

## СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ»



<b>Ступінь освіти</b>	бакалавр
<b>Спеціальність</b>	121 Інженерія програмного забезпечення
<b>Освітня програма</b>	Інженерія програмного забезпечення
<b>Загальний обсяг</b>	8 кредитів ЄКТС (240 годин)
<b>Тривалість викладання</b>	1-й семестр (1,2 чверть)
<b>Заняття:</b>	
лекції:	4 години/тиждень
лабораторні заняття:	4 години/тиждень
<b>Мова викладання</b>	українська

Сторінка курсу в СДО НТУ «ДП»: <https://do.nmu.org.ua/course/view.php?id=3586>

Кафедра, що викладає Програмного забезпечення комп'ютерних систем



**Викладач:**

Мартиненко Андрій Анатолійович  
ст. викладач

**Персональна сторінка**

<https://pzks.nmu.org.ua/ua/teachers/teachers.php>

**E-mail:** martynenko.a.a@nmu.one

### 1. Анотація до курсу

Основною метою курсу є формування бази теоретичних та практичних знань в області алгоритмізації та програмування для студентів Національного технічного університету «Дніпровська політехніка» спеціальності 121 Інженерія програмного забезпечення першого (бакалаврського) рівня вищої освіти для вирішення різноманітних практичних та теоретичних задач у професійній діяльності. В рамках курсу розглядаються такі питання: життєвий цикл програмного забезпечення, типи обчислювальних процесів, основи алгоритмізації, технології розробки програмного забезпечення, мови програмування та об'єктно орієнтоване програмування.

### 2. Мета та завдання курсу

**Мета дисципліни** – формування у здобувачів компетентностей щодо застосування методів розробки алгоритмів, конструювання програмного

забезпечення та структур даних і знань, а також вміння застосовувати інформаційні технології обробки, зберігання та передачі даних.

**Завданнями** дисципліни є:

- опанування знаннями та навичками роботи на різних етапах життєвого циклу програмного забезпечення;
- опанування знаннями методів та засобів конструювання програмного забезпечення та розробки алгоритмів та програмного забезпечення;
- опанування знаннями та вміннями використовувати на практиці основи алгоритмізації обчислювальних процесів;
- опанування знаннями та навичками розробки програмного забезпечення.

### 3. Дисциплінарні результати навчання

Дисциплінарні результати навчання сформовано на основі ПРН освітньо-професійної програми «Інженерія програмного забезпечення» першого (бакалаврського) рівня вищої освіти (ПРЗ, ПР13, ПР17):

- знати основні процеси життєвого циклу програмного забезпечення;
- знати основні поняття алгоритмізації обчислювальних процесів, вміти проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач;
- знати і застосовувати на практиці інструментальні засоби для реалізації методів та алгоритмів розв'язання задач в галузі інженерії програмного забезпечення;
- володіти навичками розробки програмних моделей предметних середовищ з використанням сучасних технологій, методів та інструментів;
- вміти застосовувати методи компонентної розробки програмного забезпечення.

### 4. Структура курсу

Види та тематика навчальних занять	Внесок в загальну оцінку, %
<b>ЛЕКЦІЇ</b>	<b>40</b>
<p><b>Тема 1. Етапи розробки ПЗ. Життєвий цикл ПЗ</b>            Поняття і завдання дисципліни «Алгоритмізація та програмування»            Етапи розробки ПЗ.            Процеси життєвого циклу ПЗ</p>	
<p><b>Тема 2. Основні методи та засоби програмування та алгоритмізації на ЕОМ</b>            Особливості технології програмування            Методи та засоби конструювання програмного забезпечення та розробки алгоритмів та ПЗ.            Види і склад систем програмування</p>	
<p><b>Тема 3. Алгоритмізація обчислювальних процесів</b>            Поняття алгоритму і його основні властивості            Основні поняття алгоритмізації обчислювальних процесів            Способи завдання алгоритмів            Структури алгоритмів            Приклади способів завдань структур алгоритмів            Приклади розв'язання задач на складання алгоритмів</p>	

Види та тематика навчальних занять	Внесок в загальну оцінку, %
<b>Тема 4. Введення в мову програмування C#</b> Загальна характеристика мови Технологія розробки програм Типи даних і вирази Алфавіт і ідентифікатори Операції, вирази й оператори Класифікація типів даних Оголошення змінних Завдання констант Час існування і область видимості змінних	
<b>Тема 5. Програмування розгалужень</b> Умовний оператор <i>if</i> та оператор вибору <i>switch case</i> Приклади використання операторів <i>if</i> та <i>switch case</i>	
<b>Тема 6. Програмування циклів</b> Оператор <i>while</i> Оператор <i>do... while</i> Оператор <i>for</i> Приклади використання операторів. Вкладені цикли Рекомендації по вибору циклів Приклади використання	
<b>Тема 7. Масиви</b> Оголошення та ініціалізація масивів Одновимірні та двовимірні масиви Приклади використання	
<b>Тема 8. Методи (Функції)</b> Загальні відомості про методи Рекурсивні і підставляються функції Області дії змінних Масиви в якості параметрів функцій (одномірні, багатовимірні та динамічні масиви)	
<b>Тема 9. Файлове введення виведення даних</b> Робота з файлами засобами мови C#	
<b>Тема 10. Введення в об'єктно-орієнтоване програмування</b> Структурний підхід в програмуванні Концепції об'єктно-орієнтованого програмування Етапи об'єктно-орієнтованого програмування Поняття класу та об'єкту	
<i>Контрольна робота №2 (теми 1-10)</i>	20
<b>Тема 11. Візуальне та компонентне програмування</b> Основні компоненти .NET Framework Середовище виконання (CLR) та бібліотека класів .NET Framework Огляд ADO.NET, ASP.NET, Windows Forms та Windows Presentation Foundation (WPF).	
<b>Тема 12. Розробка пайпростішої програми з екранною формою і елементами управління (Мова програмування C#)</b>	

<b>Види та тематика навчальних занять</b>	<b>Внесок в загальну оцінку, %</b>
<p>Приклад 1. Форма, кнопка, мітка та діалогове вікно</p> <p>Приклад 2. Подія MouseHover</p> <p>Приклад 3. Вибір потрібної дати</p> <p>Приклад 4. Введення даних через текстове поле TextBox із перевіркою типу методом TryParse</p> <p>Приклад 5. Введення пароля в текстове поле та зміна шрифту</p> <p>Приклад 6. Керування стилем шрифту за допомогою керування CheckBox</p> <p>Приклад 7. Побітовий оператор "що виключає АБО"</p> <p>Приклад 8. Вкладки TabControl та перемикачі RadioButton</p> <p>Приклад 9. Властивість Visible та підказка ToolTip в стилі Balloon</p> <p>Приклад 10. Калькулятор на основі комбінованого списку ComboBox</p> <p>Приклад 11. Виведення грецьких літер, символів математичних операторів. Кодова таблиця Unicode</p>	
<p><b>Тема 13. Ініціювання та обробка подій миші і клавіатури (Мова програмування C#)</b></p> <p>Приклад 1. Координати курсору миші щодо екрану та елемента керування</p> <p>Приклад 2. Створення елемента управління Button "програмним" способом та підключення події для нього</p> <p>Приклад 3. Обробка кількох подій однією процедурою</p> <p>Приклад 4. Калькулятор</p> <p>Приклад 5. Посилання на інші ресурси LinkLabel</p> <p>Приклад 6. Обробка подій клавіатури</p> <p>Приклад 7. Введення у текстову полі лише цифри, а також роздільник цілої та дробової частини числа</p> <p>Приклад 8. Виклик і обробка події "кляцання на кнопці"</p>	
<p><b>Тема 14. Читання, запис текстових і бінарних файлів, розробка текстового редактора (Мова програмування C#)</b></p> <p>Приклад 1. Читання/запис текстового файлу у кодуванні Unicode. Обробка винятків try...catch</p> <p>Приклад 2. Читання/запис текстового файлу у кодуванні Windows 1251</p> <p>Приклад 3. Простий текстовий редактор. Відкриття та збереження файлу. Подія форми Closing</p> <p>Приклад 4. Програма тестування знань студента з якогось предмета</p> <p>Приклад 5. Простий RTF-редактор</p> <p>Приклад 6. Програма введення каталогу координат (числових даних) із текстового файлу</p> <p>Приклад 7. Друк текстового документа</p> <p>Приклад 8. Читання/запис бінарних файлів за допомогою потоку даних</p>	
<p><b>Тема 15. Редагування графічних даних (Мова програмування C#)</b></p> <p>Приклад 1. Найпростіший висновок відображення графічного файлу у форму</p> <p>Приклад 2. Використання PictureBox для відображення растрового файлу з можливістю прокручування</p> <p>Приклад 3. Малювання у вигляді графічних примітивів (фігур)</p> <p>Приклад 4. Вибір кольору за допомогою ListBox</p>	

<b>Види та тематика навчальних занять</b>	<b>Внесок в загальну оцінку, %</b>
Приклад 5. Екранна форма із трикутником прозорості Приклад 6. Друк графічних примітивів Приклад 7. Друк BMP-файлу Приклад 8. Створення JPG-файлу "на льоту" та виведення його відображення у форму Приклад 9. Зміна виведеного зображення за допомогою оновлення форми Приклад 10. Малювання у формі покажчиком миші Приклад 11. Управління сплайном Безьє Приклад 12. Побудова графіка за методами класу Graphics	
<b>Тема 16. Управління буфером обміну з даними в текстовому і графічному форматах (Мова програмування С#)</b> Приклад 1. Буфер обміну даних у текстовому форматі Приклад 2. Елемент керування PictureBox. Буфер обміну з растровими даними Приклад 3. Імітація натискання клавіш <Alt>+<PrintScreen> Приклад 4. Запис вмісту буфера обміну в BMP-файл Приклад 5. Використання таймера Timer Приклад 6. Запис у файли поточного стану екрана кожні п'ять секунд	
<i>Контрольна робота №2 (теми 11-16)</i>	20
<b>ЛАБОРАТОРНІ ЗАНЯТТЯ</b>	<b>60</b>
<b>Лабораторна робота 1</b> Основні поняття алгоритмізації обчислювальних процесів. Способи завдання алгоритмів. Структури алгоритмів. Приклади способів завдань структур алгоритмів. Приклади розв'язання задач на складання алгоритмів. Розробка алгоритмів різних типів та форм представлення різних обчислювальних процесів.	4
<b>Лабораторна робота 2</b> Знайомлення з середовищем розробки Microsoft Visual Studio. Типи проектів. Створення проекту в Microsoft Visual Studio. Розробка програми з лінійним обчислювальним процесом.	4
<b>Лабораторна робота 3</b> Розробка програми з розгалужуючим процесом.	4
<b>Лабораторна робота 4</b> Розробка програми з циклічним процесом.	4
<b>Лабораторна робота 5</b> Розробка програми по роботі з масивами. Пошук елементів, сортування масивів.	4
<b>Лабораторна робота 6</b> Розробка програми по роботі з рядками.	4
<b>Лабораторна робота 7</b> Розробка програми по роботі з методами (функціями).	4
<b>Лабораторна робота 8</b> Розробка програми по роботі з файловим введенням-виведенням.	4

<b>Види та тематика навчальних занять</b>	<b>Внесок в загальну оцінку, %</b>
<b>Лабораторна робота 9</b> Розробка програми по роботі з класами.	4
<b>Лабораторна робота 10</b> Розробка консольного додатку.	4
<b>Лабораторна робота 11</b> Розробка найпростішої програми з екранною формою і елементами управління.	4
<b>Лабораторна робота 12</b> Ініціювання та обробка подій миші і клавіатури.	4
<b>Лабораторна робота 13</b> Читання, запис текстових і бінарних файлів, розробка текстового редактора.	4
<b>Лабораторна робота 14</b> Редагування графічних даних.	4
<b>Лабораторна робота 15</b> Управління буфером обміну з даними в текстовому і графічному форматах.	4
<b>ВСЬОГО:</b>	<b>100</b>

## **5. Технічне обладнання та/або програмне забезпечення.**

Використовуються лабораторії кафедри програмного забезпечення комп'ютерних систем (комп'ютерне та мультимедійне обладнання). Дистанційна платформа Moodle, MS Office 365, Microsoft Teams, Microsoft Visual Studio 2022.

## **6. Система оцінювання та вимоги**

**6.1. Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:**

<b>Рейтингова шкала</b>	<b>Інституційна шкала</b>
90 – 100	відмінно
74 – 89	добре
60 – 73	задовільно
0 – 59	незадовільно

**6.2.** Здобувачі вищої освіти можуть отримати **підсумкову оцінку** з навчальної дисципліни **на підставі поточного оцінювання знань** за умови, якщо набрана кількість балів з поточного тестування та виконання і захисту лабораторних робіт складатиме не менше 60 балів.

**Теоретична частина** оцінюється за результатами здачі двох контрольних тестових робіт, кожна з яких містить тестові закриті запитання з однією вірною

відповіддю (максимальна кількість – 20 балів за кожною тестовою роботою). Загалом за дві контрольні тестові роботи отримується **максимум 40 балів**, тобто 40% від оцінки за дисципліну.

**Лабораторні роботи** (15 робіт – у вигляді індивідуального завдання з кожної, розподіл % див. в таблиці розділу 4) виконуються у письмовому вигляді (звіт з кожної роботи оцінюється в межах балів, представлених в таблиці розділу 4, загалом лабораторні враховуються як 60% (максимум 60 балів). При несвоєчасному здаванні роботи оцінка знижується вдвічі. Лабораторні роботи захищаються у вигляді опитування за звітом, і захист враховується, як 50% від оцінки за роботу. У сумі за лабораторну частину курсу при поточному оцінюванні отримується **максимум 60 балів**.

Отримані бали за теоретичну частину та лабораторні роботи додаються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за поточною успішністю здобувач вищої освіти може набрати 100 балів.

**6.3. Критерії оцінювання підсумкової роботи.** У випадку якщо здобувач вищої освіти за поточною успішністю отримав менше 60 балів та/або прагне поліпшити оцінку проводиться **підсумкове оцінювання**.

**Підсумкове оцінювання** проводиться у вигляді комплексної контрольної роботи, яка включає запитання з теоретичної та практичної частини курсу. Білет складається з **30 тестових завдань** з чотирма варіантами відповідей, одна правильна відповідь оцінюється в 2 бали (**разом 60 балів**) та **2 завдань** з практичної частини, кожне з запитань оцінюється максимум у 20 балів (**разом 40 балів**).

Отримані бали за тестові завдання та завдання з практичної частини додаються і є підсумковою оцінкою за вивчення навчальної дисципліни в 1 семестрі. Максимально за підсумковою роботою здобувач вищої освіти може набрати 100 балів.

## 7. Політика курсу

**7.1. Політика щодо академічної доброчесності.** Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням "Положення про систему запобігання та виявлення плагіату у Національному технічному університеті "Дніпровська політехніка" <https://cutt.ly/MCfh5kv>

У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

**7.2. Комунікаційна політика.** Здобувачі вищої освіти повинні мати активовану корпоративну університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

**7.3. Політика щодо перескладання.** Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання підсумкового оцінювання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

**7.4 Політика щодо оскарження оцінювання.** Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

## 8 Рекомендовані джерела інформації

1. Програмування мовою С# 6.0 / Коноваленко І.В. Навчальний посібник для технічних спеціальностей вищих навчальних закладів. Тернопіль. ТНТУ. 2016 – 217с.

2. Мова програмування С# [Електронний ресурс]. – Режим доступу: <http://www.znannya.org/?view=csharp>

3. ISO/IEC/IEEE 12207:2017 «Systems and software engineering — Software life cycle processes» [Режим доступу]: <https://www.iso.org/standard/63712.html>

4. C Sharp – вікіпідручник [Електронний ресурс]. – Режим доступу: [https://uk.wikibooks.org/wiki/C\\_Sharp](https://uk.wikibooks.org/wiki/C_Sharp)

5. Голуб Б.М. С#. Концепція та синтаксис. Навч. посібник. – Львів 2006 – 136с. [Електронний ресурс].– Режим доступу: [https://www.dut.edu.ua/uploads/1\\_1569\\_93311357.pdf](https://www.dut.edu.ua/uploads/1_1569_93311357.pdf)

6. Daniel Solis. Illustrated C# 2012. – APress, 2012. – 732p.

7. Standard ECMA-334. C# Language Specification. 4th Edition. – Ecma International. – June 2006.

8. International standard ISO/IEC 23270:2006. Information technology – Programming languages – C#. Second edition. – ISO/IEC. – 2006.

9.

10. <https://kleban.page/courses/csharp-basics>

11. [https://uk.wikibooks.org/wiki/C\\_Sharp](https://uk.wikibooks.org/wiki/C_Sharp)

12. <https://programm.top/uk/c-sharp>

13. <https://www.uzhnu.edu.ua/uk/infocentre/get/5868>

14. <http://www.znannya.org/?view=csharp>

15. <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/tutorials>

16. <https://dotnet.microsoft.com/en-us/learn/csharp>