ІНДИВІДУАЛЬНІ ЗАВДАННЯ для лабораторної роботи №2

з дисципліни «Програмування-ІІ»

(«Об’єктно-орієнтоване програмування»)

студентки групи ПС-21-1

Ірини Байдачної

кафедра комп’ютерних технологій

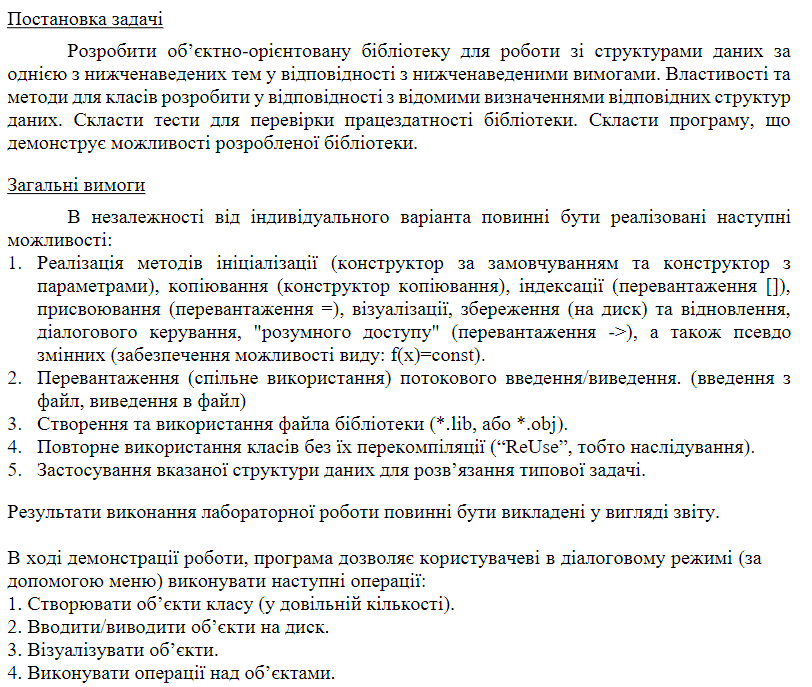
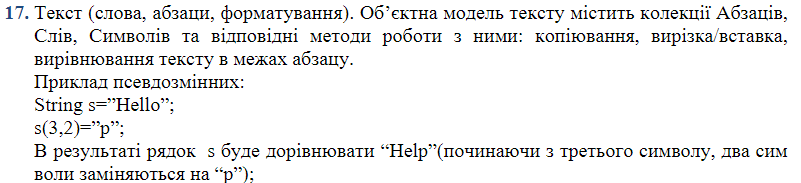
2022/2023 н.р.

Тема: «Використання класів в мові С++:

Інкапсуляція, наслідування та перенавантаження операцій»

Варіант – 17

**Постановка задачі**



**Опис розв’язку:**

Користувачу необхідно дати можливість обирати с чим і як він буде працювати, тому необхідно створити декілька меню. Але перед цим, програма повинна запитати користувача про те, скільки він бажає створити об’єктів та з якого він бажає почати працювати.

Якщо користувач обере перший пункт меню, йому необхідно обрати якого типу об’єкт він забажає створити (символ, слово, абзац).

Далі, з другим пунктом меню, необхідно надати користувачу вводити та виводити текстову інформацію на диск чи з нього. Якщо користувач не вписав до об’єкту якийсь текст, тобто об’єкт порожній, то при записуванні інформації на диск, користувачу програма повинна повідомити, що об’єкт пустий і записувати на диск немає що. Тому, користувачу необхідно надати можливість ввести текст безпосередньо все в другому пункті меню, якщо користувач його введе, то програма буде вважати цей текст з об’єкт, та працювати з цим текстом.

З третім пунктом меню, все доволі неважко. Користувач обирає цей пункт меню, з думками, що йому виведеться текст, що міститься в об’єкті. Тому, потрібно опрацювати той випадок, коли користувач ще нічого не вписав в об’єкт і він порожній. Тобто необхідно створити метод в одному з трьох класів, який буде мати перевірку на те, чи порожній об’єкт. Якщо порожній, то користувачу необхідно видати відповідне повідомлення, якщо не порожній, то текст, що міститься в об’єкті.

В четвертому пункті меню користувач може обрати декілька пунктів меню: (1) – довжина тексту; 2) - змінити рядок; 3) – обрати колір фону та колір шрифту). Якщо користувач обере 1-й пункт меню, то для його реалізації необхідно створити метод strlen, який буде рахувати довжину тексту. Для реалізації другого пункту меню, необхідно перевантажити оператор () (з двома параметрами) та оператор =. Для реалізації третього пункту меню, необхідно перевантажити оператор + та оператор = (так, в другий раз, тому що в другому пункті меню ми працюємо з псевдозмінною, вигляду: s(3, 2) = “p”. А в третьому пункті меню, ми працюємо з канкатенацією строк: str = a + b), користувачу необхідно ввести лише цифри кольорів, що будуть представлені у вигляді (цифра – колір) та ввести їх. Для зміни кольору тексту та кольору фону, необхідно в програмі скористатися system(color AB); Де А – цифр кольору фону, а В – цифра кольору шрифту.

Також, програма має додатковий пункт меню для зміни об’єкта. Тому що користувач з самого початку має можливість створити безліч об’єктів і мати можливість працювати з ними, а цей додатковий пункт меню дозволить переключатися між об’єктами. Для цього, користувачу лише необхідно ввести номер бажаного об’єкту. Потрібно передбачити можливість вводу користувачем некоректних даних, таких як символів та індексів об’єктів, що перевищують задану кількість об’єктів, що вів напочатку користувач.

Кожне меню має можливість повернутися до попереднього меню, це дає відчуття зручності для користувача.

**Код програми:**

Код програми буде викладений в GitHab’і. Він буде розділений на три модулі та бібліотеку.

**Опис інтерфейсу користувача:**

При стартуванні програми, користувачу програма пропонує ввести число об’єктів, з якими користувач може мати можливість працювати. Якщо користувач введе число менше 1, то програма попередить користувача що менше 1-го об’єкта створити не можна в програмі. Та програма надасть ще можливість ввести користувачеві коректні дані.

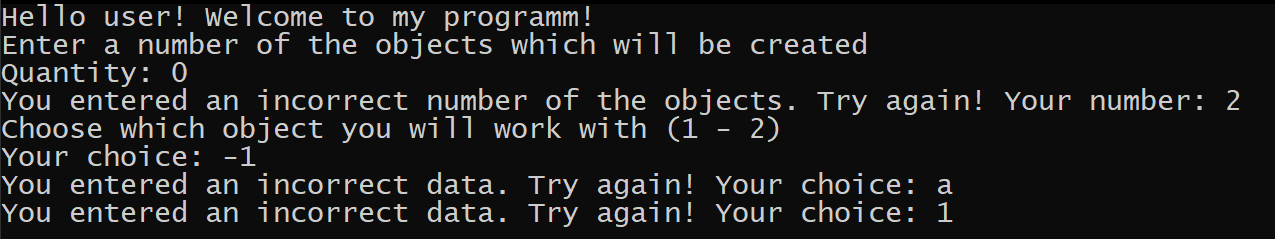
Далі, користувача програма просить вести номер об’єкту з яким користувач буде працювати. Якщо користувач введе некоректні дані, у вигляді літери, числа, що менше за 1, числа, що перебільшує кількість створених об’єктів для користувача, то при всіх цих випадках, користувач отримає відповідне повідомлення про те, що такого робити не можна та програма надасть ще раз можливість ввести коректні дані користувачеві.

Далі, користувачеві дається вибір обрати 6 пунктів меню:

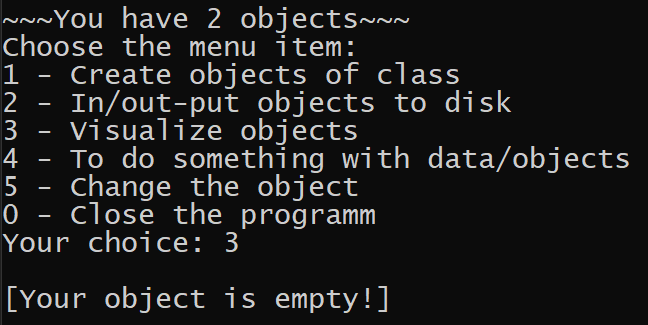
1. Користувач може обрати якого типу буде його об’єкт: символом, словом чи абзацем. Якщо користувач вже обрав та заповнив свій об’єкт якимось текстом і заходить знову в перший пункт меню, то програма запитує користувача, про те, чи дійсно він бажає видалити з об’єкта попередню інформацію. Якщо користувач обирає «так», то його об’єкт перезаписується в те, що обрав знову користувач (символ, слово, абзац), а якщо користувач обере «ні», то його перекине на попередній пункт меню.
2. Тут користувачу дається можливість обрати: чи він бажає вивести текст з файлу, чи навпаки внести текст у файл. Якщо користувач обирає пере (вивести текст), то йому необхідно буде ввести ім’я вхідного файлу, після чого, програма перевірить це ім’я на коректність. Якщо користувач ввів некоректні дані, то програма скаже, що такого файлу не існує і надасть можливість ввести ім’я вхідного файл ще раз. При правильному вводу користувачем, весь текст, що містився в файлі, тепер має об’єкт з яким працює користувач. Якщо об’єкт до цього був створений за якимось типом (символ, слово, абзац), то тут об’єкт автоматично перезаписується, та з ним можна в подальшому працювати в програмі. Якщо користувач обрав другий пункт меню: ввести в файл текст, то його тут чекає ще один вибір меню: ввести новий текст, чи старий. Старий текст – це той текст, що ввів користувач в першому пункті меню, коли обирав тип об’єкту. А в тому випадку, якщо наприклад об’єкт є порожнім, то вже не спрацює той пункт меню «ввести старий текст». Для цього вже існує «ввести новий текст» і у користувача є можливість ввести в файл рандомний текст, але він не буде записаний до об’єкту.
3. В третьому пункті меню користувач може лише побачити, що містить в собі його об’єкт (символ, слово, абзац). Якщо об’єкт порожній, то користувачу виведеться відповідне повідомлення про те, що об’єкт порожній виводити нічого.
4. В цьому пункті меню користувач може обрати три пункти меню: дізнатися про довжину тексту в об’єкті, трохи редагувати рядок та змінити кольори фону та шрифту. При обранні першого пункту, користувач лише про довжину дізнається і все. При обранні другого пункту меню, користувачу необхідно буде ввести замінюючий символ и слово для тексту, потім індекс з якого буде почато замінювання та скільки символів потрібно замінити. Після вводу трьох даних, користувачу на екран виведеться повідомлення про те, що він може побачити зміни лише в третьому пункті загального меню. Якщо користувач обере третій пункт меню, то йому необхідно буде ввести два числа, для кольорів. Якщо ці числа будуть виходити за рамки дозволеного, тобто користувач введе некоректні дані, то програма попередить користувача про помилку та надасть можливість ввести коректні дані ще раз.
5. В цьому пункті меню користувач може змінити об’єкт на інший, якщо він з самого початку створював декілька об’єктів.
6. В останньому пункті меню користувач може завершити виконання програми.

**Тестові приклади:**

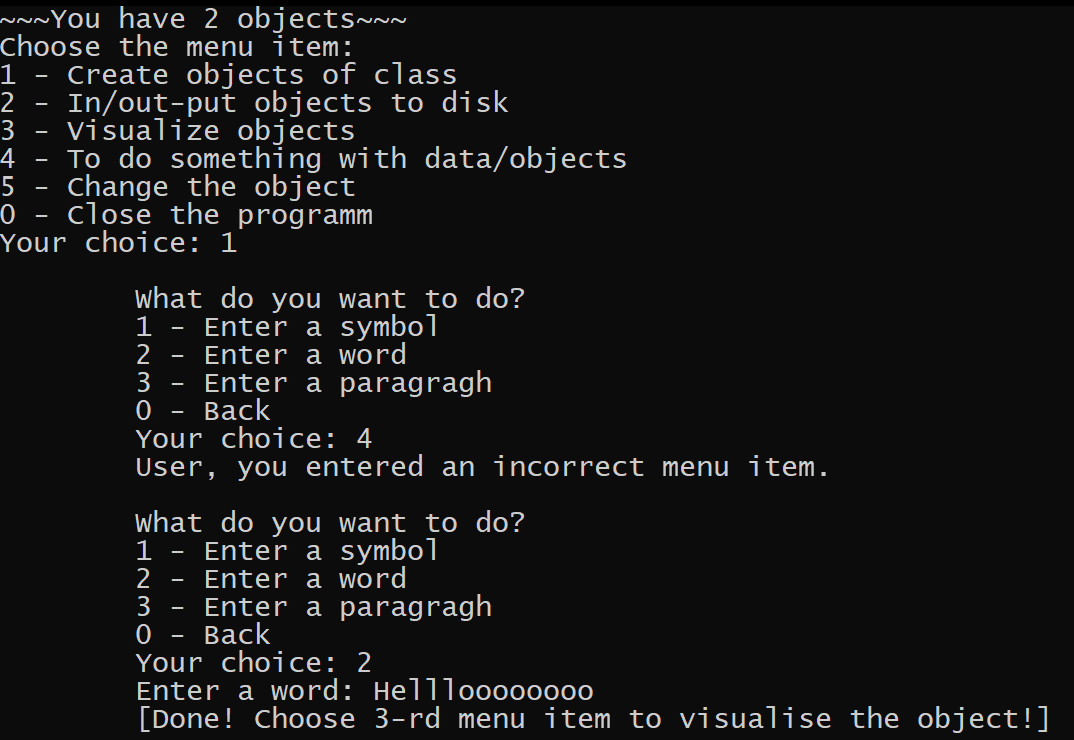
Ось користувач відкрив програму, та з самого початку він допустився деяких помилок, але програма його попередила про це та надала можливість вести ще раз коректні дані.



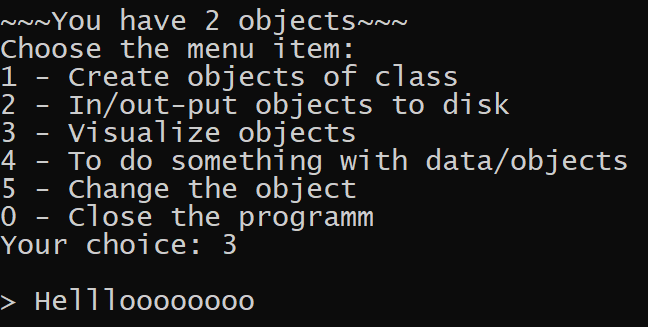
Ось той випадок, що буде, якщо об’єкт буде порожнім і користувач буде намагатися його вивести:



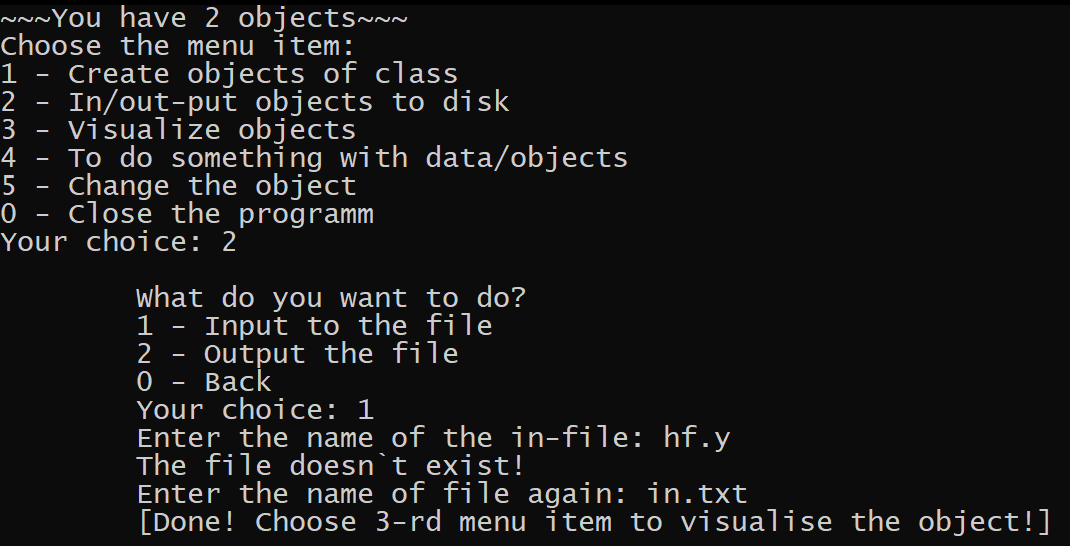
Ось користувач обрав перший пункт пеню та ввів четвертий пункт меню, якого не існує, програма надала можливість користувачеві ввести дані ще раз. Та користувач обрав все ж таки тип «слово» та ввів його, після чого програма повідомила користувача про те, що він може побачити змість об’єкту в третьому пункті меню:



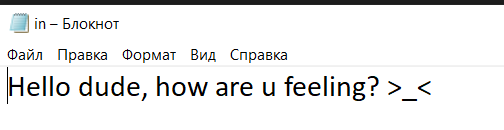
Ось третій пункт меню, як бачимо, об’єкт дійсно записався коректно:



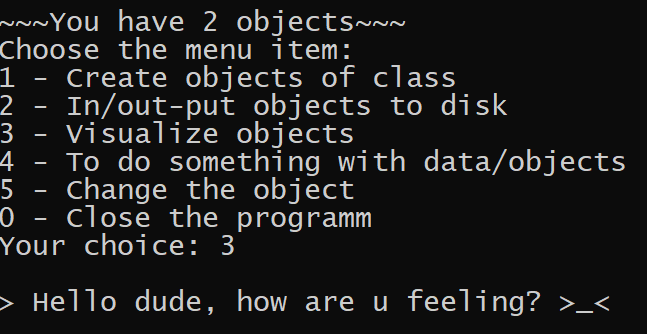
Ось користувач обирає другий пункт меню та обрав в другому меню перше, ввів спочатку неіснуюче ім’я файлу, після чого програма попередила користувача про помилку та користувач ввів ім’я заново:



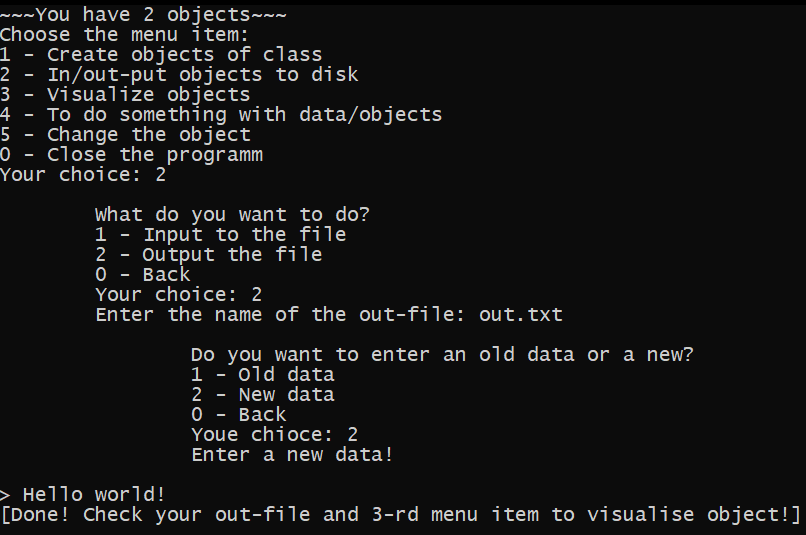
Ось що було в файлі:



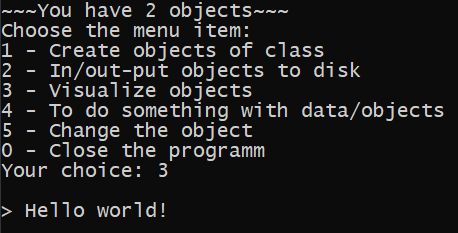
А ось, що було в третьому пункті меню:



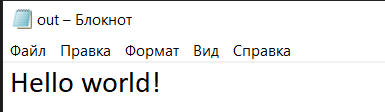
Ось користувач обрав ввести новий текст в вихідний файл, ввівши перед цим ім’я файлу. Програма пропонує подивитись в 3-му пункті та в вихідному файлі, що відбулось:



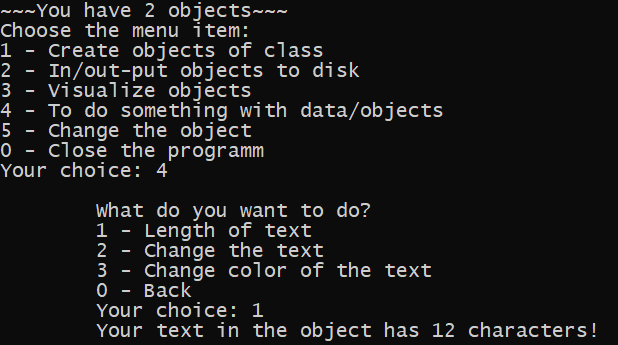
Ось третій пункт:



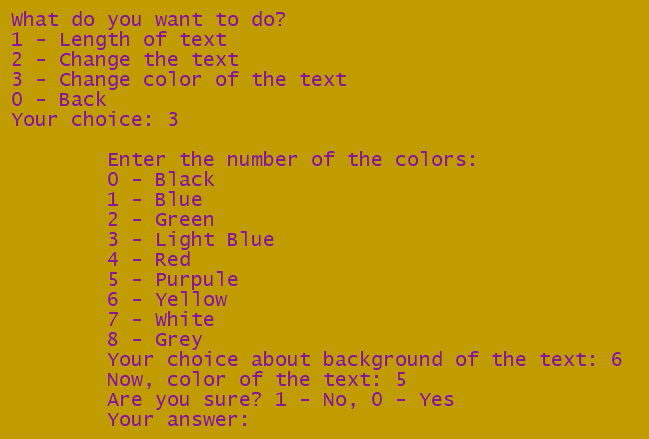
А ось вихідний файл:



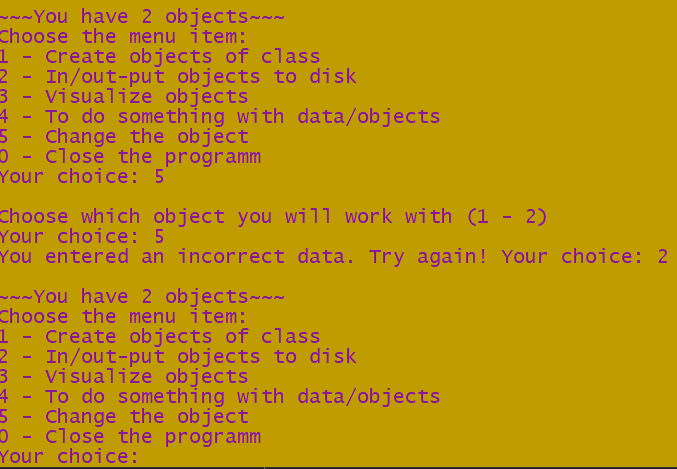
Ось користувач обрав четвертий пункт меню, та вирішив подивитися довжину тексту, що міститься в обєкті:



Тепер, користувач бажає змінити кольори фону та шрифту:

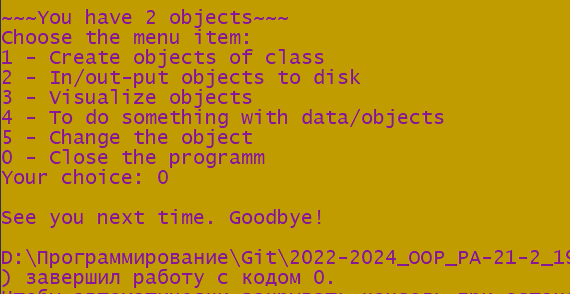


Тепер, в користувача є можливість, переобрати кольори чи залишити все як і було. Припустимо, що користувач вирішив залишити кольори, та нічого більше не міняти:



Тут користувач вирішив змінити об’єкт, з яким він працював, але перед цим, він зробив помилку, обравши той об’єкт, якого не існує, програма зробила йому попередження, та користувач ввів дані ще раз, і успішно змінив об’єкт класу. Тепер, програма буде працювати аналогічно і для цього нового об’єкту.

Якщо користувач забажає вийти з програми та завершити її виконання, то користувач обере останній пункт меню.



**Висновки:**

В цій лабораторній роботі було отримано досвід роботи з класами на мові С++. Було розглянуто тему інкапсуляція, наслідування класів та перевантаження функцій. Було розроблено програму на мові С++, було написано звіт, в якому викладено опис розв'язку задачі, опис інтерфейсу користувача і тестові приклади. Роботу було викладено на гіт хаб.