МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

**МЕТОДИЧНІ ВКАЗІВКИ**

до виконання курсових робіт

з дисципліни «**Об'єктно-орієнтоване програмування**»

для студентів напряму 6.121 «Інженерія програмного забезпечення»

Затверджено

на засіданні методичної комісії

Протокол № \_\_ від \_\_\_\_\_\_\_\_\_\_\_\_

Львів – 2020

**Об'єктно-орієнтоване програмування**: методичні вказівки до виконання курсових робіт для студентів напряму 6.121 «Інженерія програмного забезпечення» / Укл. Коротєєва Т.О., Дяконюк Л.М.– Львів: Видавництво Національного університету “Львівська політехніка ”, 2020. – с.

Укладачі: Коротєєва Т.О.,

Дяконюк Л.М

Відповідальний за випуск Яковина В.С., док. техн. наук, проф.

Рецензенти:

**1. Мета курсової роботи**

У методичних вказівках розглядається структура та обсяг основних розділів курсової роботи, подаються вимоги до їхнього змістового наповнення. Описано вимоги до оформлення звіту з курсової роботи і послідовність її виконання та захисту.

Метою виконання курсової роботи є закріплення теоретичних знань та практичних навичок, набутих при вивченні дисципліни «Об'єктно-орієнтоване програмування».

У ході виконання курсової роботи студенти повинні навчитись самостійно працювати з літературою та сучасними засобами ООП.

**2. Структура та обсяг звіту до курсової роботи**

Титульний аркуш до звіту ідентифікує студента, що його виконав та тему варіанта завдання. Взірець титульного аркушу наведено в додатку Б.

Звіт до курсової роботи має містити наступні розділи:

* завдання до курсової роботи;
* зміст;
* алгоритм розв‘язку задачі у покроковому представленні;
* діаграми UML [клас](http://vns.lpnu.ua/mod/glossary/showentry.php?eid=54317&displayformat=dictionary)ів, прецедентів, послідовності виконання;
* код розробленої програми з коментарями;
* протокол роботи програми для кожного пункту завдання
* інструкція користувача та системні вимоги;
* опис виняткових ситуацій;
* структура файлу вхідних даних;
* висновки;
* список використаних джерел.

Обсяг курсової роботи повинен становити не менше 20 сторінок друкованого тексту.

**Завдання до курсової роботи**

Варіант завдання на курсову роботу обирається зі списку варіантів у додатку В. У бланку «Завдання» вказується номер варіанту, тема роботи, яку студент формулює на основі функціоналу обраного варіанту. Також вказується прізвище студента, номер групи, перелік основних задач, які необхідно розв'язати у ході виконання курсової роботи та календарний план їх виконання.

Бланк завдання, підписаний керівником і студентом, підшивається до пояснювальної записки курсової роботи. Приклад завдання на курсову роботу наведений у **додатку** Б.

**Зміст**

Зміст курсової роботи повинен відповідати темі та завданню до курсової роботи. У змісті вказуються номери і назви основних розділів звіту до курсової роботи та номери сторінок, де вони починаються. Висновки, список літератури не нумеруються.

**Алгоритм розв‘язку задачі у покроковому представленні**

Після осмислення функціоналу завдання необхідно розробити алгоритм розв'язку задачі загалом та кожного пункту функціоналу зокрема. Алгоритми зображуються у покроковому представленні (назва алгоритму, коротка характеристика, перелік кроків).

**Діаграми UML** [**клас**](http://vns.lpnu.ua/mod/glossary/showentry.php?eid=54317&displayformat=dictionary)**ів, прецедентів, послідовності виконання.**

У цьому пункті наводяться UML діаграми розроблених класів (класу), їх взаємозв’язок, діаграма прецедентів та діаграма послідовності виконання. Діаграми оформлюються як рисунки за відповідними вимогами.

**Код розробленої програми з коментарями**

Наводиться весь код розробленої програми з коментарями. Якщо програма складається з кількох файлів, то вказується назва файлу і далі відповідний код.

**Протокол роботи програми для кожного пункту завдання**

У вигляді скрін-шотів наводяться результати роботи програми для кожного пункту функціоналу. Оформлення за вимогами до рисунків.

**Інструкція користувача та системні вимоги**

За результатами роботи програми оформляється детальна інструкція користувача (див. дисципліну «Вступ до програмної інженерії»). Крім того, наводяться зображення головного вікна програми з пронумерованими кнопками, полями даних та результатів, а нижче надається роз’яснення цих номерів згідно функціоналу.

**Опис виняткових ситуацій**

У програмі має бути передбачено не менше 3-х виняткових ситуацій з відповідними повідомленнями. В цьому пункті звіту наводяться результати спрацювання винятків у вигляді скрін-шотів та роз'яснення кожного з них.

**Структура файлу вхідних даних**

У програмі необхідно передбачити введення вхідних даних в двох режимах: 1) з клавіатури; 2) з файлу. Для другого режиму у цьому пункті звіту необхідно зобразити приклад структури файлу вхідних даних з відповідними уточненнями змісту. Оформити у вигляді рисунка.

**Висновки**

У висновках перераховуються основні результати курсової роботи, можливість розвитку теми.

**Список використаної літератури**

Наводиться список літературних джерел ( не менше 5 джерел), які були використані при виконанні курсової роботи, оформлених згідно з Загальними положеннями та правилами складання бібліографічних джерел, затверджених Міністерством освіти України.

**3. Вимоги до оформлення**

Курсова робота є індивідуальною роботою кожного студента та оформляється окремо кожним студентом згідно із виданим завданням.

Курсова робота оформляється на аркушах формату A4, які заповнюються з однієї сторони.

Текст на сторінці розміщується рівномірно з дотриманням відступів: зверху – 2,0 см, знизу – 2,0 см, зліва– 2,0 см, справа – 1,0 см. Шрифт Times New Roman, розмір шрифту – 14, міжрядковий інтервал – 1,5 , вирівнювання абзаців – по ширині, перший рядок – відступ 1 см.

Нумерацію сторінок, розділів, підрозділів, пунктів, підпунктів, рисунків, таблиць, формул подають арабськими цифрами без знака №.

Першою сторінкою роботи є титульний аркуш. На титульному аркуші номер сторінки не ставлять, на наступних сторінках номер проставляють внизу справа.

Зміст, висновки, список використаних джерел не мають порядкового номера, але всі аркуші, на яких розміщені згадані структурні частини курсової роботи, нумерують звичайним чином. Не нумерують лише їх заголовки, тобто не можна друкувати «1. ВСТУП», або «Розділ 9. ВИСНОВКИ».

Кожен розділ повинен мати свій номер, який записується перед його назвою. Після номера розділу ставиться крапка, наприклад: 6. Опис виняткових ситуацій.

Розділ може складатися з підрозділів. Підрозділи нумерують у межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, між якими ставлять крапку. У кінці номера підрозділу повинна стояти крапка, наприклад: ''2.2.'' ( третій підрозділ другого розділу). Потім, у тому ж рядку йде заголовок підрозділу, наприклад: 2.2. Діаграми UML послідовності виконання.

Рисунки і таблиці необхідно подавати у роботі безпосередньо після тексту, де вони згадані вперше, або на наступній сторінці. Рисунки позначають словом «Рис.» і нумерують послідовно у межах розділу. Номер рисунку повинен складатися з номера розділу і порядкового номера рисунку, між якими ставиться крапка. Наприклад: Рис. 1.2 ( другий рисунок першого розділу). Номер рисунку, його назва і пояснювальні підписи розміщують послідовно під ним. Рисунки підписують і нумерують під рисунком по центру тексту. Підписи під рисунками виконують шрифтом Times New Roman ( розмір – 14), курсив.

Формули у роботі (якщо їх більше одної) нумерують у межах розділу. Номер формули складається з номера розділу і порядкового номера формули у розділі, між якими ставлять крапку. Формули подають у форматі Equation 3–5. Номери формул пишуть біля правої строни аркуша на рівні відповідної формули у круглих дужках, наприклад: (3.1) (перша формула третього розділу). Нумерація формул може бути також наскрізною.

Список літератури виконується згідно з вимогами стандартів. Літературні джерела можуть бути розміщені за алфавітом або у порядку посилання на них. У списку можна наводити тільки ту літературу, яка була використана при виконанні курсової роботи і на яку є посилання у тесті звіту. Посилання у тексті звіту на джерела слід зазначати порядковим номером за переліком посилань, виділеним двома квадратними дужками, наприклад, «... у працях [1-7]...».

Звіт до курсової роботи необхідно скріпити промисловим степлером в двох місцях зліва (книжковий варіант).

**Вимоги до програмного забезпечення.**

Програма може бути написана на мовах С++, С# або Java за вибором студента. Код повинен бути оформлений згідно міжнародних конвенцій стандарту обраної мови. Кожний клас повинен мати назву, яка повністю описує його суть, і інформативний склад, а також повинен бути розміщений в окремому файлі. Конструктори, деструктор, атрибути і методи класів слід визначити самостійно. Успадкування класів потрібно використовувати тільки тоді, коли воно має сенс. При реалізації слід застосовувати стандартні колекції та алгоритми, якщо умовою завдання не передбачено інше. Вибір тих, чи інших структур даних має бути обгрунтований. Програма повинна мати зручний інтерфейс користувача, зокрема, містити головне та контексні меню, панель інструментів, що відтворюють основні пункти меню, а також систему підказок.

**4. Організація та контроль виконання роботи**

**Організація виконання роботи**

Курсова робота виконується самостійно кожним студентом у відповідності з графіком, який встановлюється при видаванні завдання на курсову роботу. Графік виконання роботи контролюється викладачем згідно з розкладом консультацій з курсової роботи.

Комп’ютерна реалізація курсової роботи здійснюється у домашніх умовах або у лабораторіях кафедри.

**Захист курсової роботи**

У призначений час студент демонструє роботу комп'ютерної програми та електронний варіант оформленого звіту до курсової роботи. При захисті роботи студент повинен дати відповіді на питання за темою роботи та засобами реалізації. Відповіді на запитання мають довести викладачу самостійність виконання роботи та знання засобів ООП.

Якщо до електронного варіанту звіту до курсової роботи у викладача є зауваження, то студенту надається право виправити їх і після того роздрукувати та скріпити звіт.

Захист роботи здійснюється студентом тільки один раз. При незадовільній оцінці студенту видається нове завдання на курсову роботу. Незахищена курсова робота вважається академічною заборгованістю.

Електронна версія курсової роботи (програма та звіт) записуються на відповідний носій, спільний для всієї навчальної групи студентів.

Додаток А

**Варіанти завдань до курсової роботи**

**Варіант 1**

Створити програму, що дозволяє здійснювати бухгалтерські обрахунки для IT-компанії.

Визначити класи працівників, програмістів та лідерів різного рангу. Обов’язковими атрибутами працівника є його персональні дані, а саме прізвище, ім’я та по-батькові, дата народження, стать, дата вступу на роботу та займана посада, а також стаж та категорія.

Вважати, що 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 для заданої категорії. Але якщо програміст відпрацював менше годин на день, ніж призначено (теж існує відповідна норма) – нараховується пеня, що обчислюється як заданий відсоток від нарахувань. Для лідерів - заробітна плата нараховується, як оклад + відсоток від заробітної плати кожного працівника, що перебуває в його підпорядкуванні. Також можливі разові штрафні та преміальні санкції, які може задавати керівництво. Для всіх працівників існує нормативна надбавка 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. Зміна для зазначених працівників посади чи категорії.
7. Збір інформації за різні часові інтервали зі збереженням в файлах, а також таблицях.

**Варіант 2.**

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 в локаційний журнал з зазначеними часовими даними початку кінця розмови (Генерувати випадковим чином на заданому проміжку часу)
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, тимчасове автоматичне відключення чи навпаки включення відповідно до зарахувань абонентської плати.

**Варіант 3.**

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

Програма передбачає роботу з продовольчими товарами, які мають назву, вагу, вартість, кількість, дату створення, термін придатності, виробника. Непродовольчі товари характеризуються назвою, вартістю, кількістю, виробником, габаритними розмірами, групою призначення (одяг, техніка, меблі, посуд іт. ін. 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.

Потрібно створити наступні сервіси:

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 з зазначенням критеріїв порівняння на еквівалентність двох товарів.
8. Визначення сумарної кількості товарів заданої назви.
9. Визначення товарів заданої категорії, обмежених за ціновим діапазоном.

**Варіант 4.**

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. Сортування повідомлень за часом відправлення.
5. Пошук повідомлень заданого відправника.
6. Знаходження користувачів, що не відправляли жодного повідомлення.
7. Виведення адреси останнього відправленого повідомлення для заданого користувача.
8. Пошук всіх різних електронних адрес, за якими були здійснені або, навпаки, були отримані відправлення заданого користувача.

**Варіант 5.**

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. Також вони поділяються на різні категорії, що мають можливі додаткові характеристики: особняк, квартира, пентхаус, гуртожиток і т.д. 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, тощо (Не менше 4 варіантів)
4. Визначення приміщень, що мають максимальні та мінімальні характеристики площ, вартості.
5. Пошук власників, які зареєстрували більше 2 приміщень для продажу.
6. Долучення та вилучення нових приміщень з фіксацією дати внесених змін.
7. Обчислення загальної суми проданої нерухомості за заданий період.

**Варіант 6**.

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. У 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 діток.

**Варіант 7.**

Створити програму, що відтворює модель організації роботи готелю.

Користувач має мати змогу зняти в користування готелю з відповідною категорією(кількістю зірочок) простий номер, чи номер люкс.

У готелі може бути наповнення номерами різного класу. Кожен клас мусить мати набір зручностей згідно вимог “зірковості” готелю. Номер має містити інформацію про кількість кімнат, кількість місць, кількість вікон, площу, наявність побутової техніки, дати бронювання, вартість.

Номер класу люкс 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 та класом.
4. Визначення користувача, який проживає в готелі найдовше.
5. Визначення користувачів, які забронювали більше 2 номерів.
6. Знаходження найпопулярнішого запиту в діапазоні заданих дат.
7. У випадку не знаходження вільних номерів на вказані дати передбачити список пропозицій, які б частково задовольняли запит.

**Варіант 8.**

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.

Пасажир 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. Здійснення аналізу частоти зайнятості маршрутів.
4. Визначення найдовшого чи найкоротшого маршруту.
5. Пошук власника проданого квитка за прізвищем.
6. Повернення квитка.
7. Обчислення прибутку каси за заданий період.

**Варіант 9.**

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 зареєструватись, 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 заданої кількох заявок.
4. Сортування картин за жанром, автором, частотою оформлення заявки.
5. Пошук картин за заданими критеріями.
6. Пошук заявника, що підписався на найбільшу суму.
7. Збереження історії заявника на заданому аукціоні.

**Варіант 10.**

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 в заданий день з можливою пропозицією для оптимізації процесу.
6. Пошук спеціалістів за заданою категорією.

**Варіант 11.**

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. Сортування списку викладачів за посадою та прізвищем.
6. Визначити всіх викладачів, що мають тільки один курс.
7. Пошук не зайнятих спеціалістів на задану дату.
8. Автоматичне очищення історії від застарілих дат.

**Варіант 12.**

Створити програму, що відтворює модель роботи тренажерного залу. Реалізувати ієрархію тренажерів, які присутні в залі, розраховані на певний тип вправ, що тривають визначену тривалість часу та зазначають список об’єктів для тренування та обмежень(наприклад, на біговій доріжці можлива вправа для схуднення, яка триває 20 хвилин, тренує заданий перелік м’язів, дихання і т.д. Задані вікові обмеження, обмеження на вагу і т.д.).

Потрібно створити наступні сервіси:

1. Наповнення інформацією про існуючі тренажери.
2. Наповнення інформацією про типові програми.
3. Сортування тренажерів за заданими параметрами за спаданням та за зростанням.
4. Визначення норм роботи за заданим тренажером та розцінок за задану одиницю часу.
5. Реалізація можливості підбору необхідних тренажерів для тренувальної програми (програма складається з вправ).

**Варіант 1У.**

Створити імітаційну модель руху суден в певній водоймі. Програма повинна задавати довільну кількість суден, що рухаються заданою користувачем лінійною траєкторією, відслідковуючи можливість перетину траєкторій. У випадку імовірного зіткнення кораблів, повинна бути здійснена зупинка їх руху. Судна можуть бути різних категорій: пасажирські, військові та вантажні. Кожен з видів суден мають мати не менше 5 атрибутів. Різні категорії суден повинні по-різному відображатись. Також користувач має мати можливість отримати інформацію про кожне з суден. Вважатимемо, що кораблі рухаються рівномірно, проте з різною швидкістю. Програма має надавати можливість змінювати ці параметри.

Також в водоймі може бути обрана область у формі опуклого многокутника, перетин якої будь-яким судном має бути відслідкованим шляхом фіксації часу в журналі історій і супроводом звукового сигналу.

**Варіант 2У.**

Створити імітаційну модель руху автомашин на регульованому світлофорами перехресті з пішохідними переходами. Вважати, що машини та пішоходи рухаються рівномірно. Користувач програми має мати можливість задавати мінімальну та максимальну частоту появи машин, які мають виникати на закінченні доріг випадковим чином. На дорогах рух може бути організований в декілька смуг. Час для зміни сигналів світлофорів може бути керованим, але не виходити за межі нормативних діапазонів. Робота світлофорів на різних вітках має бути синхронізована. Вважатиметься, що всі учасники руху дотримуються правил.

**Варіант 3У.**

Створити програмний застосунок для обчислення значень математичних виразів, в яких можуть бути задані довільні ідентифікатори, дійсні числа, дужки, операції додавання, віднімання, множення, ділення та піднесення до степені, а також стандартні математичні функції, список яких користувач має мати змогу поповнювати. При цьому вважатимемо, що вираз задано правильно, без помилок.

Перед обчисленням виразу створити список не визначених ідентифікаторів та надати можливість ініціалізувати їх.

**Варіант 4У.**

Написати програму, що реалізує роботу з розрідженими матрицями. Матриці зберігати у вигляді списку збережених значень ненульових елементів з їх індексами. Реалізувати операції стиску та розгорнення матриць, а також їх додавання та множення, звернувши увагу на ефективність реалізації цих операцій.

**Варіант 4У.**

Створити імітаційну модель руху ліфтів в багатоповерхових будівлях. Користувач має мати змогу задавати кількість поверхів в будівлі, типи ліфтів та їх кількість. У будівлі випадковим чином можуть генеруватися майбутні пасажири з заданою масою та з наміром їхати на той чи інший поверх. Пасажири можуть вишиковуватись у черги до ліфтів за принципом якомога більш рівномірного розподілу в чергах. При завантаженні ліфта враховується сумарна вага, яка може бути перевезена. У випадку перевантаження ліфта його рух буде не дозволеним. При неповному завантаженні ліфта він може обслужити виклик на одному з проміжних поверхів свого шляху. Виклики ліфта групуються в черги з пріоритетами. Програма повинна в графічній формі відображати запропоновану модель.

Додаток Б

**Структура титульної сторінки**

Національний університет “Львівська політехніка”

Кафедра програмного забезпечення

**КУРСОВА РОБОТА**

**з дисципліни «Об’єктно-орієнтоване програмування»**

**На тему:  
«***формулюється студентом індивідуально згідно обраного варіанту* **»**

Стедента групи *nn*

спеціальності 6.121

“Програмна інженерія”

*прізвище ініціали*.

Керівник: доцент кафедри ПЗ,

к.т.н., доцент Коротєєва Т. О.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів\_\_\_Оцінка ECTS\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Львів 20*nn*

Додаток В

**Приклад завдання на курсову роботу**

ЗАВДАННЯ

на курсову роботу з дисципліни «Об’єктно-орієнтоване програмування»

студента групи ПЗ-*21 Іванова Олександра*

**Тема: «*Телефонний довідник*»**

**Завдання**: автоматизувати процесс формування та опрацювання телефонного довідника.

**Зміст завдання та календарний план його виконання**

|  |  |  |
| --- | --- | --- |
| **№ з/п** | **Зміст завдання** | **Дата** |
| 1 | Здійснити аналiтичний огляд лiтератури за заданою темою та обгрунтувати вибір інструментальних засобів реалізації. |  |
| 2 | Побудова UML діаграм |  |
| 3 | Розробка алгоритмів реалізації |  |
| 4 | Реалізація завдання (кодування) |  |
| 5 | Формування інструкції користувача |  |
| 6 | Оформлення звіту до курсової роботи згідно з вимогами Міжнародних стандартів, дотримуючись такої структури:   * зміст; * алгоритм розв‘язку задачі у покроковому представленні; * діаграми UML [клас](http://vns.lpnu.ua/mod/glossary/showentry.php?eid=54317&displayformat=dictionary)ів, прецедентів, послідовності виконання; * код розробленої програми з коментарями; * протокол роботи програми для кожного пункту завдання * інструкція користувача та системні вимоги; * опис виняткових ситуацій; * структура файлу вхідних даних; * висновки; * список використаних джерел. |  |

Завдання прийнято до виконання: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Прізвище І.П.)

( пiдпис студента )

Керівник роботи: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Прізвище І .П./