

Міністерство освіти і науки України
Національний університет водного господарства та
природокористування
Навчально-науковий інститут автоматики, кібернетики та
обчислювальної техніки
Кафедра комп'ютерних технологій та економічної
кібернетики

04-05-58М

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи з навчальної дисципліни
«Програмування»
для здобувачів вищої освіти першого (бакалаврського) рівня
за освітньо-професійними програмами «Інформаційні
системи та технології» спеціальності 126 «Інформаційні
системи та технології»; «Комп'ютерні технології»
спеціальності 015.10 «Професійна освіта. Комп'ютерні
технології»; «Цифрові технології дистанційної освіти»
спеціальності 015.39 «Професійна освіта. Цифрові технології»
денної та заочної форми навчання

Рекомендовано науково-
методичною радою
з якості ННІ АКOT
Протокол № 1 від 11.11.2021 р.

Рівне – 2021

Методичні вказівки до виконання курсової роботи з навчальної дисципліни «Програмування» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійними програмами «Інформаційні системи та технології» спеціальності 126 «Інформаційні системи та технології»; «Комп'ютерні технології» спеціальності 015.10 «Професійна освіта. Комп'ютерні технології»; «Цифрові технології дистанційної освіти» спеціальності 015.39 «Професійна освіта. Цифрові технології» денної та заочної форми навчання [Електронне видання] / Грицюк П. М., Шевченко І. М. – Рівне : НУВГП, 2021. – 36 с.

Укладачі:

Грицюк П. М., д.екон.н., професор, завідувач кафедри комп'ютерних технологій та економічної кібернетики;

Шевченко І. М., старший викладач кафедри комп'ютерних технологій та економічної кібернетики.

Відповідальний за випуск:

Грицюк П. М., д.екон.н., професор, завідувач кафедри комп'ютерних технологій та економічної кібернетики.

Керівник (гарант) ОП «інформаційні системи та технології» спеціальності 126 «Інформаційні системи та технології» Гладка О. М., доцентка кафедри комп'ютерних технологій та економічної кібернетики, к.т.н., доцентка.

Керівник (гарант) ОП «Комп'ютерні технології» спеціальності 015.10 «Професійна освіта. Комп'ютерні технології» Парфенюк О. В., к.пед.н., старший викладач кафедри комп'ютерних технологій та економічної кібернетики.

Керівник (гарант) ОП «Цифрові технології дистанційної освіти» спеціальності 015.39 «Професійна освіта» Рощенюк А. М., к.пед.н., доцентка кафедри комп'ютерних наук та прикладної математики.

© Грицюк П. М. Шевченко І. М., 2021
© НУВГП, 2021

ЗМІСТ

ВСТУП.....	3
МЕТА КУРСОВОЇ РОБОТИ.....	3
ТЕМАТИКА КУРСОВИХ РОБІТ.....	4
ВИМОГИ ДО ПРОГРАМИ.....	17
СТРУКТУРА ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....	18
РЕКОМЕНДАЦІЇ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ ...	19
ЗАХИСТ КУРСОВОЇ РОБОТИ ТА КРИТЕРІЇ ЇЇ ОЦІНЮВАННЯ.....	27
РЕКОМЕНДОВАНА ЛІТЕРАТУРА.....	30
ДОДАТКИ.....	33

ВСТУП

Курсова робота – це перша самостійна наукова праця майбутнього фахівця з інформаційних технологій. При її виконанні студент поглиблює знання з фундаментальних та професійно-орієнтованих дисциплін, набуває навичок використання інформаційних і комунікаційних технологій, засвоює методику експериментальних досліджень, оволодіває навичками співставлення результатів своїх досліджень із літературними даними, аналізу, узагальнення і літературного оформлення одержаних результатів з теми дослідження, набуває вміння вести науковий пошук, що розвиває в молодого фахівця творчий підхід до роботи.

Курсова робота призначена для практичного засвоєння студентами основних розділів дисципліни «Програмування», закріплення принципів розробки комп'ютерних програм, знань мови програмування C#, розвитку практичних навичок програмування прикладних додатків.

В результаті виконання курсової роботи студент повинен здобути навички вирішення практичних завдань, пов'язаних з розробкою програм на C#, навчитися працювати з науково-технічною літературою в галузі програмування, а також використовувати стандарти, довідники з програмного забезпечення ЕОМ.

Курсова робота з програмування передбачає знання студентами об'єктно-орієнтованого проектування та програмування і є завершальним етапом вивчення у межах курсу та виконується здобувачами вищої освіти першого (бакалаврського) рівня протягом 3 семестру

МЕТА КУРСОВОЇ РОБОТИ

Метою курсової роботи є закріплення, поглиблення та узагальнення знань, якими студент оволодів під час вивчення курсу «Програмування», в набутті навичок використання основ алгоритмізації та програмування на

алгоритмічних мовах високого рівня з використанням принципів об'єктно-орієнтованого проектування та програмування, методики розробки програмних модулів та програмного забезпечення комп'ютерних систем управління, набуття студентами теоретичних знань та практичних навичок в області використання сучасних систем візуального проектування програмних засобів комп'ютерних систем управління та освоєння принципів та методів сучасних технологій програмування, набуття навичок професійних та практичних наукових досліджень з оформленням відповідного текстового, програмного та ілюстративного матеріалу у формі наукового звіту.

Основним завданням КР з дисципліни «Програмування» є:

- засвоєння концепцій об'єктно-орієнтованого програмування;
- вибір і обґрунтування необхідності та доцільності створення додатку, визначення його функціональності;
- застосування основ уніфікованої мови моделювання UML під час проектування додатку.
- розроблення алгоритмів;
- використання спеціалізованих засобів розробки програмного забезпечення мовою C#;
- підготовка необхідної проектної і організаційно-технологічної документації.

ТЕМАТИКА КУРСОВИХ РОБІТ

Тематика курсових робіт повинна відповідати завданням навчальної дисципліни і тісно пов'язуватися з практичними потребами фаху.

Тема курсової роботи може бути обрана (або запропонована) самим студентом відповідно до своїх інтересів за погодженням з викладачем. В ході обговорення тема курсової роботи може бути уточнена. Якщо студенти не може обрати тему самостійно, то викладач сам призначає її на власний розсуд.

У курсовій роботі має бути розроблена комп'ютерна програма на мові C# з використанням засобів об'єктно-орієнтованого програмування. У розробленій програмі мають бути оголошені не менше 6 об'єктів. Між об'єктами повинні підтримуватися різні типи зв'язків. У тому числі відношення – спадкоємство з глибиною не менш 3-х.

Курсова робота має розроблятися з використанням Windows Forms та її компонентної бази. Інструментальним засобом при розробці програми може служити пакет Microsoft Visual Studio.

Початковими даними для виконання курсової роботи є особливості предметної області, специфікація мови C#, особливості інструментального середовища C#.

Орієнтовна тематика курсових робіт:

1. Описати клас, що реалізує стек. Написати програму, що використовує цей клас для моделювання T-подібного сортувального вузла на залізниці. Програма повинна розділяти на два напрями склад, що складається з вагонів двох типів (на кожен напрям формується склад з вагонів одного типу). Передбачити можливість формування складу з файлу і з клавіатури.

2. Описати клас, що реалізує бінарне дерево, яке має можливість додавання нових елементів, видалення існуючих, пошуку елемента по ключу, а також послідовного доступу до усіх елементів. Написати програму, що використовує цей клас для представлення англо-російського словника. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу. Передбачити можливість формування словника з файлу і з клавіатури.

3. Побудувати систему класів для опису плоских геометричних фігур: круг, квадрат, прямокутник. Передбачити методи для створення об'єктів, переміщення на площині, зміни розмірів і обертання на заданий кут. Написати програму, що демонструє роботу з цими класами. Програма повинна містити меню, що дозволяє здійснити перевірку усіх методів класів.

4. Побудувати опис класу, що містить інформацію про поштову адресу організації. Передбачити можливість роздільної зміни складових частин адреси, створення і знищення об'єктів цього класу. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

5. Скласти опис класу для представлення комплексних чисел. Забезпечити виконання операцій складання, віднімання і множення комплексних чисел. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

6. Скласти опис класу для об'єктів-векторів, що задаються координатами кінців в тривимірному просторі. Забезпечити операції складання і віднімання векторів з отриманням нового вектору (суми або різниці), обчислення скалярного добутку двох векторів, довжини вектору, косинуса кута між векторами. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку усіх методів класу.

7. Скласти опис класу прямокутників із сторонами, паралельними осям координат. Передбачити можливість переміщення прямокутників на площині, зміни розмірів, побудови найменшого прямокутника, що містить два задані прямокутники, і прямокутники, що є загальною частиною (перетином) двох прямокутників. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

8. Скласти опис класу для визначення одновимірних масивів цілих чисел (векторів). Передбачити можливість звернення до окремого елемента масиву з контролем виходу за межі масиву, можливість завдання довільних меж індексів при створенні об'єкту і виконання операцій поелементного складання і віднімання масивів з однаковими межами індексів, множення і ділення всіх елементів масиву на скаляр, виводу на екран елементу масиву по заданому індексу і всього масиву. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

9. Скласти опис класу для визначення одновимірних масивів рядків фіксованої довжини. Передбачити контроль виходу за межі масиву, можливість звернення до окремих рядків масиву по індексах, виконання операцій поелементного зчеплення двох масивів з утворенням нового масиву, злиття двох масивів з виключенням елементів, що повторюються, а також вивід на екран елементу масиву по заданому індексу і всього масиву. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

10. Скласти опис класу многочленів від однієї змінної, що задаються мірою многочлена і масивом коефіцієнтів. Передбачити методи для обчислення значення многочлена для заданого аргументу, операції складання, віднімання і множення многочленів з отриманням нового об'єкту-многочлена, вивід на екран опису многочлена. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

11. Скласти опис класу одновимірних масивів рядків, кожен рядок задається завдовжки і покажчиком на виділену для неї пам'ять. Передбачити контроль виходу за межі масивів, можливість звернення до окремих рядків масиву по індексах, виконання операцій поелементного зчеплення двох масивів з утворенням нового масиву, злиття двох масивів з виключенням елементів, що повторюються, а також вивід на екран елементу масиву і всього масиву. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

12. Скласти опис класу, що забезпечує представлення матриці довільного розміру з можливістю зміни числа рядків і стовпців, виводу на екран підматриці будь-якого розміру і всієї матриці. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

13. Написати клас для ефективної роботи з рядками, що дозволяє форматовувати і порівнювати рядки, зберігати в рядках числові значення і витягати їх. Для цього необхідно реалізувати:

- первантажені оператори привласнення і конкатенації;

- операції порівняння і приведення типів;
- перетворення в число будь-якого типу;
- форматне виведення рядка.

Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

14. Описати клас «домашня бібліотека». Передбачити можливість роботи з довільною кількістю книг, пошуку книги з якої-небудь ознаки (наприклад, за автором або за роком видання), додавання книг у бібліотеку, видалення книг з неї, сортування книг з різних полів. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

15. Описати клас «записник». Передбачити можливість роботи з довільною кількістю записів, пошуку запису за якою-небудь ознакою (наприклад, прізвище, дата народження або номер телефону), додавання і видалення записів, сортування по різних полях. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

16. Описати клас «студентська група». Передбачити можливість роботи зі змінним числом M студентів, пошуку студента за якою-небудь ознакою (наприклад, прізвище, дата народження або номер телефону), додавання і видалення записів, сортування по різних полях. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

17. Описати клас, що реалізує тип даних «дійсна матриця» і роботу з ними. Клас повинен реалізовувати наступні операції над матрицями:

- складання, віднімання, множення, ділення (множення і ділення як на іншу матрицю, так і на число);
- комбіновані операції присвоювання ($+=$, $-=$, $*=$, $/=$);
- операції порівняння на рівність/нерівність;
- операції обчислення зворотної і транспонованої матриці, операцію піднесення до степеня;
- методи обчислення детермінанта і норми;
- методи, що реалізують перевірку типу матриці (квадратна, діагональна, нульова, одинична, симетрична, верхня трикутна, нижня трикутна);
- операції введення/виводу в стандартні потоки.

Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

18. Описати клас «множина», що дозволяє виконувати основні операції – додавання і видалення елемента, перетин, об'єднання і різницю великих кількостей. Написати програму, що демонструє роботу з цим класом.

Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класу.

19. Описати клас, що реалізує стек. Написати програму, що використовує цей клас для відшукування проходу по лабіринту. Лабіринт представляється у вигляді матриці, що складається з квадратів. Кожен квадрат або відкритий, або закритий. Вхід в закритий квадрат заборонений. Якщо квадрат відкритий, то вхід в нього можливий з боку, але не з кута. Кожен квадрат визначається його координатами в матриці. Після відшукування проходу програма друкує знайдений шлях у вигляді координат квадратів.

20. Створити клас CFile, що інкапсулює в собі такі функції роботи з файлами, як Open, Close, Seek, Read, Write, GetPosition і GetLength. На базі цього класу створити похідний клас CMyDataFile — файл, що містить в собі дані деякого певного типу MyData, а також заголовок, що полегшує доступ до цього файлу. Написати програму, що демонструє роботу з цими класами. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класів.

21. Створити клас CPoint — точка. На його основі створити класи CcoloredPoint і CLine. На основі класу CLine створити клас CColoredLine і клас CPolyLine — багатокутник. Усі класи повинні мати методи для установки і набуття значень усіх координат, а також зміни кольору і отримання поточного кольору. Написати демонстраційну програму, в якій використовуватиметься список об'єктів цих класів в динамічній пам'яті.

22. Створити абстрактний клас CVehicle. На його основі реалізувати класи CPlane, CCar і CShip. Класи повинні мати можливість задавати і отримувати координати, параметри засобів пересування (ціна, швидкість, рік випуску). Для літака має бути визначена висота, для літака і корабля — кількість пасажирів. Для корабля — порт приписки. Написати програму, що створює список об'єктів цих класів в динамічній пам'яті. Програма повинна містити меню, що дозволяє здійснити перевірку всіх методів класів.

23. Описати базовий клас «Елемент». Поля:

- ім'я елемента (покажчик на рядок символів);
- кількість входів елемента;
- кількість виходів елемента.

Методи:

- конструктор класу;
- деструкція класу;
- метод, який задає ім'я елемента.

На основі класу «Елемент» описати похідний клас «Комбінаційний», що є комбінаційним елементом (двійковий вентиль), який може мати декілька входів і один вихід.

Поля:

- покажчик, використовуваний для динамічного розміщення полів, входів, що містять значення.

Методи:

- конструктор;
- конструктор копіювання;
- деструкція;
- метод, що задає значення на входах екземпляра класу;
- метод, що дозволяє опитувати стан окремого входу екземпляра класу;
- метод, що обчислює значення виходу.

24. Моделювання роботи бухгалтерії на фірмі

Створити програму, що дозволяє моделювати роботу бухгалтерії на фірмі.

Визначити класи

”Працівник”(абстрактний), в якому визначені наступні дані: прізвище, ім’я, по-батькові, стаж роботи)

“Програміст”(кількість відпрацьованих годин) «Лідер»(список людей, що перебувають в підпорядкуванні).

Фірма складається з програмістів та лідерів. Їх може бути довільна кількість.

Для всіх працівників фірми нараховується заробітна плата, яка залежить від посади. Для програміста вона нараховується згідно норми оплати за годину. Але якщо програміст відпрацював менше годин на день, ніж призначено(теж існує відповідна норма) – нараховується пеня, що обчислюється як заданий відсоток. Для лідерів - заробітна плата нараховується, як оклад + відсоток від заробітної плати кожного працівника, що перебуває в його підпорядкуванні. Також можливі разові штрафні санкції, які може задавати керівництво. Для всіх працівників існує надбавка 10% за кожні 10 років стажу.

Потрібно створити наступні сервіси:

1. Наповнення фірми конкретними працівниками.
2. Нарахування заробітної плати для всіх працівників.
3. Визначення працівників, що мають максимальну та мінімальні заробітну плату.
4. Сортування працівників за різними критеріями: прізвищем, посадою, заробітною платою.
5. Долучення та вилучення працівників до списку.
6. Збір інформації за різні часові інтервали.

25. Модель абонентської телефонної станції

Створити програму, що створює модель абонентської телефонної станції.

Програма передбачає роботу з абонентами, які мають персональні дані: прізвище, ім’я, по-батькові, адресу, номер телефону, абонентську плату.

Також передбачено, що існують абоненти-юридичні особи, в яких на одну адресу може бути зафіксовано кілька телефонних номерів. Оплата за послуги нараховується згідно визначених тарифів і є різною для юридичних і неюридичних абонентів. В юридичних абонентів існує пільга за кожні 5 номерів -10%.

Потрібно створити наступні сервіси:

1. Наповнення телефонної станції конкретними номерами.
2. Фіксація телефонних дзвінків, здійснених абонентами.
3. Нарахування абонентської плати та здійснення аналізу боргів.
4. Сортування інформації за різними критеріями: прізвищем, адресою, номером, абонентською платою, боргом.
5. Долучення та вилучення абонентів.

26. Модель складу продовольчих та непродовольчих товарів

Створити програму, що створює модель складу продовольчих та непродовольчих товарів.

Програма передбачає роботу з продовольчими товарами, які мають назву, вагу, вартість, кількість, термін придатності, виробника. Непродовольчі товари мають наступні характеристики: назву, вартість, кількість, виробника, габаритні розміри, групу призначення. Для всіх товарів існує функція зміни вартості, яка передбачає зниження вартості у випадку, якщо термін придатності вичерпаний на 80% для продовольчих товарів, додаткову надбавку за перевищення норм габаритів – у випадку непродовольчих товарів.

1. Наповнення складу конкретними товарами.
2. Розподіл товарів за різними групами.
3. Пошук товарів за різними критеріями.
4. Зміну вартості всіх товарів.
5. Сортування інформації за різними критеріями.
6. Долучення та вилучення нових товарів.
7. Визначення унікальних товарів, які не мають повторів.
8. Визначення сумарної кількості заданої назви.

27. Модель поштового сервісу.

Створити програму, що відтворює модель поштового сервісу.

Програма передбачає роботу з відправниками та отримувачами повідомлень.

Відправники та отримувачі мають наступні характеристики: ім'я, місце реєстрації, електронну адресу скриньки. Поштове повідомлення має отримувача та відправника, текст повідомлення, розмір, час відправлення та отримання.

Потрібно створити наступні сервіси:

1. Наповнення поштового сервісу користувачами.

2. Здійснення відправки повідомлення.
3. Сортування повідомлень за розміром.
4. Пошук повідомлень заданого відправника.
5. Знаходження користувачів, що не відправляли жодного повідомлення.

6. Роздрук адреси останнього відправленого повідомлення.

28. Модель роботи агентства з продажу нерухомості

Створити програму, що відтворює модель продажу нерухомості.

Передбачається, що нерухомість є 2 типів: житлові та нежитлові приміщення. Всі приміщення мають загальну площу, кількість кімнат, адресу, власника, вартість. Житлові приміщення мають площі кухні, кімнат, стан ремонту. Нежитлові приміщення – ще додатково мають цільове призначення.

Потрібно створити наступні сервіси:

1. Наповнення магазину інформацією.
2. Здійснення пошуку приміщення за заданими параметрами.
3. Сортування приміщень за цільовим призначенням, адресою, кількістю кімнат.
4. Визначення приміщень, що мають максимальну та мінімальну площу.
5. Пошук власників, які зареєстрували більше 2 приміщень для продажу.
6. Долучення та вилучення нових приміщень.
7. Обчислення загальної суми проданої нерухомості.

29. Модель організації свята Миколая

Створити програму, що відтворює модель організації свята Миколая.

Дорослі користувачі можуть надсилати повідомлення, в яких вказувати ім'я дитини, її вік, стать, кількість хороших і поганих вчинків, тип подарунку, який хоче отримати дитина: їстівний, чи неїстівний. Їстівні подарунки можуть бути кількох видів: цукерки, печиво чи шоколад. Неїстівні подарунки можуть бути 2 видів: машинки - для хлопчиків та ляльки - для дівчаток. А також серед подарунків може бути різка. В залежності від кількості добрих і поганих вчинків Миколай формує подарунки.

Потрібно створити наступні сервіси:

1. Наповнення інформації про дітей.
2. Формування подарунку відповідно до статі дитини та її поведінки.
3. Визначити перелік адресатів і вилучити повідомлення, які поступали на ім'я дитини кілька разів, залишивши тільки останнє.
4. Долучення та вилучення нових повідомлень.
5. Обчислення загальної кількості подарунків кожного виду.

6. Визначення наймолодших дітей.

30. Модель організації роботи готелю

Створити програму, що відтворює модель організації роботи готелю.

Користувач має мати змогу зняти в користування простий номер, чи номер люкс.

В готелі може бути наповнення номерами різного класу. Номер має містити інформацію про кількість кімнат, кількість місць, кількість вікон, площу, наявність побутової техніки, дати бронювання, вартість.

Люксівський номер має мати також список додаткових зручностей.

При бронюванні номера фіксується особу власника.

Потрібно створити наступні сервіси:

1. Наповнення інформації про номери.
2. Пошук вільних номерів на вказані дати.
3. Пошук номерів за вказаним користувачем діапазоном вартості.
4. Визначення користувача, який проживає в готелі найдовше.
5. Визначення користувачів, які забронювали більше 2 номерів.
6. Знаходження найпопулярнішого номеру в діапазоні заданих дат.

31. Модель роботи залізничної каси

Створити програму, що відтворює модель роботи залізничної каси.

У касі зафіксовані маршрути, в яких вказано станція відправлення, станція прибуття та проміжні станції з їх віддаленістю в км від станції відправлення, а також час початку подорожі та прибуття на кожну з станцій. Також до маршруту прикріплений транспортний засіб, який обслуговує його з визначеною кількістю місць.

Пасажира може купувати квиток на довільну проміжну станцію маршруту.

У випадку можливості задовольнити запит, пасажиру видається квиток з вказівкою станції прибуття, часу прибуття, вартості, обчисленої згідно норми за км пробігу. При цьому фіксується прізвище пасажира.

Потрібно створити наступні сервіси:

1. Наповнення інформацією про маршрути.
2. Пошук вільних місць на найближчий час за заданим користувачем маршрутом.
3. Аналіз частоти зайнятості маршрутів.
4. Визначення найдовшого маршруту.
5. Пошук власника проданого квитка за прізвищем.
6. Повернення квитка.
7. Обчислення прибутку каси.

32. Модель роботи аукціону картин

Створити програму, що відтворює модель роботи аукціону картин.

На аукціон можуть бути подані картини, про яких фіксуються наступні дані:

Назва картини, її автор, власник, жанр, вартість.

Учасник аукціону може зареєструватися, зафіксувавши реквізити рахунку та ім'я та оформити заявку на картину.

Потрібно створити наступні сервіси:

1. Наповнення інформацією про картини.
2. Надання можливості отримання заявки та її відміни.
3. Перевірка платоспроможності заявника у випадку оформлення кількох заявок.
4. Сортуння картин за жанром, автором, частотою оформлення заявки.
5. Пошук заявника, що підписався на найбільшу суму.
6. Визначення унікальних картин.
7. Повернення квитка.

33. Модель роботи поліклініки

Створити програму, що відтворює модель роботи поліклініки. У закладі є перелік лікарів заданих профілів. Відомі їх прізвища, стаж, звання та години прийому в кожний день тижня. Пацієнт може зареєструватися, вказавши прізвище, вік, стать. Він має можливість обрати лікаря та вільний час прийому.

Потрібно створити наступні сервіси:

1. Наповнення інформацією про лікарів.
2. Створення форми заявки пацієнта.
3. Створення звіту відвідуваності, у випадку зайнятості всіх лікарів – створення списку не виконаних заявок .
4. Визначення найпопулярнішого лікаря.
5. Створення для лікаря списку пацієнтів на тиждень, а також створення для пацієнта можливого списку лікарів, які він планує відвідати на тижні.

34. Модель навчання в приватній школі

Створити програму, що відтворює модель навчання в приватній школі. У закладі є перелік викладачів та дисциплін, які вони викладають. Відомі прізвища викладачів, їх посада та стаж роботи.

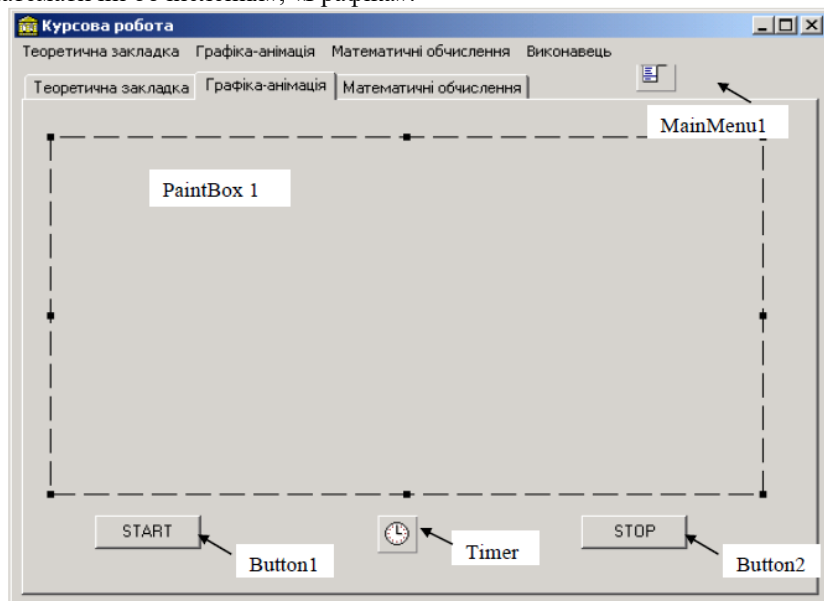
Студент, що має прізвище, номер курсу та групи, в якій навчається, та перелік дисциплін, відповідно до курсу. Він має мати змогу вибрати викладача, що може викладати цю дисципліну.

Потрібно створити наступні сервіси:

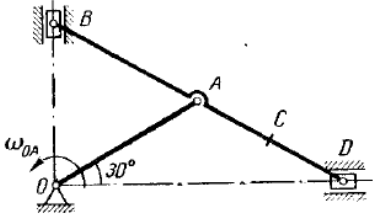
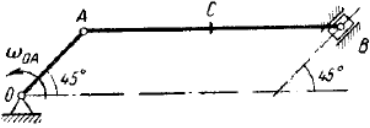
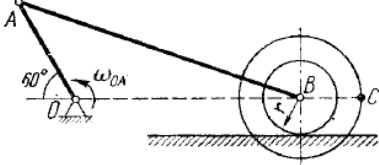
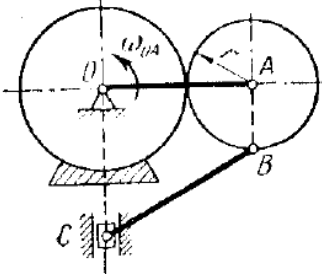
1. Наповнення інформацією про викладачів та дисципліни.

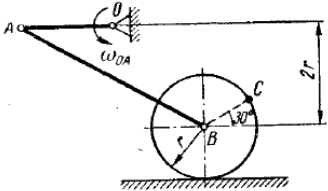
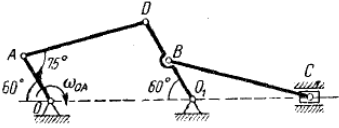
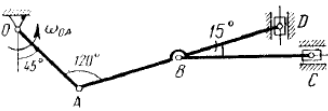
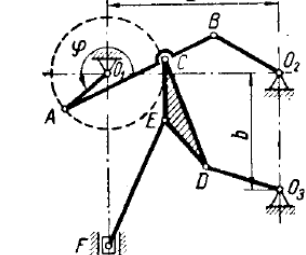
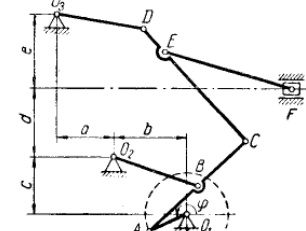
2. Можливість вибору викладачів за дисциплінами.
3. Можливість перегляду для викладачів студентів курсу та всіх студентів, що записалися за курсами викладача.
4. Визначення найпопулярнішого викладача за заданою дисципліною.
5. Сортунання списку викладачів за посадою та прізвищем.
6. Визначити всіх викладачів, що не мають тільки один курс.

35. Створення додатку, якій містить 3 вкладки: «Теоретичні відомості», «Математичні обчислення», «Графіка»:



Завдання для вкладок: (графіка орієнтовна. Студент може запропонувати свій варіант)

№ аріанту	Компонент	Математичний вираз	Схема для анімації
1	2	3	4
1	OpenDialog	$y = \frac{\cos(x) + 1}{\sin(x) \cdot \cos(x)}$ $y = x + \cos(x)$	
2	Chart	$y = \sin(x + 1)$ $y = 3x \cdot \sin x$	
3	PopupMenu	$y = \sin(2x) + 3$ $y = x \cdot 2 \sin x$	
4	MainMenu	$y = \sin(x) \cos(x) + 1$ $y = \ln(x^3)$	

5	RadioButton	$y = \frac{1}{1 + \sqrt{x}}$ $y = \frac{\sqrt[3]{x}}{\sqrt[4]{x}}$	
6	RichEdit	$y = (x+1)\sqrt{x^2+1}$ $y = x^5 + x^4$	
7	ComboBox	$y = \frac{1}{x \cdot \ln^2 x}$ $y = \frac{x^4}{x^3}$	
8	ProgressBar	$y = \frac{1}{(x+1) + \sqrt{x}}$ $y = \frac{\cos(x)}{(x+1)^4}$	
9	Timer	$y = \arctg\left(\frac{\sqrt{x}}{\sqrt{x(1-x)}}\right)$ $y = (x^2 - 1)^{\frac{1}{3}}$	

36. Утворити інформаційно-пошукову систему як файл (або зв'язані файли) з записами фіксованої структури будь-якого наповнення:

- Бібліотека
- Деканат
- Записник
- Склад

Передбачити можливість створення нового запису, редагування вмісту існуючого, видалення, сортування та пошуку за однією або декількома ознаками, виведення потрібної інформації у вигляді звіту певного змісту.

37. Навчальні програми та програми тестування.

Програми повинні містити частини: теоретичні відомості, власне навчання., тестування з окремого розділу або теми.

Приклади тем:

1. Системи числення та робота з ними
2. Навчальна програма роботи з клавіатурою.
3. Навчальна програма роботи в обраному програмному середовищі.
4. Програма тестування з обраної області.
5. Психологічні тести.

Психологічні тести повинні містити реалізацію декількох різних тестів, бажано за різними принципами (відповідь на запитання, вибір із можливих варіантів).

38. Створення спеціалізованих математичних пакетів.

Програми повинні містити частини: теоретичні відомості, можливість вибору варіантів розв'язування задачі. Приклади тем:

- Пакет роботи з векторами та матрицями.
- Пакет розв'язку систем лінійних рівнянь (різними методами).
- Пакет для обчислення визначення визначеного інтегралу (різними методами).
- Мінімізація булевих виразів.

39. Програми для роботи зі спеціальними структурами даних.

Програми повинні містити можливість побудови та візуалізації усїєї структури, а також пошуку, перестановки, видалення окремих елементів або частин. Для деяких програм варто передбачити реалізацію класичних алгоритмів, зв'язаних з даною структурою. Приклади тем:

- Однозв'язні списки та робота з ними.
- Двозв'язні списки та робота з ними.
- Бінарні дерева та робота з ними.
- Орієнтовані та неорієнтовані графи. Їх обробка.

40. Ігрові програми.

ВИМОГИ ДО ПРОГРАМИ

Програма повинна відповідати наступним вимогам:

1. Програма повинна розроблятися на C# з використанням Windows Forms.

2. Кількість класів не менше 6.
3. Наявність зв'язків між класами, у тому числі стосунків спадкоємства завглибшки не менше 3.
4. Для ініціалізації об'єктів повинні використовуватися конструктори різних видів.
5. У програмі повинні використовуватися віртуальні функції.
6. Повинен використовуватися механізм перевантаження функцій і операторів.
7. Слід використовувати абстрактні класи.
8. У класах повинні використовуватися різні види доступу до полів і методів (public, private, protected).

СТРУКТУРА ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Пояснювальна записка до курсової роботи складається зі вступу, основної частини, висновків, списку літератури, додатків. Обсяг пояснювальної записки 20-30 сторінок (не включаючи додатки).

Титульний аркуш повинен відповідати встановленому зразку (**додаток А**).

Зміст включає найменування всіх розділів курсової роботи, а також підрозділів і пунктів. Для кожного розділу, підрозділу і пункту вказується номер сторінки. Приклад змісту наведений в **додатку Б**.

Вступ має містити такі елементи: актуальність, мету, завдання, практичне значення, використане програмне забезпечення, опис структури роботи.

Основна частина складається з трьох розділів: постановка задачі, розробка програмного продукту та опис розробленого програмного продукту.

В *першому розділі* проводиться аналіз предметної області, огляд існуючих підходів до розв'язання поставленої задачі, аналізуються алгоритми, здійснюється детальний опис постановки задачі та обґрунтовується обраний варіант реалізації. Постановка задачі може здійснюватися в різний спосіб: математично або словесно (опис правил гри, функціональність інформаційної системи, тощо). Також висуваються вимоги до програмних та технічних засобів.

В *другому розділі* потрібно описати, в залежності від теми, використовуваний математичний апарат, фізичне явище, основні структури даних та алгоритми розв'язку. Цей розділ також має містити опис загального алгоритму роботи програми, UML-діаграми класів та їх взаємодій.

В *третьому розділі* міститься опис основної програми і користувацьких програмних модулів. Всі створені функції, структури даних та змінні повинні бути описані відповідно до заголовкових файлів. Приклад оформлення таблиці дивиться в **додатку В**. Потрібно надавати опис всіх розроблених функцій. Опис кожної функції містить її прототип, призначення та семантику кожного параметра під час виклику. В підрозділі «Опис інтерфейсу» описується дії користувача під час роботи з продуктом. Описується робота з усіма меню та підменю програмного продукту. В підрозділі «Результати роботи програмного продукту» описуються результати тестування програми, виявлені недоліки та рекомендації до використання.

Висновки містять стисле викладення підсумків дослідження: що і як вдалося реалізувати, переваги та недоліки розробки, шляхи подальшого вдосконалення.

Список літератури містить перелік джерел, що були використані при виконанні курсової роботи. Вказуються тільки ті джерела, на які є посилання в тексті пояснювальної записки.

Додаток містить допоміжний матеріал (лістинг програм, інструкції з користування і тому подібне).

РЕКОМЕНДАЦІЇ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Робота повинна бути написана державною мовою.

Матеріал пояснювальної записки повинен бути викладений грамотно, чітко та стисло. При цьому в тексті мають бути обов'язковими посилання на використані літературні та інші джерела.

У тексті пояснювальної записки не рекомендується вживати звороти із займенниками першої особи, наприклад: «Я вважаю», «Ми вважаємо ...» тощо. Рекомендується вести виклад, не вживаючи займенників, наприклад: «Вважаємо знаходимо ...» тощо.

Пояснювальна записка до курсової роботи виконується на аркушах білого паперу (з одного боку) формату А4 (210 x 297 мм) за формами 5 і 5а (ГОСТ 2.106-68), згідно вимог ГОСТу 2.105-79 та ДСТУ 3008-95, українською мовою із застосуванням друкуючих і графічних пристроїв виведення ПК (ГОСТ 2.004-88).

Обсяг пояснювальної записки повинен складати 20-30 сторінок машинописного тексту (без урахування додатків), надрукованого через 2 інтервали (що відповідає міжрядковому комп'ютерному інтервалу - 1,5) (до 30 рядків на аркуші А4).

Основні вимоги до комп'ютерного набору тексту курсової роботи:

текстовий редактор	WORD
гарнітура шрифту	Times New Roman
кегель шрифту (розмір)	14
кількість символів у рядку	не менше 60
Абзацний відступ для першого рядка	1,2-1,27 см
міжрядковий інтервал	полоторний (точно 24)
кількість рядків на сторінці	до 30
Параметри сторінки:	
поля верхнє і нижнє	по 2 см
Праве поле	1,5 см
Ліве поле	3 см
вирівнювання:	
слова “ЗМІСТ”, “ВСТУП”, НАЗВИ РОЗДІЛІВ, “ВИСНОВКИ”, “СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ”, “ДОДАТКИ”	по центру
назви підрозділів	по ширині
назви параграфів (за необхідності)	по ширині
Текст вступу, основної частини, висновків, списку використаних джерел додатків	по ширині
інтервали між структурними частинами	
між словами “ЗМІСТ”, “ВСТУП”, НАЗВАМИ РОЗДІЛІВ, “ВИСНОВКАМИ”, “СПИСКОМ ВИКОРИСТАНИХ ДЖЕРЕЛ”, “ДОДАТКАМИ” і текстом	2 рядки
між назвами підрозділів і текстом	1 рядок
між текстом попереднього підрозділу і назвою наступного підрозділу	2 рядки
між параграфами (за необхідності)	1 рядок
назви структурних частин роботи: “ЗМІСТ”, “ВСТУП”, “РОЗДІЛ”, “ВИСНОВКИ”, “СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ”, “ДОДАТКИ”	Друкують великими літерами (шрифт – 14 напівжирний), вирівнюючи по центру

заголовки підрозділів	друкують маленькими літерами, крім першої – великої, з абзацного відступу, шрифт 14 – напівжирний. Крапку в кінці заголовка не ставлять
структурні частини роботи: “ЗМІСТ”, “ВСТУП”, “РОЗДІЛ”, “ВИСНОВКИ”, “СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ”, “ДОДАТКИ”	Починають з нової сторінки
новий підрозділ	продовжується на тій же сторінці, на якій закінчився попередній

Нумерація структурних частин роботи

Нумерацію сторінок, розділів, підрозділів, пунктів, підпунктів, рисунків, таблиць, формул подають арабськими цифрами **без знака №**. Першою сторінкою курсової роботи є титульний аркуш, який включають до загальної нумерації сторінок. На титульному аркуші номер сторінки не ставлять, на наступних сторінках номер проставляють у правому верхньому куті сторінки без крапки в кінці.

Зміст, перелік умовних позначень, вступ, висновки, список використаних джерел не нумерують. Номер розділу ставлять після слова "РОЗДІЛ", після номера крапку не ставлять, потім з нового рядка друкують заголовок розділу.

Підрозділи нумерують у межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, між якими ставлять крапку. В кінці номера підрозділу повинна стояти крапка, наприклад: «2.3.» (третій підрозділ другого розділу). Потім у тому ж рядку йде заголовок підрозділу.

Пункти нумерують у межах кожного підрозділу. Номер пункту складається з порядкових номерів розділу, підрозділу, пункту, між якими ставлять крапку. В кінці номера повинна стояти крапка, наприклад: «1.3.2.» (другий пункт третього підрозділу першого розділу). Потім у тому ж рядку йде заголовок пункту. Пункт може не мати заголовка. Підпункти нумерують у межах кожного пункту за такими ж правилами, як пункти. Примітки до тексту і таблиць, в яких вказують довідкові і пояснювальні дані, нумерують послідовно в межах однієї сторінки. Якщо приміток на одному аркуші декілька, то після слова «Примітки» ставлять двокрапку.

Якщо є одна примітка, то її не нумерують і після слова «Примітка»

ставлять крапку.

Ілюстрації

Ілюстрації (фотографії, креслення, схеми, графіки, карти) і таблиці необхідно подавати в магістерській роботі безпосередньо після тексту, де вони згадані вперше, або на наступній сторінці. Ілюстрації і таблиці, які розміщені на окремих сторінках курсової роботи, включають до загальної нумерації сторінок. Таблицю, рисунок або креслення, розміри якого більше формату А4, враховують як одну сторінку і розміщують у відповідних місцях після згадування в тексті або у додатках.

Ілюстрації позначають словом “Рис.” і нумерують послідовно в межах розділу, за винятком ілюстрацій, поданих у додатках.

Наприклад:

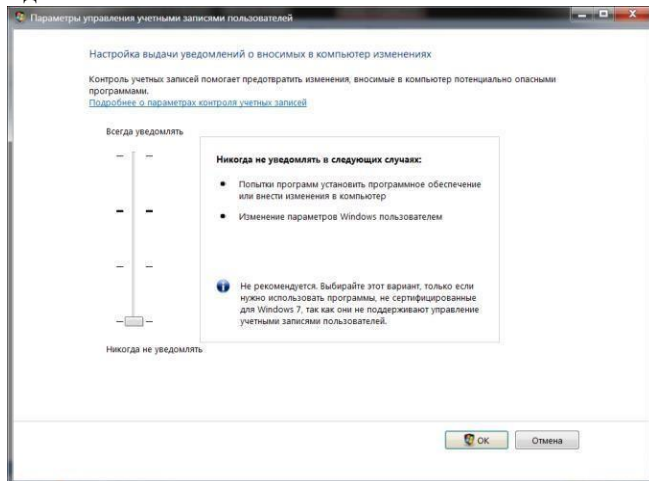


Рис. 1. Настроювання адміністрування параметрів керування користувачів

Номер ілюстрації повинен складатися з номера розділу і порядкового номера ілюстрації, між якими ставиться крапка. Наприклад: Рис.1.2 (другий рисунок першого розділу).

Ілюстрації повинні мати назву, яку розміщують після номера ілюстрації. При необхідності ілюстрації доповнюють пояснювальними даними (підрисунковий текст).

Номер ілюстрації, її назва і пояснювальні підписи розміщують послідовно під ілюстрацією. Якщо в курсовій роботі подано одну ілюстрацію, то її нумерують за загальними правилами.

Фотознімки та інші ілюстрації, розміром меншим за формат А4, можуть

бути наклеєні на стандартні аркуші білого паперу формату А4. Фото та інші ілюстрації доцільно сканувати і вставляти безпосередньо в текст програмою-редактором.

Таблиці

Для систематизації матеріалу та подачі цифрових даних, як правило використовують оформлення у вигляді таблиць.

Таблиці нумерують послідовно (за винятком тих, які подані у додатках) в межах розділу. В правому верхньому куті над відповідним заголовком таблиці розміщують напис «Таблиця» із зазначенням її номера. Номер таблиці повинен складатися з номера розділу і порядкового номера таблиці, між якими ставиться крапка, наприклад: «Таблиця 1.2» (друга таблиця першого розділу).

Якщо в курсовій роботі одна таблиця, її нумерують за загальними правилами.

При перенесенні частини таблиці на інший аркуш (сторінку) слово «Таблиця» і номер її вказують один раз справа над першою частиною таблиці, над іншими частинами пишуть слова «Продовження табл.» і вказують номер таблиці, наприклад: “Продовження табл. 1.2”.

Наприклад:

Таблиця 2.1 Відмінності систем штучного інтелекту (СШІ) від звичайних програмних систем

Характеристи ка програмування	Програмування в СШІ	Традиційне програмування
Тип обробки	Символьна	Числова
Методи	Евристичний пошук	Алгоритм
Задання кроків розв’язання	Неявне	Точне
Шуканий розв’язок	Задовільний	Оптимальний
Управління і дані	Змішані	Роздільні
Знання	Неточні	Точні
Модифікації	Часті	Рідкі

Кожна таблиця повинна мати назву, яку розміщують над таблицею і друкують симетрично до тексту. Назву і слово «Таблиця» починають з великої літери. Назву не підкреслюють.

Заголовки граф повинні починатися з великих літер, підзаголовки – з маленьких, якщо вони складають одне речення із заголовком, і з великих, якщо вони є самостійними. Графу з порядковими номерами рядків до таблиці включати не слід. Таблицю розміщують після першого згадування

про неї в тексті таким чином, щоб її можна було читати без повороту переплетеного блоку курсової роботи або з поворотом за годинниковою стрілкою. Таблицю з великою кількістю рядків можна переносити на інший аркуш. При перенесенні таблиці на інший аркуш (сторінку) назву вміщують тільки над її першою частиною. Таблицю з великою кількістю граф можна ділити на частини і розміщувати одну частину під іншою в межах однієї сторінки. Якщо рядки або графи таблиці виходять за формат сторінки, то в першому випадку в кожній частині таблиці повторюють її головку (заголовки граф), в другому випадку – боковик (перша змістовна графа таблиці).

Якщо текст, який повторюється в графі таблиці, складається з одного слова, його можна замінити лапками; якщо з двох або більше слів, то при першому повторенні його замінюють словами «Те ж», а далі лапками. Ставити лапки замість цифр, марок, знаків, математичних і хімічних символів, які повторюються, не слід. Якщо цифрові або інші дані в якому-небудь рядку таблиці не подають, то в ньому ставлять прочерк.

Формули

Формули в курсовій роботі (якщо їх більше однієї) нумерують у межах розділу. Номер формули складається з номера розділу і порядкового номера формули в розділі, між якими ставлять крапку. Нумери формул пишуть біля правого берега аркуша на рівні відповідної формули в круглих дужках, наприклад: (3.1) (перша формула третього розділу). Наприклад

«Прямолінійне рівняння регресії має вигляд

$$Y=a+bx, \quad 3.1$$

де x – факторна ознака,

Y – результативна ознака,

a – вільний член,

b – (кутовий) коефіцієнт регресії».

Пояснення значень символів і числових коефіцієнтів треба подавати безпосередньо під формулою в тій послідовності, в якій вони дані у формулі. Значення кожного символу і числового коефіцієнта треба подавати з нового рядка. Перший рядок пояснення починають зі слова «де» без двокрапки.

Рівняння і формули треба виділяти з тексту вільними рядками. Вище і нижче кожної формули потрібно залишити не менше одного вільного рядка. Якщо рівняння не вміщується в один рядок, його слід перенести після знака рівності (=) або після знаків плюс (+), мінус (-), множення (\times) і ділення (:).

Посилання

При написанні курсової роботи слід давати посилання на джерела та матеріали, результати з яких наводяться у роботі або на ідеях і висновках яких розроблюються проблеми і задачі курсової роботи. Такі посилання дають змогу відшукати документи і перевірити вірогідність відомостей про цитування документа, дають необхідну інформацію щодо нього, допомагають з'ясувати його зміст, мову тексту, обсяг. Посилатися слід на останні видання публікацій. На більш ранні видання можна посилатися лише в тих випадках, коли в них є матеріал, який не включено до останнього видання.

Якщо використовують відомості, матеріали з монографій, оглядових статей, інших джерел з великою кількістю сторінок, тоді в посиланні необхідно точно вказати номери сторінок, ілюстрацій, таблиць, формул з джерела, на яке дано посилання в курсовій роботі.

Посилання в тексті курсової роботи на джерела слід зазначати порядковим номером за переліком посилань, виділеним двома квадратними дужками, наприклад, «... у працях [1–7]...» або «... [14, с.24]...».

Допускається наводити посилання на джерела у виносках.

Посилання на ілюстрації курсової роботи вказують порядковим номером ілюстрації, наприклад, «рис.1.2».

Посилання на формули курсової роботи вказують порядковим номером формули в дужках, наприклад «... у формулі (2.1)».

На всі таблиці курсової роботи повинні бути посилання в тексті, при цьому слово «таблиця» в тексті пишуть скорочено, наприклад: «...в табл. 1.2».

У повторних посиланнях на таблиці й ілюстрації треба вказувати скорочено слово "дивись", наприклад: "див.табл.1.3".

Заключна частина роботи

Включає в себе висновки, додатки та перелік літературних джерел, використаних при написанні даної роботи.

Логічним завершенням курсової роботи є висновки. Бажано, щоб висновки до курсової роботи були добре продуманими, конкретними, містили пропозиції щодо використання результатів роботи. Головна їх мета – підсумки проведеної роботи.

Завершуючи написання курсової роботи, слід систематизувати ілюстративний матеріал. Ілюстрації можна подавати у тексті або оформляти у вигляді додатків. Усі додатки повинні мати буквену нумерацію та назви, що відповідають їхньому змісту. Нумерація аркушів з додатками продовжує загальну нумерацію сторінок основного тексту роботи.

Додатки оформлюють як продовження курсової роботи на наступних її сторінках або у вигляді окремої частини (книги), розміщуючи їх у порядку

появи посилань у тексті роботи.

Якщо додатки оформлюють на наступних сторінках роботи, кожний такий додаток повинен починатися з нової сторінки. Додаток повинен мати заголовок, надрукований угорі малими літерами з першої великої симетрично відносно тексту сторінки. Посередині рядка над заголовком малими літерами з першої великої друкується слово «Додаток « і велика літера, що позначає додаток.

Додатки слід позначати послідовно великими літерами української абетки, за винятком літер Г, Є, И, І, Ї, О, Ч, Ъ, наприклад, додаток А, додаток Б і т.д. Один додаток позначається як додаток А.

Текст кожного додатка за необхідності може бути поділений на розділи й підрозділи, які нумерують у межах кожного додатка. У цьому разі перед кожним номером ставлять позначення додатка (літеру) і крапку, наприклад, А.2 – другий розділ додатка А; В.3.1 – підрозділ 3.1 додатка В. Ілюстрації, таблиці і формули, які розміщені в додатках, нумерують у межах кожного додатка, наприклад: рис. Д.1.2 – другий рисунок першого розділу додатка Д; формула (А.1) – перша формула додатка А.

Список використаних джерел

Літературне оформлення курсової роботи є важливим елементом її виконання і одним із багатьох чинників, який враховується при оцінюванні роботи. Передусім звертається увага на змістовний аспект викладу матеріалу (логічність і послідовність, повнота і репрезентативність, тобто широта використання наукових джерел, загальна грамотність та відповідність стандартам і прийнятим правилам), а також на текст роботи, список літератури, додатки та на зовнішнє оформлення.

Список використаної літератури складається на основі використаних джерел інформації і відображає обсяг використаних джерел та ступінь вивченості досліджуваної теми, є «візитною карткою» автора роботи, його професійним обличчям, свідчить про рівень володіння навичками роботи з науковою літературою.

Літературу до роботи слід оформити таким чином. Якщо це *підручник, монографія*, то пишуться прізвище та ініціали автора, повна і точна назва книги, місце видання (для міст Київ, Харків, Москва – в скороченій формі: К; Х; М), назва видавництва, рік видання, загальна кількість сторінок.

Наприклад:

1. Бондаренко В. Г., Канівська І. Ю., Парамонова С. М. Теорія ймовірностей і математична статистика. Ч. 1 / К. : НТУУ "КПІ", 2006. 125 с.

Статті, опубліковані у періодичних виданнях описуються так:

2. Валькман Ю. Р. Моделирование НЕ-факторов — основа

интеллектуализации компьютерных технологий / Ю. Р. Валькман, В. С. Быков, А. Ю. Рыхальский. *Системні дослідження та інформаційні технології*. 2007. № 1. С. 39–61.

Якщо на титульному аркуші книги відсутнє прізвище автора, то книгу описують, починаючи з її назви, після чого вказують ініціали, прізвище редактора. Подальший порядок опису такий самий, як і для книги, що описується під прізвищем автора:

3. Кібернетика в сучасних економічних процесах: зб. текстів виступів на республік. міжвуз. наук.-практ. конф. / Держкомстат України, Ін-т статистики, обліку та аудиту. К. : ІСОА, 2002. 147 с.

Дуже часто потрібно використати електронні ресурси, але потрібно правильно робити посилання на них. На це варто звернути увагу як самим авторам робіт, так і їх науковим керівникам, оскільки оформлення роботи оцінюється. Для прикладу:

1. Розподіл населення найбільш численних національностей за статтю та віком, шлюбним станом, мовними ознаками та рівнем освіти [Електронний ресурс]: за даними Всеукр. перепису населення 2001 р. / Держ. ком. статистики України ; ред. О.Г. Осауленко. К. : CD-вид-во "Інфодиск", 2004. 1 електрон. опт. диск (CD-ROM): кольор.; 12 см. (Всеукр. перепис населення, 2001). Систем. вимоги: Pentium-266 ; 32 Mb RAM ; CD-ROM Windows 98/2000/NT/XP. Назва з титул. екрану.
2. Бібліотека і доступність інформації у сучасному світі: електронні ресурси в науці, культурі та освіті : підсумки 10-ї Міжнар. конф. „Крим-2003”. [Електронний ресурс] / Костенко Л. Й., Чекмарьов А. О., Бровкін А. Г., Павлуша І. А. *Бібліотечний вісник*, 2003. №4. С. 43. URL: <http://www.nbu.gov.ua/articles/2003/03klinko.htm>. (дата звернення: 11.11.2021).

ЗАХИСТ КУРСОВОЇ РОБОТИ ТА КРИТЕРІЇ ЇЇ ОЦІНЮВАННЯ

Курсову роботу виконують впродовж семестру. За 10 днів до початку сесії студент подає на кафедру курсову роботу в друкованому вигляді. На титульному аркуші фіксується термін подання. На останній сторінці висновків студент проставляє свій підпис та дату задачі роботи на кафедрі.

Студенти, що не подали в строк курсові роботи, не допускаються до захисту. У цьому випадку захист переноситься на кінець сесії. Студент повинен подати на кафедру готову роботу протягом першого тижня сесії і

захистити її у встановлений термін. Оцінка за курсову роботу, подану пізніше встановленого терміну, знижується.

Якщо курсова робота має суттєві недоліки, то робота може повертатися студенту на доопрацювання. Терміни доопрацювання встановлюються керівником. Якщо студент не згодний із оцінкою викладача і бажає підвищити свої результати, то він може також отримати роботу і її доопрацювати. Це можливо у випадку, коли роботу було подано на кафедру вчасно, а після її перевірки залишається певний час до захисту.

Протягом семестру студент повинен звітувати керівнику про стан виконання КР. Викладач оцінює поданий матеріал і за необхідності може надавати зауваження та рекомендації щодо курсової роботи та програми.

Заключним етапом виконання КР є її захист, який проводиться комісією. Процес захисту відбувається у формі доповіді студента і відповідей на питання членів комісії. Доповідь на захисті повинна займати 5-7 хвилин. Студент повинен провести демонстрацію програмного рішення, розробленого в процесі виконання КР, і стисло висловити основні положення роботи (мета досліджень і розробки програмного рішення, короткий огляд одержаних результатів, ефект від впровадження). Доповідь супроводжується демонстрацією слайдів. Слайдів, до яких немає посилань в доповіді, бути не повинно.

Після доповіді студент повинен дати відповіді на питання членів комісії. Відповіді повинні бути чіткими, повними і аргументованими.

При виставлянні оцінки комісія виходить з повноти і глибини виконаного завдання, обґрунтованості відповідей на запитання присутніх, якості оформлення пояснювальної записки, а також уміння грамотно, переконливо, логічно, повно і стисло продемонструвати результати своєї роботи.

Рейтингова оцінка з курсової роботи матиме дві складові. Перша (стартова) характеризує роботу студента з курсової роботи та її результат – якість пояснювальної записки та графічного матеріалу. Друга складова характеризує якість захисту студентом курсової роботи. Сума балів двох складових переводиться до залікової оцінки.

Оцінювання роботи студента з курсовою роботою:

складність, науковість, повнота розкриття дослідження та аргументованість висновків	10 балів
актуальність та елемент творчості	5 балів
стиль, грамотність	3 бали
якість оформлення	2 бали
практичні результати	40 балів
Максимальна оцінка	60 балів

Захист науково–дослідних робіт проходить у вигляді наукової доповіді. На доповідь дається 10-15 хвилин.

Оцінюється:

аргументоване доведення проблеми з урахуванням власного внеску, чіткість, логічність, лаконічність викладення матеріалу	20 балів
повнота, вичерпність відповідей, культура мовлення	10 балів
якість наочних матеріалів та вміння користуватися ними	10 балів
Максимальна оцінка	40 балів

Курсова робота		Захист	Сума
Теоретична частина	Практична частина	40	100
20	40		

За результатами захисту курсової роботи виставляється оцінка.

У екзаменаційній відомості результати навчання проставляються за двома шкалами – 100-бальною та національною. Позитивні оцінки виставляються тільки тим студентам, які виконали всі вимоги до оформлення та захисту курсової роботи.

Шкала загальної оцінки курсової роботи

Сума балів за всі види навчальної діяльності	Оцінка за національною шкалою для екзамену
90–100	відмінно
82–89	добре
74–81	
64–73	задовільно
60–63	
0–59	незадовільно

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Національний стандарт України ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання».
3. Коноваленко І. В. Програмування мовою C# 6.0. : навчальний посібник для технічних спеціальностей вищих навчальних закладів. Тернопіль ТНУ. 2016. 229 с. URL: https://exam.nuwm.edu.ua/pluginfile.php/103770/mod_resource/content/1/Konovalenko_I_Programuvannja_movoju_C%23_6_0_%282016%29.pdf (дата звернення: 11.11.2021).
2. Павловская Т. А. C#. Программирование на языке высокого уровня. Учебник для ВУЗов. СПб. : Питер, 2009. 432 с. URL: https://exam.nuwm.edu.ua/pluginfile.php/103773/mod_resource/content/1/C%23_%D0%9F%D0%B0%D0%B2%D0%BB%D0%BE%D0%B2%D1%81%D0%BA%D0%B0%D1%8F.pdf (дата звернення: 11.11.2021).
3. Основи об'єктно-орієнтованого програмування у C# : методичні вказівки до лабораторних робіт для студентів I-го курсу математичного факультету спеціальності "Прикладна математика" / Брила А. Ю. та ін. Ужгород, 2014. 73 с. URL: https://exam.nuwm.edu.ua/pluginfile.php/103774/mod_resource/content/1/C%23_%D0%9E%D0%9E%D0%9F.pdf (дата звернення: 11.11.2021).
4. Методичні вказівки до виконання лабораторних робіт з курсу "Об'єктно-орієнтоване програмування" / Уклад. Коноваленко І. В. Тернопіль : ТНТУ, 2017. URL: https://exam.nuwm.edu.ua/pluginfile.php/103775/mod_resource/content/1/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%96%20%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B8%20%D0%9E%D0%9E%D0%9F.pdf (дата звернення: 11.11.2021).
5. Полное руководство по языку программирования C#. URL: <https://metanit.com/sharp/tutorial/> (дата звернення: 11.11.2021).
6. Крушельницька О. В. Методологія та організація наукових досліджень : навч. посіб. для вищ. навч. закладів. К. : Кондор, 2003. 190 с.
7. Васильев А. C#. Объектно-ориентированное программирование М. : Питер, 2012. 320 с.
8. Бабенко Л. П., Лаврищева К. М. Основи програмної інженерії : навч. посіб. К. : Знання, 2001.
9. Албахари Д. C# 5.0. Карманный справочник. М. : Диалектика / Вильямс, 2013. 951 с.
10. Калянов Г. Н. CASE. Методы и средства системного структурного анализа и проектирования. М. : МГУ, 2000.

11. Вендров А. М. CASE-технологии. Современные методы и средства проектирования информационных систем. М. : Финансы и статистика, 1998.
12. Марченко А. Л. Основы программирования на C# 2.0. М. : Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2007. 552 с.
13. Ларман К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ и проектирование. 3-е изд. СПб. : Вильямс, 2012. 736 с.
14. Матвеева Л. Є., Волков В. А. Процес розробки програмного забезпечення. Від теорії до практики. К. : Інформаційні програмні системи, 2008. 117 с.
15. Рудаков А. В. Технология разработки программных продуктов : уч. пособие. М. : Академия, 2006. 208 с.
16. Скит Д. C#. Программирование для профессионалов. М. : Вильямс, 2011. 544 с.
17. Сфер Ч., Дэвис С. Р. C# 2008 для "чайников". М. : Вильямс, 2009. 592 с.
18. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. М. : Диалектика / Вильямс, 2015. 409 с.
19. Методичні вказівки до виконання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» (для студентів напряму підготовки 6.050101 «Комп'ютерні науки») / Укл. С. С. Стоянченко, Т. М. Терещенко. Луганськ : Вид-во СНУ ім. В. Даля, 2013. 17 с.
20. Донченко В. Ю. Курсовий проект: Методичні рекомендації до виконання курсового проекту з об'єктно-орієнтованого програмування для студентів спеціальності 123 Комп'ютерна інженерія. Старобільськ : Альма-матер, 2018. 38 с. К.
21. Методичні вказівки до виконання курсового проекту для студентів галузі знань 12 «Інформаційні технології» спеціальність 122 «Комп'ютерні науки та інформаційні технології», 124 «системний аналіз» / Уклад. : І. В. Назарчук, Г. Г. Швачко. К. : НТТУ «КПІ», 2017 32 с.
22. Методичні вказівки до виконання курсової роботи з навчальної дисципліни «Проектування інформаційних систем» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Інформаційні системи та технології» спеціальності 126 «Інформаційні системи та технології» денної та заочної форми навчання [Електронне видання] / Волошин В. С. Рівне : НУВГП, 2020. 17 с.

Інформаційні ресурси

1. Стандарт вищої освіти України: перший (бакалаврський) рівень, галузь знань 12 – Інформаційні технології, спеціальність 126 – Інформаційні

- системи та технології. (Затверджено і введено в дію наказом Міністерства освіти і науки України від 12.12.2018 № 1380.)
2. Гладка О. М., Грицюк П. М., Барановський С. В., Назарук, В. Д. (2021) ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «Інформаційні системи і технології» першого рівня вищої освіти за спеціальністю 126 «Інформаційні системи та технології» галузі знань 12 «Інформаційні технології». Кваліфікація: бакалавр з інформаційних систем та технологій. URL: <http://ep3.nuwm.edu.ua/21061/> (дата звернення: 11.11.2021).
 3. Грицюк П. М., Парфенюк О. В., Роценюк А. М. (2021) Освітньо-професійна програма "Цифрові технології дистанційної освіти" першого рівня вищої освіти за спеціальністю 015.39 "Професійна освіта (цифрові технології)" галузі знань 01 "Освіта / Педагогіка". Кваліфікація: бакалавр з професійної освіти (за спеціалізацією "Цифрові технології"). URL: <http://ep3.nuwm.edu.ua/21062/> (дата звернення: 11.11.2021).
 4. Грицюк П. М., Тадеєв П. О., Гладка О. М., Роценюк А. М. (2020) Освітньо-професійна програма "Цифрові технології дистанційної освіти" першого рівня вищої освіти за спеціальністю 015.39 "Професійна освіта (цифрові технології)" галузі знань 01 "Освіта / Педагогіка". Кваліфікація: бакалавр з професійної освіти (за спеціалізацією "Цифрові технології дистанційної освіти"). URL: <http://ep3.nuwm.edu.ua/18805/> (дата звернення: 11.11.2021).
 5. Національна бібліотека ім. В. І. Вернадського. URL: <http://www.nbuv.gov.ua/e-resources/>, <http://www.nbuv.gov.ua/webnavigator/> (дата звернення: 11.11.2021).
 6. Рівненська обласна універсальна наукова бібліотека (м. Рівне, майдан Короленка, 6). URL: <http://www.lib.rv.ua/> (дата звернення: 11.11.2021).
 7. Рівненська централізована бібліотечна система (м. Рівне, вул. Київська, 44). URL: <http://cbs.rv.ua/>
 8. Наукова бібліотека НУВГП (м. Рівне, вул. Олекси Новака, 75). URL: <http://nuwm.edu.ua/naukova-biblioteka/>, http://nuwm.edu.ua/MySql/page_lib.php (дата звернення: 11.11.2021).
 9. Цифровий репозиторій НУВГП. URL: <http://ep3.nuwm.edu.ua>.
 10. Express API 4.x. URL: <http://expressjs.com/ru/4x/api.html> [Цит. 25.03.2020]. (дата звернення: 11.11.2021).
 11. Docs NPM. URL: <https://docs.npmjs.com/all> [Цит. 25.03.2020].
 12. Node.js body parsing middleware. URL: <https://github.com/expressjs/body-parser> [Цит. 25.03.2020]. (дата звернення: 11.11.2021).
 13. Multer. URL: <https://ewigggin.gitbooks.io/expressjs-middleware/content/multer.html> [Цит. 06.06.2019]. (дата звернення: 11.11.2021).

ДОДАТКИ

Додаток А

Зразок оформлення титульної сторінки звіту

Міністерство освіти і науки України
Національний університет водного господарства та
природокористування
Навчально-науковий інститут автоматики, кібернетики та
обчислювальної техніки
Кафедра комп'ютерних технологій та економічної кібернетики

Курсова робота

з дисципліни «Програмування»
на тему

(назва теми курсової роботи)

Виконав(ла) студент(тка) ННІАКОТ

(курс, група, спеціальність,

прізвище, ім'я та по-батькові)

Керівник:

(вчений ступінь, звання, посада,

прізвище та ініціали)

Рівне – 20__

Приклад оформлення змісту

ВСТУП

РОЗДІЛ 1. Постановка задачі

- 1.1. Огляд існуючих підходів до розв'язання поставленої задачі
- 1.2. Уточнена постановка задачі на розробку програмного забезпечення

РОЗДІЛ 2. Розробка програмного продукту

- 2.1. Метод розв'язання задачі
- 2.2. Алгоритм розв'язання задачі

РОЗДІЛ 3. Опис розробленого програмного продукту

- 3.1. Опис головних структур і змінних програми
- 3.2. Опис головних функцій
- 3.3. Опис інтерфейсу
- 3.4. Результати роботи програмного продукту

ВИСНОВКИ

СПИСОК ЛІТЕРАТУРИ

Додаток А. Текст програми

Таблиця з описом функцій

№	Синтаксис	Призначення параметрів	Опис
1.	<i>FILE* fOp(char* fName, char* typ);</i>	<i>fName</i> – ім'я файлу <i>typ</i> – режим відкриття	Відкриває файл <i>fName</i> та пов'язує його з відповідним потоком <i>stream</i> .

Таблиця опису змінних та структур даних

№	Синтаксис	Призначення	Опис
1.	<i>int count</i>		Глобальна змінна для підрахунку кількості об'єктів
2.	<code>class Queue { public int inf; public Queue next; }</code>	<code>inf</code> – поле для збереження інформації <code>next</code> – поле зв'язку	Клас для створення черги

Приклади оформлення списку літератури

Підручника, навчального посібника (один, два або більше авторів):

1. Гради Буч, Джеймс Рамбо, Ивар Якобсон. Введение в UML от создателей языка. М. : ДМК Пресс, 2011 г., 496 с.
2. Гради Буч, Роберт А. Максимчук, Майкл У. Энгл, Бобби Дж. Янг, Джим Коналлен, Келли А. Хьюстон Объектно-ориентированный анализ и проектирование с примерами приложений. М. : Вильямс, 2010, 720 с.
3. Спинеллис Д., Гусиос Г. Идеальная архитектура. Ведущие специалисты о красоте программных архитектур / Пер. с англ. СПб. : Символ#Плюс, 2010. 528 с., ил.
4. Майк Кон Scrum. Гибкая разработка ПО. М. : Вильямс, 2011., 576 с.
5. Эрик Эванс Предметно-ориентированное проектирование (DDD). Структуризация сложных программных систем. 2010. 448 с.
6. Мартин Фаулер Рефакторинг. Улучшение существующего кода. СПб. : Символ-Плюс, 2008 г., 432 с.
7. Роберт Мартин Чистый код.Создание, анализ и рефакторинг: Питер, 2010 г., 464 с.
8. Roman Pichler Agile Product Management with Scrum: Creating Products that Customers Love: Addison-Wesley Professional, 2010. 160 с.

Збірника наукових праць:

1. Вісник. Економіка. Сучасні методи управління підприємством : зб. наук. праць. Вип.4(22).Ч.1. Рівне : НУВГП, 2004. 254 с.

Статті з журналу:

2. Захарін С. В. Кредитування інвестиційної діяльності. *Фінанси України*. 2004. № 4. С. 97–105.