

УКРАЇНА
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до виконання курсового проекту
з дисципліни «Програмна технологія .NET»
для студентів спеціальності 121
«Інженерія програмного забезпечення»
освітньо-кваліфікаційного рівня «Бакалавр»

Київ 2023

УДК 378.022.51

Викладено загальні методичні рекомендації до виконання курсового проекту, вимоги до змісту підрозділів, правила оформлення пояснювальної записки, список базової, допоміжної літератури та інформаційних ресурсів. У додатках наведені зразки оформлення пояснювальної записки та орієнтовні теми для курсового проекту.

Рекомендовано вченою радою факультету інформаційних технологій Національного університету біоресурсів і природокористування, протокол №
від _____ р.

Укладач: ст. викладач Ю.О. Міловідов

Рецензенти: канд. техн. наук Голуб Б.Л.
 канд. техн. наук Бородкіна І.Л.

Навчальне видання

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ДО ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ
З ДИСЦИПЛІНИ «ПРОГРАМНА ТЕХНОЛОГІЯ .NET»
Для студентів спеціальності
121 – «Інженерія програмного забезпечення»
(денна форма навчання)

Укладач: Міловідов Юрій Олеголич

ЗМІСТ

1 ЗАГАЛЬНІ МЕТОДИЧНІ ВКАЗІВКИ	4
1.1 Мета курсового проекту	4
1.2 Основні задачі курсового проекту	5
1.3 Структура і зміст курсового проекту.....	5
1.4 Порядок виконання курсового проекту.....	6
1.5 Основні вимоги до оформлення комплексу документації на курсовий проект	6
1.6 Порядок захисту курсового проекту.....	7
2 ОСОБЛИВОСТІ ВИКОНАННЯ РОЗДІЛІВ КУРСОВОГО ПРОЕКТУ	7
2.1 Аналіз предметної галузі задачі.....	7
2.2 Постановка задачі.....	7
2.3 Структури даних та діаграми класів для розв'язання задачі	8
2.4 Виконання схем алгоритмів та ієрархії класів.....	9
2.5 Текст програми	9
2.6 Програма і методика випробувань.....	9
2.7 Інструкція користувачеві.....	11
3 ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ ДО КУРСОВОГО ПРОЕКТУ	13
3.1 Загальні вимоги	13
3.2 Нумерація сторінок	14
3.3 Нумерація розділів, підрозділів, пунктів, підпунктів	14
3.4 Ілюстрації.....	15
3.5 Таблиці	16
3.6 Перерахування.....	17
3.7 Примітки	18
3.8 Винесення.....	19
3.9 Формули	19
3.10 Посилання	20
3.11 Список використаних джерел	21
3.12 Додатки.....	22
КРИТЕРІЇ ОЦІНЮВАННЯ	24
ПЕРЕЛІК НАВЧАЛЬНО-МЕТОДИЧНОЇ ЛІТЕРАТУРИ	25
ДОДАТОК А.....	26
ДОДАТОК Б	27
ДОДАТОК В.....	27
ДОДАТОК Г	28

1 ЗАГАЛЬНІ МЕТОДИЧНІ ВКАЗІВКИ

1.1 Мета курсового проекту

Мета виконання курсового проекту полягає у закріпленні у студентів знань та практичних навичок отриманих при вивчанні дисципліни «Програмна технологія .NET». Мета досягається при виконанні об'єктного аналізу завдання, розробці ієрархії класів і алгоритму взаємодії їх об'єктів, складанні програми на мові C#.

У ході виконання курсового проекту студенти можуть:

- Навчитися виконувати аналіз предметної області задачі, виділяти понятійний базис та створювати логіко-функціональну і об'єктну моделі фізичних та логічних процесів на прикладах завдань з моделями програмування, реалізованими FCL (Framework Class Library), включаючи Windows Forms, Web Forms, WPF, ASP.NET, ADO.NET;
- Отримати навички прикладного програмування на єдиній універсальній платформі (framework) Microsoft .NET в інтегрованому середовищі Microsoft Visual Studio .NET з використанням мови програмування C#.
- Отримати практичний досвід роботи з MS SQL Server, знання Entity Framework, базовий рівень знань технології ASP.NET MVC.

При виконанні курсового проекту студент розробляє схеми алгоритмів та ієрархії класів, створює безпосередньо програму та налагоджує її. Результатом, що подається на захист є текст пояснювальної записки та презентація програми.

Текст пояснювальної записки повинен містити: аналіз завдання та відповідної предметної області, обґрунтування обраних рішень, спроектовані класи та алгоритми, текст програми, результати її тестування, результати виконання програми у вигляді контрольних прикладів.

Ця робота підсумовує знання і навички студентів у галузі програмування, які здобуті студентами на поточному та попередніх курсах. При її виконанні

студент має проявити свої творчі здібності до самостійного аналізу, алгоритмізації та створення програм.

1.2 Основні задачі курсового проекту

1. Ознайомитися з основними сучасними технологіями програмування платформи Microsoft .Net.
2. Одержати практичні навички розробки алгоритмів і програм.
3. Навчитися використовувати нормативні документи в інженерній діяльності.
4. Одержати практичні навички з обґрунтування методів вирішення різноманітних задач інформатики.
5. Засвоїти основні етапи розробки програмних виробів, їх зміст і форму виконання в програмних документах.

1.3 Структура і зміст курсового проекту

Курсовий проект оформлюється у вигляді комплексу програмної документації в який входять:

Курсовий проект:

1. титульна сторінка (див. Додаток А);
2. завдання на виконання роботи (див. Додаток Б);
3. графік виконання роботи (див. Додаток В);
4. Текст роботи:
 - 4.1 зміст роботи;
 - 4.2 аналіз предметній галузі задачі;
 - 4.3 постановку задачі;
 - 4.4 структури даних та діаграми класів для розв'язання задачі;
 - 4.5 схеми взаємодії об'єктів та алгоритми виконання їх методів;
 - 4.6 опис розробленої програми;
 - 4.7 контрольний приклад (опис, результати його виконання у вигляді роздруківки);
 - 4.8 висновки;

4.9 список літератури;

4.10 додатки:

4.10.1 роздруківка тексту програми з детальними коментарями;

4.10.2 комплект тестів, на яких було перевірено функціонування програми;

4.10.3 інструкція по використанню програми.

1.4 Порядок виконання курсового проекту

В процесі виконання курсового проекту необхідно:

- Вивчити основні етапи розробки програмних засобів;
- Уточнити постановку завдання та, якщо це потрібно, сформулювати математичну модель задачі і обґрунтувати вибір методу її розв'язання;
- Розробити функціональну структуру програми (перелік та призначення її режимів);
- Розробити структури даних задачі;
- Розробити алгоритм виконання задачі;
- Виконати кодування та препарування програми, її синтаксичне відлагоджування;
- Спроектувати тести та виконати тестування та відлагоджування програми;
- Оформити пояснювальну записку на курсовий проект;
- Захистити курсовий проект.

1.5 Основні вимоги до оформлення комплексу документації на курсовий проект

Індивідуальні особливості виконання кожного з розділів курсового проекту описані в розділі 2.

Правила оформлення комплексу документації на курсовий проект відповідно до вимог ЄСПД наведені в розділі 3.

1.6 Порядок захисту курсового проекту

Для захисту курсового проекту необхідно подати:

- комплект програмної документації, який було розроблено під час курсового проектування;
- в окремому файлі (або файлах) вихідний текст розробленого програмного виробу;
- EXE-файл розробленого програмного виробу, готового до використання.

Під час захисту курсового проекту необхідно:

- вміти обґрунтувати метод розв'язання задачі;
- вміти оформлювати документи на програмні вироби відповідно до стандартів;
- твердо знати основи програмування;
- вміти розробляти алгоритми вирішених задач.

2 ОСОБЛИВОСТІ ВИКОНАННЯ РОЗДІЛІВ КУРСОВОГО ПРОЕКТУ

2.1 Аналіз предметній галузі задачі

Розділ «Аналіз предметній галузі задачі» містить аналітичний огляд літературних джерел та прототипів щодо розкриття теми курсового проекту (2-3 стор.). Він повинен характеризувати ступінь знайомства студента із задачею, її математичним змістом та різними методами її розв'язання та існуючими програмними системами, в яких вирішуються аналогічні задачі. Літературними джерелами можуть бути підручники, монографії, статті, електронні джерела (Інтернет, телеконференції тощо). Посилання на літературні джерела є обов'язковими.

2.2 Постановка задачі

На основі попереднього розділу студент повинен формалізувати «Постановку задачі» та надати обґрунтований вибір варіанту її рішення (якщо такі існують), відзначити особливості та переваги обраного варіанту рішення.

Перед розробкою програми необхідно визначити основні математичні

співвідношення та розробити математичний алгоритм рішення задачі, тобто побудувати математичну модель процесів, що відбуваються при вирішенні завдання. Для цього рекомендується користуватися довідниками з математики, фізики, а також науковою і довідковою літературою з предметної галузі.

Потім необхідно представити собі сценарій роботи програми: що треба вводити в програму, що і як виводити на екран, які перетворення даних здійснюються тощо. Для розробки алгоритму використовувати метод низхідного проектування: спочатку визначити загальний підхід до вирішення задачі, потім виділити окремі блоки, які виконують закінчений процес обробки даних, далі конкретизувати блоки, які були виділені раніше. Цей процес продовжувати до тих пір, поки не визначаться прості операції.

2.3 Структури даних та діаграми класів для розв'язання задачі

Розробка програмного продукту - багатоступеневий процес, що повторюється. Основний цикл цього процесу: намітити загальний проект; знайти стандартні компоненти; створити нові компоненти; створити проект. Для проектування компонент необхідно виконати такі дії:

- знайти основні поняття та об'єктні типи (класи) і їх основні взаємозв'язки в ієрархії успадкування;
- уточнити об'єктні типи, задавши множину операцій над ними. Для цього розбити операції на групи, проаналізувати потребу в конструкторах, деструкторах, інших операціях. проаналізувати операції в залежності від того, як вони змінюють внутрішній стан об'єктів і визначити базисні операції - конструктори, функції-методи та функції-властивості. Взяти до уваги повноту, мінімальність, зручність до застосування, зрозумілість і наявність коментарів до коду, що розробляється;
- уточнити об'єктні типи, визначивши, як вони залежать від інших об'єктних типів, для цього визначити об'єктні типи, які успадковують властивості інших об'єктних типів;
- визначити об'єктні типи, які підпорядковуються іншим об'єктним типам.

Програмний код кожного класу виокремити в файл, доданий до проекту.

2.4 Виконання схем алгоритмів та ієрархії класів

Схеми повинні дати наочну уяву послідовності дій, що запрограмовані. Розробляються такі схеми:

- схема алгоритму, яка відображає процес знаходження рішення задачі,
- схема ієрархії об'єктів, яка показує взаємозв'язок і успадкованість класів,
- структурна схема програми або схема взаємозв'язку функцій і процедур програми, яка відображає послідовність викликів процедур і функцій.

Обов'язково додається специфікація функцій і процедур, де наводиться формат оголошення функцій і процедур з поясненням змісту параметрів. Приклад схеми алгоритмів наведено на рис. 2.1. Приклади схем ієрархії класів та відношень між класами наведені на рис.2.2 .

2.5 Текст програми

Текст програми складається з текстів одного або декількох поименованих розділів. У текст програми рекомендується включати коментарі, що можуть відображати функціональне призначення і структуру програми.

2.6 Програма і методика випробувань

Програма і методика випробувань містить в собі такі розділи.

Анотація.

Зміст.

Об'єкт випробувань. Вказується найменування, область застосування і позначення програми, що випробовується.

Мета випробування. Вказується мета проведення випробувань.

Вимоги до програми. Повинні бути наведені вимоги на програму які підлягають перевірці під час випробувань.

Засоби і порядок випробувань. Повинні бути зазначені технічні і програмні засоби, які використовуються під час випробувань, а також порядок проведення випробувань.

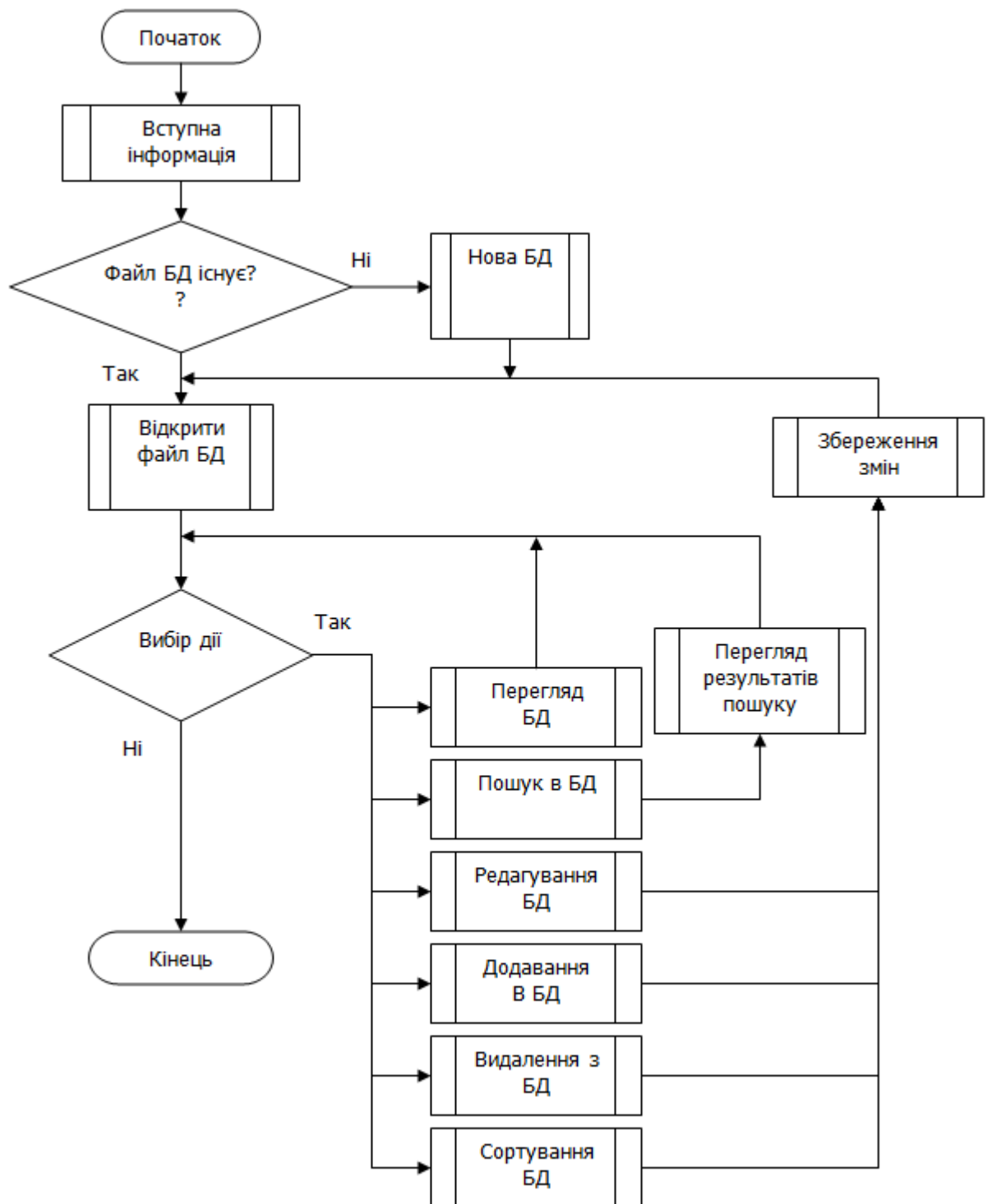


Рис. 2.1 Приклад блок-схеми алгоритму

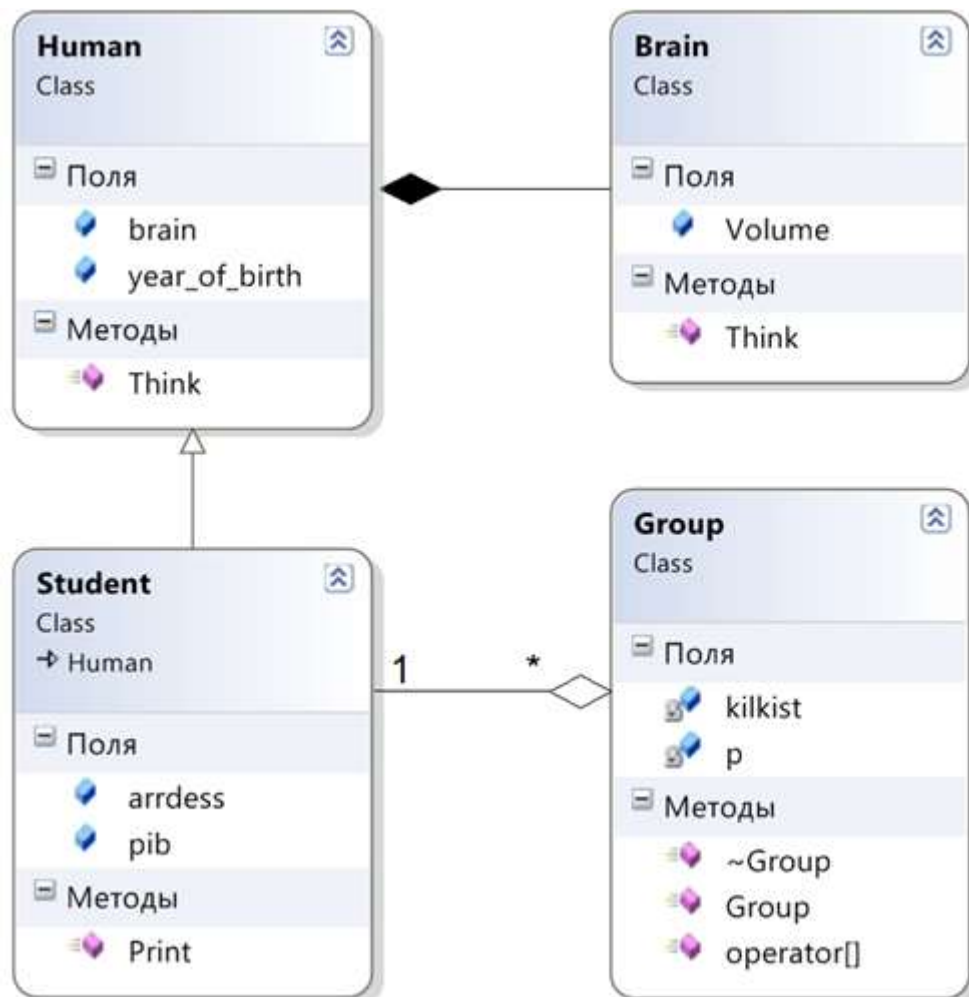


Рис. 2.2. Відносини між класами типу спадкування, агрегація і композиція.

Методи випробувань. Приводяться описи методів, які використовуються під час випробувань.

У методах випробувань повинні бути наведені описи перевірок, із вказівкою результатів проведення випробувань.

Для кожного тесту наводяться:

- призначення тесту;
- перелік тестових даних;
- контрольні результати тестових прикладів.

2.7 Інструкція користувачеві

Інструкція користувачеві містить такі розділи.

Анотація.

Зміст.

Загальні відомості. У цьому розділі повинні бути зазначені:

- Позначення і найменування програми.
- Програмне забезпечення, необхідне для функціонування програми.
- Мови програмування, на яких було написано програму.

Функціональне призначення. Містить клас розв'язуваних задач і (або) призначення програми і відомості про функціональні обмеження її застосування.

Призначення й умови застосування програми. Вказуються призначення й функції, які реалізуються програмою, умови, необхідні для її виконання (тип ЕОМ, операційна система, обсяг оперативної пам'яті, вимоги до складу і параметрів периферійних пристроїв, вимоги до програмного забезпечення і т.ін.).

Характеристики програми. Наводиться опис основних характеристик і особливостей програми (потрібний обсяг пам'яті, часові характеристики, режим роботи, засоби контролю правильності виконання і т.ін.).

Виклик і завантаження. Вказуються засіб виклику програми з відповідного носія даних та вхідні точки в програму.

Вхідні дані. Вказуються характер, організація, попередня підготовка, формат, опис і засіб кодування вхідних даних.

Вихідні дані. Характер і організація вихідних даних.

Повідомлення. Вказуються тексти повідомлень, що видаються програмісту або оператору в ході виконання програми, опис їхнього змісту і дії, які необхідно виконати у випадку появи цих повідомлень.

3 ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ ДО КУРСОВОГО ПРОЕКТУ

3.1 Загальні вимоги

Пояснювальну записку до курсового проекту оформляють на аркушах формату А4 (210х297). Допускається використання аркушів формату А3 (297х420), коли це необхідно.

Пояснювальну записку до курсового проекту виконують машинописним або машинним способом (за допомогою комп'ютера) на одній стороні аркуша.

При машинописному способі текст друкується через півтора інтервали; при машинному розміщують не більш 40 рядків на сторінці при умові рівномірного її заповнення і висоті символів не менше 1,8 мм.

Текст пояснювальної записки до курсового проекту необхідно друкувати, залишаючи поля наступних розмірів: ліве - не менше 20 мм, праве - не менше 10, верхнє - не менше 20, нижнє - не менше 20 мм.

Шрифт тексту належний бути чітким, рядок - чорного кольору середньої щільності, однакової по всьому тексту.

Вписувати в текст пояснювальної записки окреме іноземне слово, формули, умовні позначення можна чорнилами або тушшю, тільки чорного кольору, при цьому щільність вписаного тексту належна бути наближена до щільності основного тексту.

Текст основної частини роботи розбивають на розділи, підрозділи, пункти та підпункти.

Заголовки структурних частин пояснювальної записки “ЗМІСТ”, “ВСТУП”, “ВИСНОВКИ”, “СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ”, “ДОДАТКИ” не нумерують.

Заголовки структурних елементів роботи і заголовки розділів необхідно розташовувати в середині рядка і друкувати прописними літерами без крапки в кінці, не підкреслюючи.

Заголовки підрозділів, пунктів і підпунктів необхідно починати з абзацного відступу і друкувати рядковими літерами, окрім першої. Крапка в

кінці заголовка не ставиться. Якщо заголовок складається з двох або більш речень, вони розділяються крапкою. Заголовки пунктів друкують рядковими літерами (крім першої прописної) врозрядку, без відступу від тексту. У кінці заголовка, надрукованого без відступу від тексту, ставиться крапка.

Відступ між заголовком (за винятком заголовка пункту) і наступним або попереднім текстом належний бути:

- при машинописному способі - не менше 3 інтервалів;
- при машинному способі - не менше двох рядків. Відстань між основами рядків заголовка, а також між двома заголовками приймають таку ж, як у тексті.

Не допускається розміщати найменування підрозділу, пункту, підпункту в нижній частині сторінки, якщо після нього розташовано менше 3 рядків тексту.

Кожну структурну частину необхідно починати з нової сторінки.

Роздруковані на ЕОМ програмні документи повинні відповідати формату. Їх включають у загальну нумерацію сторінок і розміщують, як правило, у додатку.

3.2 Нумерація сторінок

Сторінки нумерують арабськими цифрами, дотримуючись наскрізної нумерації по всьому тексту. Номер сторінки проставляється внизу сторінки, по центру, без крапки в кінці.

Першою сторінкою є титульний аркуш, який включають у загальну нумерацію сторінок. На титульному аркуші номер сторінки не ставлять.

Ілюстрації і таблиці, які розміщуються на окремих сторінках, включають у загальну нумерацію сторінок.

3.3 Нумерація розділів, підрозділів, пунктів, підпунктів

Розділи, підрозділи, пункти, підпункти нумерують арабськими цифрами.

Розділи повинні мати порядкову нумерацію в межах матеріалу, який викладається, і позначатися арабськими цифрами без крапки. По центру рядка друкується номер розділу без крапки та заголовок розділу.

Підрозділи нумерують у межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, розділених крапкою. У кінці номера підрозділу крапка не ставиться, наприклад: “3.2” (другий підрозділ третього розділу). Потім у тому ж рядку йде заголовок підрозділу.

Пункти нумерують у межах кожного підрозділу. Номер пункту складається з порядкового номера розділу, підрозділу, пункту. Між цифрами і у кінці номера ставиться крапка, наприклад: “1.2.4 ” (четвертий пункт другого підрозділу першого розділу). Заголовок пункту йде в тому ж рядку. Пункт може не мати заголовка.

Підпункти нумерують у межах кожного пункту по правилам нумерації пунктів.

3.4 Ілюстрації

Ілюстрації (фотографії, рисунки, схеми, графіки, карти) необхідно розміщати безпосередньо після першого нагадування про них в тексті або на наступній сторінці. На всі ілюстрації належні бути посилання.

Графічний матеріал повинен відповідати вимогам стандартів “Єдиної системи конструкторської документації” та “Єдиної системи програмної документації”.

Фотознімки розміром менше ніж А4 необхідно наклеювати на аркуші формату А4.

Після ілюстрації розміщують її назву. При необхідності ілюстрації доповнюють пояснюючими даними (підрисунковий текст).

Ілюстрація позначається словом “Рисунок _” з вказанням номера. Далі через тире розміщують назву ілюстрації. При необхідності далі може бути пояснювальний текст.

Ілюстрації необхідно нумерувати арабськими цифрами по порядку в межах розділу.

Номер ілюстрації складається з номера розділу та порядкового номера ілюстрації, розділених крапкою, наприклад: “Рисунок 2.1 - Структурна схема

генератора прямокутних імпульсів”.

Якщо в роботі одна ілюстрація, її нумерують за загальними правилами.

Якщо ілюстрація не вміщується на одній сторінці, її можна переносити на інші, при цьому назву ілюстрації розміщують на першій, а пояснюючі дані - на кожній сторінці (у відповідності з розміщеними на ній елементами). Ілюстрації на наступних аркушах підписують так: “Рисунок __, аркуш __”.

Ілюстрації при необхідності можуть бути перераховані в змісті з вказівкою їхніх номерів, найменування і номерів сторінок, на яких вони розташовані.

3.5 Таблиці

Цифровий матеріал оформляють у вигляді таблиць.

Горизонтальні та вертикальні лінії, які розмежовують рядки таблиці, а також лінії, що обмежують таблиці з боків, можна не проводити, якщо їхня відсутність не заважає користуванню таблицею.

Таблиця розташовується безпосередньо після тексту, у якому вона згадується в перший раз або на наступній сторінці. На всі таблиці в тексті належні бути зроблені посилання.

Таблиці нумерують арабськими цифрами по порядку в межах поділу.

Номер таблиці складається з номера розділу і порядкового номера таблиці, розділених крапкою, наприклад: **Таблиця 2.3** - третя таблиця другого поділу.

Якщо в роботі одна таблиця, її нумерують згідно вищевказаним вимогам.

Таблиця може мати назву, яку друкують рядковими літерами (перша прописна) і поміщають над таблицею. Назва повинна бути короткою та відображувати зміст таблиці.

Якщо рядки або графи таблиці виходять за формат сторінки, таблицю необхідно розділити на частині, розміщуючи одну частину під іншу, або поруч, або переносючи частину таблиці на наступну сторінку. При цьому в кожній частині таблиці повторюють її голівку і боковик.

Наприклад:

Таблиця 2.3 Специфікація функцій

			Заголовки
			граф
			Підзаголовки
			граф
			Рядки
			(горизонтальні ряди)

Ряди (колонки)

При розподілі таблиці на частини допускається її голівку або боковик замінити відповідно номерами граф або рядків. При цьому графи і/або рядки першої частини таблиці нумерують арабськими цифрами.

Слово “Таблиця _” вказують один раз зліва з абзацного відступу над першою частиною таблиці, над іншими частинами пишуть: “Продовження таблиці _” з вказівкою її номера.

Заголовки граф таблиці друкують із прописних літер, а підзаголовки - із рядкових, якщо вони складають одне речення з заголовком.

Підзаголовки, що мають самостійне значення, пишуть з прописної літери. У кінці заголовків і підзаголовків таблиць крапки не ставлять. Заголовки і підзаголовки граф вказують у єдиному числі.

Таблиці при необхідності можуть бути перераховані в змісті, з вказівкою їхніх номерів, назви і номерів сторінок, на яких вони розміщуються.

3.6 Перерахування

Перерахування при необхідності можуть бути приведені усередині пунктів або підпунктів. Перед перерахуванням ставлять двокрапку.

Перед кожною позицією перерахування необхідно ставити рядкову літеру українського алфавіту зі скобкою або, не нумеруючи, - дефіс (перший рівень деталізації).

Для подальшої деталізації перерахування необхідно використовувати

арабські цифри зі скобкою (другий рівень деталізації).

Наприклад:

- а) форма і розмір клітин;
- б) живий уміст клітин;
 - 1) частина клітин;
 - 2) неживі вмикання протопластів;
- в) утворення тканини.

Перерахування першого рівня деталізації друкують рядковими літерами з абзацного відступу, другого - із відступом щодо місцеположення перерахувань першого рівня.

3.7 Примітки

Примітки поміщають у тексті при необхідності пояснення змісту тексту, таблиці або ілюстрації.

Одна примітка не нумерується.

Слово «Примітка» друкують з прописної літери, з абзацного відступу, після слова «Примітка» ставлять крапку і з прописної літери в тому ж рядку дають текст примітки.

Наприклад:

Примітка. _____.

Декілька приміток нумерують послідовно арабськими цифрами з крапкою. Після слова «Примітки» ставлять двокрапку і з нового рядка з абзацного відступу після номера примітки з прописної літери дають його текст.

Наприклад:

Примітки:

- 1. _____.
- 2. _____.
- 3. _____.

3.8 Винесення

Пояснення до окремих даних, приведені у тексті або таблицях, допускається оформляти винесеннями.

Винесення позначають надрядковими арабськими цифрами по порядку у режимі надрядкового символу зі скобкою (¹). Нумерація винесень ведеться в межах сторінки.

Знаки винесень ставлять безпосередньо після того слова, числа, символу або іншого елемента тексту, до якого дається пояснення, і перед текстом пояснення.

Текст винесення поміщають під таблицею або в кінці сторінки і відокремлюють від таблиці або тексту лінією довжиною 30-40 мм, проведеною у лівій частині сторінки.

Текст винесення починають з абзацного відступу і друкують при машинописному способі виконання курсових і дипломних проектів через один інтервал, при машинному - із мінімальним міжрядковим інтервалом.

Наприклад:

У тексті: “Теорема Каратеодора ¹⁾. Нехай області...”

Відповідне уявлення винесення:

¹⁾ Константин Каратеодорн (1873 - 1950) - німецький математик грецького походження.

3.9 Формули

Формули і рівняння розташовують безпосередньо після їхнього нагадування в тексті, посередині сторінки.

Вище і нижче формули (рівняння) треба залишити не менше одного вільного рядка.

Формули і рівняння нумеруються порядковою нумерацією в межах розділу.

Номер формули або рівняння включає в себе номер розділу та порядковий номер формули (рівняння), розділених крапкою, наприклад: (2.3) - третя

формула другого розділу.

Номер формули (рівняння) вказують у тому ж рядку у крайньому правому положенні.

Пояснення значень символів і числових коефіцієнтів необхідно розміщати безпосередньо під формулою і у тієї ж послідовності, у якій вони використовуються у формулі.

Пояснення значення кожного символу і числового коефіцієнта необхідно давати з нового рядка. Перший рядок пояснення починають з абзацу словом “де” без двокрапки.

Наприклад:

Питома провідність плазми

$$\lambda = \frac{ne^2\tau}{m}, \quad (3.5)$$

де n – концентрація електронів;

e – заряд електрона;

m – його маса;

τ – інерційний час.

Переносити формули і рівняння на наступний рядок можна тільки на знаках операцій (+, -, =, , :). При переносі виразу на знаку множення його заміняють на “×”.

Формули, що впливають одна за другою і не розділені текстом, розміщують у стовпець і відокремлюють одну від другої комою.

3.10 Посилання

Посилання дають можливість перевірити достовірність зведень, приведених у курсових і дипломних проектах. Посилатися слід на останні видання публікацій, а на більш ранні - тільки у випадках, якщо приведений в них матеріал відсутній в наступних виданнях.

Якщо використовуються матеріали з монографій, оглядових статей, інших джерел з великою кількістю сторінок, необхідно вказувати номер сторінки, рисунка, таблиці і т.д., на які дані посилання.

Посилання на джерела необхідно вказувати порядковим номером переліку використаних джерел, відокремлюючи квадратними дужками, наприклад: [1 - 4].

Допускається приводити посилання у виносках, при цьому оформлення повинне відповідати їхньому бібліографічному опису по переліку використаної літератури з указівкою номера.

Наприклад:

Цитата в тексті: "Кінетичну систему можна охарактеризувати як сукупність перемінних та параметрів, що виражаються через вимірні величини, котрі в кожен момент часу приймають визначені числові значення " [12] 1).

Опис у переліку використаних джерел:

12. Рубін А.Б. Біофізика. - К.: Вища школа, 1987. - С. 13.

Відповідне уявлення винесення:

1) [12] Рубін А.Б. Біофізика. - К.: Вища школа, 1987. - С. 13.

Посилання на ілюстрації вказують їхнім порядковим номером, наприклад: рисунок 1.2; на формули - (2.3); на таблиці - таблиця 1.5.

При повторному звертанні до таблиці або ілюстрації вказують скорочено: «див. рисунок 2.3»

3.11 Список використаних джерел

Список використаних джерел можна розміщати наступним способом:

- у порядку появи джерела в тексті;
- в алфавітному порядку прізвищ перших авторів або заголовків;
- у хронологічному порядку.

Для книги необхідно вказати:

- порядковий номер;
- автор або автори;
- назва;
- місто та назва видавництва;

- рік видавництва;
- кількість сторінок.

Для статті необхідно вказати:

- порядковий номер;
- автор або автори;
- назва статті;
- назва збірки в якій надруковано статтю;
- рік видавництва та номер збірки;
- номери сторінок на яких надруковано статтю.

3.12 Додатки

Додатки необхідно оформляти як продовження пояснювальної записки на наступних сторінках, або, якщо це доцільно, окремим документом. Розміщують додатки в порядку їхньої появи в тексті.

Додатки нумерують літерами українського алфавіту за винятком літер Г, Є, З, І, Ї, О, Ч, Ї. Якщо додаток один, то він позначається як “А”.

Якщо додатки оформляють на сторінках після основного тексту записки, то кожен з них повинен починатися з нової сторінки. Додаток оформляється наступним способом: серед рядка вгорі сторінки друкується слово “ДОДАТОК _” з вказівкою його порядку (літери), далі з нового рядка друкується заголовок рядковими літерами з першої прописної і розміщується по центру щодо тексту сторінки.

Якщо додатки оформляються окремим документом, на титульному аркуші під назвою теми курсового проекту прописними літерами друкують слово “ДОДАТКИ”.

При необхідності текст додатка може мати розділи, підрозділи, пункти і підпункти, які необхідно нумерувати в межах кожного додатка, наприклад: А.5 - п'ятий розділ додатка А.

Загальним вимогам нумерації підкоряються і таблиці (“Таблиця Б.3” - третя таблиця додатка Б”), рисунки, формули і т.д.

При посиланні в тексті на матеріали, приміщені в додатку, рекомендується

писати: “на рис. А.1”, “у табл. В.2”, “у формулі (Д.3)”.

Джерела, що цитуються тільки в додатку, належно розглядати незалежно від основної частини і повинні бути перераховані в кінці кожного додатка в переліку використаних джерел.

Форма цитування, правила упорядкування переліку використаних джерел, і виноски повинні бути аналогічними приведеним в основній частині. Перед номером цитати і відповідним номером в переліку і виносках ставлять позначення додатка.

КРИТЕРІЇ ОЦІНЮВАННЯ

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 уведення в дію від 27.12.2019 р. No 1371)

Рейтинг здобувача вищої освіти, бали	Оцінка національна
90-100	відмінно
74-89	добре
60-73	задовільно
0-59	незадовільно

ПЕРЕЛІК НАВЧАЛЬНО-МЕТОДИЧНОЇ ЛІТЕРАТУРИ

Базова

1. Эндрю Троелсен, Филипп Джепикс. Язык программирования C# 7 и платформы .NET и .NET Core - 8-е изд. - М. : ООО "И.Д. Вильямс", 2016. - 1328с.
2. Джеймс Чамберс, Дэвид Пэккетт, Саймон Тиммс. ASP.NET Core. Разработка приложений . - СПб. : "Питер Пресс", 2017. - 600с.
3. Програмна технологія .NET: Навчальний посібник. / Ю.О. Міловідов. – К. НУБіП України, 2020. – 253с.
4. Гросс, К. C# и платформа NET 3.5 Framework: базовое руководство : учебное пособие / К. Гросс. - 2-е изд. - М. : ООО "И.Д. Вильямс", 2009. - 480 с.

Допоміжна

1. ДСТУ 3008-95 „Документація. Звіти в сфері науки і техніки. Структура і правила оформлення” [Текст] – Державний стандарт України.
2. Мэтью Мак-Дональд Silverlight 3 с примерами на C# для профессионалов М. «Вильямс», 2010– 560с.
3. Бен Ватсон. М. «Вильямс», 2010 C# 4.0 на примерах. – 610с

Інформаційні ресурси

1. .NET Framework [Електронний ресурс]. – Режим доступу: [https://msdn.microsoft.com/ru-ru/library/w0x726c2\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx) – Назва з екрана.
2. Павловский В.Е. Задачи динамики и управления мобильными роботами [Електронний ресурс]. – Режим доступу: <http://posp.raai.org/data/posp2005/SIR/Pavlovsky/pavlovsky.html>
3. ЕНК по дисципліні «Програмна технологія .NET» знаходиться за електронною адресою: <http://elearn.nubip.edu.ua/course/view.php?id=876>

ДОДАТОК В

Зразок складання календарного плану виконання курсового проекту

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів курсової роботи	Термін виконання етапів роботи	Підписи керівника, студента
1	Отримання теми	10.09.23	
2	Узгодження постановки задачі з керівником	19.09.23	
3	Пошук та вивчення джерел з питань курсової роботи	20.09.23	
4	Розробка сценарію роботи програми	28.09.23	
6	Узгодження сценарію з керівником	01.10.23	
5	Розробка алгоритму рішення задачі	04.10.23	
6	Узгодження алгоритму з керівником	06.10.23	
7	Узгодження з керівником інтерфейсу користувача	28.10.23	
8	Розробка інформаційного забезпечення	03.11.23	
9	Розробка програмного забезпечення	10.11.23	
10	Тестування програми	25.11.23	
14	Підготовка пояснювальної записки	26.11.23	
15	Здача курсового проекту (роботи) на перевірку	01-12.12.23	
16	Захист	14-16.12.23	

Студент _____
(підпис)

Керівник _____
(підпис)

(прізвище, ім'я, по батькові)

« _____ » _____ 2023 р

**Завдання для курсових робіт з дисципліни
«Програмна технологія .NET»**

1. Спроекувати та реалізувати систему класів, в основу якої покладено логічну структуру даних, наведену у варіанті, для обробки даних відповідно варіанту курсового проекту.
2. Перед виходом із застосування накопичені дані зберегти у файл, на початку роботи з застосуванням дані завантажити із файлу (якщо дані існують).
3. При написанні коду застосувати правила «хорошого» стилю програмування. А саме: створювати мнемонічні ідентифікатори, чітко формувати код, застосувати при необхідності коментарі. Не використовувати public членів-даних у класах. За необхідності доступу до них використати властивості, індексатори чи окремі методи-аксесори.
4. Пояснювальна записка має містити наступні складові:
 - 1) зміст роботи;
 - 2) аналіз предметній галузі задачі;
 - 3) постановку задачі;
 - 4) структури даних та діаграми класів для розв'язання задачі;
 - 5) схеми взаємодії об'єктів та алгоритми виконання їх методів;
 - 6) опис розробленої програми;
 - 7) контрольний приклад (опис, результати його виконання у вигляді роздруківки);
 - 8) висновки;
 - 9) список літератури;
 - 10) додатки:

Орієнтовний перелік тем (теми узгоджуються з керівником)

1. Розробка програмного конвертора

Визначити інтерфейс `IConvertible`, який вказує, що клас, який його реалізує, може конвертувати блок коду `C#` або `VB` код. В інтерфейсі визначити два методи `ConvertToCSharp` і `ConvertToVB`, кожен з яких приймає і повертає рядок.

Створити клас `ProgramHelper`, який реалізує інтерфейс `IConvertible`. При написанні методів замість перетворення рядка використовувати прості рядкові повідомлення для імітації перетворення.

Створити новий інтерфейс `ICodeChecker`, визначивши в ньому метод `CodeCheckSyntax`, який приймає два рядки: рядок для перевірки та мову, що використовується. Метод має повертати тип `bool`. Додати до класу `ProgramHelper` функціональність нового інтерфейсу `ICodeChecker`

Створити клас `ProgramConverter`, який реалізує інтерфейс `IConvertible`. Змінити клас `ProgramHelper`, успадковуючи його від класу `ProgramConverter` та інтерфейсу `ICodeChecker`.

Протестувати клас, створивши масив об'єктів `ProgramConverter`, одні з яких мають тип `ProgramConverters`, інші - тип `ProgramHelpers`. Потім викликати методи перетворення та перевірки коду для кожного елемента масиву, щоб перевірити, які з них реалізують інтерфейс `ICodeChecker`, а які ні.

2. Розробка системи класів для обробки матриць

Розробити власний тип для роботи з матрицями.

Реалізувати методи, що дозволяють виконувати операції складання, віднімання та множення матриць, передбачивши можливість їх виконання, інакше має генеруватися виняток.

Як UI-інтерфейс використовувати консольний додаток з інтерфейсом командного рядка, `WinForms` або `WPF`-додаток.

Створити `unit`-тести для тестування методів розробленого типу.

3. Розробка додатку для роботи з файлами

Розробити просту WPF-програму, яка може використовуватися для відкриття, відображення, редагування і, якщо це необхідно, збереження зміненого вмісту текстових файлів.

Додаток повинен бути простим у використанні і включати повну обробку винятків.

Створити unit-тести для тестування розробленої програми.

4. Програма для імітації годинника зі зворотним відліком

Розробити клас для імітації годинника зі зворотним відліком, що реалізує можливість після закінчення призначеного часу (кількість секунд очікування, що поставляються користувачем) передавати повідомлення будь-якому типу, що підписався на подію.

Для створення ефекту очікування використовувати метод `Thread.Sleep` простір імен `System.Threading`.

Передбачити можливість підписки на подію у кількох класах.

Для обробки подій використовувати

- анонімні методи;
- лямбда-вирази.

Як UI-інтерфейс використовувати консольний додаток з інтерфейсом командного рядка, WinForms або WPF-додаток.

5. Програмна реалізація алгоритма Евкліда

Розробити тип, що реалізує алгоритм Евкліда для обчислення НОД двох цілих чисел (http://en.wikipedia.org/wiki/Euclidean_algorithm).

Додати до розробленого типу додаткову функціональність у вигляді перевантажених методів обчислення НОД для трьох, чотирьох або п'яти цілих чисел.

Додати до розробленого типу метод, що реалізує алгоритм Стейна (бінарний алгоритм Евкліда) для розрахунку НОД двох цілих чисел (http://en.wikipedia.org/wiki/Binary_GCD_algorithm). Метод повинен приймати

вихідний параметр, що містить значення часу, необхідне виконання розрахунків.

Змінити метод, що реалізує алгоритм Евкліда обчислення НОД двох параметрів, додавши аналогічну функціональність.

Розробити метод для відображення результатів роботи із використанням гістограм. Метод повинен приймати як два обов'язкові параметри час виконання, і як необов'язкові параметри - орієнтацію гістограм (вертикальна або горизонтальна) і колір для відображення її смуг.

Створити unit-тести для тестування розроблених методів.

Семувати перевищення ліміту часу виконання unit-тесту.

Отримати результати аналізу покриття unit-тестами коду, що тестується.

Як UI-інтерфейс використовувати консольний додаток з інтерфейсом командного рядка, WinForms або WPF-додаток.