# 1. Порядок виконання курсового проекту

Основні етапи виконання курсового проекту:

1. формулювання вимог до програми;
2. проектування програми;
3. кодування програми;
4. тестування програми;
5. оформлення пояснювальної записки;
6. захист роботи.

***Формулювання вимог до програми***

На етапі формулювання вимог студент повинен:

* сформулювати призначення програми;
* визначити перелік функцій, які виконує програма;
* запропонувати інтерфейс користувача, який надасть можливість останньому застосувати дані функції.

Формулювання вимог є самостійним етапом роботи і завершується створенням «Специфікації програми». Специфікація програми складається з послідовного опису усіх виділених функцій, та пов’язаного з ними графічного інтерфейсу користувача.

## *****Проектування програми*****

**На етапі проектування визначається архітектурний стиль, об’єктна структура програми (**перелік класів, що включають поля, методи, властивості, конструктори**; взаємодію класів згідно принципів ООП), а також формат похідних,** вихідних та внутрішніх даних.

***Кодування програми***

На етапі кодування створюється вихідний код програми. Він повинен бути прокоментований.

***Тестування програми***

Функціональне тестування перевіряє, як програма виконує свої функції в умовах коректних та некоректних дій користувача.

***Пояснювальна записка***

Оформлення пояснювальної записки є останнім етапом виконання курсового проекту і має надати студентові навичок документування програмного продукту.

## ****2. Розробка програми****

### 

### ****Загальні правила****

При розробці індивідуального завдання курсовго проекту використати 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, глобальних функцій, створення об'єктів та організацію взаємодії між ними в головній програмі для досягнення поставленої мети.

### 

### 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 спільне (дані, властивості, функціональність). У більшості випадків об'єкти будуть більш чи менш подібні між собою. Деколи в групах можна 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 продемонструвати гру 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:*

1. З чого вона складається?

Сюди входять різноманітні видимі елементи: дошка, клітинки на ній, різні фігури, позначення рядів на дошці.

1. 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.

1. Які дії повинні виконувати об'єкти?

Вони повинні реагувати або не реагувати на натискання мишкою чи при настиканні відповідної клавіші клавіатури, відображатися, зникати, деякі ще повинні переміщуватись, причому по-різному.

1. 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 «х», «у», «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.

1. 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 приклад, далеко не повний, він дає можливість зрозуміти лише основний підхід до проектування класів. Пофантазувавши трохи, можна отримати 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.

Етап «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.

### Розробка методів

На даному етапі слід описати принцип дії основних методів та відмінності поліморфних методів від методів базових класів. 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) – запис тих дій, 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.

### Створення об'єктів і розробка головної програми

На даному етапі слід описати процес створення самих об'єктів у програмі, функціональність та дані яких розроблені на попередніх етапах. 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».

Стосовно розглянутого прикладу із шахами, даний етап 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 етапу. Такий опис повинен бути достатнім для розуміння всіх деталей реалізації алгоритму та можливих випадків, що зустрічаються при роботі програми.

### Опис файлів даних та інтерфейсу програми

Якщо програма використовує файли як джерело вхідних даних або для зберігання проміжних чи кінцевих результатів роботи, то на даному етапі слід навести опис формату цих файлів.

Особливо важливий даний етап 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 етапі слід описати призначення елементів меню, роботу з ними, параметри, що вибираються в діалогових вікнах тощо. Тут же слід описати інтерфейс програм, які працюють з параметрами командного рядка.

1. **Вимоги до програми**

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 загальноприйнятою: F1 – допомога; Entег – згода, завершення введення; Еsс – відмова, повернення до попереднього вузла гілки алгоритму (з відновленням екранної форми); Таb – перехід до наступного поля, вікна і т.д.; Shift & Tab – повернення до попереднього.

## *Вимоги до курсового проекту (на 3, 4, 5)*

Вимоги до оцінки 3:

* Програмна реалізація має складатися не менше ніж з 1 класу і 1 структури.
* Дані мають бути захищенні (ніяких public даних). Зміну і доступ до кожного поля реалізувати через get i set.
* Обов’язкове збереження даних у файл та зчитування даних з файлу.
* Реалізувати сортування даних (згідно будь-якого алгоритму сортування, розглянутого на парах).
* Реалізувати пошук і редагування даних за будь-яким полем класу.
* Програма повинна містити декілька пунктів меню та здійснювати перехід між ними.

Вимоги до оцінки 4:

* Програмна реалізація має складатися не менше ніж з 3 класів.
* Дані мають бути захищенні (ніяких public даних). Зміну і доступ до кожного поля реалізувати через get i set.
* Обов’язкове збереження даних у файл та зчитування даних з файлу. Використання декількох файлів (дані згруповані за деякою ознакою).
* Обов’язкове використання наслідування при реалізації взаємозв’язків між класами.
* Реалізувати сортування даних (згідно будь-якого алгоритму сортування, розглянутого на парах).
* Реалізувати пошук і редагування даних за декількома полями класу.
* Обов’язкове використання графічного інтерфейсу.
* Обов’язкова перевірка коректності вводу даних та обробка виключень.
* Програма повинна містити декілька пунктів меню та здійснювати перехід між ними.

Вимоги до оцінки 5:

* Програмна реалізація має складатися не менше ніж з 4-5 класів.
* Дані мають бути захищенні (ніяких public даних). Зміну і доступ до кожного поля реалізувати через get i set.
* Обов’язкове використання наслідування при реалізації взаємозв’язків між класами, застосування віртуального наслідування та інтерфейсів.
* Обов’язкове використання графічного інтерфейсу.
* Програма повинна містити декілька пунктів меню та здійснювати перехід між ними.
* Обов’язкова наявність «посібника користувача» – інструкції з використання програми.

1. **Теми курсових проектів**

Таблиця 1 – Тематика курсових проектів (орієнтовна оцінка 3)

|  |  |  |
| --- | --- | --- |
| **№** | **Тема** | **Опис** |
| 1 | Система управління базою даних “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 за різними критеріям; інвентаризація бібліотеки. |
| 2 | Система управління базою даних “Обмін житла” | База пропозицій по обміну: район, площа, планування і т.д.; вимоги до варіантів обміну. Реєстрація клієнтів, вибір відповідних варіантів, видалення даних після обміну або відмови. |
| 3 | Система управління базою даних “Довідник споживача” | База підприємств побутового обслуговування міста: назва, розряд, адреса та телефони, спеціалізація, перелік надаваних послуг, форма власності, години і дні роботи. Пошук підприємств по заданій послузі та іншими ознаками. |
| 4 | Система управління базою даних “Довідник покупця” | Бази торгових точок міста: назва, адреса і телефони, спеціалізація, форма власності, час роботи, відстань. Вибір магазинів за довільним шаблоном. |
| 5 | Система управління базою даних “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. |
| 6 | Система управління базою даних “Довідник поштової індексації” | Область, район, населений пункт, поштовий індекс. Пошук за будь-якими полями (крім останнього); ієрархічний зв'язок між полями (зверніть увагу, що, однакові назви міст та сіл є у різних областях). |
| 7 | Система управління базою даних “Довідник фірм” | Назва, адреса та телефони, перша особа, статус (форма власності), сировина, продукція. Вибір даних за довільним шаблоном. |
| 8 | Система управління базою даних “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. |
| 9 | Система управління базою даних “Довідник фаната” | База спортсменів: анкетні і антропологічні дані, громадянство, походження, вид спорту, клуб або команда, дані про особові рекорди або перемоги і т.д. Вибір даних за довільним шаблоном. Пошук рекордсмена в заданому виді спорту. |
| 10 | Система управління базою даних “Довідник радіоаматора” | Бази довідкових даних транзисторів, діодів, тиристорів і т.д.: марка, характеристики, гранично допустимі умови експлуатації, ціна, облік взаємозамінності і т.д. Підбірка по заданим вимогам. |
| 11 | Система управління базою даних “Довідник комерційних банків” | Найменування, адрес, статус (форма власності), умови зберігання коштів на особовому рахунку (річні відсотки для різних видів вкладів). Вибір банку з найбільшим відсотком для заданого типу вкладів. |
| 12 | Система управління базою даних “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. |
| 13 | Система управління базою даних “Крилаті фрази” | Довідник прислів'їв, приказок, афоризмів, каламбурів, інших словесних курйозів. Класифікація за авторами і джерелами, пошук по темах і ключовим словам. |
| 14 | Система управління базою даних оголошень | База оголошень (радіо-, газетних, в біжучому рядку) по всіх рубриках. Пошук для... (наприклад, «репетиторство & математика»). |

Таблиця 2 – Тематика курсових проектів (орієнтовна оцінка 4)

|  |  |  |
| --- | --- | --- |
| **№** | **Тема** | **Опис** |
| 15 | Система управління картотекою 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 законів, 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. |
| 16 | Система управління 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. |
| 17 | Система автоматизації центра зайнятості | 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 |
| 18 | Система управління власною довідниковою книгою | Анкетні дані, адреси, телефони, місце роботи або навчання, посада знайомих, колег і родичів, характер знайомства, ділові якості і т.д. Автоматичне формування привітання з днем народження (до поточної дати). Впорядкування за алфавітом і за датою останнього коригування. Пошук по довільному шаблону. |
| 19 | Система управління касою 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 (із зменшенням числа вільних місць), оформлення посадкової відомості. |
| 20 | Система управління магазином з одним продавцем | Комп'ютер замість касового апарату. База наявності товарів: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завозу. 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. |
| 21 | Система автоматизації відділу 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.  Сортування за прізвищем, окладом та стажем. Формування зарплатної відомості. Скорочення штатів: вибір для звільнення осіб пенсійного та передпенсійного віку, підготовка наказу. |
| 22 | Система створення генеалогічного дерево | Паспортні дані членів деякого родового клану; посилання на дітей (або на батьків). Пошук всіх нащадків або всіх предків для вказаної особи. Сортування за прізвищем, кількістю дітей, днем народження. Створення шаблонного дерева для зберігання генеалогічного дерева. |
| 23 | Система автоматизації складу товарів | База товарів, що зберігаються на складі: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завозу. Сортування за ціною. Реєстрація надходження товару (формування прибуткової накладної) і відвантаження (видаткова накладна). Вивід інвентарної відомості. |
| 24 | Система автоматизації робочого місця касиру автовокзалу | Розклад автобусів: номер рейсу, кінцевий і проміжний пункти, час відправлення. Сортування за часом та місцем. 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. |
| 25 | Система автоматизації робочого місця адміністратора 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 за довільною ознакою. |
| 26 | Система управління власним щоденним розкладом | База намічуваних заходів - дата, час і протяжність, місце проведення, з ким зустріч, пріоритетність. Сортування за пріоритетністю, назвою, часом, прізвищем. Автоматичне нагадування найближчої справи: за поточною датою і часом, видалення вчорашніх справ або перенесення на майбутнє. Аналіз "накладок" - перетинань планованих справ. Перегляд справ на завтра, післязавтра і т.д. |
| 27 | Система управління базою даних термінології деякої предметної області | База визначень будь-якої науки: вводиться термін, його тлумачення (визначення), посилання на використовувані терміни. Можливість перегляду всього ланцюжка від заданого терміна до первинних понять. |
| 28 | Система автоматизації робочого місця Шеф-кухара | База рецептур страв: розкладка, рецепт приготування. База продуктів на складі: найменування, ціна, кількість. Формування меню на день (на задане число осіб); звану вечерю. Підрахунок вартості. Перевірка достатності запасів: формування видаткової накладної на склад, коригування запасів. |
| 29 | Система автоматизації робочого місця лікаря (фармацевта) | 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. |
| 30 | Система управління базою даних зарахування абітурієнтів | База абітурієнтів; анкетні дані, сукупність оцінок за ЗНО, готовність вчитися на договірній основі, наявність оригіналів документів, подані документи за спеціальностями. Вибір для зарахування заданої кількості абітурієнтів; формування для співбесіди списку тих, хто набрав граничний прохідний бал, але не може платити за освіту. Сортування за прізвищами та за сумою балів. Складання рейтингу на зарахування, його коригування, зарахування до груп за спеціальностями. |
| 31 | Інформаційна система «Довідник абітурієнта» | База вузів; найменування, адреса, перелік спеціальностей, конкурс минулого року по кожній спеціальності (денна, вечірння, заочна форми), розмір оплати при договірному навчанні. Вибір за різними критеріями: все про даному вузі; все по даній спеціальності, пошук мінімального конкурсу з даної спеціальності та загалом. |
| 32 | Моделювання роботи банківської системи засобами ООП | Змоделювати декілька банків (2-3). Назва банку, види і умови рахунків та кредитування. Відомості про вкладників банку: номер особового рахунку, категорія вкладу, процент, паспортні дані, поточна сума внеску, дата останньої операції. Операції внесення та зняття суми, кредитування, автоматичне нарахування відсотків. Мультивалютний рахунок, зберігання у різних валютах, різні відсотки, перевід між банками (одного та різних клієнтів). |
| 33 | Система управління базою даних ломбарда | База клієнтів. 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. Вибір товару за різними критеріями. |
| 34 | Система управління базою даних операцій купівлі – продажу товарів (Інтернет магазин) | База продавців: найменування товару, обсяг партії при оптовому продажі, ціна, умови продажу - відвантаження, форма оплати, контактна адреса або телефон, примітка (наприклад, «посередників прошу, не турбувати»). База покупців: найменування товару, обсяг покупки, прийнятна ціна і форма оплати, контактна адреса 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. |
| 35 | Система управління базою даних автосалону | 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. |
| 36 | Система управління базою даних ріелтерської контори (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 40 тис. доларів. |
| 37 | Система управління базою даних “Довідник нумізмата” | Монети: країна, номінал, рік випуску, кількість випущених монет, особливості. Колекціонери: країна, ім'я, контактні координати, наявність рідкісних монет в колекції. Власна колекція. |
| 38 | Система управління базою даних “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. |
| 39 | Система бронювання номерів в готелі | Список готелів за містами, клас готелю. Список вільних номерів на кожен день: клас, число місць. Список людей, що забронювали номер: паспортні дані, дати приїзду і від'їзду, номер. Пошук людей, що бронювали та номера за довільною ознакою. Пошук вільних місць у готелях за довільними комбінаціями ознак. |
| 40 | Система управління базою даних “Довідник філателіста” | Марки: країна, національна валюта, рік випуску, тираж, особливості. Філателісти: країна, ім'я, контактні координати, наявність рідкісних марок в колекції. Власна колекція. |
| 41 | Система автоматизованого бронювання квитків на потяг | Розклад потягів: номер потягу, кінцевий і проміжний пункти, час відправлення, кількість вільних місць кожного класу, номер вагону і місце, вартість квитка. Сортування за напрямом, класом, вартістю і т.д. Вибір і бронювання рейсу до заданого пункту по заданій даті (при наявності вільних місць), оформлення квитків. Зняття броні і повернення коштів (різний процент в залежності від часу, що залишився). |
| 42 | Система управління базою даних  військовослужбовців | Анкетні дані, цивільна професія, військова спеціальність, освіта, звання та дата його отримання, посада, підрозділ, форма служби, особливості характеру і ставлення до служби. Формування списків мобілізації за військовою спеціальністю, місцем проживання, віком до заданого підрозділу, пошук офіцерського складу, новобранців і т.д. База даних військовослужбовців заданого підрозділу, зарахування призовників, звільнення у запас. |
| 43 | Система управління базою даних “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 житла: задоволення підходящих черговиків, переселення наступних в квартири, що звільнилися. Додавання заявників та видалення вибулих і задоволених. |
| 44 | Система управління базою даних “Довідник туриста” | База туристичних агенцій (назва, адреса) та пропонованих послуг: країна, місто (або маршрут круїзу), умови проживання та проїзду, дата відправлення та повернення, рівень готелю, умови харчування, додаткові вигоди (басейн), вартість путівки. Вибір путівки за ціновими межами, датами, умовами проживання, країною і т ін. Збереження результату вибору. |
| 45 | Система бронювання апартаментів | База оголошень про здавання апартаментів деякої країни: місто, умови проживання, кількість місць, вартість за добу, тиждень по періодам сезону, умови бронювання, додаткові вигоди (WiFi, кухня, пральна машина, стоянка, басейн т.ін.). Вибір апартаментів за ціновими межами, датами, умовами проживання, містом, бронювання та оплата, відмова від бронювання та повернення коштів. Збереження результату вибору. |
| 46 | Система управління базою даних “Каталог запчастин автомобіля” | Склад магазину запчастин. База деталей: країна, фірма-виробник, марка автомобіля, агрегат, вузол, деталь. Облік взаємозамінності. Пошук деталі, сортування за ціною, інвентаризація, формування замовлення. |
| 47 | Система електронного управління футбольною командою | База даних гравців. анкетні і антропологічні дані, громадянство, походження, роль на полі, клуб або команда з якої перейшов, умови трансферу. Результативність за сезон, кількість проведених ігор, травми. База даних вільних футболістів та тих що можна обміняти чи купити. Покупка і продаж гравців. |
| 48 | Інформаційна система навчального закладу | Система дає можливість вести базу даних студентів, викладачів та предметів. Шукати, змінювати дані і додавати студентів, викладачів та предмети і здійснювати запис студентів у групу вивчення деякого предмета. Один предмет – один викладач. |
| 49 | Система електронного каталогу товарів | База даних товарів за категоріями. База даних замовників. Можливість сформувати замовлення на декілька товарів. Формування відправної накладної. |
| 50 | Система управління базою даних “Довідник селекціонера” | База даних декількох селекціонерів. Найменування сорту будь-якої культури, автор, батьківські сорти, врожайність, характеристики плодів, морозостійкість, стійкість до шкідників і хвороб. Власна колекція. Пошук сортів, наявність сортів в того чи іншого селекціонера. Обмін культурами, інформація про урожайність. Вибір сортів, що володіють заданими властивостями. |
| 51 | Система управління студентською бібліотекою | База документів: автор, назва, рік видання, форма (електронна, друкована), кількість аркушів, носій. База студентів: прізвище, імя, номер залікової, курс, група, картка читача. Пошук, облік видачі та повернення книг. Список боржників. |
| 52 | Система замовлення страв в ресторані | Управління стравами: категорія, назва, ціна, вага, наявність, час приготування. Формування меню. Управління замовленнями, розрахунок вартості. Можливість зміни, пошуку для замовлення і страви. |
| 53 | Система створення тестів | Система що допомагає створювати тести, які складаються не більше ніж з 10 питань. Питання та відповіді можна додавати, змінювати, шукати, групувати і зберігати. Система запуску тесту із файлу з підрахунком кількості правильних відповодей та даних хто проходив із запам’ятовуванням. |
| 54 | Система управління базою даних  військомату | Військовослужбовці: анкетні дані, адреса 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. |

Таблиця 3 – Тематика курсових проектів (орієнтовна оцінка 5)

|  |  |  |
| --- | --- | --- |
| **№** | **Тема** | **Опис** |
| 55 | Гра – аркада «Космічна дуель» | *Правила.* 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 напрямку, 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 сонце. Снаряди літають навколо сонця як кораблі. Коли енергія закінчується, міна падає на сонце. Мінам, 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 і в 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 яку-небудь клавішу напрямку руху. Перший рух буде відбуватися в цей бік. |
| 56 | Гра «Break House» | *Правила.* Гравець переходить від рівня до рівня (передбачити не менше 3 рівнів). Кожен рівень являє собою клітинне ігрове поле – лабіринт. Клітка може бути прохідною, або на ній 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.  Гра вважатиметься завершеною коли гравець пройде останній рівень.  Приклад: <http://r-games.net/328-break-house.html> |
| 57 | Гра «KSnake (Черв’як)» | *Правила.* Щоб виграти в KSnake, вам потрібно з'їсти всі яблука в кімнаті і вийти через двері, які відкриються вгорі. З кожним з'їденим яблуком ви стаєте довшим. Якщо ви вріжетеся в стіну, ви помрете. Якщо ви вріжетеся в себе, ви помрете. Якщо вам в голову потрапить м'яч, ви помрете. Якщо ви занадто довго не будете їсти яблука, з'являться нові. |
| 58 | Гра «KBounce (невгамовні кульки)» | *Правила.* Два гравці 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 ніякій кулі не вдається потрапити в в стіну, що будується. Нову 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. 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.  Приклад: <http://bimball.com/bounce> |
| 59 | Гра «Манкала» | *Правила.* Старовина африканська гра. Ігрове поле складається з двох рядів, по 6 лунок у кожному. Кожна лунка на початку гри містить 4 камені. 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 кожну наступну лунку, рухаючись проти годинникової стрілки. Початкова лунка завжди пропускається. Якщо гравець кладе останній камінь у лунку супротивника і лунка містить 2 або 3 камені (включаючи щойно покладений ) то гравець збирає камені з лунок, рухаючись за годинниковою стрілкою, до тих пір, поки дві вище наведених умови дотримуються. Якщо гравець не може зробити черговий хід, він збирає всі камені, що залишилися, гра закінчується і визначається переможець.  Приклад: https://www.youtube.com/watch?v=4GYvxrrlZG4 |
| 60 | Гра «Водопровідник» | *Правила.* 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. Передбачити не менше 5 видів вузлів трубопроводу. 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. |
| 61 | Гра «Lines» | *Правила.* Ігрове поле 10х10 клітинок. Суть гри: на ігровому полі відображаються кульки різного кольору, гравцеві пропонується створити лінію з кульок однакового кольору, розташованих по горизонталі або по вертикалі. Створена лінія згорає, якщо кількість елементів в ній дорівнює 3. Після цієї процедури порожні місця на ігровому полі заповнюються новими елементами. Лінію можна створити шляхом перестановки по горизонталі або по вертикалі сусідніх елементів. Гра закінчується тоді коли не можливо створити жодної лінії. Додаток повинен забезпечувати можливість задання кількості кольорів елементів. Слід врахувати, що кольори кульок вибираються довільним чином виходячи з заданої кількості. Крім того, лінії (3 і більше елементів) одержувані при довільній розстановці елементів «згоряють». Обовязково програма повинна рахувати кількість кульок, що згоріли.  *Мета гри* полягає в тому, щоб набрати максимальну кількість очок.  Приклад: http://shariki.online/game/linii-98.html |
| 62 | Гра «Хрестики-нулики» на необмеженому полі | *Правила.* У даній реалізації передбачається гра двох гравців на необмеженому полі. 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 невиправданого збільшення розміру поля слід встановити максимально допустиму відстань (не більше п'яти) від вже існуючих фішок до нової, що розміщується гравцем.  *Мета гри* побудувати безперервну лінію з п'яти або більше фішок (хрестиків або нуликів) по горизонталі, вертикалі або діагоналі. |
| 63 | Гра «Хід слоном» | *Правила.* Гравці по черзі рухають слона. Напрямок ходу довільний, але має відповідати правилам руху шахового слона в шахматах. До суми очок гравця додається число, вказане в клітинці, на яку він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник. Гра завершується, коли оброблені всі клітинки або неможливо здійснити хід.  *Мета гри* набрати більше очок, ніж суперник. |
| 64 | Гра «Так-тіклі» | *Правила.* Ця гра схожа на шашки і на хрестики нулики. Ігрове поле – це 16 клітин (4х4). Кожен з гравців ставить свої чотири фішки по обидва боки поля, через одну з фішками противника. 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.  *Мета гри* полягає в тому, щоб розташувати три фішки свого кольору в один ряд по вертикалі, горизонталі або діагоналі швидше противника. |
| 65 | Гра «Блокада» | *Правила.* Ігрове поле має розмір 7х7. Спочатку фішки розташовуються двома блоками 3х3 в протилежних кутках дошки. Через порожні сусідні поля можна просувати фішку до трьох клітинок 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.  *Мета гри* пересунути всі шашки в протилежний кут швидше противника. |
| 66 | Гра «Куточки» | *Правила.* Ігрове поле має розмір 8х8. Спочатку фішки розташовуються двома блоками 3х3 в протилежних кутках дошки. 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 шашки 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.  *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.  Приклад: http://www.gambiter.ru/ugolki/item/22-pravila-igry-ugolki.html |
| 67 | Гра «Сенегальські шашки» | *Правила.* Розмір ігрового поля – 5х6 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 розміщаються 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.  Приклад: https://www.youtube.com/watch?v=kdV4RAP-p78 |
| 68 | Гра «Хід амазонки» | *Правила.* Амазонка – уявна шахова фігура, яка об'єднує властивості ферзя і коня. Гравці по черзі рухають амазонку. Напрямок ходу довільний, але має відповідати правилам ходу ферзя і коня (в шахматах). До суми очок гравця додається число, вказане в клітинці, на яку він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник. Гра завершується, коли оброблені всі клітинки або неможливо здійснити хід.  *Мета гри* набрати більше очок, ніж суперник. |
| 69 | Гра «Double-dealing fellows» | *Правила.* Ігрове поле має розмір 6х6. Можна рухати будь-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 cупротивника. Гра завершується, коли оброблені всі клітинки або неможливо здійснити хід.  *Мета гри* зайняти на дошці простір більше, ніж супротивник. |
| 70 | Гра «Діагоналі» | *Правила.* Ігрове поле має розмір 6х6. Коли настає ваша черга ходу, ви можете поставити одну фішку на будь-яку клітину дошки. Якщо після Вашого ходу, одна чи кілька фішок супротивника опинилися між вашими фішками, всі фішки суперника переходять до Вас.  Гра завершується, коли оброблені всі клітинки або неможливо здійснити хід.  *Мета гри* зайняти якомога більше місця на дошці. |
| 71 | Гра «Решітка» | *Правила.* Ігрове поле має розмір 10х10 з випадково розподіленими числами в них. Перший гравець ходить по горизонталі, другий – по вертикалі. Перший гравець може вибрати будь-яку клітину з цифрою на поточній горизонталі. Якщо колір клітини червоний, то кількість очок, вказана 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 поточній вертикалі і т.д. Гра завершується, коли оброблені всі клітинки або неможливо здійснити хід.  *Мета гри* набрати більше очок, ніж суперник. |
| 72 | Гра «Хід конем» | *Правила.* Гравці по черзі рухають коня. Напрямок ходу довільний, але має відповідати правилам руху шахового коня. До суми очок гравця додається число, вказане в клітинці, на яку він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник. Гра завершується, коли оброблені всі клітинки або неможливо здійснити хід.  *Мета гри* набрати більше очок, ніж суперник. |
| 73 | Гра «Шибениця» | *Правила.* Задається слово, яке потрібно розгадати за *n*-кроків (в два рази більше ніж кількість літер у слові). Користувач обирає літеру, яку він би хотів відкрити. Якщо обрана літера присутня у слові, то вона відкривається і всі аналогічні букви в слові будуть також відкриті. Гра завершується, коли слово вгадане або перевищено кількість кроків.  *Мета гри* вгадати слово, загадане комп'ютером. |
| 74 | Гра «Зворотній тетріс» | *Правила.* Тетріс навпаки. Грають два учасники. 1-й гравець вибирає фігуру і кидає, 2-й гравець намагається встановити її у склянку (прямокутне поле розміром *n*).  *Мета гри* – перемогти другого гравця. |
| 75 | Гра «Морський бій» | *Правила.* Додаток повинен забезпечувати: розстановку кораблів на ігровому полі 10х10, вибір противника (людина, комп'ютер), реалізацію ходу, фіксацію імен супротивників і число перемог. |
| **76** | **Гра «Шашки»** | *Правила.* 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 шашки 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. |
| **77** | Гра «Хід тури» | *Правила.* Гравці по черзі рухають туру. Напрямок ходу довільний, але має відповідати правилам руху шахової тури. До суми очок гравця додається число, вказане в клітинці, на яку він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник. Гра завершується, коли оброблені всі клітинки або неможливо здійснити хід.  *Мета гри* набрати більше очок, ніж суперник. |
| **78** | Гра «KURTAN» | *Правила.* Гравець переходить від рівня до рівня. Кожен рівень являє собою клітинне ігрове поле – лабіринт, в якому безладно розставлені ящики. Передбачити мінімум 2 рівні. Гравець повинен штовхаючи ящики чоловічком розташувати їх на своїх місцях (місця відзначені на лабіринті). Чоловічок може штовхати тільки один ящик. Якщо ящик впирається в стіну, то далі його штовхати не можна.  *Мета гри* знайти сховані під ящиками ключі від дверей або/і звільнити шлях для переходу на інший рівень.  Приклад: [https://www.youtube.com/watch?v=e0vjLZKVwik](http://old-game.org/kurtan-kurtan.html) |
| **79** | **Гра «Реверсі (Отелло)»** | Правила перегляньте у файлі, що додається.  Приклад:  <https://www.youtube.com/watch?v=0-s1SPFd3VA> |
| **80** | **Гра «Go»** | Правила перегляньте у файлі, що додається.  Приклад: <https://www.youtube.com/watch?v=W6Up4q86q7Y> |
| **81** | Гра «Китайські шашки» | Правила перегляньте у файлі, що додається.  Приклад: <https://www.youtube.com/watch?v=O3niNUg_M_c> |

# Приклад специфікації

**Тема: Система електронного каталогу товарів**

*Функціональні вимоги:*

1. Управління категоріями

* Можливість додавати категорію
* Можливість видаляти категорію
* Можливість змінитикатегорію
* Можливість перегляду категорії
* Можливість перегляду всіх категорій

1. Управління товарами

* Можливість додавати товар у категорію
* Можливість видаляти товар з категорії
* Можливість змінювати дані товару
* Можливість змінювати кількість товару на складі
* Можливість переглянути дані конкретного товару
* Можливість переглянути список всіх товарів
* Можливість відсортувати список по назві
* Можливість відсортувати список по бренду
* Можливість відсортувати список по ціні

1. Управління замовниками

* Можливість додавати замовників
* Можливість видаляти замовників
* Можливість змінювати дані замовників
* Можливість переглянути дані конкретного замовника
* Можливість переглянути список всіх замовників
* Можливість відсортувати список по імені
* Можливість відсортувати список по прізвищу

1. Управління замовленнями

* Можливість замовити товар у будь-якій кількості
* Можливість замовити декілька товарів
* Можливість відмінити замовлення
* Можливість переглянути дані конкретного замовлення

1. Пошук

* Можливість пошуку по ключовому слову серед товарів
* Можливість пошуку по ключовому слову серед замовників
* Пошук найактивнішого замовника
* Пошук замовника, що найбільше витратив коштів
* Можливість переглянути довідку по командам

*Нефункціональні вимоги:*

1. Вимоги до зовнішнього інтерфейсу

* Зовнішній інтерфейс користувача має бути командним
* Додаток повинен бути консольним АБО із використанням графічного інтерфейсу

1. Дані повинні зберігатись у файлах після виходу з програми
2. Система має забезпечити правильність введених даних
3. Логічна структура даних (інформаційний об’єкт)

* Категорія: назва категорії, код категорії
* Товар: назва товару, ціна товару, штрих-код товару
* Замовлення: товар, кількість товару, сума замовлення
* Замовник: ПІБ замовника, телефон