

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ

БАКАЛАВР

Кафедра інформаційних технологій

«ЗАТВЕРДЖУЮ»

Голова НМР факультету автоматизації і  
інформаційних технологій

Олександр ТЕРЕНТЬЄВ/

« \_\_\_\_\_ » \_\_\_\_\_ 2024 року

РОБОЧА ПРОГРАМА ОСВІТНЬОЇ КОМПОНЕНТИ

ОК 12 «Інструментальні засоби програмування»

(шифр та назва освітньої компоненти)

шифр	назва спеціальності, назва освітньої програми
122	<b>Комп'ютерні науки, Інформаційні управляючі системи і технології</b>

Мова викладання: українська

Розробники:

Юлія РЯБЧУН, PhD, доцент кафедри ІТ



(підпис)

Ірина Пороховниченко, асистент кафедри ІТ



(підпис)

(ім'я та прізвище, науковий ступінь, звання)

Робоча програма затверджена на засіданні кафедри інформаційних технологій,  
протокол № 13 від «25» червня 2024 року

Завідувач кафедри ІТ



(підпис)

Тетяна ГОНЧАРЕНКО

(ім'я та прізвище)

Схвалено гарантом

освітньої програми «Інформаційні управляючі системи і технології»

Гарант ОП



(підпис)

Олександр ПОПЛАВСЬКИЙ


(ім'я та прізвище)

Розглянуто на засіданні науково-методичної комісії спеціальності

122 «Комп'ютерні науки»,

протокол № 3 від «28» червня 2024 року

ВИТЯГ З РОБОЧОГО НАВЧАЛЬНОГО ПЛАНУ 2024-2025 рр.

шифр	Назва спеціальності, освітньої програми	Кількість кредитів	Форма здобуття ВО: денна						Кількість індивідуальних робіт	Форма контролю	Семестр	Погодженя заступником декана факультету
			Обсяг годин			Са м. роб	Конт. роб					
			Вс	аудиторних	у тому числі		КП	КР				
122	Комп'ютерні науки, Інформаційні управляючі системи і технології	3	90	40	20	20	50	1	залік	2		

## **Мета та завдання освітньої компоненти**

**Пререквізити:** Вступ до фаху, Програмування та алгоритмічні мови

**Посилання на сторінку електронного навчально-методичного комплексу дисципліни:** <https://org2.knuba.edu.ua/course/section.php?id=2896>

**Мета дисципліни** є забезпечити здобувачів теоретичними знаннями та практичними навичками використання сучасних інструментів для розробки програмного забезпечення.

### **Завдання курсу:**

*Ознайомлення з інструментами програмування:*

- середовища розробки (IDE) (Visual Studio);
- системи контролю версій (Git, GitHub, GitLab);
- інструменти автоматизації збірки (Maven, Gradle).

*Формування навичок практичного застосування:*

- налаштування та використання IDE;
- робота з системами контролю версій.

*Розвиток алгоритмічного та логічного мислення:*

- застосування мов програмування (C++);
- ефективна розробка та налагодження коду.

*Забезпечення компетенцій командної роботи:*

- принципи Agile та Scrum;
- інструменти для спільної розробки (JIRA, Trello).

## Компетенції здобувачів, що формуються в результаті засвоєння дисципліни

Код	Зміст компетентності
<b>Інтегральна компетентність</b>	
<b>ІК</b>	Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.
<b>Загальні компетентності</b>	
<b>ЗК2</b>	Здатність застосовувати знання у практичних ситуаціях.
<b>ЗК3</b>	Знання та розуміння предметної області та розуміння професійної діяльності.
<b>ЗК6</b>	Здатність вчитися й оволодівати сучасними знаннями.
<b>ЗК7</b>	Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
<b>ЗК9</b>	Здатність працювати в команді
<b>ЗК12</b>	Здатність оцінювати та забезпечувати якість виконуваних робіт
<b>Фахові компетентності</b>	
<b>СК10</b>	Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.
<b>СК15</b>	Здатність до аналізу та функціонального моделювання бізнес-процесів, побудови та практичного застосування функціональних моделей організаційно-економічних і виробничо-технічних систем, методів оцінювання ризиків їх проектування.

**Програмні результати здобувачів освітньої програми, що формуються в  
результаті засвоєння освітньої компоненти**

<b>Код</b>	<b>Програмні результати</b>
<b>ПР 9</b>	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.
<b>ПР 10</b>	Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.
<b>ПР 11</b>	Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).
<b>ПР 15</b>	Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

## Структура навчальної дисципліни

Назва теми	Кількість годин		
	Лекції	Практичні заняття	Самостійна робота
<b>Змістовий модуль 1.</b>			
<b>Базові інструменти програмування</b>			
<b>Лекція 1.</b> Вступ до курсу «Інструментальні засоби програмування». Поняття інформаційної систем, основні властивості та їх класифікація. Призначення і функції інструментальних засобів інформаційних систем, архітектури сучасних інструментальних засобів, проблеми сполучення. Базові та прикладні програмні засоби інформаційних систем: операційні системи, мови програмування, програмні середовища, системи управління базами даних. Парадигми програмування (об'єктно-орієнтоване, функціональне, процедурне).	2	2	5
<b>Лекція 2.</b> Системи контролю версій. Введення в Git, SVN, Mercurial. Інструменти для спільної роботи з кодом: GitHub, GitLab.	2	2	5
<b>Лекція 3.</b> Основи Git. Основні команди та концепції Git. Створення та управління локальними репозиторіями. Робота з гілками: створення, злиття, вирішення конфліктів. Відновлення змін та управління історією. Робота з віддаленими репозиторіями (GitHub).	2	2	5
<b>Лекція 4.</b> Інтегровані середовища розробки (IDE). Основні можливості та функції IDE.	2	2	5
<b>Змістовий модуль 2.</b>			
<b>Інструменти для забезпечення якості та автоматизації</b>			
<b>Лекція 5.</b> Автоматизація процесів розробки (CI/CD). Огляд Jenkins, GitHub Actions, GitLab CI/CD. Автоматичне тестування, збірка та розгортання. Хмарні платформи (AWS, Azure, Google Cloud).	2	2	5
<b>Лекція 6.</b> Тестування та забезпечення якості. Написання автоматизованих тестів (JUnit, PHPUnit, Selenium). Інструменти для статичного аналізу коду.	2	2	5

<b>Лекція 7.</b> Методології розробки. Agile, Scrum, Kanban як основи командної роботи. Управління проєктами та задачами. Використання Jira, Trello, Asana для управління робочим процесом.	2	2	5
<b>Лекція 8.</b> Інструменти для документації. Використання Markdown, Sphinx, Swagger для створення документації. Інтеграція документації в процеси розробки.	2	2	5
<b>Лекція 9.</b> Бібліотеки та фреймворки. Прискорення розробки за допомогою фреймворків. Використання бібліотек для роботи з даними, мережею та UI.	2	2	5
<b>Лекція 10.</b> Розвиток інструментальних засобів оброблення інформації. Методи розвитку та дослідження інформаційних систем. Тенденції розвитку інформаційних технологій	2	2	5
<b>РАЗОМ</b>	<b>20</b>	<b>20</b>	<b>50</b>

### Теми практичних завдань

№	Назва теми	К-сть годин	К-сть балів
1	Предмет і зміст дисципліни	2	5
2	Створення та налаштування репозиторію на GitHub.	2	5
3	Робота з гілками в Git, робота зі знімками змін, перегляд історії, розв'язання конфліктів.	2	5
4	Використання GitHub для командної роботи	2	5
5	Розгортання простого проєкту за допомогою GitHub Actions та GitHub Pages	2	5
6	Інтеграція статичного аналізатору коду як крок частина CI/CD.	2	5
7	Як створювати, призначати, та закривати задачі в GitHub.	2	5
8	Написання документації до проєкту засобами Markdown і створення README файлу у GitHub.	2	5

9	Тестування та супровід. Ч.1	2	5
10	Тестування та супровід. Ч.2	2	5
	<b>Разом</b>	<b>20</b>	<b>60</b>

### Розподіл годин самостійної роботи здобувачів

№	Вид роботи	Кількість годин	Форми звітності
1	Опрацювання лекційного матеріалу	5	конспекти лекцій
2	Підготовка до практичних занять	10	усні відповіді та виконання практичних завдань
3	Опрацювання тем, винесених на самостійну підготовку	5	робота на практичних заняттях та виконання творчих завдань
4	Робота з інтернет-ресурсами	12	індивідуальні завдання, підготовка до РГР та практичних занять
5	Виконання індивідуальної роботи Завдання повинно включати: 1. Формування робіт для розміщення на GitHub 2. Створення портфоліо на GitHub 3. Написання пояснювальної записки 4. Публічний захист	12	Демонстрація розробки: активного репозиторія на GitHub та захист роботи
6	Підготовка до заліку	6	
<b>Разом за семестр</b>		<b>50</b>	

## **Індивідуальне завдання**

### **Розрахункова-графічна робота на тему**

### **«Розробка власного портфоліо робіт на GitHub»**

Важливим елементом успішного засвоєння дисципліни та здобуття навичок, які передбачені освітньою програмою, є практичні роботи з індивідуальним завданням.

Головна мета – на прикладі індивідуального завдання засвоїти та навчитися використовувати можливості GitHub для створення активного репозитарію виконаних робіт і проєктів, який можна використовувати як власне портфоліо, що сприяє формуванню навичкам самопрезентації для майбутнього роботодавця.

### **Політика щодо академічної доброчесності**

Розрахунково-графічна робота (РГР) є важливою складовою навчального процесу, що дозволяє здобувачам закріпити та поглибити теоретичні знання, отримані під час лекційних занять, шляхом практичного застосування їх для вирішення конкретних задач. РГР сприяє розвитку аналітичного мислення, навичок програмування, самостійного пошуку та обробки інформації, а також навичок ефективного представлення результатів своєї діяльності.

Основними цілями виконання РГР є:

- Засвоєння теоретичних знань.
- Розвиток практичних навичок.
- Формування навичок самостійної роботи.
- Оцінка знань та вмінь.

Успішне виконання РГР вимагає від здобувачів вміння аналізувати задачу, розробляти ефективні алгоритми, писати якісний код, проводити тестування та оптимізацію, а також представляти результати своєї роботи у вигляді звіту та презентації. Всі ці етапи є невід'ємною частиною процесу навчання і розвитку компетенцій, необхідних для майбутньої професійної діяльності.

Здобувачі можуть обрати будь-яку з запропонованих тем для виконання своєї роботи. Якщо у здобувача є інша ідея для проєкту, яка здається більш цікавішою, будь ласка, обговоріть її з викладачем заздалегідь, щоб отримати дозвіл на її реалізацію

### **Політика щодо відвідування**

Здобувач, який пропустив аудиторне заняття з поважних причин, має продемонструвати викладачу та надати до деканату факультету документ, який засвідчує ці причини.

За об'єктивних причин (хвороба, міжнародне стажування, наукова та науково-практична конференція (круглий стіл) тощо) навчання може відбуватись в он-лайн формі за погодженням з керівником курсу.

### **Методи контролю**

Основні форми участі Здобувачів у навчальному процесі, що підлягають поточному контролю: виступ на практичних заняттях; доповнення, опонування до

виступу; участь у дискусіях; аналіз першоджерел; письмові завдання (тестові, індивідуальні роботи у формі рефератів). Кожна тема курсу, що винесена на лекційні та практичні заняття, відпрацьовується Здобувачами у тій чи іншій формі, наведеній вище. Обов'язкова присутність на лекційних заняттях, активність впродовж семестру, відвідування/відпрацювання усіх аудиторних занять, виконання інших видів робіт, передбачених навчальним планом з цієї дисципліни.

При оцінюванні рівня знань Здобувача аналізу підлягають:

- характеристики відповіді: цілісність, повнота, логічність, обґрунтованість, правильність;

- якість знань (ступінь засвоєння фактичного матеріалу): осмисленість, глибина, гнучкість, дієвість, системність, узагальненість, міцність;

- ступінь сформованості умінь поєднувати теорію і практику під час розгляду ситуацій, практичних завдань;

- рівень володіння розумовими операціями: вміння аналізувати, синтезувати, порівнювати, абстрагувати, узагальнювати, робити висновки з проблем, що розглядаються;

- досвід творчої діяльності: вміння виявляти проблеми, розв'язувати їх, формувати гіпотези;

- самостійна робота: робота з навчально-методичною, науковою, допоміжною вітчизняною та зарубіжною літературою з питань, що розглядаються, вміння отримувати інформацію з різноманітних джерел (традиційних; спеціальних періодичних видань, ЗМІ, Internet тощо).

**Тестове опитування** може проводитись за одним або кількома змістовими модулями. В останньому випадку бали, які нараховуються Здобувачу за відповіді на тестові питання, поділяються між змістовими модулями.

**Підсумковий контроль** здійснюється під час проведення залікової сесії з урахуванням підсумків поточного та модульного контролю. Під час семестрового контролю враховуються результати здачі усіх видів навчальної роботи згідно зі структурою кредитів.

Оцінювання проводиться за 100-бальною шкалою.

### Розподіл балів для дисципліни з формою контролю залік

Поточне оцінювання			Залік	Сума балів
Змістовий модуль 1	Змістовий модуль 2	Індивідуальне завдання		
30	30	20	20	100

### Шкала оцінювання індивідуальної роботи

Оцінка за національною шкалою	Кількість балів	Критерії
<b>відмінно</b>	20	відмінне виконання (розкриття теми, посилання та цитування сучасних наукових джерел (не старше 2019 року), <b>дотримання норм доброчесності</b> )
	18	відмінне виконання з незначною кількістю помилок виконання (розкриття теми, посилання та цитування сучасних наукових джерел (більшість з яких не старше 2019 року), <b>дотримання норм доброчесності</b> )
<b>добре</b>	17	виконання вище середнього рівня з кількома помилками (розкриття теми в межах об'єкта та завдань роботи, посилання та цитування сучасних наукових джерел (серед яких є такі, що не старше 2019 року), <b>дотримання норм доброчесності</b> )
	13	виконання з певною кількістю помилок (розкриття теми в межах об'єкта та завдань роботи, наявність посилань та цитувань наукових джерел, <b>дотримання норм доброчесності</b> )
<b>задовільно</b>	12	виконання роботи задовольняє мінімальним критеріям помилок (розкриття теми в основному в межах об'єкта роботи, наявність концептуального апарату роботи, присутність не менше 5 посилань та цитувань наукових джерел, <b>дотримання норм доброчесності</b> )

## Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90-100	<b>A</b>	відмінно
82-89	<b>B</b>	добре
74-81	<b>C</b>	
64-73	<b>D</b>	задовільно
60-63	<b>E</b>	
35-59	<b>FX</b>	Не зараховано з можливістю повторного складання
0-34	<b>F</b>	Не зараховано з обов'язковим повторним вивченням дисципліни

### Умови допуску до підсумкового контролю

Здобувачу, який має підсумкову оцінку за дисципліну від 35 до 59 балів, призначається додаткова залікова сесія. В цьому разі він повинен виконати додаткові завдання, визначені викладачем.

Здобувач, який не виконав вимог робочої програми по змістових модулях, не допускається до складання підсумкового контролю. В цьому разі він повинен виконати визначене викладачем додаткове завдання по змісту відповідних змістових модулів в період між основною та додатковою сесіями.

Здобувач має право на опротестування результатів контролю (апеляцію). Правила подання та розгляду апеляції визначені внутрішніми документами КНУБА, які розміщені на сайті КНУБА та зміст яких доводиться Здобувачам до початку вивчення дисципліни.

## Методичне забезпечення дисципліни

### Програмне забезпечення

1. <https://visualstudio.microsoft.com/ru/downloads/>
2. <https://github.com/>

### Навчальні посібники

1. Карпенко М.Ю., Манакова Н.О., Гавриленко І.О. Технології створення програмних продуктів та інформаційних систем : навч. Посібник. Харків, ХНУМГ ім. О. М. Бекетова, 2017. 93 с.
2. Scott Meyers. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. (2014)
3. Цибульник, С. О. Технології розробки програмного забезпечення-1. Комп'ютерний практикум [Електронний ресурс]: навчальний посібник / С. О. Цибульник ; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім. Ігоря Сікорського, 2021. – 125 с.
4. Горгош Л.І. Текстові редактори та текстові процесори: навч.-метод. посіб. / Л.І. Горгош, Ю.Ю. Білак. – Ужгород: ПП «АУТДОР - ШАРК», 2016. – 128 с.//[Електронний ресурс]. Режим доступу: <https://www.uzhnu.edu.ua/en/infocentre/get/44991>.

### Додаткові джерела

1. Вступ до систем контролю версій: Git та SVN. <https://www.artofba.com/uk/post/version-control-system-git-vs-svn>
2. Serhii Dolhpolov, Tetyana Honcharenko, Pavlo Kruk, Iryna Porokhovnichenko and Vladyslav Hots. YOLOv8, YOLOv9, and YOLOv10: A Study in Automated Vehicle Damage Detection. INFORMATION TECHNOLOGIES: THEORETICAL AND APPLIED PROBLEMS (ITTAР-2024) The 4th International Workshop (Scopus)
3. Погребний М.В., Рябчун Ю.В. ІНТЕГРАЦІЯ ШТУЧНОГО ІНТЕЛЕКТУ В ОСВІТНІЙ ПРОЦЕС УКРАЇНИ: ВИКЛИКИ ТА ПЕРСПЕКТИВИ НА ШЛЯХУ ДО ЄВРОПЕЙСЬКИХ СТАНДАРТІВ. VII Міжнародна конференція «Актуальні проблеми освітнього процесу в контексті європейського вибору України», 14 листопада 2024 року. <https://www.knuba.edu.ua/wp-content/uploads/2024/10/inf-lyst-mizhn-konf-14.11.knuba-2024.pdf>
4. E. Gorda, Yu. Riabchun, V. Khrolenko. Cognitive Technologies for Object Detection and Topology of the Environment's Information Space. 2024 IEEE 4th International Conference on Smart Information Systems and Technologies, Kazakhstan, Astana 2024, pp. 48-54 (Scopus). DOI: 10.1109/SIST61555.2024.10629620

### Інформаційні ресурси:

1. <http://library.knuba.edu.ua>  
<https://org2.knuba.edu.ua/course/view.php?id=3625>
2. Омельчук Л.Л. Інструментальні середовища та технології програмування. Лабораторний практикум / Л.Л. Омельчук, Н.Г. Русіна. - Одеса: Айс Прінт, 2020. - 175 с.

3. GitHub dox

<https://docs.github.com/en/repositories/working-with-files/using-files/working-with-non-code-files>