

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ
ЗАКЛАД «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ»**

Кафедра комп'ютерної інженерії

МЕТОДИЧНІ ВКАЗІВКИ ДО КУРСОВОЇ РОБОТИ
з дисципліни "**Об'єктно-орієнтовне програмування**"

для студентів денної та заочної форм навчання
за спеціальністю:

121- Інженерія програмного забезпечення

122 – Комп'ютерні науки

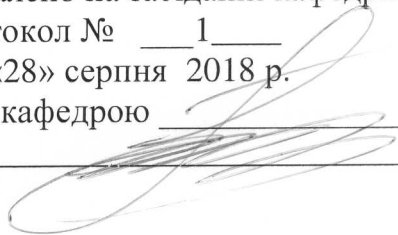
123 - Комп'ютерна інженерія

Запоріжжя
2019-2020 навчальний рік

Методичні вказівки до курсової роботи з дисципліни "Об'єктно-орієнтовне програмування" для студентів денної та заочної форм навчання за спеціальністю: 121 – Інженерія програмного забезпечення, 122 – Комп'ютерні науки, 123 - Комп'ютерна інженерія.

Укладач: доцент кафедри КІ Жеребцов О.А.

Ухвалено на засіданні кафедри
протокол № 1
від «28» серпня 2018 р.
Зав. кафедрою _____



ЗМІСТ

ВСТУП.....	5
1 ЦІЛІ ТА ЗАВДАННЯ КУРСОВОЇ РОБОТИ.....	6
2 ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ	8
2.1 Формулювання вимог до програми.....	8
2.2 Проектування програми.....	8
2.3 Кодування програми	9
2.4 Тестування програми	9
2.5 Пояснювальна записка	9
3 ТИТУЛЬНИЙ ЛИСТ І ЗАВДАННЯ НА КУРСОВУ РОБОТУ	10
3.1. Аналіз технічного завдання	11
3.2. Обґрунтування алгоритму й структури програми.....	12
3.3. Розробка програми	13
3.3.1. Загальні правила	14
3.3.2 Розробка системи класів	14
3.3.4 Розробка методів	17
3.3.5 Створення об'єктів і розробка головної програми.....	18
3.3.5 Опис файлів даних та інтерфейсу програми.....	19
3.4. Тестування програми і результати її виконання.....	19
4 ВИМОГИ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ КУРСОВОЇ РОБОТИ.....	21
4.1 Структурні елементи та розділи.....	21
4.2 Нумерація сторінок	22
4.3 Нумерація розділів та підрозділів	22
4.4 Формули і рівняння	23
4.5. Перелік посилань.....	24
5 ВАРІАНТИ ЗАВДАНЬ НА КУРСОВУ РОБОТУ	26
6 ПОРЯДОК ЗАХИСТУ КУРСОВОЇ РОБОТИ ТА ЇЇ ОЦІНКА	42
7 ПЕРЕЛІК ДЖЕРЕЛ, ЩО РЕКОМЕНДОВАНІ ДЛЯ ВИКОНАННЯ КУРСОВОЇ РОБОТИ	43
8 ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	45

ДОДАТОК А	46
ДОДАТОК Б	47
ДОДАТОК В. Приклад оформлення реферату.....	48
ДОДАТОК Г. Приклади специфікації	49

ВСТУП

Курсова робота - це перша самостійна наукова праця майбутнього програмного інженера. При її виконанні студент поглиблює знання з фундаментальних та професійно-орієнтованих дисциплін, 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 інженера творчий підхід до роботи.

Курсова робота дає можливість використовувати набуті теоретичні знання для вирішення конкретних прикладних задач. Основна мета виконання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» — закріплення на практиці вміння використовувати основні концепції об'єктно-орієнтованого підходу (ООП) — класи, засоби UML- моделювання, інкапсуляцію, успадкування, поліморфізм, перевантаження методів і операцій, шаблони методів і класів та один із найпотужніших інструментаріїв ООП і програмної інженерії — STL- бібліотеку (Стандартну бібліотеку шаблонів C++) при створенні складних програмних проектів.

Студент повинен вміти використовувати та практично застосовувати переваги об'єктно-орієнтованої парадигми програмування і проектування при вирішенні поставленого завдання.

Курсова робота з об'єктно-орієнтованого програмування передбачає знання студентами об'єктно-орієнтованого проектування та програмування і є завершальним етапом вивчення у рамках курсу об'єктно-орієнтоване програмування підготовки студентів.

1 ЦІЛІ ТА ЗАВДАННЯ КУРСОВОЇ РОБОТИ

Курсова робота (КР) з курсу “Об’єктно-орієнтоване програмування” передбачає виконання комплексного завдання науково – дослідницького характеру. Метою КР є закріплення, поглиблення та узагальнення знань, якими студент оволодів під час вивчення курсу, 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 практики наукових досліджень з оформленням відповідного текстового, програмного та ілюстративного матеріалу у формі наукового звіту.

Використанням принципів об’єктно-орієнтованого проектування ПЗ різних задач на ЕОМ необхідно для подальшого використання для розробки програмного забезпечення сучасних складних технічних систем та комп’ютерних систем управління.

В результаті вивчення дисципліни та роботи над курсовою роботою студенти повинні

знати: загальні питання побудови та використання сучасних систем об’єктно-орієнтованого та візуального проектування програмних засобів комп’ютерних систем управління; основні поняття та методи об’єктно-орієнтованого проектування та програмування, основні засоби розробки програм з використанням багато ієрархічних структур об’єктів та класів об’єктів;

вміти: самостійно вибрати середовище проектування програмних засобів, проектувати, розробляти, налагоджувати та супроводжувати сучасне програмне забезпечення комп'ютерних систем управління.

Під час виконання курсової роботи студент повинен продемонструвати:

– вміння збирати і аналізувати відповідні матеріали про об'єкт дослідження, використовуючи сучасні джерела інформації, включаючи Інтернет ресурси;

– спроможність проводити необхідні обґрунтування для розробки програмних засобів систем управління різного призначення, тощо;

– здатність доводити розв'язання поставленої задачі до логічного кінця;

– вміння аналізувати отримані результати і робити з них висновки.

Курсова робота є самостійною роботою студента.

Відповідальність за правильність аналітичних висновків, результатів розрахунків і моделювання, а також оформлення несе студент - автор КР.

2 ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Основні етапи виконання курсової роботи:

- а) формулювання вимог до програми;
- б) проектування програми;
- в) кодування програми;
- г) тестування програми;
- д) оформлення пояснювальної записки;
- е) захист роботи.

2.1 Формулювання вимог до програми

Формулювання вимог є самостійним етапом роботи і завершується створенням двох документів: індивідуального плану проекту та розділу пояснювальної записки “Специфікація програми”. Індивідуальний план складає студент на початку роботи над проектом и копію плану у той же час надає викладачу. За суттєве відставання від календарного плану викладач повинен знижувати оцінку курсової роботи.

На етапі формулювання вимог студент повинен:

- 1) сформулювати призначення програми;
- 2) визначити перелік функцій, які виконує програма;
- 3) запропонувати інтерфейс користувача, який надасть можливість останньому дістатися цих функцій.

Специфікація програми складається з послідовного опису усіх виділених функцій, та пов'язаного з ними графічного інтерфейсу користувача. У специфікації програми треба обов'язково надати зображення усіх вікон програми.

2.2 Проектування програми

На етапі проектування визначається архітектурний стиль, об'єктна структура програми, а також формат похідних, вихідних та внутрішніх даних.

Співвідношення між класами та об'єктами відображується за допомогою діаграм, бажано на мові UML. Під час проектування можна розробити план функціонального тестування, та необхідні тести. Завершується проектування створенням проектної специфікації програми.

2.3 Кодування програми

На етапі кодування створюється похідний код програми. Він повинен бути гідним чином прокоментований і повністю задовольняти вимогам до похідного коду, наведеним у додатку ж цих вказівок. Увесь похідний код програми має бути наведений у додатку до пояснювальної записки. У тексті основних розділах записки можливо наводити лише невеликі за розміром окремі фрагменти коду.

2.4 Тестування програми

Розрізняють два основних різновиду тестування: функціональне та модульне. Функціональне тестування перевіряє, як програма виконує свої функції в умовах коректних та некоректних дій користувача. Воно планується на етапі проектування і виконується у міру того, як кодуються окремі функції програми і перед захистом програми. Модульне тестування робиться одночасно з кодуванням і має за мету оперативну перевірку окремих частин програми. Виконання модульного тестування вельми бажано, але не є обов'язковим.

2.5 Пояснювальна записка

Оформлення пояснювальної записки є останнім етапом виконання курсової роботи і має надати студентіві навичок документування програмного продукту.

3 ТИТУЛЬНИЙ ЛИСТ І ЗАВДАННЯ НА КУРСОВУ РОБОТУ

Титульний лист пояснювальної записки курсової роботи оформляється на листіформату А4 згідно з ДСТУ 3008:2015 [1]. Приклад оформлення титульного листа наведено у додатку А. Титульний лист має бути підписаний студентом та керівником роботи.

Завдання на курсову роботу є початковим документом для розробки роботи і має бути виконане за установленою формою.

Завдання складається керівником курсової роботи, узгоджується із студентом, що виконує роботу. Тема курсової роботи затверджується на засіданні кафедри. До завдання додається графік виконання курсової роботи. Форма завдання на курсову роботу наведена в додатку Б.

Титульний лист і завдання на курсової роботи вважаються відповідно першим і другим листом пояснювальної записки, однак номери листів на них не проставляються.

Реферат повинний містити:

- відомості про обсяг, кількість ілюстрацій, таблиць, використаних джерел, додатків;
- перелік ключових слів;

У тексті реферату коротко описують прийняті в проекті технічні, організаційні, економічні рішення, а також пристрої, їх ефективність і міра впровадження. У короткій лаконічній формі має бути представлений опис основних частин проекту. У формі, що стверджує, наголошується, що було виконано, розраховано, сконструйовано, запропоновано, модернізовано і так далі. Реферат рекомендується складати після написання пояснювальної записки проекту.

Перелік ключових слів повинен характеризувати зміст реферованої записки і включати від 5 до 15 ключових слів в називному відмінку прописними буквами, викладених в рядки через коми.

Ключові слова в сукупності повинні поза текстом давати достатньо повне уявлення про вміст записки і забезпечувати можливість інформаційного пошуку.

Обсяг реферату не більше 2000 знаків (500 слів) і не повинний перевищувати одного листа. Приклад змісту реферату наведено у додатку В.

Реферат є першим текстовим листом пояснювальної записки виконується на листі формату А4 з основним написом згідно ДСТУ 3008:2015.

Зміст є обов'язковим розділом пояснювальної записки курсового проекту, який включає назву розділів і підрозділів записки (починаючи зі вступу), список використаної літератури, додатки і їх найменування, а також номер сторінки, на якій розміщений або починається розділ або додаток. Зміст записки розміщують на першому (заголовному) листі розділу, що має основний напис. Зміст поміщають після титульного листа і реферату і включають в загальну чисельність листів пояснювальної записки.

3.1. Аналіз технічного завдання

Основною метою даного розділу пояснювальної записки є аналіз вимог завдання на курсову роботу й формулювання додаткових технічних вимог, які безпосередньо впливають з нього, та мети дослідження.

На основі аналізу потрібно конкретизувати основні параметри програми, її можливості та функції, засоби їх реалізації, інтерфейс із користувачем, середовище, в якому програма може працювати.

Якщо у попередньому розділі лише формулюють загальну методику розв'язування задач, то тут визначають конкретний план роботи. Просте переписування вимог завдання на курсову роботу не допускається, потрібно творчо осмислити його та сформулювати конкретні технічні вимоги і шляхи їх досягнення.

Обсяг розділу — 1..2 сторінки.

3.2 Обґрунтування алгоритму й структури програми

Виходячи із поставленого завдання та технічних вимог до програми, потрібнорозробити алгоритм її роботи. Він може бути відображений у вигляді UML-діаграми (при необхідності блок-схеми, оформленій згідно з вимогами стандартів, або словесним описом, що включає послідовність узагальнених операцій та переходів між ними).

Такий опис може мати ієрархічну структуру, де спочатку записують більш загальні операції, які потім розшифровують на нижчому ієрархічному рівні.

Опис алгоритму при використанні об'єктно-орієнтованого підходу проектування програмного забезпечення зручніше проводити в термінах об'єктів, що взаємодіють між собою за допомогою UML діаграм відношень, повідомлень і подій. До таких програм можуть відноситися різноманітні редактори (текстові, нотні, графічні), програми керування інформаційними сховищами (базами даних), інформаційно-довідкові системи, системи опрацювання переривань, ігрові програми тощо. Щодо останніх, то у таких випадках алгоритм може розпадатися на кілька незалежних одна від одної послідовностей операцій, що реагують на події та керують опрацюванням повідомлень.

Для систем, що працюють у реальному часі, важливим є часова послідовність опрацювання повідомлень і подій. Для них, крім опису алгоритму, застосовують ще й часові UML-діаграми, що характеризують послідовність виконання різних процесів, та їх взаємодію з апаратними засобами.

В загальному, спосіб опису алгоритму вибирають, виходячи з конкретної задачі, однак він повинен бути повним, несуперечливим і достатнім для розуміння суті та ілюстрації запропонованого способу її вирішення та принципу роботи спроектованої програми.

Аналізуючи отриманий алгоритм, потрібно визначити в ньому основні структурні частини, які дозволять здійснити декомпозицію (розбиття) проекту

на завершені програмні одиниці (модулі, програми, бібліотеки). Це в більшості випадків спрощує як процес розробки, так і відлагодження програмного проекту.

Найчастіше окремі методи (функції), класи та змінні групують за функціональним призначенням та розміщують в окремих програмних одиницях (модулях), які можна окремо компілювати. Для програмування на мові C++ кожен із модулів, як правило, розбивають на два файли: заголовний файл із розширенням *.h, що містить інтерфейсну частину модуля, та файл реалізації модуля з розширенням *.cpp.

Крім модулів, потрібна головна програма, яка їх викликає. Для випадку, якщо завданням передбачено розробку повноцінної програми, то ця головна програма й реалізує необхідний алгоритм роботи. Якщо ви розробляєте лише набір модулів або бібліотеку функцій (наприклад, реалізуєте побудову графіків на основі даних, сформованих у програмі користувача, або створюєте бібліотеку для реалізації різноманітних операцій з новим типом даних, таким як комплексні числа), то така головна програма буде лише тестовою, призначеною для відлагодження, перевірки та демонстрування можливостей модулів.

Для деяких задач виникає необхідність створення ще й допоміжних програм (утиліт), що виконують певні сервісні функції (наприклад, конвертування файлів, формування тестових даних, виведення графіків тощо). У такому випадку розв'язок задач реалізується у вигляді програмного пакета, що містить головну програму, модулі та допоміжні програми.

Згідно з указаною вище схемою потрібно провести декомпозицію проекту та відобразити її в даному розділі.

Обсяг розділу — 2..4 сторінки.

3.3. Розробка програми

3.3.1. Загальні правила

Цей розділ є основним у курсовій роботі. Його обсяг — 8.. сторінок.

У залежності від обраного стилю програмування, можливі різні підходи до розробки програми. У найпростіших випадках вона йде послідовно, алгоритм програми реалізується за допомогою послідовності операторів, включно з умовними та операторами циклу.

У структурному програмуванні застосовується ієрархічний підхід, у якому окремі структурні одиниці перетворюються на функції, що можуть приймати параметри та повертати результат. Опис функції в такому випадку виконується один раз у певному місці програми, а використовується вона багато разів, при цьому лише змінюються параметри, і не потрібно щоразу повторювати одні й ті ж оператори, які включені у функцію.

Найвищу продуктивність роботи при розробці складних програмних продуктів дає об'єктно-орієнтований підхід, який завдяки принципам інкапсуляції, успадкування та поліморфізму дає можливість виділити всі спільні та відмінні риси окремих елементів поставленого завдання і записати їх у програмі в найлаконічнішій формі.

Розробка програми при об'єктно-орієнтованому підході розбивається на розробку класів та глобальних типів даних, реалізацію методів, дружніх та глобальних функцій, створення об'єктів та організацію взаємодії між ними в головній програмі для досягнення поставленої мети.

3.3.2 Розробка системи класів

Після того, як виконана декомпозиція програми (розроблена її структура) і відомі алгоритми розв'язку задач, можна приступати до реалізації наміченого плану. Потрібно чітко уявити, з якими даними буде працювати програма (чи окремий модуль) та які дії над ними доведеться виконувати. Дані та дії над ними утворюють об'єкт. Також можна уявити програму як набір об'єктів (реально існуючих фізичних об'єктів або абстрактних понять, що

існують лише в нашій уяві), які взаємодіють між собою. Наступним етапом буде поділ об'єктів на групи, що мають хоча б щось (дані, властивості, функціональність). У більшості випадків об'єкти будуть більше чи менше подібні. Деколи в групах можна буде виділити підгрупи і т. д., аж до складного ієрархічного дерева. У кожній групі виділяється найзагальніші властивості, притаманні всім без винятку об'єктам у групі та дії над ними. Отже, отримаємо базовий клас для цієї групи об'єктів, даними якого будуть загальні параметри об'єктів групи, а методами — дії, які можна проводити над даними будь-якого об'єкта групи. Далі для кожної з підгруп (у межах виділеного класу об'єктів) додаємо притаманні лише їй властивості, як поля нового, породженого від базового класу. Додаткові дії, які властиві кожній підгрупі, стають методами породжених класів, а параметри, яких не мали об'єкти базового класу, — даними породжених (похідних) класів. Породження класів продовжується до тих пір, поки не будуть описані всі параметри об'єктів та дії, які необхідно над ними здійснювати для розв'язку поставлених задач. Описана послідовність повторюється для кожної з Групи об'єктів у програмі.

Для прикладу розглянемо програму, яка повинна грати в шахи. У ній можна виділити такі дві Групи об'єктів: перша — реалізує графічне відображення на екрані шахової дошки та фігур на ній, друга — реалізує логіку роботи програми — генерує ходи, перевіряє їх правильність, слідкує за часом і т. д. Розглянемо першу групу. З чого вона складається? Сюди входять різноманітні видимі елементи: дошка, клітинки на ній, різні фігури, позначення рядів на дошці. Які в них параметри? Це — координати, розміри, колір для фігур, для клітинки — ще тип фігури, яка на ній знаходиться, для дошки — ще масив клітинок на ній і т. д. Які дії повинні виконувати об'єкти? Вони повинні реагувати або не реагувати на натискання мишкою чи на клавіатуру, відобразитися, зникати, деякі ще повинні переміщуватись, причому по-різному. Що ж є спільним для них усіх? Серед параметрів — напевне, координати центра, ще колір і розмір по горизонталі та вертикалі, з

дій — кожен із об'єктів повинен уміти створюватися, знищуватися та відображати себе на екрані. Таким чином, базовий клас буде містити такі поля даних: координати "x", "y", "ширина", "висота", "колір"; методи — конструктор, що створює об'єкт, деструктор, що його знищує, та метод "відобрази себе". Оце й усе! Але описане лише один раз (!) для всіх видимих об'єктів. Далі можна розділити видимі об'єкти на пасивні та активні. Пасивними будуть написи на дошці, що позначають назви рядів, активними — всі решта, вони повинні мати метод, який змушує їх реагувати на натискання мишкою. Тепер виділимо ще підклас фігур, що породжується від класу "активні видимі елементи", який має додатковий метод "перемісти мене" та поле даних або метод, що визначає допустимі ходи для даної фігури. Так, дошка містить 64 клітинки, а кожна клітинка — може містити якусь фігуру. Тому до даних цього класу, також породженого від "активних видимих елементів", слід додати ще поле даних (напевне, вказівник), який містить інший об'єкт(-и) класу — "видимі елементи". Також необхідні методи, які ці об'єкти "вставляють" у клас "Група" або видаляють з "Групи". Метод "відобрази себе" для "групи" завдяки поліморфізму можна доповнити відображенням об'єктів "групи" після того, як відобразилася сама "група". Далі від "групи" можна успадкувати нові класи: "шахова дошка", "шахове поле", а від "шахового поля" — "поле останнього рядка", яке крім методів свого попередника може ще й перетворювати "пішака" у "королеву". Цей, далеко не повний, приклад дає можливість зрозуміти лише основний підхід до проектування класів. Пофантазувавши трохи, можна отримати досить елегантну й, головне, завершену ієрархію об'єктів, які будуть уміти робити все, що від них може вимагати головна програма для розв'язування поставлених задач. Основне — кожна спільна для кількох об'єктів дія (і спільний елемент даних) описана лише один раз у базовому класі і більше ніде її не потрібно описувати повторно. Якщо деякі дії передбачають виконання операцій над кількома об'єктами і при цьому необхідно мати безпосередній доступ до закритих даних класу, то такі дії робимо дружніми функціями для

того класу, над даними якого вони працюють. Деякі типи даних, які не доцільно робити класом, оскільки вони не мають власної функціональності, але які неодноразово використовуються в програмі (наприклад, масиви фіксованої довжини, списки чи структури) можемо описати, використавши `typedef` помістити в заголовок одного з модулів.

Підрозділ "Розробка системи класів" відображає основну роботу, яка виконується при об'єктно-орієнтованому програмуванні, в результаті чого отримуються описи всіх класів з даними та методами, причому для методів уже задані їх параметри та тип результату, який вони повертають. Тобто, з точки зору інтерфейсу, робота над класами на цьому етапі вже завершена. Залишилося тільки "навчити" методи виконувати свою роботу. Глобальні функції, які не є методами ні одного з класів, також поміщаються в один із модулів, а їх інтерфейсна частина описується в цьому підрозділі.

3.3.4 Розробка методів

Після того, як з інтерфейсною частиною "закінчено", залишилося лишерозробити реалізацію методів, дружніх та глобальних функцій. Це завдання є досить простим, якщо на попередньому етапі було правильно спроектовано систему класів та детально пророблено інтерфейсні частини. Реалізація будь-якої функції (чи методу) розміщується в *.cpp - частині модуля, на відміну від інтерфейсу, яка записується у *.h- файл. Потрібно лише акуратно записати ті дії, які має виконувати метод чи функція, не забуваючи користуватися вже готовими функціями та методами замість того, щоб увесь час змінювати дані за допомогою одних і тих же послідовностей операцій. Так, метод "перемісти мене" (див. приклад у попередньому пункті) повинен лише змінити координати активного видимого елемента, а не намагатися перемалювати його самостійно — метод "відобрази себе" зробить це набагато краще. Аналогічно метод "відобрази себе" для "групи" (шахової дошки чи клітинки), який повинен відображати об'єкти Групи після того, як відобразилася сама група, реалізується з використанням поліморфізму та

успадкування. При цьому додається лише один-два оператори, оскільки і група, і видимі елементи вже самі вміють відображатися, необхідно лише "сказати" їм, щоб вони це зробили один за одним. У методі "відобрази себе" групи спочатку викликається метод "відобрази себе" базового класу ("видимий елемент"), який відображає саму дошку або клітинку, а далі — методи "відобрази себе" для всіх видимих елементів, що входять до групи. Для клітинки Це буде виклик "відобрази себе" для фігури, що стоїть на клітинці, а для дошки — виклик "відобрази себе" для всіх позначок та клітинок, які, у свою чергу, відобразять фігури, що на них знаходяться. У даному підрозділі слід описати принцип дії основних методів та відмінності поліморфних (віртуальних) методів від методів базових класів.

3.3.5 Створення об'єктів і розробка головної програми

Нарешті, У цьому підрозділі слід описати процес створення самих об'єктів у програмі, функціональність та дані яких розроблені в попередніх підрозділах. Якщо кількість об'єктів, що будуть створені наперед невідома, або обсяг пам'яті, яку вони займають, є досить значним, тоді пам'ять для об'єктів слід виділяти "динамічно" з використанням оператора `new`. Стосовно розглянутого прикладу із шахами, даний підрозділ, очевидно буде описувати, як створюється шахова дошка з 32 фігурами на ній та як перетворюються вибрані партнерами ходи в повідомлення шаховим фігурам. Створивши об'єкти, можемо реалізувати певну послідовність дій або схему взаємодії між об'єктами, яка призводить до вирішення поставленого завдання. Опис цієї послідовності дій чи схеми взаємодії, можливо, з прикладами для реальних вхідних даних чи повідомлень і є основою даного підрозділу. Такий опис повинен бути достатнім для розуміння всіх деталей реалізації алгоритму та можливих випадків, що зустрічаються при роботі програми.

3.3.5 Опис файлів даних та інтерфейсу програми

Якщо програма використовує файли як джерело вхідних даних або для зберігання проміжних чи кінцевих результатів роботи, то в даному підрозділі слід навести опис формату цих файлів. Він може бути виконаний у текстовому, табличному чи графічному вигляді. Можна додавати до нього приклади реальних файлів із даними програми. Особливо важливий даний підрозділ для програм, що працюють із базами даних, адже структура таблиць бази даних, перелік, типи полів, засоби взаємодії з базою даних, перетворення даних є основою таких програм. Для інтерактивної програми з розвинутою системою меню та діалогових вікон у цьому підрозділі слід описати призначення елементів меню, роботу з ними, параметри, що вибираються в діалогових вікнах тощо. У цьому ж розділі слід описати інтерфейс програм, які працюють з параметрами командного рядка. Якщо темою роботи є розробка модулів або бібліотеки, то слід детально описати порядок їх використання, навести в алфавітному порядку всі доступні для користувача глобальні типи даних і змінні, функції та класи.

3.4. Тестування програми і результати її виконання

У даному розділі треба описати методику тестування програми, тестові дані та навести результати роботи програми. Якщо програма працює в графічному режимі, то слід роздрукувати копію графічного вікна програми. Якщо результатом роботи програми є текстовий файл, то необхідно вивести вміст цього файлу. Для програм з розвинутою системою діалогових вікон і меню слід обмежитися друком лише найсуттєвіших результатів, які демонструють правильну роботу програми, а не передруковувати весь екран для кожного відкритого пункту меню. Перелік усіх пунктів меню в такому випадку та вміст неосновних діалогових вікон можна подати в текстовому вигляді. Якщо для відображення роботи програми необхідна значна кількість роздруків, то їх можна подати в додатках. Обсяг розділу — 1..5 сторінок. У розділі потрібно зробити висновок, який підтверджує (або заперечує)

працездатність програми. У висновках по КР в реферативній формі повинні бути описані результати, отримані студентом на кожному з етапів виконання роботи (аналітичному, етапу проектування ПЗ системи, експериментальному дослідженню, аналізу отриманих результатів), а також висновків щодо досягнення мети курсової роботи, перспективи розвитку даної галузі тощо. Висновки пишуться на 1-2 сторінки і як розділ не нумеруються. Тут наводиться перелік джерел, на які були посилання в тексті. Список повинен формуватися в порядку посилань за текстом і вміщувати бібліографічні відомості офіційно виданих книжок, статей, патентів, депонованих рукописів тощо. Як розділ перелік літератури не нумерується. Формат опису літературних джерел повинен відповідати стандарту. У додатки включають логічні схеми програм та їх лістинги, а також інші документи. Крім цього, в додатки поміщуються таблиці, графіки та методики, які з якихось причин не увійшли до пояснювальної записки, але потрібні для пояснень.

4 ВИМОГИ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ КУРСОВОЇ РОБОТИ

Загальними вимогами до пояснювальної записки є логічна послідовність викладання матеріалу, стислість, чіткість і конкретність викладання теоретичних і практичних результатів роботи, доказовість висновків і обґрунтованість рекомендацій. Записку виконують із розрахунку не більше 40 рядків на сторінці за умови рівномірного її заповнення і висотою літер і цифр не менше 1,8 мм. При написанні тексту необхідно дотримуватись таких розмірів полів: верхнє, лівє і нижнє - не менше 20 мм, правє - не менше 10 мм. Прізвища, назви додатків, організацій, фірм та інші імена наводяться мовою оригіналу. Пояснювальна записка виконується машинним способом на одній стороні аркушу білого паперу формату А4 (210 x 297 мм) відповідно до ДСТУ 3008-2015 "Документація. Звіти в сфері науки і техніки. Структура і правила оформлення". Наведемо деякі положення цього документу.

4.1 Структурні елементи та розділи

Треба розрізняти структурні елементи записки, такі як "РЕФЕРАТ", "ЗМІСТ", "ВСТУП", "СУТЬ" і розділи записки. Деякі елементи, а саме "РЕФЕРАТ", "ЗМІСТ", "ВСТУП", "ВИСНОВКИ", "ПЕРЕЛІК ПОСИЛАНЬ", створюють розділи записки і їхні найменування служать заголовками окремих розділів. Такі розділи не мають нумерації. Структурний елемент записки „СУТЬ” сам може поділятися на розділи, які нумеруються и мають власні заголовки. Розділи можуть поділятися на підрозділи, пункти та підпункти. Розділи і підрозділи повинні мати заголовки, пункти і підпункти можуть мати заголовки. Заголовки структурних елементів записки і заголовки розділів необхідно розташовувати посередині рядка і друкувати прописними літерами без крапки наприкінці, не підкреслюючи. Заголовки підрозділів, пунктів та підпунктів записки необхідно починати з абзацного відступу і друкувати малими літерами, крім першої прописної, не підкреслюючи, без крапки наприкінці. Абзацний відступ повинен бути однаковим по всьому тексту

записки і має дорівнювати п'яти знакам. Якщо заголовок складається з двох або більше речень, їх розділяють крапкою. Переноси слів у заголовку розділу не припускаються. Відстань між заголовком і наступним або попереднім текстом повинна бути не менше двох рядків. Не припускається розміщати найменування розділу, підрозділу, а також пункту і підпункту в нижній частині сторінки, якщо після нього розташований тільки один рядок тексту.

4.2 Нумерація сторінок

Нумерувати сторінки необхідно арабськими цифрами, дотримуючи наскрізну нумерацію по всьому тексті. Номер сторінки проставляють у правому верхньому кутку без крапки наприкінці.

4.3 Нумерація розділів та підрозділів

Розділи, підрозділи, пункти, підпункти пояснювальної записки варто нумерувати арабськими цифрами. Розділи пояснювальної записки повинні мати порядкову нумерацію в межах суті пояснювальної записки і позначатися арабськими цифрами без крапки, наприклад, 1, 2, 3. Підрозділи повинні мати порядкову нумерацію в межах кожного розділу. Номери підрозділів складаються з номерів розділів і підрозділів, що розділяються крапкою, наприклад, 1.1, 1.2, 1.3.

За необхідністю можуть бути використані переліки. Перед переліком ставлять двокрапку. Перед кожною позицією переліку треба ставити малу літеру алфавіту з дужкою або дефіс (перший рівень деталізації). Для подальшої деталізації переліку треба використовувати арабські цифри з дужкою (другий рівень деталізації). Ілюстрації треба розташовувати в пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, або на наступній сторінці. На всі ілюстрації повинні бути посилання в пояснювальній записці. Ілюстрації можуть мати назву, що розташовують під ілюстрацією. При необхідності під ілюстрацією ще

розташовують пояснювальні дані. Ілюстрація позначається словом “Рисунок”, що разом із назвою поміщають після пояснювальних даних, наприклад, “Рисунок 2.1 - Схема алгоритму”. Ілюстрації необхідно нумерувати арабськими цифрами в межах розділу, за винятком ілюстрацій, наведених у додатках. Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, розділених крапкою. Якщо ілюстрація не вміщується на одній сторінці, можна переносити її на інші сторінки, при цьому назву ілюстрації розташовують на першій сторінці, пояснювальні дані - на кожній сторінці, і під ними вказують: “Рисунок, аркуш”. Таблиці необхідно розташовувати в пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, або на наступній сторінці. На всі таблиці повинні бути посилання в пояснювальній записці. Таблиця може мати назву, яку друкують малими літерами (крім першої прописної) і розміщують над таблицею. Якщо рядки або графи таблиці виходять за формат сторінки, таблицю поділяють на частини, розташовуючи одну частину під іншою, або поруч, або переносючи частини таблиці на наступну сторінку. Слово “Таблиця”, вказують один раз зліва над першою частиною таблиці, над іншими частинами пишуть: “Продовження таблиці ”з вказівкою номера таблиці. Таблиці треба нумерувати арабськими цифрами в межах розділу, за винятком таблиць, наведених у додатках. Номер таблиці складається з номера розділу і порядкового номера таблиці, розділених крапкою.

4.4 Формули і рівняння

Формули і рівняння розташовують у пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, посередині сторінки. Вище і нижче кожної формули повинно бути залишено не менше одного вільного рядка. Формули і рівняння треба нумерувати арабськими цифрами порядковою нумерацією в межах розділу, за винятком формул і рівнянь, наведених у додатках. Номер формули або рівняння складається з номера розділу і порядкового номера формули або рівняння, розділених

крапкою. Номер формули або рівняння вказують на рівні формули або рівняння в дужках у крайньому правому положенні на рядку. Пояснення символів і числові коефіцієнти, що входять до формули або рівняння, варто робити безпосередньо після формули або рівняння в тій послідовності, у якій вони надані у формулі або рівнянні. Пояснення значення кожного символу або числового коефіцієнта необхідно давати з нового рядка. Перший рядок пояснення починають з абзацу словом "де" без двокрапки. Переносити формули або рівняння на інший рядок припускається тільки на знаках виконуваних операцій, причому знак операції на початку такого рядка повторюють. При переносі формули або рівняння на знаку операції множення застосовують знак "х". Формули, що прямують одна за одною, і не розділені текстом, відокремлюють комою. Посилання в тексті на джерела необхідно наводити в послідовності їх згадування в записці, вказуючи порядковий номер, виділений двома квадратними дужками, наприклад: [5]. У тексті пояснювальної записки припускається робити тільки загальноживані скорочення (наприклад "і т.д., і т.п., та ін.") або поширені аббревіатури. Якщо в пояснювальній записці прийнята специфічна термінологія, або вживаються малопоширені скорочення, нові символи, позначення і ін., їх перелік необхідно надати в окремому списку. Перелік повинен розташовуватися стовпчиком, у якому слова за абеткою наводяться скорочено, а справа - їхнє докладне тлумачення. Якщо спеціальні символи, терміни, скорочення, і позначення наводяться менше трьох разів, перелік їх не складається, а їхнє тлумачення роблять у тексті при першому згадуванні.

4.5. Перелік посилань

Перелік посилань оформлюється як розділ пояснювальної записки з нової сторінки відповідно до вимог ДСТУ 8302:2015 - "Бібліографічне посилання. Загальні положення та правила складання" [2].

Для оформлення бібліографічних посилань, правильного запису загальноприйнятих скорочень рекомендується використовувати методичні вказівки [3], які складені з урахуванням відповідних стандартів.

Додатки необхідно оформляти як продовження пояснювальної записки на наступних її сторінках, або в окремій частині, розташовуючи додатки в порядку появи посилань на них у тексті пояснювальної записки. Кожний додаток повинен починатися з нової сторінки і мати заголовок, надрукований вгорі малими літерами з першої прописної, симетрично щодо тексту сторінки. Посередині рядка над заголовком малими літерами з першої прописної повинно бути надруковано слово “Додаток ” і прописна буква, що позначає додаток. Додатки слід позначати послідовно прописними буквами українського алфавіту, за винятком букв Ѓ, Є, З, І, Ї, Й, О, Ч, Ь. Один додаток позначається як А. Додатки повинні мати загальну з іншою частиною пояснювальної записки наскрізну нумерацію. Текст кожного додатка, при необхідності, може бути розділений на підрозділи і пункти, що нумеруються арабськими цифрами в межах кожного додатка.

5 ВАРІАНТИ ЗАВДАНЬ НА КУРСОВУ РОБОТУ

Незалежно від теми програма, що розробляється, повинна задовольняти наступним загальним вимогам.

Стійкість програми. Програма не повинна втрачати працездатності ні при яких, навіть некоректних, діях користувача. Всякі дії, що загрожують втратою інформації, мають бути підтверджені користувачем. Інформація, що вводиться, скрізь, де це можливо, піддається логічному забезпеченню цілісності даних. При будь-яких діях користувача не повинні втрачатися дані або їх цілісність (некоректність індексів, втрата посилань в зв'язках після видалення-додавання записів і т. д.).

Функціональна повнота. Мають бути реалізовані усі функції, вказані в специфікації програми.

Інтерфейс користувача. Використовуються тільки терміни, зрозумілі користувачеві, і не використовуються терміни розробника («запис», «індексація» і т. д.). Поява службових англомовних повідомлень неприпустима. У повідомленнях користувача слід дотримуватися норм ввічливості, колірна гамма повинна наслідувати загальноприйняті рекомендації.

Використання клавіатури. На будь-якому етапі натиснення будь-якої клавіші повинне ігноруватися або викликати передбачені дії, описані в засобах допомоги.

Прив'язка дій до клавіш має бути загальнопринятною: F1 допомога; Enter - згода, завершення введення; Esc - відмова, повернення до попереднього вузла гілки алгоритму (з відновленням екранної форми); Tab - перехід до наступного поля, вікна і т. д.; Shift - Tab - повернення до попереднього Усі приклади тем узяті з книги [1].

1. Особиста бібліотека. Картотека домашньої бібліотеки.

Вихідні дані книги (Автори, назва, видавництво і т.д.), розділ бібліотеки (спеціальна література, хобі, домашнє господарство, белетристика і т.д.),

походження і наявність книги в даний момент, суб'єктивна оцінка книги. Вибір книг по безпідставному замовленню; Інвентаризація бібліотеки.

2. Картотека Інтерполу.

Дані по кожному зареєстрованому злочинцю: прізвище, ім'я, кличка, зріст, колір волосся і очей, особливі прикмети, громадянство, місце і дата народження, останнє місце проживання, знання Зиков, злочинна професія, остання справа і т.д. Злочинні і мафіозні угруповання (дані про співучасників). Вибірка по будь-якій підмножині ознак. Перенесення «Зав'язали в архів; Видалення лише після смерті.

3. Бюро знайомств.

База потенційних наречених: стать, реєстраційний номер, дата реєстрації, відомості про себе, вимоги до партнера. Вибір підмножини підходящих кандидатур, підготовка зустрічей (формування запрошення для знайомства). Перенесення в архів пар, які вирішили свої сімейні проблеми, видалення клієнтів, що відмовилися від послуг.

4. Біржа праці.

База безробітних: анкетні дані, професія, освіта, місце і посада останньої роботи, причина звільнення, сімейний стан, житлові умови, контактні координати, вимоги до майбутньої роботи. База вакансій: фірма, посада, умови праці та оплати, житлові умови, вимоги до фахівця. Пошук і реєстрація варіантів з того та іншого боку, формування оголошень для друку, видалення в архів після працевлаштування, повне видалення при відмові від послуг.

5. Записна книжка.

Анкетні дані, адреси, телефони, місце роботи або навчання, посаду знайомих, колег і родичів, характер знайомства, ділові якості і т.д. Автоматичне формування поздоровлення з днем народження (за поточної дати). Впорядкування за алфавітом і за датою останнього коригування. Пошук по довільному шаблону.

6. Каса Аерофлоту.

Розклад: номер рейсу, маршрут, пункти проміжної посадки, час відправлення, дні польоту. Кількість вільних місць на кожному рейсі. Вибір найближчого рейсу до заданого пункту (при наявності вільних місць), оформлення заданого числа квитків за погодженням з пасажиром (з зменшенням числа вільних місць), оформлення посадкової відомості.

7. Довідник споживача (служба побуту).

База підприємств побутового обслуговування міста: назва, розряд, адреса та телефони, спеціалізація, перелік надаваних послуг, форма власності, години і дні роботи. Пошук підприємств по заданій послугі та іншими ознаками.

8. Довідник покупця.

Бази торгових точок міста: назва, адреса і телефони, спеціалізація, форма власності, час роботи. Вибір магазинів за довільним шаблоном.

9. Магазин з одним продавцем.

Комп'ютер замість касового апарату. База наявності товарів: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завою. Реєстрація надходження товару (як старих, так і нових найменувань). Оформлення покупки: виписка чека, коректування бази. Проблема уцінки і списання. Інвентаризація залишків товару з обчисленням сумарної вартості.

10. Відділ кадрів.

База даних про співробітників фірми: паспортні дані, освіта, спеціальність, підрозділ, посада, оклад, дати надходження у фірму і останнього призначення і т.д. Вибір за довільним шаблоном. Скорочення штатів: вибір для звільнення осіб пенсійного та передпенсійного віку, підготовка наказу.

11. Генеалогічне дерево.

Паспортні дані членів деякого родового клану; посилання на дітей (або на батьків). Пошук всіх нащадків або всіх предків для вказаної особи.

12. Склад.

База товарів, що зберігаються на складі: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завозу. Реєстрація надходження товару (формування прибуткової накладної) і відвантаження (видаткова накладна). Вивід інвентарної відомості.

13. Каса автовокзалу.

Розклад автобусів: номер рейсу, кінцевий і проміжний пункти, час відправлення. Кількість вільних місць на кожному рейсі. Вибір найближчого рейсу до заданого пункту (при наявності вільних місць), оформлення квитків, оформлення посадкової відомості. Попередній продаж, повернення квитків.

14. Адміністратор готелю.

Список номерів: клас, число місць. Список гостей: паспортні дані, дати приїзду і від'їзду, номер. Поселення гостей: вибір відповідного номера (за наявності вільних місць), реєстрація, оформлення квитанції. Від'їзд: вибір всіх постояльців, що від'їжджають сьогодні, звільнення місця або оформлення затримки з випискою додаткової квитанції. Можливість дострокового від'їзду з перерахунком. Пошук гостя по довільною ознакою.

15. Довідник меломана.

База груп і виконавців: база пісень: база дисків з переліком пісень (у вигляді посилань). Вибір всіх пісень заданої групи; всіх дисків, де зустрічається задана пісня.

16. Щоденник.

База намічуваних заходів - дата, час і протяжність, місце проведення. Автоматичне нагадування найближчої справи: за поточною датою і часу: видалення вчорашніх справ або перенесення на майбутнє. Аналіз "накладок" - перетинань планованих справ. Перегляд справ на завтра, післязавтра і т.д.

17. Термінологія.

База визначень будь-якої науки: вводиться термін, його тлумачення (визначення), посилання на використовувані терміни. Можливість перегляду всього ланцюжка від заданого терміна до первинних понять.

18. Шеф-кухар.

База рецептур страв: розкладка, рецепт приготування. База продуктів на складі: найменування, ціна, кількість. Формування меню на день (на задане число осіб); звану вечерю. Перевірка достатності запасів: формування видаткової накладної на склад, коригування запасів.

19. Довідник лікаря.

База хвороб: назва, симптоми, процедури, перелік рекомендованих ліків із зазначенням необхідної кількості. База медикаментів на складі: назва, кількість, взаємозамінність. Формування рецепта після огляду хворого, перевірка наявності ліків, коригування запасів.

20. Зарахування абітурієнтів.

База абітурієнтів; анкетні дані, сукупність оцінок на вступних іспитах, готовність вчитися на договірній основі. Вибір для зарахування заданої кількості абітурієнтів; формування для співбесіди списку тих, хто набрав граничний прохідний бал, але не може платити за освіту.

21. Довідник фірм.

Назва, адреса та телефони, перша особа, статус (форма власності), сировина, продукція. Вибір за довільним шаблоном.

22. Обмін житла.

База пропозицій по обміну: район, площа, планування і т.д.; вимоги до варіантів обміну. Реєстрація клієнтів, вибір відповідних варіантів, видалення при відбулому обміні або відмову.

23. Довідник абітурієнта.

База вузів; найменування, адреса, перелік спеціальностей, конкурс минулого року за кожною спеціальністю (денною, вечірньою, заочною форм), розмір оплати при договірному навчанні. Вибір за різними критеріям: все по даному вузі; все про даній спеціальності, пошук мінімального конкурсу по даній спеціальності та загалом.

24. Довідник поштової індексації.

Республіка, область (край), район, населений пункт, поштовий індекс. Пошук по будь-якій сукупності полів (крім останнього); ієрархічна зв'язок між

полями (зверніть увагу, що, наприклад, Новомосковськ є і в Тульській, і в Дніпропетровській областях).

25. Ощадбанк.

Відомості про вкладників банку: номер особового рахунку, категорія вкладу, паспортні дані, поточна сума внеску, дата останньої операції. Операції прийому і видачі будь-якої суми, автоматичне нарахування відсотків.

26. Ломбард.

База збережених товарів та нерухомості: анкетні дані клієнта, найменування товару, оціночна вартість; сума, видана під заставу, дата здачі, термін зберігання. Операції прийому товару, повернення, продажу - по закінченню терміну зберігання.

27. Довідник селекціонера.

Найменування сорту будь-якої культури, автор, батьківські сорти, врожайність, характеристики плодів, морозостійкість, стійкість до шкідників і хвороб, наявність в тому чи іншому фонді. Вибір сортів, що володіють заданими властивостями.

28. Довідник власника відеотеки.

База відеофільмів: назва, студія, жанр, рік випуску, режисер, виконавці головних ролей, короткий зміст, суб'єктна оцінка фільму. Факт наявності фільму у відеотеці. Оформлення видачі та повернення касети.

29. Купи – продай

База продавців: найменування товару, обсяг партії при оптовому продажі, ціна, умови продажу - відвантаження, форма оплати, контактну адресу або телефон, примітка (наприклад, «посередників прошу, не турбується»). База покупців: найменування товару, обсяг покупки, прийнятна ціна і форма оплати, контактну адресу або телефон, примітка. Пошук і реєстрація варіантів з тієї та іншої сторони; формування оголошень для друку, видалення в архів після купівлі - продажу (можливо, один із клієнтів залишається незадоволеним), повне вилучення при відмові від послуг.

30. Довідник фаната.

База спортсменів: анкетні і антропологічні дані, громадянство, походження, вид спорту, клуб або команда, дані про особовий рекорд або перемоги і так далі. Вибір за довільним ознакою. Пошук рекордсмена в заданому виді спорту.

31. Довідник радіоаматора.

Бази довідкових даних транзисторів, діодів, тиристорів і так далі: марка, характеристики, граничнодопустимі умови експлуатації, ціна, облік взаємозамінності і т.д. Підбір за заданими вимогам.

32. Довідник комерційних банків.

Найменування, адреса, статус (форма власності), умови зберігання коштів на особовому рахунку (річні відсотки на різних видів вкладів). Вибір банку з найбільшим відсотком для заданого типу вкладів.

33. Довідник начальника в'язниці.

Анкетні дані ув'язнених, стаття, термін, дата взяття під варту, місце в тюремній ієрархії, камера, відомості про родичів, особливості характеру. Формування статистичних зведень про склад, вибір за довільною ознакою.

34. Довідник військомата.

Військовослужбовці: анкетні дані, адресу батьків, цивільна професія, освіта, звання та дата його отримання, посада, підрозділ, форма служби, особливості характеру і ставлення до служби. Формування списків: заданого підрозділи, офіцерського складу, новобранців і т.д.

35. Ріелтерська контора (купівля-продаж житла). База пропозицій:

Район та адреса, характеристика будинку і квартири, запитувана вартість, координати заявника. База попиту: вимоги покупця до житла (можливо кілька варіантів, допустимі діапазони), фінансові можливості, координати заявника. Підбір варіантів для того чи іншого боку, автоматизований пошук взаємоприйнятних варіантів. Приклад запиту покупця: однокімнатна, до 100 тис. руб., Інкерман не пропонувати.

36. Черга на житло.

Список черговиків на отримання і поліпшення житла: дата надходження у фірму, дата подачі заяви, склад сім'ї, житлові умови на дату подання заяви, пільги на додаткову площу, позачергове і першочергове поліпшення, побажання на район і ін. Реалізація розподілу одержуваного муніципального житла: задоволення підходящих черговиків, переселення наступних в звільнилися квартири. Додавання заявників та видалення вибулих і задоволених.

37. Автосалон.

База нових і старих вітчизняних та іноземних автомобілів: марка, рік випуску, технічні характеристики, особливості виконання, технічний стан, запитувана ціна. База покупців: контактні координати, вимоги до марки, технічним характеристикам і технічним станом, фінансові можливості.

Автоматизація підбору варіантів для покупця, формування заявки для постачальників і перегонників.

38. Довідник туриста.

Туристичні агенції та пропоновані послуги: країна, місто (або маршрут круїзу), умови проживання та проїзду, екскурсійне обслуговування, сервіс прийнятої боку, вартість путівки.

39. «Купи-продай».

База оголошень (радіо-, газетних, в біжучому рядку) по всіх рубриках. Пошук для... (наприклад, «репетиторство & математика»).

40. Крилаті фрази.

Довідник прислів'їв, приказок, афоризмів, каламбурів, інших словесних курйозів. Класифікація за авторам і джерелам, пошук по темах і ключовим словам.

41. Каталог запчастин автомобіля.

В автомобілі налічується кілька тисяч деталей; деякі використовуються в різних марках. Таблиці: країна, фірма-виробник, марка автомобіля, агрегат, вузол, деталь. Облік взаємозамінності. Користувачі: працівники автосервісу, магазинів запчастин; постачальники-оптовики.

42. Каталог радіодеталей (довідник телевізійного обладнання).

Моделі побутової аудіо-, та відеотехніки; для кожної моделі - каталог радіодеталей, використаних в ній (резистори, конденсатори, котушки індуктивності, чіпи тощо). Багато деталей використовуються в різних моделях; деякі деталі однобічно (наприклад, замість резистора 10К 0,25 Вт, але не навпаки) або двостороннє (транзистори з близькими характеристиками) замінювані. Облік наявності деталей на складі, пошук відповідних деталей для ремонту.

43. Довідник нумізмата

Монети: країна, номінал, рік випуску, кількість випущених монет, особливості. Колекціонери: країна, ім'я, контактні координати, наявність рідкісних монет в колекції. Власна колекція.

44. Довідник філателіста.

Марки: країна, національна вартість, рік випуску, тираж, особливості. Філателісти: країна, ім'я, контактні координати, наявність рідкісних марок в колекції. Власна колекція.

45. Довідник любителя живопису.

Художники з анкетними даними та стилями. Картини з посиланням на художників, датою створення, жанром. Колекціонери та музеї: наявність оригіналів картин і копій. Аукціонери і комісіонки: дата проведення, список виставлених шедеврів і ціни на них. Власна колекція.

46. Так-тіклі. Ця гра схожа на шашки і на хрестики нулики.

Ігрове поле - це 16 клітин (4x4). Кожен з гравців ставить свої чотири фішки по обидва боки поля, через одну з фішками противника, як показано на малюнку. За один хід можна пересунути свою фішку на одну вільну клітину вгору або вниз, вправо або вліво, але не по діагоналі. Знімати фішки супротивника не можна. Завдання полягає в тому, щоб розташувати три фішки свого кольору в один ряд по вертикалі, горизонталі або діагоналі швидше противника.

47. Хід конем. Мета гри: набрати більше очок, ніж суперник.

Правила гри: Гравці по черзі рухають коня. Напрямок ходу довільно, але має відповідати правилам руху шахового коня. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

48. Хід слоном. Мета гри: набрати більше очок, ніж суперник.

Правила гри: Гравці по черзі рухають слона. Напрямок ходу довільно, але має відповідати правилам руху шахового слона. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

49. Хід королеви.

Мета гри: набрати більше очок, ніж суперник. Правила гри: Гравці по черзі рухають ферзя. Напрямок ходу довільно, але має відповідати правилам руху шахового ферзя. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

50. Хід тури. Мета гри: набрати більше очок, ніж супротивник.

Правила гри: Гравці по черзі рухають туру. Напрямок ходу довільно, але має відповідати правилам руху шахової тури. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

51. Хід амазонки.

Мета гри: набрати більше очок, ніж суперник. Правила гри: Амазонка - уявна шахова фігура, яка об'єднує властивості ферзя і коня. Гравці по черзі рухають амазонку. Напрямок ходу довільний, але має відповідати зазначеним правилам. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

52. Шибениця.

Мета гри: вгадати слово, загадане комп'ютером. Правила гри: Можна вибрати будь-яку літеру. Якщо така буква є, всіаналогічні букви в слові будуть відкриті.

53. Решітка.

Ігрове поле має розмір 10x10 з випадково розподіленими числами в них. Мета гри: Набрати більше очок, ніж супротивник. Правила гри: Перший гравець ходить по горизонталі, другий - по вертикалі. Перший гравець може вибрати будь-яку клітину з цифрою на поточному горизонталі. Якщо колір клітини червоний, то кількість очок, вказане на ній, додається до рахунку гравця, якщо синій - віднімається. Після ходу першого гравця вертикаль, на якій знаходиться вибрана ним клітина, стає поточною і хід переходить до супротивника, який може вибрати будь-яку клітину з цифрою на поточному вертикалі і т.д.

54. Діагоналі.

Ігрове поле має розмір 6x6. Мета гри: Зайняти якомога більше місця на дошці. Правила гри: Коли настає Ваша черга ходу, Ви можете поставити одну фішку на будь-яку клітину дошки.

Якщо після Вашого ходу, один чи кілька фішок супротивники опинилися між вашими фішками, все етіфішки суперника переходять до Вас.

55. Double-dealing fellows.

Ігрове поле має розмір 6x6. Мета гри: зайняти на дошці простір більше, ніж супротивник. Правила гри: можна рухати яку свою фішку на наступну клітину дошки, при цьому фішка розділиться на двоє, тобто з'явиться в новій позиції і залишиться на старій одночасно. Також можна перестрибувати через одну клітку, але в цьому випадку роздвоєння фішки не відбувається. Якщо сусідні клітини зайняті суперником, вони переходять до Вас.

56. Сенегальські шашки. Розмір ігрового поля - 5x6 клітин.

Кожен гравець отримує по 14 шашок (черепашки і каміння), які розставляються так, щоб заповнилися дві крайні верхні і нижні лінії шашками

відповідного кольору. У центрі шашки розміщуються з країв, так щоб утворилося два порожніх поля, з яких починається гра. Рухатися можна тільки на порожнє поле попереду, праворуч або ліворуч, по одній клітці за хід. Не можна рухатися назад або по діагоналі. Якщо з'являється можливість перестрибнути через шашку супротивника, гравець зобов'язаний зробити це. У такому випадку, шашка супротивника вважається убитою і знімається з дошки. Гра закінчується, коли шашки одного з партнерів просунулися так далеко, що більше не можуть з'їсти фішок. Тоді підраховується кількість з'їдених шашок і визначається переможець.

57. Японські шашки.

Ігрове поле має розмір 10x10. Мета гри: поставити 5 фішок в ряд (по вертикалі, горизонталі, діагоналі). Правила гри: Фішку можна ставити на будь-який вільний місце ігрового поля.

58. Куточки.

Ігрове поле має розмір 7x7. Спочатку фішки розташовуються двома блоками 3x3 в протилежних кутках дошки. Мета гри - перемістити всі фішки в протилежний кут. Шашку можна пересувати двома способами: на одну клітку вліво, вправо, вгору або вниз, або перестрибувати через інші фішки.

59. Блокада.

Ігрове поле має розмір 7x7. Спочатку фішки розташовуються двома блоками 3x3 в протилежних кутках дошки. Мета гри: пересунути всі шашки в протилежний кут швидше противника. Правила гри: Через порожні сусідні поля можна просувати фішку до трьох клітин за хід. Можна перестрибувати, через фішку. У цьому випадку фішка, через яку гравець перестрибнув, відправляється в дім - в кут, з якого вона почала свій рух. Кожен гравець може зробити не більше 10 стрибків за гру.

60. ЛОГІЧНА ГРА " хрестики-нулики " на необмеженому полі.

Додаток є реалізацією відомої логічної гри " Хрестики-нулики ". У даній реалізації передбачається гра двох гравців на необмеженій полі. Мета гри : побудувати безперервну лінію з п'яти або більше фішок (хрестиків або

нуликів) по горизонталі, вертикалі або діагоналі. Можливі два режими гри, коли виграє той, хто першим збудує лінію, або хто більше набере очок за певну кількість часу. У другому випадку кількість побудованих ліній не обмежена, кожна фішка в лінії приносить гравцеві одне очко. Додаток повинен забезпечувати початок нової гри на чистому полі, а також перевірку відповідності дій гравців правилами гри та умови закінчення гри. Роль одного з гравців (за вибором користувача) може виконувати комп'ютер. Для виключення можливості утворення ізольованих ігор на одному полі і невинуватого збільшення розміру поля слід встановити максимально допустиму відстань (не більше п'яти) від вже існуючих фішок до нової, що розміщується гравцем.

61. ЛОГІЧНА ГРА "МОРСЬКИЙ БІЙ".

Розроблюється, являє собою програмну реалізацію відомої логічної гри. Додаток повинен забезпечувати : розстановку, кораблів на ігровому полі 10x76.2. Вибір противника (людина, комп'ютер). Зміна інтерфейсу залежно від вибору противника. Фіксацію імен супротивників і число перемог.

62. ЛОГІЧНА ГРА "ВОДОПРОВІДНИК"

Суть гри полягає в споруді трубопроводу від одного краю ігрового поля до іншого за певний час. Грає один гравець. Комп'ютер пропонує черговий вузол трубопроводу, гравець може приєднати його до кінця трубопроводу, попередньо повернувши його потрібною стороною. У певний момент часу з початку гри, в трубопровід починає подаватися вода, яка повільно заповнює трубопровід. Якщо трубопровід буде заповнений до того, як гравець закінчить будівництво - гравець програв..

63. ЛОГІЧНА ГРА " BREAK HOUSE "

Гравець переходить від рівня до рівня. Кожен рівень являє собою клітинне ігрове поле- лабіринт. Клітка може бути прохідний, або на неї може стояти блок. Блоки можуть бути багатоклітинними. Також, є рухомі по простих траєкторіях (вперед-назад) блоки, якщо рухомий блок притисне гравця до стіни або іншого блоку - гравець буде розчавлений. Ігрове поле

містить джерело лазера, набір цілей, набір дзеркал. Дзеркала й цілі виставляються на полі випадковим чином на початку гри. Кожне дзеркало може повертатися в будь-якому напрямку з кроком 30гр. Гравець повинен повертаючи дзеркала по черзі спалити всі цілі.

64. ЛОГІЧНА ГРА "Манкала "

Варіація африканської гри " Авеля " (інша назва " Манкала "). Ігрове поле складається з двох рядів, по 6 лунок у кожному. Кожна лунка на початку гри містить 4 каменю. Нижній ряд належить гравцю, верхній - його противникові. Гравці роблять ходи по черзі. Для ходу гравець бере камені з будь-якої своєї не порожньої лунки, і розкладає їх по одному в кожную наступну лунку, рухаючись проти годинникової стрілки. Початкова лунка завжди пропускається. Якщо гравець кладе останній камінь у лунку супротивника і лунка містить 2 або 3 каменю (включаючи щойно покладений) то гравець збирає камені з лунок, рухаючись за годинниковою стрілкою, до тих пір, поки два вищенаведених умови дотримуються. Якщо гравець не може зробити черговий хід, він збирає всі залишилися камені, гра закінчується і визначається переможець.

65. ЛОГІЧНА ГРА " ЗВОРОТНІЙ ТЕТРІС "

Тетріс навпаки. Гравець вибирає фігуру і кидає, комп'ютер намагається встановити її усюди. Мета гри - завалити комп'ютер. Розроблюється, являє собою програмну реалізацію логічної гри " Шашки ". Необхідно реалізувати підмножина правил класичних шашок на дошці 8x8.

66. ЛОГІЧНА ГРА "ШАШКИ "

Відома гра на шашковому полі. Противники розставляють свої шашки симетрично в протилежних кутках дошки у вигляді прямокутників 3x4. Ходи здійснюються по черзі. За один хід можна рушити одну шашку на одну вільну клітину. Якщо до шашці примикає шашка супротивника, а за нею є вільне поле, можна « стрибнути » через шашку супротивника на це вільне поле. За один хід можна робити кілька послідовних стрибків однією шашкою. Виграє той, хто швидше вибудує свої шашки на позиції противника.

67. ЛОГІЧНА ГРА "KBounce "

У KBounce грають на полі, оточеному стіною, з двома або більше м'ячами, які відскакують від стін. Розмір поля зменшується, якщо ви створюєте стіну і при цьому ніякої куля не чиниться потрапили в пастку. Щоб пройти рівень, гравцеві за даний час потрібно зменшити розмір поля принаймні на 75 %. На кожному наступному рівні до гри додається по одній кулі, а гравцеві дається більше життів і часу. Підрахунок очок залежить від того, наскільки ви зменшили площу. Нові стіну будуються по клацанню лівою кнопкою миші на вільному просторі поля. Після клацання, починаючи від клітини, де він був зроблений, в протилежних напрямках починають будуватися дві частини стіни. За один проміжок часу може будуватися тільки одна стіна.

67. ЛОГІЧНА ГРА "KSnake "

Правила гри: Щоб виграти в KSnake, вам потрібно з'їсти всі яблука в кімнаті і вийти через двері, яка відкриється вгорі. З кожним з'їденим яблуком ви стаєте довшим. Якщо ви вріджеться в стіну, ви помрете. Якщо ви вріджеться в себе, ви помрете. Якщо вам в голову потрапить м'яч, ви помрете. Якщо ви занадто довго не будете їсти яблука, з'являться нові.

75. ГРА - АРКАДА " КОСМІЧНА ДУЕЛЬ ".

Кожен гравець керує одним кораблем. Корабель може повертатися, прискорюватися, стріляти і закладати міни. У кожного корабля є певна кількість енергії. Кораблю потрібна енергія для поворотів, прискорення, стрільби та закладання мін. Корабель отримує її за допомогою своїх сонячних батарей. Кількість енергії, яку корабель отримує, залежить від відстані та напрямки, в якому знаходиться сонце. Корабель отримує більше енергії поблизу сонця і менше біля кордону. Він отримує повну кількість енергії, якщо сонце світить прямо на батареї, частина енергії, якщо воно світить під кутом і, зовсім не отримує її, якщо сонце світить на торець батареї. Якщо у корабля закінчилася енергія, він втрачає керування і не може стріляти. Зіткнення зі своїми або чужими снарядами і мінами зменшує здоров'я корабля.

Якщо зіткнуться два кораблі, то слабший корабель буде знищений, і здоров'я сильнішого зменшиться на значення здоров'я слабкого корабля, плюс деяке значення (Збиток при аварії). Корабель знищується, коли він влітає у сонці. Снаряди літають навколо сонця як кораблі. У хвилинах є деяка кількість енергії, щоб залишатися на одному місці. Коли енергія закінчується, міна падає на сонці. Мінах, розташованим біля сонця, потрібно більше енергії, ніж розташованим далеко від нього. Міни можуть бути знищені снарядами. Час від часу на полі бою з'являються заправки.

Мета гри прожити довше, ніж супротивник. Тому не можна врізатися в стіни, себе і супротивника. Після початку раунду гравці вже не можуть зупинити свій рух (якщо не натиснута пауза). Ви можете тільки намагатися уникати зіткнення, постійно змінюючи напрямок руху. Крім того, ви можете заважати своєму противнику. Ви можете збільшувати швидкість свого руху, натискаючи на клавішу прискорення. Раунд починається, коли всі гравці натиснуть на якого-небудь клавішу напрямку руху. Перший рух буде відбуватися в цей бік.

6 ПОРЯДОК ЗАХИСТУ КУРСОВОЇ РОБОТИ ТА ЇЇ ОЦІНКА

До захисту курсові роботи студент допускаються викладачем, який керує курсовим проектуванням. Необхідними умовами допуску до захисту є:

- а) наявність діючої програми, яка відповідає;
- б) правильно оформлений похідний код програми;
- в) відповідність пояснювальної записки стандарту.

Захист відбувається перед комісією з трьох викладачів у присутності студентів з академічної групи автора;

Студент під час доповіді повинен викласти основні результати курсової роботи. Орієнтовний час виступу - 10 хвилин. Після доповіді студент повинен відповісти на поставлені запитання. Якщо з відповідей на запитання з'ясується, що студент не є єдиним автором курсової роботи, вона відхиляється, а студент отримує нове завдання.

Курсова робота оцінюється за наступними критеріями:

- програма – до 60 балів;
- пояснювальна записка – до 20 балів;
- захист до 20 балів.

7 ПЕРЕЛІК ДЖЕРЕЛ, ЩО РЕКОМЕНДОВАНІ ДЛЯ ВИКОНАННЯ КУРСОВОЇ РОБОТИ

1. Бадд Т. Объектно-ориентированное программирование в действии: Пер. с англ./Тимоти Бадд. – СПб.: Питер, 1997.– 487 с.
2. Бондарев В.М., Программирование на C++. Учебное пособие.– Харьков: СМИТ, 2004г. – 294 с.
3. Гамма Э. И др. Приемы объектно-ориентированного программирования. - СПб. :Питер, 2001г.
4. Бублик В.В. Б90 Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТкнига, 2015. – 624 с.: іл.
5. Stroustrup, Bjarne. The C++ programming language / Bjarne Stroustrup.—Fourth edition. Addison-Wesley publishing. 2013 – 1366 p.
6. Вирт Никлаус. Алгоритмы и структуры данных : пер. с англ. / Н. Вирт. — 2-е изд., испр. — СПб. : Невский Диалект, 2001. — 351 с.
7. Герберт Шилдт. C++ A Beginner's Guide: Step by Step. – М.: ЭКОМ Паблицерз., 2013. – 640 с. ISBN - 978-5-9790-0127-2, 0-07-219467-7.
8. C++. Теорія та практика : Навч. посібник / [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.] ; за ред.О. Г.Трофименко. – Одеса: ОНАЗ ім.. О.С. Попова, 2011. – 588 с.
9. Липпман Стенли Б., Лажойе Жози, Му Барбара Э. Язык программирования C++. Базовый курс. М.: Вильямс, 2014. — 1120 с. — 5-е изд. — ISBN: 9785845918390, 9780321714114.
10. Основи програмування : навчальний посібник для вузів / Броніслав Генріхович Пекарський . – Київ : Кондор, 2008 . – 363 с. : іл. - Бібліогр.: с.362-363(17 назв) . – На укр. яз. - ISBN 966-351-075-0 .
11. Дейтел Х., Дейтел П. Как программировать на C++. Третье издание.-М.: ЗАО „Издательство БИНОМ”, 2001г.-1152с.
12. Дьюхарт С. Старк К. Программирование на C++. Киев: Диалог 6. Ирэ Пол. Объектно-ориентированное программирование с использованием C++: Пер. с англ. - Киев: НИИПФ ДиаСофт Лтд, 1995. 480с.

13. Страуструп Б. Язык программирования С++. Специальное издание.-М., СПб.: "Издательство БИНОМ" – „Невский Диалект”, 2001г.-1099 с.
14. Шмак Н.К. Основы С++ и объектно-ориентированного программирования. Киев: Диалектика, 1996г.

8 ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Звіти у сфері науки і техніки. Структура та правила оформлювання[Електронний ресурс]. Режим доступу [www.
URL=\[http://www.knmu.kharkov.ua/attachments/3659_3008-2015.PDF\]\(http://www.knmu.kharkov.ua/attachments/3659_3008-2015.PDF\)](http://www.knmu.kharkov.ua/attachments/3659_3008-2015.PDF) - 08.08.19.
2. Бібліографічне посилання Загальні положення та правила складання[Електронний ресурс]. Режим доступу [www.
URL=\[http://library.nlu.edu.ua/Biblioteka/sait/DSTU_8302-2015.pdf\]\(http://library.nlu.edu.ua/Biblioteka/sait/DSTU_8302-2015.pdf\)](http://library.nlu.edu.ua/Biblioteka/sait/DSTU_8302-2015.pdf) - 08.08.19.

ДОДАТОК А

Зразок титульної сторінки курсової роботи

ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО «ПРИВАТНИЙ ВИЩИЙ
НАВЧАЛЬНИЙ ЗАКЛАД «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ
ТА ІНФОМАЦІЙНИХ ТЕХНОЛОГІЙ»

ДО ЗАХИСТУ ДОПУЩЕНИЙ

Зав. Каф. КІ

д.т.н., проф. Переверзєв А.В.**КУРСОВА РОБОТА**

З дисципліни: «Об'єктно-орієнтовне програмування»

ТЕМА: Розробка чату за технологією Web-socket

Виконав

ст.гр. ІПЗ-113

А.В. Шевченко

Керівник

Доц.

О.А. Жеребцов

Поточний рік

ДОДАТОК Б

Зразок титульної сторінки завдання на курсову роботу

ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО «ПРИВАТНИЙ ВИЩИЙ
НАВЧАЛЬНИЙ ЗАКЛАД «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА
ІНФОМАЦІЙНИХ ТЕХНОЛОГІЙ»

Кафедра комп'ютерної інженерії

ЗАВДАННЯ:

1. Тема роботи
2. Термін подання студентом закінченої роботи на кафедрі
3. Вихідні дані до роботи (проекту) (визначаються кількісні або (та) якісні показники, яким повинен відповідати об'єкт розробки)
4. Вимоги до змісту (перелік питань, що їх належить розробити) (визначаються назви розділів або (та) перелік питань, які повинні увійти до тексту роботи)
5. Індивідуальний план виконання

Керівник (ПБ, підпис)

З планом ознайомлений (ПБ, підпис)

ДОДАТОК В. Приклад оформлення реферату

РЕФЕРАТ

Пояснювальна записка складається з: 30 с., 12 рисунків, 3 таблиць, 5 лістингів, 7 формул, 8 джерел.

Об'єкт дослідження – візуалізатори алгоритмів.

Мета роботи – аналіз візуалізаторів та розробка додатку візуалізації алгоритмів сортування.

Методи дослідження – методи теорії алгоритмів та мов програмування, методи комп'ютерної графіки.

Розглянуто опис поточного стану в області візуалізаторів і систем візуалізації. Зокрема, розглянуто застосування візуалізаторів в навчальному процесі і виникаючі при цьому вимоги до візуалізаторів.

Наводиться аналіз візуалізаторів алгоритмів з точки зору висунутих вимог. Розглядається оцінка складеності алгоритмів сортування.

Проводиться аналіз алгоритмів сортування. Обґрунтовується вибір технології та середовища розробки додатку. Розроблено додаток для візуалізації алгоритмів сортування.

ВІЗУАЛІЗАЦІЯ, ВІЗУАЛІЗАТОР, АЛГОРИТМ СОРТУВАННЯ, JAVASCRIPT.

ДОДАТОК Г. Приклади специфікації

Тема: Інформаційна система навчального закладу

Спрощені вимоги до програмного забезпечення

Функціональні вимоги:

1. Управління викладачами
 - 1.1. Можливість додавати викладача
 - 1.2. Можливість видаляти викладача
 - 1.3. Можливість змінити данні викладача
 - 1.4. Можливість перегляду списку всіх викладачів
 - 1.5. Можливість перегляду даних вказаного викладача
2. Управління студентами
 - 2.1. Можливість додавати студента
 - 2.2. Можливість видаляти студента
 - 2.3. Можливість змінити данні студента
 - 2.4. Можливість перегляду списку всіх студентів
 - 2.5. Можливість перегляду даних вказаного студента
3. Управління групами
 - 3.1. Можливість додавати групу
 - 3.2. Можливість видаляти групу
 - 3.3. Можливість змінювати дані групи
 - 3.4. Можливість перегляду списку даних групи
 - 3.5. Можливість перегляду даних певної групи
4. Управління навчальним процесом
 - 4.1. Можливість додавання предмета вивчення до даних груп
 - 4.2. Можливість видалення предмета вивчення з даних груп
 - 4.3. Можливість додавання викладача даного предмета вивчення
 - 4.4. Можливість заміни викладача даного предмета вивчення
 - 4.5. Можливість додавання студента до існуючої групи
 - 4.6. Можливість видалення студента з існуючої групи

5. Пошук

5.1. Можливість пошуку студента за його даним(прізвище, ім'я)

5.2. Можливість пошуку студентів певної групи

5.3. Можливість пошуку студентів певного викладача

5.4. Можливість пошуку студентів за предметом вивчення

*викладач викладає один предмет

**група повинна мати список викладачів та студентів

Нефункціональні вимоги:

1. Вимоги до зовнішнього інтерфейсу

1.1. Зовнішній інтерфейс користувача має бути командним

1.2. Застосування має бути консольним

2. Дані повинні зберігатись у файлах після виходу з програми.

3. Система має забезпечити правильність введених даних

4. Логічна структура даних

4.1. Інформаційний об'єкт викладач

4.2. Інформаційний об'єкт студент Номер залікової книжки Адреса Телефон

4.3. Інформаційний об'єкт група

4.4. Інформаційний об'єкт предмет Групи