає умовам до «задовільно» – 0 балів.

Сума стартових балів та балів за екзаменаційну роботу переводиться до екзаменаційної оцінки згідно з таблицею:

<b>Бали</b>	<b>Оцінка</b>
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

### **Робочу програму навчальної дисципліни (силабус):**

#### **Складено**

доцент, к.т.н., Шпурик Вадим Вадимович,  
старший викладач Оленєва Ксенія Миколаївна

**Ухвалено**

кафедрою ІПЗЕ (протокол № 34 від 10.05.2024 р.)

**Погоджено**

Методичною комісією ННІАТЕ (протокол № 9 від 31.05.2024 р.)