

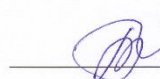
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кафедра комп'ютерних наук


«ЗАТВЕРДЖУЮ»
Декан факультету інформаційних
технологій
Олена ГЛАЗУНОВА
« 12 » 20 23 р.



«СХВАЛЕНО»
на засіданні кафедри комп'ютерних наук
Протокол № 12 від «01» 06 2023
р.
Завідувач кафедри
Белла ГОЛУБ



«РОЗГЛЯНУТО»
Гарант ОП «Комп'ютерні науки»
Гарант ОП
Олена ГЛАЗУНОВА



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ПРОГРАМУВАННЯ

спеціальність 122 «Комп'ютерні науки»
освітня програма «Комп'ютерні науки»
Факультет інформаційних технологій
Розробник: к.е.н., Густера О.М.

Київ – 2023

**Опис навчальної дисципліни
«ПРОГРАМУВАННЯ»**

Галузь знань, напрям підготовки, спеціальність, освітньо-кваліфікаційний рівень	
Галузь знань	12 “Інформатика та обчислювальна техніка”
Спеціальність	122 “Комп’ютерні науки”
Освітня програма	“Комп’ютерні науки”
Освітній ступінь	бакалавр
Характеристика навчальної дисципліни	
Вид	Обов’язкова
Загальна кількість годин	300
Кількість кредитів ECTS	10
Кількість змістових модулів	4
Курсовий проект (робота) <small>(якщо є в робочому навчальному плані)</small>	
Форма контролю	залік (I семестр), екзамен (II семестр)
Показники навчальної дисципліни для денної та заочної форм навчання	
	денна форма навчання
Рік підготовки	1
Семестр	1,2
Лекційні заняття	60 год.
Практичні, семінарські заняття	
Лабораторні заняття	60 год.
Самостійна робота	180 год.
Кількість тижневих годин для денної форми навчання: аудиторних самостійної роботи студента —	4 год. 6 год.

1. Мета, завдання та компетентності навчальної дисципліни

Мета: надання теоретичних та практичних знань з використання комп'ютерної техніки, сучасного програмного забезпечення, яке буде основою для отримання студентами знань з області розробки алгоритмів та програмування. Оволодіння такими знаннями дозволить реалізовувати задачі автоматизації обробки інформації, автоматизації керування об'єктами, обробки інформації при вивченні професійно-орієнтованих дисциплін та розробки практичних проектів.

Завдання: полягає в отриманні навичок студентом користування комп'ютерною технікою і сучасними прикладними програмами, що сприятиме професійній адаптації в інформаційному просторі для створення програмних засобів вирішення практичних задач.

У результаті вивчення навчальної дисципліни студент повинен

знати: основні поняття та термінологію курсу, визначення алгоритмів та технологій їх розробки, правила та алгоритми опрацювання даних, технології розробки програм мовою С, базові конструкції мови С.

вміти: розроблювати алгоритми та реалізовувати їх у вигляді програми на мові С, налагоджувати програми; самостійно опановувати нові методи та технології розробки програм.

Навчальна дисципліна забезпечує формування ряду загальних та спеціальних (фахових, предметних) компетентностей:

Загальні компетентності:

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Спеціальні (фахові, предметні) компетентності:

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК4. Здатність використовувати сучасні методи математичного моделювання об'єктів, процесів і явищ, розробляти моделі й алгоритми чисельного розв'язування задач математичного моделювання, враховувати похибки наближеного чисельного розв'язування професійних задач.

СК8. 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.

У результаті вивчення навчальної дисципліни студент повинен показати певні **програмні результати**, а саме:

ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

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.

ПР15. 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.

2. Програма та структура навчальної дисципліни

для повного терміну денної форми навчання;

Змістовий модуль 1. Організація програм

Тема лекційного заняття 1. Введення в програмування (2 год)

Алгоритми та програми. Мови програмування. Блок-схема як засіб графічного зображення алгоритмів. Лінійні алгоритми. Алгоритми з розгалуженням. Цикли та циклічні структури.

Тема лекційного заняття 2. Основні поняття мови C (6 год)

Історія мови C. Характеристика C-систем. Загальна структура програми на мові C. Директиви передпроцесора. Стандартні функції введення/виведення мови C. Використання коментарів. Об'ява змінних. Загальні поняття щодо типів даних мови C. Числові типи та символічний тип.

Загальні поняття щодо пам'яті комп'ютера. Руйнуюче та не руйнуюче зчитування. Арифметика в мові C. Пріоритет виконання операцій. Прийняття рішення: операції рівності та відношення. Пріоритет виконання розглянутих операцій. Ключові слова. Логічні операції.

Тема лекційного заняття 3. Оператори мови C. Оператори вибору (2 год)

Операції присвоєння. Повна та скорочена форма операції присвоєння. Операції інкремента та декремента. Структури управління. Структура вибору *if*. Структура вибору *if/else*.

Структура із множинним вибором *switch*. Приклади програм з реалізованими структурами управління.

Тема лекційного заняття 4. Оператори мови C. Структури повторення(6 год)

Структура повторення *for*. Блок-схема алгоритму структури *for*. Синтаксис структури *for*. Приклади структур *for*.

Структура повторення *while*. Структура повторення *do/while*. Рекомендації щодо вибору структур повторення в залежності від алгоритму. Приклади програм з реалізованими структурами повторення.

Оператори *break i continue*. Логічні операції. Пріоритет виконання розглянутих операцій.

Змістовий модуль 2.Функції та обробка даних

Тема лекційного заняття 5. Використання функцій на мові C (4 год)

Загальні поняття щодо структурного програмування. Програмні модулі мови C. Функції математичної бібліотеки. Функції бібліотеки стандартного введення/виведення. Використання функцій при розробці програм на мові C. Визначення функцій. Прототипи функцій та файли заголовків.

Розробка власних функцій. Виклик функції за значенням. Способи обміну інформацією між функціями. Локальні та глобальні змінні.

Тема лекційного заняття 6. Структуровані типи даних. Одновимірні масиви (6 год)

Масиви як структурований тип даних. Визначення масивів. Синтаксис об'яви масивів. Поняття “елемент масиву” та “індекс масиву”. Синтаксис використання масивів.

Стандартні алгоритми роботи з масивами: визначення екстремальних значень, статистичних показників, упорядкування масивів, пошук визначеного елемента.

Тема лекційного заняття 7. Структуровані типи даних. Багатовимірні масиви (6 год)

Багатовимірні масиви. Приклади програм з багатовимірними масивами. Реалізація алгоритмів, пов'язаних з обробкою матриць.

Змістовний модуль 3. Структури даних та управління пам'яттю при програмуванні

Тема лекційного заняття 8. Функції і управління пам'яттю(4 год)

Виклик функції за посиланням – передача параметрів за адресою. Приклад програми, що використовує виклик за посиланням.

Класи пам'яті. Правила області дій. Розробка програми у вигляді проекту. Зовнішні змінні. Рекурсія. Приклад використання рекурсії: числа Фібоначчі.

Тема лекційного заняття 9. Проектування програм з використанням покажчиків (8 годин)

Ідеологія використання покажчиків в програмі на мові C. Об'ява та ініціалізація покажчиків. Приклади використання покажчиків. Операції з покажчиками. Виведення значень покажчиків як адрес оперативної пам'яті. Виведення значення, на яке посилається покажчик. Вирази та арифметика з покажчиками.

Доступ до апаратних елементів комп'ютера за допомогою покажчиків. Приклади програм з використанням покажчиків.

Тема лекційного заняття 10. Робота з символами та рядками(4 год)

Визначення рядків. Бібліотека роботи з рядками. Доступ к рядку за допомогою покажчика. Введення-виведення рядків. Операції над рядками. Порівняння рядків. Перетворення рядків. Пошук символів у рядку.

Змістовний модуль 4. Структури даних і робота з файлами

Тема лекційного заняття 11. Використання структур для створення власних типів даних. Використання об'єднань (4 год)

Створення власних типів даних за допомогою структур. Об'ява структур. Об'ява структурних змінних. Доступ до елементів структури.

Використання масивів структур. Використання структур з функціями. Реалізація зв'язаних списків, стеків, черг та дерев за допомогою структур.

Створення власних типів даних за допомогою об'єднань. Об'ява об'єднань та змінних-об'єднань.

Тема лекційного заняття 12. Використання операцій з бітами(2 год)

Бітові операції. Бітові поля. Вивчення властивостей операції над бітами. Константи, що перераховуються.

Тема лекційного заняття 13. Файлова система (8 годин)

Загальні уявлення про файлову систему. Ієрархія даних. Файли і потоки. Стандартні методи для роботи з файлами на мові C.

Створення файлів послідовного доступу. Читання із файлів послідовного доступу..

Файли довільного доступу. Створення файлів довільного доступу. Довільний запис у файл довільного доступу. Послідовне читання даних із файлів довільного доступу. Приклади програм.

Назви змістових модулів і тем	Кількість годин										
	денна форма										
	всього	у тому числі									
		л	п	лаб	інд	с.р.					
1	2	3	4	5	6	7					
Змістовний модуль 1. Організація програм											
Введення в програмування	8	2				6					
Основні поняття мови C	20	6		2		12					
Оператори мови C. Оператори вибору	8	2		6							
Оператори мови C. Структури повторення	26	6		8		12					
Разом за змістовим модулем 1	62	16		16		30					
Змістовний модуль 2. Функції та обробка даних											
Використання функцій на мові C	28	4		6		18					
Структуровані типи даних. Одновимірні масиви	39	5		4		30					
Структуровані типи	21	5		4		12					

даних. Багатовимірні масиви												
Разом за змістовим модулем 2	88	14		14		60						
Всього 1 семестр	150	30		30		15						
Змістовний модуль 3. Структури даних та управління пам'яттю при програмуванні												
Функції і управління пам'яттю	18	4		2		12						
Проектування програм з використанням показчиків	34	8		8		18						
Робота з рядками та символами	26	4		4		18						
Разом за змістовим модулем 3	78	16		14		48						
Змістовний модуль 4. Структури даних і робота з файлами												
Використання структур для створення власних типів даних. Використання об'єднань	20	4		4		12						
Використання операцій з бітами	16	2		2		12						
Файлова система	36	8		10		18						
Разом за змістовим модулем 4	72	14		16		42						
Всього 2 семестр	150	30		30		90						
Всього	300	60		60		180						

3. Теми семінарських занять

Не передбачено програмою

4. Теми практичних занять

Не передбачено програмою

5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	C-системи. Змінні та типи змінних, об'ява змінних. Виведення та введення даних	2
2	Проста та скорочена форми операції присвоєння. Структура	6

	вибору if, if/else.	
3	Структура прийняття рішення switch.	2
4	Структури повторення	6
5	Створення власних функцій	4
6	Масиви. Визначення масивів.	10
7	Класи пам'ті. Правила області дії.	4
8	Показчики. Зв'язок між показчиками та масивами	8
9	Рядки й символи.	4
10	Структури та об'єднання	4
11	Операції з бітами.	2
12	Робота з файлами. Створення та читання файла послідовного доступу.	4
13	Робота з файлами. Створення та читання файла довільного доступу.	4
Всього		60

6. Теми самостійної роботи

№ з/п	Назва теми	Кількість годин
1	Порівняльна характеристика мов програмування	6
2	Методології розробки програм: низхідне та висхідне проектування	12
3	Виведення даних із визначеною шириною поля і точністю представлення. Використання прапорців в рядку управління форматів функції <i>printf</i> . Друк літералів та Esc-послідовностей	12
4	Стандартні бібліотеки компілятора мови	18
5	Алгоритми сортування великих масивів	30
6	Завершення роботи програми за допомогою макросів <i>exit</i> , <i>atexit</i>	12
7	Основні директиви передпроцесора мови C. Використання директив <i>#include</i> , <i>#define</i> , <i>#ifdef</i>	12
8	Додаткові директиви передпроцесора <i>#error</i> , <i>#pragma</i> . Операції <i>#</i> , <i>##</i>	18
9	Структури, які посилаються самі на себе. Зв'язані списки, черги, дерева	30
10	Рекурсія – переваги і недоліки	12
11	Низько рівневий доступ до файлів	18
Всього		180

7. Зразки контрольних питань, тестів для визначення рівня засвоєння знань студентами.

1. Що виконує така директива передпроцесора: `#include <math.h>` ?
2. Який оператор має бути використаний, щоб вивести на екран фразу "Hello, world" та перевести курсор на початок наступного рядка?
3. Який оператор має бути використаний, щоб створити змінну з іменем `cheeses` цілого типу?
4. Який оператор має бути використаний, щоб реалізувати введення з клавіатури цілого значення та присвоєння цього значення змінній `cheeses`?
5. Написати чотири оператори мови C, кожний із яких додає 1 до змінної цілого типу.
6. Обчислити остачу при діленні змінної `q` на змінну `divisor` та присвоїти результат змінній `q`. Використати два способи.
7. Нехай в програмі є оператори `"x=5; printf("%d", ++x);"`. Що буде виведено на екран?
8. Написати оператор, який перевіряє змінну `u`, щоб з'ясувати, чи є ця змінна додатною.
9. Що буде виведено на екран після виконання такого фрагмента програми:
`l1=3; l2=4; l3=5;`
`p=(l1+l2+l3)/2;`
`p=sqrt(p*(p-l1)*(p-l2)*(p-l3));`
`if (p>0) printf ("\nПлоща заданого трикутника = %.2f ед.",p);`
`else printf ("\nНевірно задані координати, такого трикутника не існує.");`
10. Нехай у програмі об'явлений масив `t: 'int A[125];'`. Вивести значення всіх елементів масиву.
11. Запишіть оператор, який реалізує присвоєння значення 0 п'ятьом елементам масиву цілих чисел.
12. Запишіть оператор, який реалізує введення з клавіатури 12 значень масиву, що складається з елементів типу `float`.
13. Зарезервуйте пам'ять для двадцятьох елементів масиву дійсного типу.
14. Що таке цикл?
15. Для чого використовується керуюча змінна?
16. В циклі `for`, тіло яке складається більш ніж з одного оператора, крапка з комою ставиться після:
 - а) оператора циклу `for`;
 - б) закриваючої фігурної дужки, яка обмежує тіло циклу;
 - в) кожного оператора в тілі циклу;
 - г) умови продовження циклу.

17. Скільки разів буде виконаний цикл в наступних варіантах:
 - а) for (i = 1; i <= 100; i++);
 - б) for (i = 100; i >1; i- -);
 - в) for (i = 0, j = 3; i < 10; i++, j--).
18. Дати поняття потоків і файлів. Який між ними зв'язок?
19. Які види потоків бувають? Пояснити різницю між ними.
20. Охарактеризувати стандартні потоки введення-виведення.
21. Яким чином можна оголосити і відкрити файл?
22. Що таке режими відкриття файлів і які режими ви знаєте?
23. Наведіть приклади форматowanego запису у файл і читання з нього.
24. Яким чином можна встановлювати покажчик файлу у задану позицію?
25. Яким чином можна здійснювати запис і зчитування блоками?
26. Чи можна одночасно використовувати файл для запису і читання?
27. Що таке файлові потоки? Які поняття програмування забезпечують роботу з ними? Назвіть їх.
28. Основні функції роботи з файловими потоками.
29. Які режими доступу до файлових потоків ви знаєте?
30. Яким чином можна виводити у файлові потоки і вводити великі обсяги даних (структури, масиви)?
31. Як можна перевірити, чи закінчився файл, і наявність помилок?
32. Наведіть приклади запису і читання по символах, по словах, по рядках та блоками довільного розміру.

8. Методи навчання

При викладанні навчальної дисципліни використовуються такі методи навчання:

- М1. Лекція (інтерактивна, проблемна)
- М2. Лабораторна робота
- М4. Проєктне навчання (індивідуальне, малі групи, групове)
- М5. Онлайн навчання

9. Форми контролю

При викладанні навчальної дисципліни використовуються такі методи контролю:

- МК1. Тестування
- МК2. Контрольне завдання
- МК4. Методи усного контроль
- МК5. Екзамен
- МК6. Залік

10. Розподіл балів, які отримують студенти

Поточний контроль				Рейтинг з навчальної роботи $R_{нр}$	Рейтинг з додаткової роботи $R_{др}$	Рейтинг штрафний $R_{штр}$	Підсумкова атестація (екзамен чи залік)	Загальна кількість балів
Змістовий модуль 1	Змістовий модуль 2	Змістовий модуль 3	Змістовий модуль 4					
0-100	0-100	0-100	0-100	0-70	0-20	0-5	0-30	0-100

Примітки. 1. Відповідно до «Положення про кредитно-модульну систему навчання в НУБіП України», затвердженого ректором університету, № 8 від « 26 » квітня 2023 р., рейтинг студента з навчальної роботи $R_{нр}$ стосовно вивчення певної дисципліни визначається за формулою

$$R_{нр} = 0,7 \cdot (R^{(1)зм} \cdot K^{(1)зм} + \dots + R^{(n)зм} \cdot K^{(n)зм}) + R_{др} - R_{штр},$$

$$K_{дис}$$

де $R^{(1)зм}, \dots, R^{(n)зм}$ – рейтингові оцінки змістових модулів за 100-бальною шкалою;

n – кількість змістових модулів;

$K^{(1)зм}, \dots, K^{(n)зм}$ – кількість кредитів ECTS, передбачених робочим навчальним планом для відповідного змістового модуля;

$K_{дис} = K^{(1)зм} + \dots + K^{(n)зм}$ – кількість кредитів ECTS, передбачених робочим навчальним планом для дисципліни у поточному семестрі;

$R_{др}$ – рейтинг з додаткової роботи;

$R_{штр}$ – рейтинг штрафний.

Наведену формулу можна спростити, якщо прийняти $K^{(1)зм} = \dots = K^{(n)зм}$. Тоді вона буде мати вигляд

$$R_{нр} = 0,7 \cdot (R^{(1)зм} + \dots + R^{(n)зм}) + R_{др} - R_{штр}.$$

$$n$$

Рейтинг з додаткової роботи $R_{др}$ додається до $R_{нр}$ і не може перевищувати 20 балів.

Він визначається лектором і надається студентам рішенням кафедри за виконання робіт, які не передбачені навчальним планом, але сприяють підвищенню рівня знань студентів з дисципліни.

Рейтинг штрафний $R_{штр}$ не перевищує 5 балів і віднімається від $R_{нр}$. Він визначається лектором і вводиться рішенням кафедри для студентів, які матеріал змістового модуля засвоїли невчасно, не дотримувалися графіка роботи, пропускали заняття тощо.

2. Оцінювання знань студента відбувається за 100-бальною шкалою і переводиться в національні оцінки згідно з табл. 1 «Положення про екзамени та заліки у НУБіП України» (наказ про введення в дію від 27.12.2019 р. № 1371).

Рейтинг здобувача вищої освіти, бали	Оцінка національна за результати складання екзаменів заліків	
	екзаменів	заліків
90-100	відмінно	зараховано
74-89	добре	

60-73	задовільно	
0-59	незадовільно	не зараховано

Для визначення рейтингу студента (слухача) із засвоєння дисципліни $R_{\text{дис}}$ (до 100 балів) одержаний рейтинг з атестації (до 30 балів) додається до рейтингу студента (слухача) з навчальної роботи $R_{\text{нр}}$ (до 70 балів): $R_{\text{дис}} = R_{\text{нр}} + R_{\text{ат}}$.

11. Навчально-методичне забезпечення

1. Електронний навчальний курс, розроблений на базі платформи LMS Moodle, розміщений на навчальному порталі факультету інформаційних технологій за адресою: <https://elearn.nubip.edu.ua/course/view.php?id=582>

2. ПОЛОЖЕННЯ про екзамени та заліки у Національному університеті біоресурсів і природокористування України, затверджене Вченою радою НУБіП України № 8 від «26» квітня 2023 р.

12. Рекомендовані джерела інформації

Базові

1. Програмування на мові C : навчальний посібник до вивчення дисципліни “Програмування” для студентів зі спеціальностей 121 – «Інженерія програмного забезпечення» 122 – «Комп’ютерні науки» 123 – «Комп’ютерна інженерія» / Б.Л. Голуб, Ю.Є. Бояринова. - К. : , 2017. - 173 с.

2. Програмування на мові C : методичні вказівки до виконання лабораторних робіт з дисципліни «Програмування» для студентів спеціальності: 121 «Інженерія програмного забезпечення», 122 «Комп’ютерні науки», 123 «Комп’ютерна інженерія» / уклад.: Б. Л. Голуб, С. В. Циба, Ю. Є. Бояринова. - К. : , 2016. - 95 с.

3. Вінник В.Ю. Алгоритмічні мови та основи програмування: мова C / В.Ю. Вінник. – Житомир : ЖДТУ, 2007. – 328 с.

4. Ковалюк Т. В. Алгоритмізація та програмування: підручник з грифом МОН України / Т.В. Ковалюк. – Львів : Магнолія-2006, 2013. – 400 с.

Додаткові

5. ANSI, American National Standard for Information Systems – Programming Language C. – New York, 1990.

6. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.

7. Пол Іре. Об’єктно-орієнтоване програмування з використанням C++: Пер. з англ. - Київ: НИПФ “ДіаСофт Лтд, 1995.

8. Bjarne Stroustrup The C++ Programming language, Addison Weasley, 1986.

9. Фейсон Т. Об’єктно-орієнтоване програмування на Borland C++ 4.5: Пер. з англ. - Київ: Діалектика, 1996. 544с. 5. Сван Т. Опанування Borland C++ 4.5: Пер. з англ. - Київ: Діалектика, 1996. 544с.

10. Object-Oriented Data Structures in C++ (Coursera). URL: <https://dev.ua/news/valid-article-1676469637-1676469637#title0>

11. Beginning C++ Programming Course — Beginner to Beyond (Udemy) URL: <https://dev.ua/news/valid-article-1676469637-1676469637#title1>

12. Coding for Everyone: C and C++ Specialization (Coursera) URL: <https://dev.ua/news/valid-article-1676469637-1676469637#title2>
13. Unreal Engine 4 Mastery: Create Multiplayer Games with C++ (Udemy) URL: <https://dev.ua/news/valid-article-1676469637-1676469637#title3>
14. Learn C++ for Game Development (Udemy) URL: <https://dev.ua/news/valid-article-1676469637-1676469637#title4>
15. C++: від початку до експерта (Udemy) URL: <https://dev.ua/news/valid-article-1676469637-1676469637#title5>
16. Learn Advanced C++ Programming (Udemy) URL: <https://dev.ua/news/valid-article-1676469637-1676469637#title6>