

**КЛАСИЧНИЙ ПРИВАТНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ**

**СИЛАБУС**

**навчальної дисципліни**  
**«СУЧАСНІ МОВИ ПРОГРАМУВАННЯ»**

**КОНТАКТНА ІНФОРМАЦІЯ ТА ТЕХНІЧНОЇ ДОПОМОГА**  
(включаючи електронну пошту, робочий час / місцезнаходження тощо).

|   |  |
|---|--|
| <b>Викладач (-і)</b>  | Хрипко Сергій Леонідович<br>Бречко Наталія Анатоліївна<br>Ковтун Володимир Андрійович  |
| <b>Контактний тел.</b>  | +38(061) 228-07-85 (внутр. 224)  |
| <b>E-mail:</b>  | <a href="mailto:ur9qq@ukr.net">ur9qq@ukr.net</a>   |
| <b>Сторінка курсу на сайті підтримки навчальних програм КПУ</b> | <a href="http://www.zhu.edu.ua/cpu_edu/course/view.php?id=4972">http://www.zhu.edu.ua/cpu_edu/course/view.php?id=4972</a><br><a href="http://www.zhu.edu.ua/cpu_edu/course/view.php?id=4919">http://www.zhu.edu.ua/cpu_edu/course/view.php?id=4919</a><br><a href="http://www.zhu.edu.ua/cpu_edu/course/view.php?id=5279">http://www.zhu.edu.ua/cpu_edu/course/view.php?id=5279</a><br><a href="http://www.zhu.edu.ua/cpu_edu/course/view.php?id=5337">http://www.zhu.edu.ua/cpu_edu/course/view.php?id=5337</a> |
| <b>Консультації</b>   | <i>Очні консультації:</i> за графіком консультацій викладача, а. 311, головний корпус КПУ<br><i>Консультації off-line:</i> шляхом повідомлення на сторінці навчальної дисципліни сайту підтримки навчальних програм КПУ <a href="http://www.zhu.edu.ua/cpu_edu/message">http://www.zhu.edu.ua/cpu_edu/message</a>  |

**АНОТАЦІЯ**

Навчальна дисципліна «Сучасні мови програмування» є нормативною для студентів першого (бакалаврського) рівня вищої освіти галузі знань 01 Освіта/Педагогіка за спеціальністю 014 Середня освіта, освітня програма: Інформатика.

Згідно з навчальним планом денної форми навчання вивчення дисципліни заплановано на 4, 5, 6, 7 семестри (2,3 та 4 курсі).

Особливістю курсу для спеціальності 014 Середня освіта освітня програма є його методична спрямованість на підготовку майбутнього вчителя до викладання сучасних мов програмування в закладах загальної середньої освіти з акцентом на наочність та інтерактивність.

Освітній процес з дисципліни здійснюється за такими формами: навчальні заняття; самостійна робота; контрольні заходи. Видами навчальних занять згідно з навчальним планом є лекції; практичні заняття, а також консультації.

В 4 семестрі 2 курсу дисципліна передбачає опанування мови програмування Python та оволодіння студентами принципами побудови графічного інтерфейсу користувача (GUI) на функціональній основі. У межах курсу вивчаються механізми подієво-орієнтованого програмування, зокрема обробка подій і організація взаємодії між елементами інтерфейсу та логікою програми. Також розглядаються підходи до створення динамічної графіки й інтерактивних візуалізацій для демонстрації

роботи алгоритмів і структур даних.

В 5 семестрі 3 курсу дисципліна передбачає опанування мови програмування Java отримання студентами базових знань і практичних навичок використання технології та мови програмування Java, вивчення реалізації об'єктно-орієнтованих підходів та технологій, а також отримання навичок використання бібліотечних класів Java при розробці програмних продуктів. Практичні заняття передбачають розробку прикладних проектів освітнього спрямування, таких як тренажери, тести, калькулятори та візуалізатори сортування даних, що готує студентів до керівництва гуртками програмування та підготовки учнів до олімпіад.

В 6 семестрі 3 курсу дисципліна спрямована на опанування сучасної платформи .NET та мови C# з орієнтацією на розробку навчальних програмних продуктів, дидактичних тренажерів, а також на методику пояснення ключових понять програмування учням. Практичні заняття передбачають виконання завдань з розробки консольних і простих GUI-застосунків (WPF), використання подій/делегатів, файлового введення-виведення та базових елементів архітектури MVVM.

В 7 семестрі 4 курсу дисципліна передбачає використання сучасних мов веб-програмування HTML, CSS та JavaScript і практичні заняття з опанування побудови HTML-документів та структури веб-сторінки, роботи з тегами й атрибутами, гіперпосиланнями та навігаційними структурами, а також створення форм і елементів взаємодії з користувачем. Особливу увагу приділено застосуванню каскадних таблиць стилів (CSS) для оформлення сторінок, використанню кольорів і шрифтів, побудові простих макетів, основам адаптивної верстки та медіа-запитам. Також розглядаються базові засоби інтерактивності за допомогою JavaScript: робота з подіями та елементами сторінки, інтеграція HTML, CSS і JavaScript у межах одного веб-проєкту та створення простої лендінг-сторінки освітнього спрямування.

Самостійна робота проводиться під час аудиторних занять та в час, вільний від обов'язкових навчальних занять, без участі викладача шляхом самостійного опрацювання лекційного матеріалу, виконання індивідуальних завдань з кожного модуля курсу. Повний курс лекційного матеріалу та методичні рекомендації до виконання індивідуальних домашніх завдань розміщено на сторінках дисципліни сайту підтримки навчальних програм університету.

Консультації призначені для роз'яснення студентам теоретичних або практичних питань.

Засвоєння навчального матеріалу перевіряється за допомогою поточного контролю, який здійснюється на практичних заняттях у формі усних відповідей, самостійних робіт та перевірки виконання завдань, виконання практичних робіт. Для визначення результатів модульного та підсумкового контролю використовується система накопичення балів, яка стимулює систематичну роботу студента протягом семестру.

Підсумковий (семестровий) контроль після завершення 4, 5, 6, 7 семестрів здійснюється у формі екзамену.

### ФОРМАТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

| Курс | Семестр | Всього кредитів | Всього годин | Аудиторних годин | Лекцій | Практич. (сем.) | Сам. робота |
|------|---------|-----------------|--------------|------------------|--------|-----------------|-------------|
| 2    | 4       | 4               | 120          | 48               | 14     | 34              | 72          |
| 3    | 5       | 4               | 120          | 48               | 16     | 32              | 72          |
| 3    | 6       | 4               | 120          | 48               | 24     | 24              | 72          |
| 4    | 7       | 4               | 120          | 48               | 24     | 24              | 72          |

### ОЗНАКИ ДИСЦИПЛІНИ

| Навчальний рік | Курс (рік навчання) | Семестр    | Цикл підготовки | Нормативна/вибіркова | Форма контролю |
|----------------|---------------------|------------|-----------------|----------------------|----------------|
| (за планом)    | 2, 3, 4             | 4, 5, 6, 7 | професійна      | нормативна           | екзамен        |

# МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

## Мета навчальної дисципліни

Метою навчальної дисципліни є формування у здобувачів освіти системи теоретичних знань і практичних навичок розробки прикладного програмного забезпечення та веб-ресурсів із використанням сучасних мов і технологій програмування Java, Python, C# (.NET), HTML, CSS та JavaScript. Дисципліна спрямована на опанування базових принципів алгоритмізації, об'єктно-орієнтованого та подієво-орієнтованого програмування, створення графічних інтерфейсів користувача (GUI), а також проектування й реалізації структурованих та інтерактивних веб-сторінок освітнього призначення з орієнтацією на подальшу професійну й педагогічну діяльність.

## Завдання навчальної дисципліни

Завданнями навчальної дисципліни є:

- опанування синтаксису, семантики та інструментальних засобів розробки програм мовами Java, Python і C#, а також набуття умінь компіляції, налагодження та документування коду;
- засвоєння принципів виконання програм у середовищі віртуальної машини Java та загальних підходів до побудови програм на платформі .NET;
- формування системного розуміння об'єктно-орієнтованого програмування (класи, об'єкти, інкапсуляція, наслідування, поліморфізм) і застосування ООП для реалізації прикладних задач;
- вивчення основ роботи з даними: примітивних типів, операторів, керувальних конструкцій, масивів, колекцій та типових структур даних;
- ознайомлення з подієво-орієнтованим програмуванням і набуття практики створення інтерактивних додатків із графічним інтерфейсом на основі бібліотек GUI (зокрема у Python);
- розвиток навичок декомпозиції задач і проектування візуальних оболонок для алгоритмічних задач, а також візуалізації роботи базових алгоритмів і структур даних;
- засвоєння принципів веб-програмування: побудови структури сторінки HTML, стилізації засобами CSS, створення базової інтерактивності за допомогою JavaScript, організації контенту та навігації;
- формування вмінь використовувати редактори коду, інструменти розробника браузера та допоміжні онлайн/офлайн засоби для створення, тестування й удосконалення програм та веб-ресурсів;
- розвиток здатності проектувати зручні інтерфейси та оцінювати ергономічні, візуальні й функціональні характеристики програмних продуктів і веб-сторінок (зокрема освітнього призначення).

## ЗАПЛАНОВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

У результаті вивчення дисципліни здобувачі освіти повинні **знати**:

- базові принципи створення та виконання програм мовами Java, Python, C#, зокрема роль JVM і загальні засади платформи .NET;
  - примітивні типи даних, оператори, керувальні конструкції та способи організації обчислень у програмах;
  - концепції об'єктно-орієнтованого програмування та способи їх реалізації в Java і C#;
  - основи роботи з масивами, колекціями, типовими структурами даних, а також базові принципи використання бібліотек і утилітарних класів;
  - механізми потоків введення-виведення, обробки текстової інформації, дат і виключень (на рівні базових бібліотечних можливостей);
  - архітектуру подієво-орієнтованого програмування, компоненти GUI, віджети, механізми обробки подій і взаємозв'язок між інтерфейсом та логікою програми;
  - принципи веб-програмування: призначення та можливості HTML, CSS, JavaScript, логіку побудови структури веб-сторінки, основи створення зрозумілих веб-інтерфейсів та узгодження елементів дизайну в межах веб-проєкту.
- Після вивчення дисципліни здобувачі освіти повинні **вміти**:
- розробляти програми в інтегрованих середовищах розробки, виконувати компіляцію, налагодження та документування коду;
  - проєктувати й реалізовувати класи та об'єкти, застосовуючи принципи ООП для розв'язання прикладних і алгоритмічних задач;
  - використовувати колекції, бібліотечні класи та методи для зберігання, обробки й аналізу даних;
  - створювати базові консольні та GUI-застосунки, організовувати взаємодію користувача з

програмою через систему подій;

- реалізовувати введення й виведення даних через графічні елементи, створювати багатовіконні інтерфейси та інтерактивні візуалізації алгоритмів і структур даних;
- самостійно створювати веб-сторінки та прості веб-проекти: формувати структуру в HTML, стилізувати за допомогою CSS, додавати базову інтерактивність засобами JavaScript;
- проєктувати навігацію та структуру веб-ресурсів з урахуванням логіки подання навчального матеріалу, потреб користувачів та базових принципів адаптивної верстки;
- аналізувати й оцінювати зручність використання та функціональні можливості створених програмних продуктів і веб-ресурсів, удосконалювати їх за результатами тестування.

Відповідно до освітньо-професійної програми підготовки бакалавра галузі знань 01 Освіта/Педагогіка за спеціальністю 014 Середня освіта, освітня програма: Інформатика вивчення дисципліни «Сучасні мови програмування» сприяє формуванню **компетентностей та програмних результатів навчання:**

**Інтегральна компетентність:**

Здатність розв'язувати складні спеціалізовані задачі у галузі середньої освіти, що передбачає застосування теоретичних знань і практичних умінь з наук предметної спеціальності, педагогіки, психології, теорії та методики навчання і характеризується комплексністю та невизначеністю умов організації освітнього процесу в закладах загальної середньої освіти.

**Загальні компетентності:**

ЗК 2. Здатність до абстрактного мислення, аналізу та синтезу. Володіння методами логічного та системного аналізу, здатність виявляти закономірності в інформаційних процесах та розв'язувати складні завдання предметної області.

ЗК 3. Інформаційно-комунікаційна компетентність. Здатність ефективно використовувати сучасні ІКТ для пошуку, обробки, критичної оцінки та поширення інформації в освітньому та соціальному просторах.

ЗК 6. Здатність до навчання впродовж життя. Здатність самостійно визначати потреби власного розвитку, опановувати нові знання та технології, бути готовим до професійної адаптації.

ЗК 7. Здатність до креативності та інновацій. Здатність генерувати нові ідеї, виявляти гнучкість у професійних ситуаціях та застосовувати творчий підхід до розв'язання практичних проблем.

ЗК 8. Здатність спілкуватися іноземною мовою. Здатність до комунікації іноземною мовою в професійній діяльності; здатність розуміти іншомовні фахові тексти.

**Спеціальні (фахові) компетенції:**

СК 3. Програмно-технологічна компетентність. Здатність розробляти програмні продукти, використовуючи сучасні мови програмування, знання архітектури ПК, операційних систем та мереж.

**Програмні результати навчання:**

РН 3. Використовує сучасні мови, методи та технології програмування для розв'язання практичних завдань і створення програмних продуктів.

РН 15. Демонструє готовність до навчання впродовж життя, професійного самовдосконалення та адаптації до змін у цифровій галузі.

РН 16. Використовує іноземну мову для аналізу фахових джерел та технічної документації, а також для професійної комунікації у цифровому середовищі.

**ПЛАН КУРСУ**  
**(в 4 семестрі 2 курсу мова Python)**

| Тема  | Лекц. | Практ. | Самостійна робота  |
|---|-------|--------|--|
| <b>Змістовий модуль № 1. Основи GUI та подієве програмування</b>            |       |        |  |
| Тема №1. Побудова віконних інтерфейсів та базові віджети                    | 2     | 2      | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Використання властивості anchor для вирівнювання тексту.<br>2) Вивчення параметрів рельєфу та меж віджетів.<br>3. Виконання індивідуального завдання № 1.<br>4. Виконання тестового завдання на сайті підтримки навчальних програм.  |
| Тема №2. Робота з даними та менеджери компонування                          | 2     | 6      | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Використання методу place для точного позиціонування за координатами. Очищення та блокування полів введення після виконання операцій.<br>3. Виконання індивідуального завдання № 2.<br>4. Виконання тестового завдання на сайті підтримки навчальних програм.  |
| Тема №3. Функції як обробники подій   | 2     | 6      | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Використання методів get та delete для керування вмістом полів.<br>2) Створення функцій для динамічної зміни тексту на мітках.<br>3. Виконання індивідуального завдання № 3.<br>4. Виконання тестового завдання на сайті підтримки навчальних програм.<br>5. Підготовка до модульного контролю за темами 1-3.          |
| <b>Змістовий модуль № 2. Візуалізація алгоритмів та складені інтерфейси</b> |       |        |  |
| Тема №4. Вибір варіантів та меню програми                                   | 2     | 6      | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Створення контекстного меню, що викликається правою кнопкою миші.<br>2) Робота з віджетом вибору числових значень Scale.<br>3. Виконання індивідуального завдання № 4.<br>4. Виконання тестового завдання на сайті підтримки навчальних програм.   |
| Тема №5. Графічне полотно та динамічне малювання                            | 2     | 6      | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Використання ідентифікаторів (ID) та тегів для керування групою фігур.<br>2) Зміна кольору фігур залежно від їхніх значень.<br>3. Виконання індивідуального завдання № 5.<br>4. Виконання тестового завдання на сайті підтримки навчальних програм.  |
| Тема №6. Візуалізація сортування та пошуку                                  | 4     | 6      | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Візуалізація алгоритму пошуку мінімального/максимального значення.<br>2) Створення інтерфейсу для порівняння швидкості двох алгоритмів.<br>3. Виконання індивідуального завдання № 6.<br>4. Виконання тестового завдання на сайті підтримки навчальних програм.<br>5. Підготовка до модульного контролю за темами 4-6. |
| Разом   | 14    | 34     | 72   |

**ПЛАН КУРСУ**  
**(в 5 семестрі 3 курсу мова Java)**

| Тема   | Лекц. | Практ. | Самостійна робота   |
|--|-------|--------|---|
| <b>Змістовий модуль № 1. Основні концепції мови</b>  |       |        |   |
| Тема № 1. Вступ до Java, Встановлення Java JDK та IntelliJ IDEA  | 2     | 4      | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Різниця між JDK, JRE та JVM.<br>2)Налаштування змінних середовища (PATH, JAVA_HOME) та перевірка встановлення через консольні команди.<br>3.Виконання індивідуального завдання №1 (налаштування середовища + створення першого проєкту “Hello, Java”).<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 1.   |
| Тема №2. Основні концепції мови. Виведення інформації. Змінні та типи даних і оператори, управління виконанням програми. | 2     | 4      | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Примітивні типи даних: діапазони значень, перетворення типів (implicit/explicit casting).<br>2)Оператори: арифметичні, порівняння, логічні, інкремент/декремент; пріоритет операторів.<br>3.Виконання індивідуального завдання №2 (обчислювальні задачі з введенням/виведенням, перевіркою коректності даних).<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 2.             |
| Тема №3. Умовні конструкції Java (if-else, switch-case)  | 2     | 4      | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Вкладені умови та складені логічні вирази (AND/OR/NOT), коротке замикання.<br>2)Особливості switch (типи, значення case, default, break), типові помилки.<br>3.Виконання індивідуального завдання №3 (задачі на розгалуження: тариф/оцінювання/категоризація даних).<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 3.<br>5.Підготовка до модульного контролю за темами 1–3. |
| <b>Змістовий модуль № 2. Об'єктно-орієнтоване програмування в Java</b>   |       |        |   |
| Тема №4. Цикли та оператори в Java (for, while, do-while).   | 2     | 4      | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Порівняння циклів for/while/do-while: коли який застосовувати.<br>2)Оператори break/continue, вкладені цикли, типові помилки (нескінченні цикли, off-by-one).<br>3.Виконання індивідуального завдання №4 (задачі на цикли: суми/добутки, табличні обчислення, пошук мін/макс, перевірки).<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 4.                                  |
| Тема №5. Функції у Java (методи)   | 2     | 4      | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Оголошення та виклик методів, параметри, повернення значення, область видимості змінних.<br>2)Перевантаження методів, передача параметрів (значення/посилання), робота з масивами у методах.<br>3.Виконання індивідуального завдання №5 (винесення логіки в методи, створення бібліотеки допоміжних методів).<br>4.Виконання тестового завдання за темою 5.  |

|   |    |    |   |
|---|----|----|---|
| Тема №6. “Створення класів та об'єктів” | 2  | 6  | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Класи/об'єкти/поля/методи, конструктори, this, інкапсуляція та модифікатори доступу.<br>2)Статичні члени класу (static), перевизначення toString(), базові принципи проектування класів (єдина відповідальність).<br>3.Виконання індивідуального завдання №6 (мініпроект: модель предметної області — “Студент”, “Товар”, “Замовлення” тощо; створення кількох класів і взаємодії між ними).<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 6.                               |
| Тема №7. “Інтерфейси в Java”.           | 4  | 6  | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Інтерфейси та їх роль у поліморфізмі: “контракт”, реалізація, множинна реалізація.<br>2)default/static методи в інтерфейсах, порівняння з абстрактними класами, типові сценарії використання.<br>3.Виконання індивідуального завдання №7 (побудова моделі з 2–3 інтерфейсами: наприклад, “Payable/Deliverable/Comparable”, реалізації та демонстрація поліморфізму).<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 7.<br>5.Підготовка до модульного контролю за темами 4–7. |
| Разом                                   | 16 | 32 | 72  |

**ПЛАН КУРСУ**  
**(в 6 семестрі 3 курсу мова C#)**

| Тема  | Лекц. | Практ. | Самостійна робота  |
|---|-------|--------|--|
| <b>Змістовий модуль № 1. Основи платформи та базовий синтаксис</b>  |       |        |  |
| Тема № 1. Вступ до .NET. Налаштування IDE. Структура програми       | 4     | 2      | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Опрацювати матеріали про .NET SDK, збірку та запуск.<br>2)Підготувати коротку інструкцію для учнів.<br>3.Виконання індивідуального завдання №1.<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 1. |
| Тема № 2. Змінні, типи даних та оператори. Робота з консоллю        | 2     | 4      | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Виконати вправи на перетворення типів, 2)валідацію введення.<br>3.Виконання індивідуального завдання №2.<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 2.  |
| Тема № 3. Умовні конструкції (if-else, switch). Логіка розгалужень. | 4     | 2      | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Розв'язати задач на розгалуження; 2)підготувати пояснення типових помилок.<br>3.Виконання індивідуального завдання №3.<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 3.                          |
| Тема № 4. Цикли (for, while,  | 2     | 4      | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Розв'язати задачі на сумування, пошук, 2)табличні   |

|  |    |    |  |
|--|----|----|--|
| foreach) та робота з масивами.   |    |    | значення.<br>3.Виконання індивідуального завдання №4.<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 4.<br>5.Підготовка до модульного контролю за темами 1–4.  |
| <b>Змістовий модуль № 2. Розширені можливості C# та розробка освітніх ресурсів</b> |    |    |  |
| Тема № 5. Методи (функції). Параметри та рекурсія в навчальних задачах..           | 4  | 2  | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Підготувати приклади: сортування, 2)пошук, агрегація<br>3.Виконання індивідуального завдання №5.<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 5.  |
| Тема № 6. Класи та об'єкти. Властивості, конструктори та інкапсуляція              | 2  | 2  | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Виконати вправи з аналізу тексту, 2)підрахунок слів/символів.<br>3.Виконання індивідуального завдання №6.<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 6.   |
| Тема № 7. Успадкування, поліморфізм та інтерфейси.                                 | 2  | 2  | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Додати керування параметрами симуляції (Slider/Buttons), 2)журнал подій.<br>3.Виконання індивідуального завдання №7.<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 7.  |
| Тема № 8. Розробка підсумкового проекту (освітній тренажер/тестер).                | 4  | 6  | 1.Опрацювання лекційного матеріалу.<br>2.Самостійне опрацювання теоретичних питань:<br>1)Перенести симулятор у MVVM: binding журналу і команд Start/Stop/Step/Reset.<br>3.Виконання індивідуального завдання №8.<br>4.Виконання тестового завдання на сайті підтримки навчальних програм за темою 8.<br>5.Підготовка до модульного контролю за темами 5–8. |
| Разом  | 24 | 24 | 72   |

**ПЛАН КУРСУ**  
(в 7 семестрі 4 курсу мова HTML, CSS, JS)

| Тема  | Лекц. | Практ. | Самостійна робота   |
|---|-------|--------|---|
| <b>Змістовий модуль № 1. Основи веб- технологій та мови розмітки HTML</b> |       |        |   |
| Тема 1. Веб-програмування як складова сучасних цифрових технологій        | 2     | 2      | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Веб як освітнє середовище: можливості та переваги<br>2) Статичні та динамічні веб-сторінки<br>3) Роль програмування в інтерфейсах<br>4) Типи освітніх веб-ресурсів.<br>3. Проходження тестування на сайті підтримки навчальних програм. |
| Тема 2. Мова HTML та принципи побудови веб-документів                     | 2     | 2      | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Призначення мови HTML<br>2) HTML як основа веб-сторінки<br>3) Вплив HTML на сприйняття контенту.<br>3. Проходження тестування на сайті підтримки навчальних програм.  |

|  |   |   |   |
|--|---|---|---|
| Тема 3. Робота з текстом та контентом HTML                                       | 2 | 2 | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Мультимедійний контент в освіті<br>2) Організація текстової інформації<br>3) Форми в навчальному процесі.<br>3. Виконання домашніх завдань, опрацювання першоджерел та навчальної літератури; виконання індивідуального завдання.   |
| Тема 4. Візуальні компоненти HTML-сторінки                                       | 2 | 2 | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Розміщення графічних матеріалів у HTML<br>2) Вбудовування посилань і навігації<br>3) Значення медіа-контенту в проєктах.<br>3. Проходження тестування на сайті підтримки навчальних програм.  |
| <b>Змістовий модуль № 2. Стилiзація, верстка та iнтерактивнiсть веб-сторiнок</b> |   |   |   |
| Тема 5. Принципи використання CSS у веб-розробці                                 | 2 | 2 | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Вплив CSS на сприйняття інформації<br>2) Обмеження стилів в освітніх ресурсах<br>3) Основні типи CSS-селекторів.<br>3. Проходження тестування на сайті підтримки навчальних програм.  |
| Тема 6. Макетування веб-сторінок засобами CSS                                    | 1 | 2 | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Використання Flexbox<br>2) Оформлення навчальних матеріалів<br>3) Верстка і сприйняття інформації.<br>3. Виконання домашніх завдань, опрацювання першоджерел та навчальної літератури; виконання індивідуального завдання.  |
| Тема 7. Особливості адаптивного дизайну освітніх ресурсів                        | 2 | 2 | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Пристрої для навчання<br>2) Поведінка користувача на різних екранах<br>3) Адаптація елементів сторінки<br>4) Як перевірити адаптивність веб-сторінки.<br>3. Проходження тестування на сайті підтримки навчальних програм.   |
| Тема 8. Базові поняття та синтаксис JavaScript                                   | 2 | 2 | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Відмінності JavaScript від HTML і CSS<br>2) JavaScript в освітніх ресурсах<br>3) Як JavaScript підвищує інтерактивність сторінки.<br>3. Виконання домашніх завдань, опрацювання першоджерел та навчальної літератури; виконання індивідуального завдання. Проходження тестування на сайті підтримки навчальних програм. |
| Тема 9. Інтерактивність у веб-ресурсах освітнього призначення                    | 2 | 2 | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Інтерактивні компоненти в навчанні<br>2) Помірність інтерактивності<br>3) Залученість користувачів.<br>3. Виконання домашніх завдань, опрацювання першоджерел та навчальної літератури; виконання індивідуального завдання. Проходження тестування на сайті підтримки навчальних програм.                               |

| <b>Змістовий модуль № 3. Проектування та використання освітнього веб-ресурсу</b> |    |    |  |
|--|----|----|--|
| Тема 10. Архітектура персонального освітнього веб-ресурсу                        | 2  | 2  | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Контент сторінки вчителя<br>2) Навігація і зручність користування веб-сторінкою<br>3) Структурування інформації на сторінці.<br>3. Проходження тестування на сайті підтримки навчальних програм.                       |
| Тема 11. Інтеграція HTML, CSS і JavaScript                                       | 2  | 2  | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Структура веб-проєкту<br>2) Типові помилки інтеграції мов<br>3) Інструменти налагодження коду.<br>3. Виконання домашніх завдань, опрацювання першоджерел та навчальної літератури; виконання індивідуального завдання. |
| Тема 12. Тестування та вдосконалення веб-сторінки                                | 1  | 1  | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Значення тестування веб-сторінки<br>2) Браузерна перевірка<br>3) Як самостійно оцінити зручність власного сайту.<br>3. Проходження тестування на сайті підтримки навчальних програм.                                   |
| Тема 13. Освітні веб-ресурси у професійній діяльності вчителя                    | 2  | 1  | 1. Опрацювання лекційного матеріалу,<br>2. Самостійне опрацювання теоретичних питань:<br>1) Залучення учнів до роботи з веб-ресурсом<br>2) Оновлення контенту<br>3) Перспективи розвитку освітнього веб-ресурсу.<br>3. Проходження тестування на сайті підтримки навчальних програм.                               |
| Разом  | 24 | 24 | 72   |

## **ФОРМИ КОНТРОЛЮ ТА КРИТЕРІЇ ОЦІНЮВАННЯ**

У процесі вивчення навчальної дисципліни «Сучасні мови програмування» використовуються наступні види контролю:

1. Поточний контроль – здійснюється протягом семестру шляхом перевірки виконання тестових завдань, виконання лабораторних робіт, модульних контрольних робіт тощо. За змістом він включає перевірку ступеню засвоєння студентом навчального матеріалу, який охоплюється темою лекційного заняття, уміння самостійно опрацьовувати навчально-методичну літературу, здатність осмислювати зміст теми, уміння публічно та письмово представити певний матеріал, а також виконання завдань самостійної роботи.

2. Підсумковий семестровий контроль – здійснюється у формі екзамену відповідно до графіка освітнього процесу шляхом визначення ступеню засвоєння студентом навчальної дисципліни за результатами виконання обов'язкових завдання поточного (модульного) контролю.

Для оцінювання студентів використовується система накопичування балів. Згідно з «Положенням про організацію освітнього процесу в КПУ» підсумкова оцінка з дисципліни виставляється за 100-бальною шкалою з наступним переведенням у національну шкалу та шкалу ECTS.

Бали нараховуються за виконання завдань аудиторної роботи, практичних, лабораторних робіт, контрольних (модульних) завдань, тестів.

Результати поточного контролю здобувачів вищої освіти є складовими елементами підсумкової оцінки з дисципліни.

**РОЗПОДІЛ БАЛІВ ПОТОЧНОГО ТА ПІДСУМКОВОГО  
(СЕМЕСТРОВОГО) ОЦІНЮВАННЯ**

|           |  |    |    |                    |    |    |         |      |
|-----------|--|----|----|--------------------|----|----|---------|------|
| 4 семестр | Поточне оцінювання під час навчальних занять та самостійної роботи |    |    |                    |    |    | Екзамен | Сума |
|           | Змістовий модуль 1   |    |    | Змістовий модуль 2 |    |    |         |      |
|           | T1   | T2 | T3 | T4                 | T5 | T6 | 20      | 100  |
|           | 13   | 13 | 14 | 13                 | 13 | 14 |         |      |

|           |  |    |    |  |                    |    |    |         |      |     |
|-----------|--|----|----|--|--------------------|----|----|---------|------|-----|
| 5 семестр | Поточне оцінювання під час навчальних занять та самостійної роботи |    |    |  |                    |    |    | Екзамен | Сума |     |
|           | Змістовий модуль 1   |    |    |  | Змістовий модуль 2 |    |    |         |      |     |
|           | T1   | T2 | T3 |  | T4                 | T5 | T6 | T7      | 20   | 100 |
|           | 10   | 10 | 12 |  | 12                 | 12 | 12 | 12      |      |     |

|           |  |    |    |    |                    |    |    |    |         |      |
|-----------|--|----|----|----|--------------------|----|----|----|---------|------|
| 6 семестр | Поточне оцінювання під час навчальних занять та самостійної роботи |    |    |    |                    |    |    |    | Екзамен | Сума |
|           | Змістовий модуль 1   |    |    |    | Змістовий модуль 2 |    |    |    |         |      |
|           | T1   | T2 | T3 | T4 | T5                 | T6 | T7 | T8 | 20      | 100  |
|           | 10   | 10 | 10 | 10 | 10                 | 10 | 10 | 10 |         |      |

|           |  |    |    |    |                    |    |    |    |                    |     |     |     |         |      |     |
|-----------|--|----|----|----|--------------------|----|----|----|--------------------|-----|-----|-----|---------|------|-----|
| 7 семестр | Поточне оцінювання під час навчальних занять та самостійної роботи |    |    |    |                    |    |    |    |                    |     |     |     | Екзамен | Сума |     |
|           | Змістовий модуль 1   |    |    |    | Змістовий модуль 2 |    |    |    | Змістовий модуль 3 |     |     |     |         |      |     |
|           | T1   | T2 | T3 | T4 | T5                 | T6 | T7 | T8 | T9                 | T10 | T11 | T12 | T13     | 20   | 100 |
|           | 6  | 7  | 6  | 6  | 6                  | 7  | 6  | 6  | 6                  | 6   | 6   | 6   | 6       |      |     |

**Шкала оцінювання: 100-бальна, національна та ЄКТС**

| Сума балів за всі види навчальної діяльності | Оцінка за шкалою ЄКТС | Оцінка за національною шкалою                              |   |
|--|-----------------------|--|---|
|  |                       | для екзамену, курсового проекту (роботи), практики         | для заліку  |
| 90-100                                       | A                     | відмінно   | зараховано  |
| 82-89  | B                     | добре  |   |
| 75-81  | C                     |  |   |
| 67-74  | D                     | задовільно   |   |
| 60-66  | E                     |  |   |
| 35-59  | FX                    | незадовільно з можливістю повторного складання             | не зараховано з можливістю повторного складання             |
| 0-34   | F                     | незадовільно з обов'язковим повторним вивченням дисципліни | не зараховано з обов'язковим повторним вивченням дисципліни |

**Критерії оцінювання**

**Відмінно (А) . Бали: 90–100**

Здобувач вільно володіє принципами алгоритмізації, концепціями ООП (інкапсуляція, наслідування, поліморфізм) та архітектурою платформ .NET і JVM. На високому рівні розробляє програми в IDE, самостійно виконує налагодження та документування коду. Вміє створювати складні GUI-застосунки з інтерактивними візуалізаціями та адаптивні веб-сторінки з використанням HTML/CSS/JavaScript. Здатний самостійно оцінювати ергономіку та функціональність програмних продуктів.

**Добре (В). Бали: 82–89**

Здобувач на високому рівні знає синтаксис мов (Python, Java, C#), розуміє різницю між типами даних та операторами, вільно пояснює концепції ООП (класи, об'єкти, методи). Впевнено розробляє програмний код у середовищах VS Code, PyCharm або Visual Studio. Самостійно реалізує обробку виключних ситуацій та роботу з файловими потоками. Вміє створювати GUI-інтерфейси з коректною обробкою подій та розділяти логіку програми й інтерфейс користувача. Створює валідний HTML/CSS код, використовує базовий JavaScript для інтерактивності.

**Добре (С). Бали: 75–81**

Здобувач добре засвоїв основні теми курсу, розуміє принципи побудови алгоритмів та структуру програмних проєктів. Вміє писати працездатний код для розв'язання типових задач, використовує стандартні класи та бібліотеки мов програмування. Створює функціональні вікна застосунків (Tkinter, WPF), проте може потребувати незначних підказок при налаштуванні складних прив'язок даних (Data Binding). Самостійно розробляє структуру веб-сторінки, розуміє блочну модель CSS, але може припускатися неточностей у кросбраузерній верстці.

**Задовільно (D). Бали: 67–74.**

Здобувач володіє теоретичним матеріалом на репродуктивному рівні: знає основні конструкції мови, але має труднощі з поясненням глибоких архітектурних рішень (наприклад, принципів роботи JVM або CLR). Може написати простий код або модифікувати готовий приклад. Виконує базові операції з масивами та рядками. При написанні програм допускає логічні або синтаксичні помилки, які часто не може знайти та виправити самостійно без допомоги викладача.

**Задовільно (Е). Бали: 60–66**

Знання фрагментарні, обмежуються знанням базового синтаксису та основних термінів програмування. Виконує лише найпростіші завдання за чітким зразком або інструкцією. Здатний запустити готовий проєкт та внести мінімальні зміни в код. Демонструє мінімально допустимий поріг володіння інструментарієм розробки, необхідний для отримання позитивної оцінки.

**Незадовільно (FX) (з можливістю перескладання). Бали: 35–59**

Здобувач виконав частину лабораторних робіт, але не продемонстрував цілісних знань на модульному контролі. Потребує додаткового часу на вивчення тем та повторну здачу.

**Незадовільно (F) (з обов'язковим повторним курсом). Бали: 0–34**

Здобувач не володіє базовим синтаксисом, не може скласти найпростіший алгоритм, проігнорував більшість практичних завдань семестру.

Оскільки курс охоплює кілька семестрів і різні технології (від Python до Web), ці критерії застосовуються до кожної мови програмування, що вивчається у відповідному блоці.

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### Підручники та навчальні посібники

1. Коноваленко І. В. Програмування мовою Python: Навчальний посібник. – Тернопіль: ТНТУ, 2021. – 232 с.
2. Глибовець М. М. Алгоритми та структури даних: Підручник. – К.: НаУКМА, 2023. – 250 с.
3. Любченко В. В. Алгоритмізація та програмування: Навчальний посібник. – Одеса: ОНПУ, 2022. – 310 с.
4. Морзе Н. В. Методика навчання інформатики: Навчальний посібник (2-е видання). – К.: Академвидав, 2019. – 320 с.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to the algorithms. – The MIT Press, 2011. – 1180p.
6. Кузьменко І. В. "Програмування на Java" Освіта. 2021. 400с. ISBN: 978-966-667-008-5
7. Чалий К. Г. «Java. Професійне програмування» Вітчизняна книга. 2020. 450с. ISBN: 978-617-7400-91-1.
8. Федоренко Г.В. Java. Базовий курс. КНЕУ. 2019. 320с ISBN: 978-617-7482-62-7
9. Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002.– 83 с.
10. Richter J. CLR via C#. – 4th ed. – Redmond : Microsoft Press, 2012. – 896 p. ISBN: 978-0735667457.
11. Федоренко Г. В. Програмування мовою C#. Базовий курс. – Київ : КНЕУ, 2020. – 320 с. ISBN: 978-617-7482-85-6.
12. Кузьменко І. В. Основи програмування на платформі .NET (C#). – Київ : Освіта, 2022. – 384 с. ISBN: 978-966-667-820-3.
13. Чалий К. Г. C#. Професійне програмування. – Харків : Вітчизняна книга, 2021. – 460 с. ISBN: 978-617-7400-98-0.
14. Онлайн-довідник HTML тегів та CSS селекторів <https://css.in.ua/>
15. Онлайн-підручник з HTML <https://w3schoolsua.github.io/html/>
16. Основи html <https://learn.microsoft.com/ru-ru/cpp/mfc/htmlbasics?view=msvc-170>
17. Правила веб-дизайну та налаштування CSS для екранів різних пристроїв <https://www.geeksforgeeks.org/css-rules-in-web-design-for-all-screens/>
18. Покроковий посібник зі створення веб-сайту <https://itproger.com/ua/news/polnoe-poshagovoe-rukovodstvo-po-sozdaniyusaytov-dlya-nachinayushtih>

### ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

1. Дистанційна академія: [d-academy.com.ua](http://d-academy.com.ua) – ресурс для підвищення кваліфікації вчителів інформатики з матеріалами по Python.
2. Prometheus: [prometheus.org.ua](http://prometheus.org.ua) – українська платформа з безкоштовними курсами з програмування та алгоритмізації.
3. IT-школа СМАРТ: [smart.edu.ua](http://smart.edu.ua) – методичні розробки для навчання дітей програмуванню та створенню інтерфейсів.
4. Освіторія: [osvitoria.media](http://osvitoria.media) – портал з інноваційними методиками викладання IT-дисциплін у школі.
5. W3Schools Tkinter: [w3schools.com/python/python\\_tkinter.asp](http://w3schools.com/python/python_tkinter.asp) – покроковий довідник з прикладами створення кожного віджета.
6. Stepik: [stepik.org](http://stepik.org) – інтерактивна платформа з курсами, що містять завдання на автоматичну перевірку коду Python.
7. Tutorialspoint: [tutorialspoint.com/python/python\\_gui\\_programming.htm](http://tutorialspoint.com/python/python_gui_programming.htm) – детальний посібник з графічного програмування для студентів.
8. Khan Academy: [khanacademy.org](http://khanacademy.org) – освітній ресурс з основами програмування та комп'ютерних наук.
9. Офіційна документація Java. [docs.oracle.com/javase/docs](http://docs.oracle.com/javase/docs)

10. Java Tutorials. [oracle.com/java/technologies/javase/tutorials](http://oracle.com/java/technologies/javase/tutorials)
11. Фесюнов В. Конспект лекцій по Java. <http://www.javable.com/tutorials/fesunov/>
12. Підручник з BlueJ (The Bluej Tutorial (Russian)), PDF, translated by Alexey Vasilchenko - <http://bluej.org/tutorial/tutorial-russian.pdf>
13. Форум. Питання та відповіді з програмування, в тому числі Java. [stackoverflow.com](http://stackoverflow.com)
14. Офіційна документація мови C#. <https://learn.microsoft.com/dotnet/csharp>
15. Документація платформи .NET. <https://learn.microsoft.com/dotnet>
16. Навчальні матеріали та туторіали з C# на Microsoft Learn. <https://learn.microsoft.com/training/paths/csharp-first-steps>
17. Документація з розробки WPF-застосунків. <https://learn.microsoft.com/dotnet/desktop/wpf>
18. Матеріали з Data Binding та MVVM у WPF. <https://learn.microsoft.com/dotnet/desktop/wpf/data>
19. Онлайн-довідник HTML тегів та CSS селекторів <https://css.in.ua/>
20. Онлайн-підручник з HTML <https://w3schoolsua.github.io/html/>
21. Основи HTML <https://learn.microsoft.com/ru-ru/cpp/mfc/htmlbasics?view=msvc-170>
22. Правила веб-дизайну та налаштування CSS для екранів різних пристроїв <https://www.geeksforgeeks.org/css-rules-in-web-design-for-all-screens/>
23. Покроковий посібник зі створення веб-сайту <https://itproger.com/ua/news/polnoe-poshagovoe-rukovodstvo-po-sozdaniyusaytov-dlya-nachinayushtih>

### **Програмне забезпечення та документація**

1. Python Tkinter Docs: [docs.python.org/3/library/tkinter.html](https://docs.python.org/3/library/tkinter.html) – офіційна документація стандартної бібліотеки для створення GUI.
2. Visual Studio Code: [code.visualstudio.com](https://code.visualstudio.com) – безкоштовний редактор коду з плагінами для розробки та налагодження додатків.
3. PySimpleGUI: [pysimplegui.com](https://pysimplegui.com) – документація надбудови над Tkinter, яка спрощує створення інтерфейсів для початківців.
4. IDLE Python: [docs.python.org/3/library/idle.html](https://docs.python.org/3/library/idle.html) – базове середовище розробки, що входить до складу Python і рекомендоване для шкіл.