МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**КУРСОВА РОБОТА**

з дисципліни «Об’єктно-орієнтоване програмування»

Варіант №7

|  |  |  |
| --- | --- | --- |
| **Керівник :**  доц. каф. БМК,  к.т.н. Алхімова С.М. |  | **Виконав:**  студент гр. БС-03, ФБМІ  Затуловський Г. А.  залікова книжка № БС-0307 |
| Допущено до захисту |  |
| І\_\_\_І \_\_\_\_\_\_\_\_\_ 2022 \_\_\_\_\_\_\_\_\_\_\_  підпис |  |
| Захищено з оцінкою  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_  оцінка підпис  І\_\_\_І \_\_\_\_\_\_\_\_\_ 2022 |  |

Київ-2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Факультет | ***біомедичної інженерії*** | | | | |
|  | (назва факультету, інституту) | | | | |
| Кафедра | ***біомедичної кібернетики*** | | | | |
|  | (назва кафедри) | | | | |
| Дисципліна | ***«Об’єктно-орієнтоване програмування»*** | | | | |
|  | (назва) | | | | |
| Курс | ***2*** | Група | ***БС-03*** | Семестр | ***4*** |

**ЗАВДАННЯ**

**на курсовий проект (роботу) студента**

|  |
| --- |
| ***Затуловського Георгія Андрійовича*** |
| (прізвище, ім’я, по батькові) |

.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Тема роботи: | | ***Розробка програмного забезпечення з використанням об’єктно-*** | | | |
| ***орієнтованого підходу.*** | | | | | |
|  | | | | | |
|  | |  | | | |
| 2. Строк здачі студентом закінченого проекту (роботи) | | | | | ***13.05.2022*** |
|  | |  | | | |
| 3. Вихідні дані до проекту (роботи): | | | | ***Варіант №7*** | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | |  | | | |
| 4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають | | | | | |
| розробці): | ***1. Визначення класів та програмування меню користувача (ієрархія*** | | | | |
| ***класів програми: абстрактний клас – Задача сервісного центру, класи-нащадки –*** | | | | | |
| ***Гарантійний ремонт техніки, регламентне обслуговування техніки). 2.*** | | | | | |
| ***Створення об’єктів та використання контейнерів (тип контейнера – Черга з*** | | | | | |
| ***пріоритетом). 3. Організація роботи з даними через файл. 4. Пошук даних у*** | | | | | |
| ***контейнері (запит для пошуку – Підрахувати загальну кількість гарантійних*** | | | | | |
| ***ремонтів техніки заданої фірми).*** | | | | | |
|  | |  | | | |
| 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): | | | | | |
| ***діаграми класів, послідовності та об’єктів в нотації UML*** | | | | | |
|  | | | | | |
|  | |  | | | |
| 6. Дата видачі завдання: | | | ***18.02.2022*** | | |

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| **№п/п** | **Назва етапів курсового проекту (роботи) та питань, які мають бути розроблені відповідно до завдання** | **Термін виконання етапу** | **Позначки керівника про виконання завдань** |
| 1. | Отримання завдання на курсову роботу | 18.02.2022 |  |
| 2. | Огляд технічної літератури за темою роботи | 25.02.2022 |  |
| 3. | Розробка першої частини курсової роботи (визначення класів та програмування меню користувача) | 11.03.2022 |  |
| 4. | Перший контроль за процесом виконання курсової роботи, консультація у викладача | 11.03.2022 |  |
| 5. | Розробка другої частини курсової роботи (створення об’єктів та використання контейнерів) | 25.03.2022 |  |
| 6. | Розробка третьої частини курсової роботи (робота з даними через файл) | 08.04.2022 |  |
| 7. | Другий контроль за процесом виконання курсової роботи, консультація у викладача | 08.04.2022 |  |
| 8. | Розробка четвертої частини курсової роботи (пошук даних у контейнері) | 22.04.2022 |  |
| 9. | Оформлення пояснювальної записки | 13.05.2022 |  |
| 10. | Захист курсової роботи | 26.05.2022 |  |

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(підпис)

**Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Алхімова Світлана Миколаївна

(підпис) (прізвище ім'я, по батькові)

**«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 р.**

**Анотація**

Затуловський Г.А. Розробка програмного забезпечення з використанням об’єктно-орієнтованого підходу.

Курсова робота з дисципліни «Об’єктно-орієнтоване програмування» присвячена питанню створення ієрархії класів, застосування спадкування та поліморфізму, розробки контейнера для зберігання даних множини об’єктів, які створюються користувачем. У курсовій роботі було виконано визначення класів (ієрархія класів програми: базовий клас – задача сервісного центру, класи-нащадки – Гарантійний ремонт техніки, регламентне обслуговування техніки), розроблено меню користувача, створено множину об’єктів та розроблено контейнер для її зберігання (тип контейнера – черга з пріоритетом), виконано серіалізацію даних елементів контейнера у файл та створення вмісту контейнера через десеріалізацію даних файлу, реалізовано пошук даних у контейнері (запит для пошуку – підрахувати загальну кількість гарантійних ремонтів техніки заданої фірми).

**Структура і обсяг роботи.** Курсова робота складається із вступу, трьох розділів, висновків, списку використаної літератури з 3 джерел і 2 додатків. Загальний обсяг курсової роботи становить 45 сторінок, основного тексту (без додатків) – 25 сторінок, ілюстрацій – 34, таблиць – 38.

**Annotation**

George Zatulovsky Software development using object oriented approach.

Coursework on the Object Oriented Programming course is devoted to the issue of creating a hierarchy of classes, applying inheritance and polymorphism, developing storage container for the collection of objects created by the user. In the coursework class definitions were performed (hierarchy of program classes: base class – the task of the service center, derived classes – warranty repair of equipment, routine maintenance of equipment), user menu was designed, set of objects was created and its storage container was developed (type of container – queue with priority), serialization of container data was done and container elements were created via deserialization of file data, data search in the container was performed (search query – сalculate the total number of warranty repairs of equipment of a given company).

**The structure and the amount of work.** Coursework consists of an introduction, three partitions, conclusions, list of used literature with 3 references, and 2 applications. The total volume of coursework is 45 pages, main text (without applications) – 25 pages, illustrations – 34, tables – 38.

**Аннотация**

Затуловський Г. А. Разработка программного обеспечения с использованием объектно-ориентированного подхода.

Курсовая работа по дисциплине «Объектно-ориентированное программирование» посвящена вопросу создания иерархии классов, применения наследования и полиморфизма, разработки контейнера для хранения данных коллекции объектов, которые создаются пользователем. В курсовой работе было выполнено определение классов (иерархия классов программы: базовый класс – задача сервисного центра, классы-потомки – гарантийный ремонт техники, регламентное обслуживание техники), разработано меню пользователя, создано множество объектов и разработан контейнер для их хранения (тип контейнера – очередь с приоритетом), выполнены сериализация данных элементов контейнера в файл и создание содержимого контейнера путем десериализации данных файла, реализован поиск данных в контейнере (запрос поиска – подсчитать общее количество гарантийных ремонтов техники заданной фирмы).

**Структура и объем работы.** Курсовая работа состоит из введения, трех разделов, выводов, списка использованной литературы из 3 источников и 1 приложения. Общий объем курсовой работы составляет 45 страниц, основного текста (без приложений) – 25 страниц, иллюстраций – 34, таблиц – 38.

Оглавление

[ВСТУП 10](#_Toc103109442)

[РОЗДІЛ 1 11](#_Toc103109443)

[СПЕЦИФІКАЦІЯ ВИМОГ ПРОГРАМНОГО ЗАСТОСУНКУ 11](#_Toc103109444)

[1.1 Функціональні вимоги 11](#_Toc103109445)

[1.2 Нефункціональні вимоги 18](#_Toc103109448)

# ВСТУП

Об'єкто-орієнтоване програмування — це модель програмування яка базується на стверджені того, що програма це сукупність об’єктів які взаємодіють між собою. Кожен об’єкт в цій моделі є незалежним, і він здатний отримувати, обробляти дані та відправляти ці дані іншим об’єктам. В ООП використано моделі успадкування, модульності, поліморфізму та інкапсуляції.

Завдання курсової робота з дисципліни «Об’єктно-орієнтоване програмування» полягає:

* + - Cтворити ієрархію класів
    - Реалізувати об'єкт так, щоб він використовував незалежну пам'ять, яка складається з інших об'єктів
    - Застосувати контейнера для зберігання об’єктів класів.
    - Розробити меню користувача, через який буде відбуватись робота з програмним застосунком.
    - Застосування спадкування та поліморфізму

Метою курсової роботи є застосування своїх знань, які ми отримали під час проходження курсу дисціпліни «Об'єктно-орієнтоване програмування».

# РОЗДІЛ 1

# СПЕЦИФІКАЦІЯ ВИМОГ ПРОГРАМНОГО ЗАСТОСУНКУ

## Функціональні вимоги

Варіанти використання програмного застосунку характеризуються у таблицях 1.1 – 1.9 та рисунках 1.1 – 1.9

* + 1. Опис створення об’єкта типу «регламентоване обслуговування техніки» та записання його у контейнер (чергу з пріоритетом)

*Таблиця 1.1*

**Створення об’єкта типу «регламентоване обслуговування техніки»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Індентифікатор | Назва | Мета | Дійові особи | Тип | Опис |
| 1 | Створення об’єкту «регламен-товане обслугову-вання техніки» | Створити об’єкт типу «регламентоване обслуговування техніки» та додати його в контейнер | Користувач | Основний | Програма створює об’єкт, користувач заповнює поля і об’єкт додається до контейнеру |

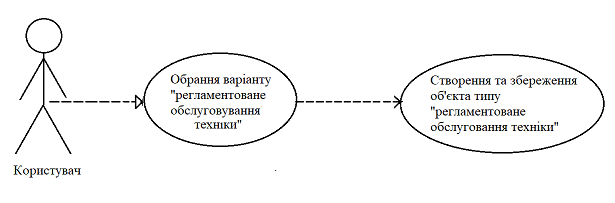


Рис. 1.1 - Створення об’єкту типу «регламентоване обслуговування техніки» та збереження в чергу з пріоритетом

* + 1. Опис створення об’єкта типу «гарантійний ремонт техніки» та записання його у контейнер (чергу з пріоритетом)

*Таблиця 1.2*

**Створення об’єкта типу «гарантійний ремонт техніки»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Індентифікатор | Назва | Мета | Дійові особи | Тип | Опис |
| 2 | Створення об’єкту «гарантійний ремонт техніки» | Створити об’єкт типу «гарантійний ремонт техніки» та додати його в контейнер | Користувач | Основний | Програма створює об’єкт, користувач заповнює поля і об’єкт додається до контейнеру |

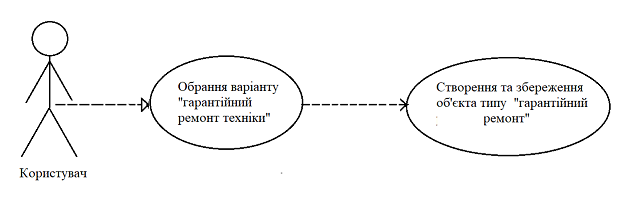


Рис. 1.2 - Створення об’єкту типу «гарантійний ремонт техніки» та збереження в чергу з пріоритетом

* + 1. Опис методу видалення першого об’єкта з контейнера (чергу з пріоритетом)

*Таблиця 1.3*

**Видалення об’єкта з контейнера**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Індентифікатор | Назва | Мета | Дійові особи | Тип | Опис |
| 3 | Видалення об’єкта з черги з пріоритетом | Видалення об’єкта з черги з пріоритетом | Користувач | Основний | У процесі роботи програми надає можливість видалити перший об’єкт з контейнеру |

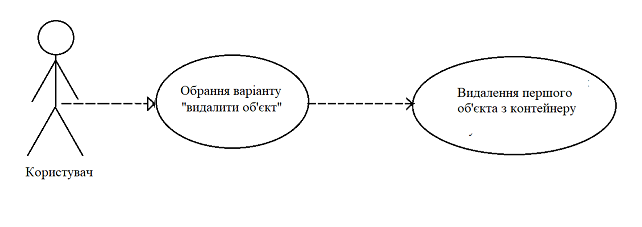


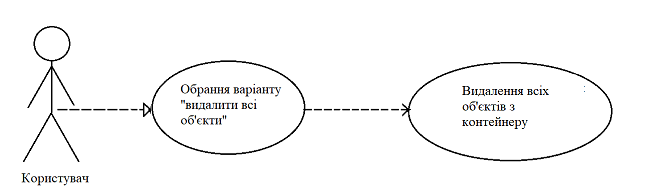
Рис. 1.3 - Видалення першого елементу в черзі з пріоритетом

* + 1. Опис методу видалення всіх об’єктів з контейнера (чергу з пріоритетом)

*Таблиця 1.4*

**Видалення всіх об’єктів з контейнера**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Індентифікатор | Назва | Мета | Дійові особи | Тип | Опис |
| 4 | Видалення всіх об’єкти з черги з пріоритетом | Видалення об’єкти з черги з пріоритетом | Користувач | Основний | У процесі роботи програми надає можливість видалити всіх об’єктів з контейнеру |

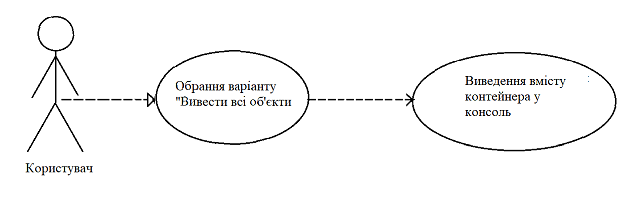
 Рис. 1.4 - Видалення всіх елементів в черзі з пріоритетом

* + 1. Опис видаленняоб’єктів з контейнера (чергу з пріоритетом)

*Таблиця 1.5*

**Виведення всіх об’єктів з контейнера**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Індентифікатор | Назва | Мета | Дійові особи | Тип | Опис |
| 5 | Виведення  всіх об’єктів  черги з пріоритетом  у консоль | Вивести у консоль всі елементи черги з пріоритетом | Користувач | Основний | Надає  можливість вивести інформацію  про  об’єкти з контейнеру  у консоль |



1.1.6. Запис об’єктів контейнера (черги з пріоритетом) в файл

*Таблиця 1.6*

**Запис об’єктів контейнера в файл**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Індентифікатор | Назва | Мета | Дійові особи | Тип | Опис |
| 6 | Запис об’єктів контейнера в файл | Зберегти всі об’єкти у файл | Користувач | Основний | Надає  можливість зберегти інформацію  про  об’єкти контейнера  у файл |

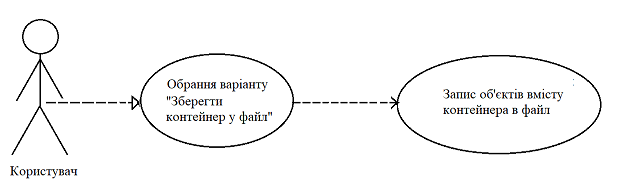
****

Рис. 1.6 - Запис об’єктів черги з пріоритетом в файл

1.1.7. Читання з файлу об'єктів та внесення їх в контейнер (черги з пріоритетом)

*Таблиця 1.7*

**Зчитування об’єктів контейнера з файлу**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Індентифікатор | Назва | Мета | Дійові особи | Тип | Опис |
| 7 | Зчитування об’єктів контейнера з файлу | Завантажити  інформацію  про об’єкти з файлу | Користувач | Основний | Надає  можливість завантажити інформацію  про всі  об’єкти контейнера  з файлу |

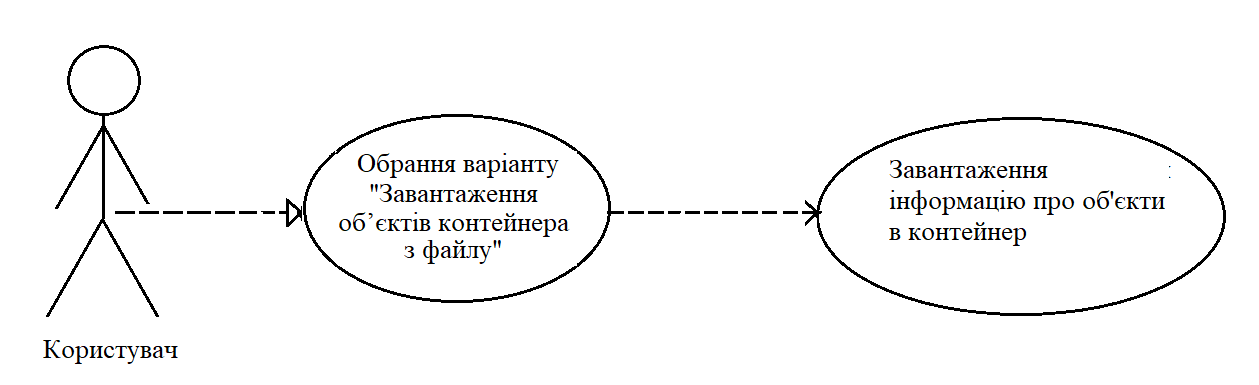
****

Рис. 1.7 - Зчитування об’єктів черги з пріоритетом в файл

1.1.8. Завантажити з файлу об'єктів та внесення їх в контейнер (черги з пріоритетом)

*Таблиця 1.8*

**Завантажити об’єктів контейнера з файлу**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Індентифікатор | Назва | Мета | Дійові особи | Тип | Опис |
| 8 | Завантаження об’єктів контейнера з файлу | Завантажити  інформацію  про об’єкти з файлу | Користувач | Основний | Надає  можливість завантажити інформацію  про всі  об’єкти контейнера  з файлу |

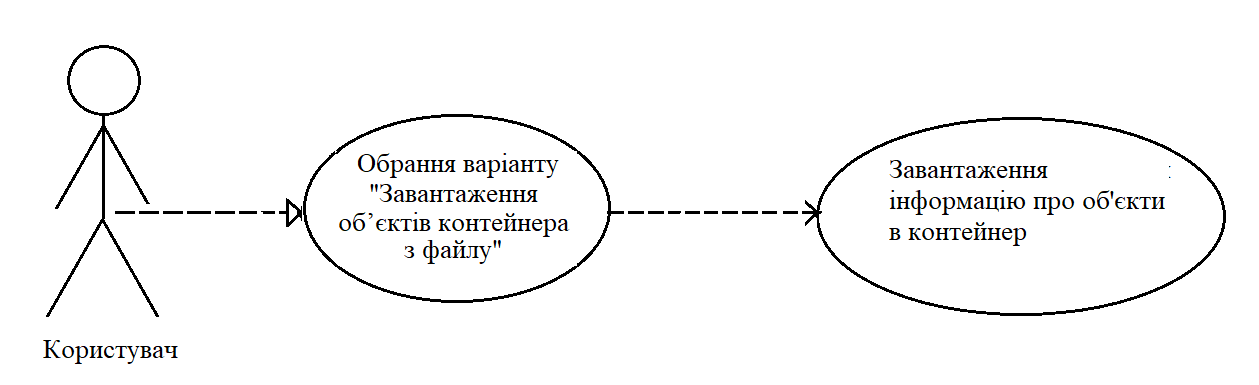
****

Рис. 1.8 - Завантаження об’єктів черги з пріоритетом в файл

1.1.9. Здійснити вихід з програми застосунку

*Таблиця 1.9*

**Вихід з програми**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Індентифікатор | Назва | Мета | Дійові особи | Тип | Опис |
| 9 | Вихід з програми | Закінчити роботу з програмного застосунку | Користувач | Основний | У процесі роботи програми надає  можливість  виходу з програмного застосунку |

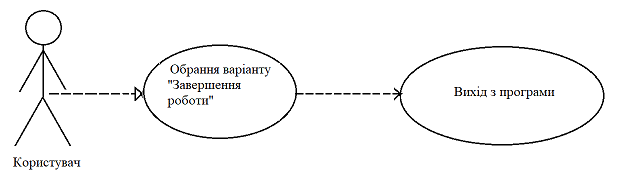
****

Рис. 1.9 – Вихід з програми

## Нефункціональні вимоги

В програмному застосунку реалізовані наступні нефункціональні вимоги:

● Можливість на повторне введення номеру вибору пункту в меню при некоректному його введені.

● Можливість на повторне введення при некоректному введені даних у заповнені полів об’єкту «регламентоване обслуговування» та «гарантійний ремонт».

● Можливість на повторне обирання файлу при невдалій перевірці на можливість відкриття файлу