МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ХПІ»

Кафедра «Обчислювальна техніка та програмування»

Розрахункове завдання з програмування

Тема: «РОЗРОБКА ІНФОРМАЦІЙНО-ДОВІДКОВОЇ СИСТЕМИ»

Пояснювальна записка

КІТ.119Д.

Розробник

Виконав:

студент групи КІТ-119Д

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Гряник Г.В./

Перевірив:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Яловега В.А./

Харків 2020

ЗАТВЕРДЖЕНО

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

Розрахункове завдання з дисципліни

«Програмування ч.2»

Тема: «РОЗРОБКА ІНФОРМАЦІЙНО-ДОВІДКОВОЇ СИСТЕМИ»

Пояснювальна записка

КІТ.119Д.

Листів 69\_

Харків 2020

ЗМІСТ  
Вступ……………………………………………………………………………. 3

1 Огляд літературних джерел…………………………………………………....4

2 Постановка завдання до розробки………………………………………….….7

3 Коротка словесна характеристика алгоритма роботи програми………….…9

4 Розробка методів класу RGZ, Scientific\_SR, kolektsiya\_SR………………...10

5 Розробка методів класу List\_SR……………………………………….….…..12

6 Розробка методів класу Dialog\_menu………………………………………...14

Висновки…………………………………………………………………………15  
Список джерел інформації………………………….…………………………..16  
Додаток А. Текст програми…………………………………………………… 17  
Додаток Б. Результати роботи програми ……………….……………………. 63

Додаток В. Алгоритми роботи класів…............................................................67

**Вступ**

Відомо, що з початку розвитку обчислювальної техніки утворилося два основних напрями її використання: для виконання чисельних розрахунків та в автоматичних або автоматизованих інформаційних системах обробки даних. Останні набули широкого впровадження в життєдіяльність людини.

Стосовно технології програмування. Натепер беззаперечним лідером є об’єктно-орієнтоване програмування, яке дозволяє розкласти проблему на пов’язані між собою завдання. Кожне з них стає самостійним об’єктом, що містить свої власні коди та дані, які стосуються цього об’єкта. У цьому випадку вихідне завдання в цілому спрощується, і програміст одержує можливість розробляти більші за обсягом програми. З погляду програмування подібний підхід значно спрощує розробку і налагодження програм.

Мета роботи: закріпити отримані знання з дисципліни «Програмування» шляхом виконання типового комплексного завдання.

1. **Огляд літературних джерел з прикладної галузі**

Програмування пройшло довгий шлях від процедурного, модульного  
до об’єктно-орієнтованого (ООП). Суттєва відмінність ООП від  
традиційного проектування у тому, що акцент проектування переноситься  
з розробки алгоритмів функціонування системи на побудову абстракцій та  
їх взаємодію.

ООП дозволяє розкласти проблему на пов’язані між собою завдання. Кожне завдання стає самостійним об’єктом, що містить свої власні коди та дані, які мають відношення до цього об’єкта. У цьому випадку завдання в цілому спрощується, і програміст одержує можливість оперувати з більшими по обсягу програмами. Таке визначення ООП відображає відомий підхід до рішення складних завдань, коли завдання розбивається на часткові завдання, які вирішуються окремо. З точки зору програмування подібний підхід значно спрощує розробку, налагодження та тестування програмних продуктів. Саме до такого підходу в програмуванні студента необхідно заохочувати з самого початку його навчання.

Тематика даної частини лабораторних робіт присвячена розробці програми на основі вивченого матеріалу впродовж навчання: від розробки одного класу до ієрархічної структури класів з реалізацією поліморфізму та шаблонізації.

Відповідно до варіанта завдання, обрано прикладну галузь «Самостійні роботи студентів». Прикладна галузь представляє з себе набір розрахунково-графічних завдання, що можуть бути запропоновані студенту під час навчання з різних предметів. Також студенту буде представлено можливість визначити кількість РГЗ, що потрібно виконати за весь період навчання в інституті відповідно до навчального плану (кафедри).

Загальні підходи до методики самостійної роботи студента та її здійснення. Самостійна робота студента (СРС) - це самостійна діяльність та навчання студента, яку науково-педагогічний працівник планує разом зі студентом, але виконує її студент за завданнями та під методичним керівництвом і контролем науково-педагогічного працівника без його прямої участі.

Під час вивчення навчальної дисципліни виокремлюють такі види самостійного учіння студента:

* слухання лекцій, участь у семінарських заняттях, виконання практичних і
* лабораторних робіт;
* відпрацювання тем лекцій та семінарських занять, виконання практичних і
* лабораторних робіт студентами заочної форми навчання (ЗФН);
* підготовка рефератів і курсових робіт, написання дипломної роботи;
* робота з літературою та ін.

Однією з форм активізації самостійної роботи студентів є система розрахунково-графічних завдань (РГЗ). Застосування системи РГЗ рекомендовано діючою програмою кафедри наприклад для інженерно-технічних спеціальностей вищих технічних навчальних закладів.

Розрахунково-графічна робота (РГР) — це самостійне дослідження студента. Виконуючи РГР студент удосконалює знання та вміння, отримані в процесі вивчення дисципліни “Медична інформатика”, а саме: визначати мету, виділяти задачі, формулювати проблеми та знаходити способи їх розв'язання. Працюючи над РГР студент отримує вміння та навички, що будуть корисними в майбутньому при виконанні більш складних завдань (дипломна робота, дисертація, наукове дослідження тощо).

Написання розрахунково-графічної роботи є однією з форм самостійної роботи з дисципліни «Медична інформатика».

Метою написання РГР є:

* систематизація, закріплення та розширення теоретичних знань і практичних умінь студента;
* надбання досвіду роботи з літературою та іншими джерелами інформації, вміння узагальнювати та аналізувати наукову інформацію, виробляти власне ставлення до проблеми;
* вироблення вміння застосовувати інформаційні та комп'ютерні технології для розв’язання прикладних медичних задач;
* розвиток навичок оволодіння спеціалізованим програмним забезпеченням;
* проведення ґрунтовного аналізу результатів власних досліджень і формування змістовних висновків стосовно якості отриманих результатів.

Виходячи з назви наукової роботи, визначеного об’єкта та предмета, формулюється мета дослідження, що характеризує, яку найбільш важливу проблему або завдання має намір вирішити дослідник.

Мета визначається відповіддю на запитання: «Для чого проводиться дослідження?». Чітке формулювання конкретної мети – одна з найважливіших методологічних вимог до програми наукового дослідження. Мета дослідження полягає у вирішенні наукової проблеми шляхом удосконалення вибраної сфери діяльності конкретного об’єкта. Поставленої мети треба обов’язково досягти, на завершальному етапі досліджень необхідно перевірити, чи відповідають висновки поставленій меті. Мета формулюється лаконічно, вона повинна точно виражати те основне, що намагається зробити дослідник.

Мета конкретизується та розвивається у завданнях дослідження. Завдання дослідження визначають для того, щоб більш конкретно реалізувати його мету. Завдання наукового дослідження, як правило, полягають у такому:

* вирішення теоретичних питань, які пов’язані з проблемою дослідження (введення до наукового обігу нових понять, розкриття їх сутності і змісту; розроблення нових критеріїв і показників; розроблення принципів, умов і факторів застосування окремих методик і методів);
* виявлення шляхів та засобів удосконалення явища, процесу, що досліджується (практичні аспекти роботи); обґрунтування системи заходів, необхідних для вирішення прикладних завдань;
* експериментальна перевірка розроблених пропозицій щодо розв’язання проблеми, підготовка методичних рекомендацій для їх використання на практиці.

«Збірник розрахунково-графічних завдань » групи авторів – викладачів кафедри охоплює навчальні програми з курсу певної дисципліни для студентів певних спеціальностей.

Збірник складається з декількох частин, кожна з яких може містити певну кількість розділів. Кожний розділ містить певну кількість індивідуалізованих завдань. Це робить можливим використання цього «Збірника» викладачами як збірника задач з курсу. Окрім варіантів завдань, кожний розділ «Збірника» містить приклади розв’язування типових задач РГЗ і відповіді до кожного завдання розроблених варіантів РГЗ. У кожному розділі «Збірника» надається список літератури, рекомендованої для самостійного ознайомлення, що містить як традиційні класичні підручники та інші навчальні видання за радянських часів, так і навчально методичні видання кафедри за останні 15 років.

1. **Постановка завдання до розробки**

Для розробки програми потрібно розробити ієрархію класів відповідно до описів в попередньому пункті.

Кожна галузь повинна мати свої унікальні поля. Прикладна галузь РГЗ представлятиме з себе клас що матиме поля, що характерезують РГЗ. Даний клас повинен мати сім полів, що будуть характерезувати саме це РГЗ. На мою думку доцільними будуть такі поля: «subject» – обєкт(обєкт РГЗ), «predmet»-предмет з якого треба зробити конкретно саме це РГЗ, «tema» – тема з якої буде писати студент РГЗ, «bal» - кількість балів за виконання всіх завданнь для того щоб студентубуло простіше розставляти пріорітети цього РГЗ, «zavdanj» - кількість завдань яке потрібно виконати у цьому РГЗ щоб воно вважалося зробленим, «termin» - строк здачі РГЗ у днях (кількість днів на виконання всіхзавдань починаючи з моменту додавання в список), «cathedra» - кафедра яка пропонує саме ці РГЗ(номер).

Головне завдання цієї частини роботи це надання інформації студенту про характерестичні дані РГЗ яке йому потрібно буде виконати або не потрібно в залежності від кафедри. Головною проблемою залишуться поля які не повністю характерезують прикладну галузь хоча на мій погляд це достатньо розкриває.

Прикладна галузь наукові завдання пропонує студенту виконати особливі завдання з дослідення. Класс повинен наслідуватися від класу РГЗ так як даний клас матиме деякі поля точнісінько такіж самі, тому повтоно їх не потрібно створювати, що спрощує наше завдання. На мійпогляд доцільно буде додати поле «surname» – прізвище викладача, що контролює виконання роботи. Поле «surname» надасть інформацію студенту яки саме викладач пропонує таку роботу і кому її треба буде здавати.

Основні вимоги до класу нічим не відрізняються від попереднього класу, тому найбільша проблема цього класу залишається його актуальність та наповнення.

Для того, щоб студенту було простіше знайти посіник з завданнями створюю клас Збірників завдань. Головне завдання даного класу це надати інформацю про реомендовану літературу до орацювання, що виконати поставлені завдання.

На мою думку доцільми будть такі поля: «subject» – (буде успадковано від галузі РГЗ), predmet (буде успадковано від галузі РГЗ ) author - автор збірника , collection - назва збірника або назва книги цього автора, Vydavnytstvo – видавництвояке видало цю літературу. data – рік публікації збірника аже можливе перевидання із точнісінько такаю назвою і атором.

Якщо підбити підсумок то виходить три прикладних галузі які я назвав:

* Class RGZ – базовий клас створений для роботи з РГЗ що пропонується студенту до виконання.
* Class Scientific\_SR – клас що наслідується від попереднього класу RGZ, та головне завдання робота з спеціальним науковим завдання, що пропонується студенту для виконання.
* Class kolektsiya\_SR – клас, головне завдання класу роботу із збірниками завдань, що пропонується студенту для виконання.

Для збереження та обробки даних створено статичний масив розміром 251 (значення розміру випадкове).

Через те що вище перераховані класи мають по більшій мірі однакові методи ли різниця в тому що методи в різних класах мали лише різні типи даних було створено шаблоний клас який виконував основні завдання класів такі як додавання даних або видалення із списку.

Для того щоб даний клас який виходить як класс-контролер працював належним чином і небуло проблем із реалізуванням методів та роботою з класами було перевантажено оператори вводу та виводу і оператори порівняння та присвоєння.

Також відповідно до завдання виконуються такі вимоги можливостей класу контролера.

а) читання даних з файлу та їх запис у контейнер;

б) запис даних з контейнера у файл;

в) сортування елементів у контейнері за вказаними критеріями:

г) пошук елементів за вказаним критерієм

Реалізація завдань а,б вдасться реалызувати лише через перевантаження потоків яке треба грамотно перевантажити.

Для зручного контролю та навігії прорамою розроблено окремо для класа-контролера окремий клас із діалоговим меню та деякими повідомленнями для роботи з програмою.

Також для визначеня кількості РГЗ, що виконує студент за весь період навчання в інституті відповідно до навчального плану розробити методи який за допомогою номера який потрібно ввети студенту і йому буде виведено число/кількість РГЗ яке порібно зробити, дотогож повинно пропонувати вивести на екран предмети з яких потрібно зробити РГЗ.

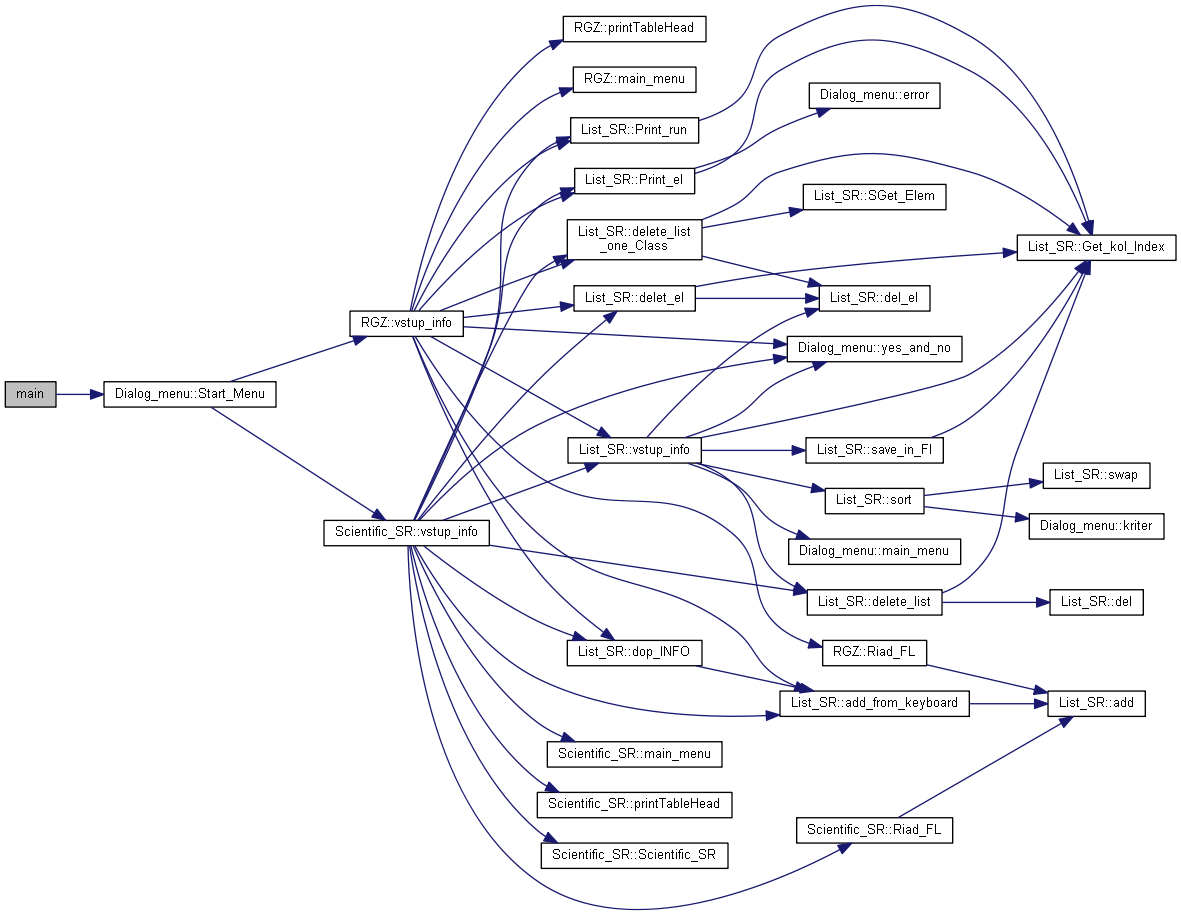
Для уникнення неочікуваних помилок треба використовувати оробки помилок наприклад генерувати та відловлювати помилки наприклад при виходу за рамки масиву або щось тому подібно.

Для перевірки введених даних на логічну коректність наприклад написання декількох слів або щось подібне використати регулярні вирази які при виявленні помилок у вводі з клавіатиру просила ввести данні ще раз або при читанні з файну записувало в поле помилку. І додатково перевіряти слова на те що вони пишуться з велкої літери ,якщо ні то програма це виправляє.

1. **Коротка словесна характеристика алгоритма роботи програми**

Основна частина алгоритма роботи програми подано на рис 1.

Робота програми починається з функції «main», але основини початок програми старт починається в фунції “Dialog\_menu::Start\_Menu”. Саме з цієї функціє починається основна дія адже тут конструюються обєкти та пропонується попрацюва з одним із класів як подано на схемі. Першим кроком кожного класу галузей – це спосіб додавання даних цієї галузі після чого відкриваєтья меню цього класу. Кожен клас реалізований таким чином щоб попри те що дані записані в один контейнер можна оперувати обєктами лише певногокласу.



Рисинок 1 – схема роботи програми

З кожного класу можна перейти до загального додаткового меню що виконує загальну роботу «List\_SR::vstup\_info» в якому пропонується користувачеві роботи з основними методами наприклад додавання (add()), видалення(delete\_list()), сортування (sort()) . По завершеню робот програма повертається на початок і перевіряє виток памяті.

Алгоритм реалізований таким чином, що можна безпроблем перемикатися між класами без втрат данних.

**4. Розробка методів класу RGZ, Scientific\_SR, kolektsiya\_SR**

Для обслуговування даних обєктів типу RGZ створено методи класу які безпроблем обслуговують дані. Для роботи з полями створені певні гетери та сетери яки забеспечують запис даних та читання даних з полів. Для видоду на екран даних створено функцію “show()”.Дана функція представляє з себе потоковий вивод даних. А для інтерфейсу було розроблено функцію «printTableHead()»для простішого орієнтуваняв виведених даних. Для того щоб користувач міг безпроблем користуватися індивідуальним функціоналом класу створено метод «vstup\_info» та для навігації функція “main\_menu”.

Принцип роботи даного методу полягає в тому що в циклі “while” який працює поки виконується умова. В циклі реалізований switch за допомогою якого проходить основне керування програмою. Switch викликається відповідно такий набір функція

* Вивод даних
* Ввод даних
* Видалити елемент за номером
* Відобразити елемент за номером
* Видалити зі списку дані цього класу
* Перехід до додаткового меню(меню контролера)
* Перейти до попереднього меню
* Завершити роботу

Метод «ЕL” створений для того щоб можна було записувати дані із файлу. Для коректної роботи було перевантажено раніше названі оператори. Оператор «=» було реалізовано виключення у разі некоректного присвоєння даних.

Для того щоб була можливість додавати елемент в масив було створено функцію клон яка перетворює тип в потрібний нам. Даний метод реалізований у кожному класі ієрархії.

За таким принципом реалізовані методи класі **Scientific\_SR, kolektsiya\_SR**

Для цих класі був розроблений абстрактний клас інтрфейс за допомогою якого контролювалося наявність даних методів які необжідні для роботи з класами.

Ієрархія класі подана на рис.2.

Абстрактний клас SR немає полів а лише не реалізовані віртуальні методи.

* SR\* clone() = 0;
* RiadFL- віртуальна функція для додавання данних із файлу
* main\_menu(number- для реалізацію головного меню для класу
* printTableHead() - віртуальна функція для реалізації вивода шапки таблиці
* show() - вивод елеменів
* vstup\_info- віртуальна функція для реалізації початку програми

Та перевантажено опереатори “new” тa “delete”. Тааке перевантаження наслідується класами наслідниками від класу предка.

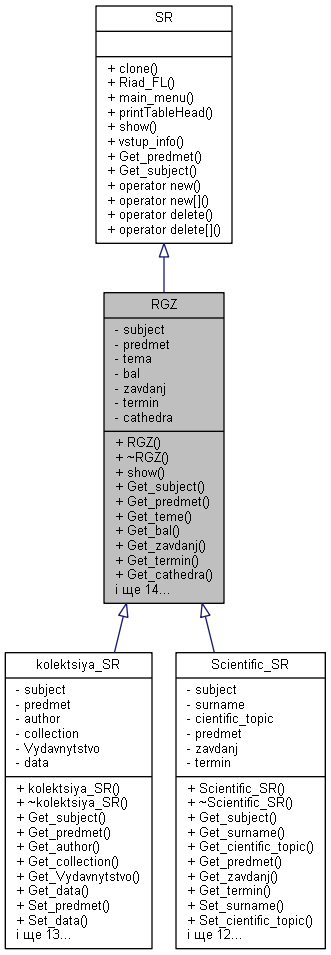


Рисунок 2 –схема наслідування ієрархії класі

**5. Розробка методів класу List\_SR**

Для того щоб у програмі уникнути повторюваності коду та зберігання данних було розроблено шаблоний клас мета якого зменшити обєм коду та збільшити читаємість.

Для зберігання даних створено два поля. Перше поле контейнер (data[ARRAY\_SIZE]) масив контейнерзберіганн елементів розміром 256 обєктів у який будуть записуватися наші дані. Друге поле індекс(index) у який беде записано індекс останьо записаного елементу. Якщо значенняіндексу буде -1 це означатиме що контейнер пустий тобто обєктів немає.

Деструктор реалізований із циклом «while» де буде знищувати обєкти поки індекс не буде -1.

Метод «vstup\_info» має такеж призначення та алгоритм як в іших класах галуях. В цому класі будуть надані такі можливості користувачеві:

* Вивести на екран список;
* Сортувати;
* Очистити список;
* Видалити елемент з списку за номереом;
* Додати з файлу ;
* Зберегти дані у файл;
* Визначити кiлькiсть РГЗ, що виконує студент за весь перiод навчання в iнститутi у вiдповiдностi до навчального плану;
* Перейти до попереднього меню ;

Для сортування елементів розроблено метод “sort”.Мета даного методу відсортувати за певним крітерієм який пропонується користувачеві та напрямом сортування (порядок від більшого до меншого або від меншого до більшого). Перевірка проводиться за допомого оператора нерівності в залежності від напрямку певна перевірка. Для перестановки обєктів розроблена функція ”swap” що збільшить читабильність.

Для виводу даних на екран не розробляється жодних функцій, а в замін перевантажується оператор виводу, що дозволяє нам вивести на екран без створення функції.

Перевантаження виводу також упростить нам задачу по зюереженню даних у файл. Метод «save\_in\_Fl» працює за подібним алгоритмом введеня даних з файлу. У функції пропонуватимеця збереження у текстовийфайл або в бінарний файл.

Додавання даних у контейнера проходитиеме поки масив матиме місце. Додавання даних проходе задопомогою клона. У масив записується клон обєкту прямим додаванням. Шаблонна функці «add\_from\_keyboard» записує дані обєкту, а потім зберігає клона у масив. Також реалізовано два способи додавання даних додавання даних

* З клавіатири
* З файлу

Шаблона функція «dop\_INFO» запрпонує више перераховані способи додавання даних. Дана функція реалізована для інших класі.

Для виконання індивідуального завдання розробити метод якийбуде визначати кількість РГЗ яке потрібно зробити студенту протягом навчального року розроблено функцію яка за номером буде визначатися кількість РГЗ. У разі якщо не буде знайдено РГЗ і число буде 0 то виведеться відповідне повідомлення. Додатковою функцією буде пропонування користувачеві вивести на екран предмети з яких треба зробити хоча предмети можуть повторюватися.

Щоб більше контролювати список розробити методи видалення даних із масиву що передбачатиме різні способи виделення даних

* Видалити всі елементи
* Видалити елемент за номером
* Видалити елементи певного класу

Видалення елементів дозволить контролювати память можливо уникнити витоку памяті, або ці методи зменшуть виток памяті що вже буде чудово.

Для зручного звертання до елементів масиву було перевантажено оператор «[]» що дозволяє нам упростити задачу із звертанням до обєктів.

**6. Розробка методів класу Dialog\_menu**

Розробка методів даного класу вважається найпростішим завданням. Клас має методи що несуть собою тексти для деяких допоміжних моментів програми.

В цьому класі реалізована функція «Start\_Menu» голвна задача якої є створити змінні за вдяки яким буде проводитися робота програма. Також тут релізован перемикач між класами за допомогою конади «switch».

**Висновок**

Під час виконання завдання закріпив знання з дисципліни «Програмування шляхом виконання комплексного завданням.

Під час виконнання завдання зіткнувся з типовими труднощями реалізації програми із використанням шаблоних функції які були вирішені за допомогою наприклад виключень.

Під час написання записки було виєвлено ще декілька підходів до тієї чи іншої проблеми. Наприклад для уникнення витоків памяті використовувати розумні покажчики та для зберігання даних використовувати вектор. Такий метод міг би зпростит мені роботу, але на момент написання цієї програми не додумався до деяки нюансів.

Навчився обєктивніше дивитися на завдання та вирішувати завдання які потребують особливого підходу. Написання поснювальної записки це також важливий досвід який знадобиться в майбутньому.

**Джерела**

* Х.М.Дейтел., П.Дж.Дейтел. Как программировать на Си++. М .: ЗАО БИНОМ, 1999, 1000 с.
* Г. Буч. Объектно – ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд./ Пер. с англ. – М.: "Издательство Бином". 1998г. – 560 с., ил.
* Семестр 2(1-18)
* [ISO/IEC JTC1/SC22/WG21 - The C++ Standards Committee](http://www.open-std.org/jtc1/sc22/wg21/) (1 September 1998). *ISO/IEC 14882:1998(E) Programming Language C++*. International standardization working group for the programming language C++. с. 40–41.
* <http://politsockaf.kpnu.edu.ua/samostiina-robota-studentiv/>
* <https://uk.wikipedia.org/wiki/Оператори_в_C_та_C%2B%2B>
* https://www.kontrolnaya-rabota.ru/s/equal-many/system-diff/?ef-\

Додаток А

Текст програми

/\*\*

\* \mainpage

\* Тема роботи: Розробка інформаційно-довідкової системи.;

\* Мета роботи: «Програмування» шляхом виконання типового комплексного завдання.

\* Індивідуальне завдання :

\* прикладна галузь «RGZ||SR»

\*

\*Типи об’єктів та їх поля : RGZ : Предмет,тема кількість завдань,термін здачі,кафедра

\*

\*Наукове РГЗ ( Scientific\_SR ): Викладач ,наукова тема кількість завдань,термін здачі.

\*

\*Збірник РГЗ ( kolektsiya\_SR ): Автор,збірник,Видавництво,предмет,рік.

\*

\author Гряник Г.В.

\date 24 - 05 - 2020

\version 1.5.1

\*\*/

/\*\*

Основиний файл з початком програми

\* @file main.cpp

\*

\*

\* \author Гряник Г.В.

\* \date 10 - 05 - 2020

\* \version 1.0.4

\*/

#include "prototaype.h"

#include "SR.h"

int main()

{

setlocale(LC\_CTYPE, "ru");

SetConsoleCP(1251);/// установка кодовой страницы win-cp 1251 в поток ввода

SetConsoleOutputCP(1251);

std::cout << "\t\t\t-----------------------------------------\n";

std::cout << "\t\t\t| | Програмування-РГЗ | |\n";

std::cout << "\t\t\t-----------------------------------------\n";

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(10));

number poku = 1;

int n = 1;

Dialog\_menu D;

poku = D.Start\_Menu(poku);

auto ram =0\* \_CrtDumpMemoryLeaks();

std::cout << "\nВитоку памятi:" << ram << std::endl;

SetConsoleTextAttribute(hConsole, (WORD)(13));

std::cout << "\nРоботу завершено\nТривалiсть роботи:" << kon << " хв " << kon2 << " сек" << std::endl;

SetConsoleTextAttribute(hConsole, (WORD)(10));

return 0;

}

/\*\*

Реалізація функцій-текст

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 2.5.4

\* @file Dialog.cpp

\*/

#include "prototaype.h"

#include "SR.h"

const int RGZ::main\_menu(number e)///<Реалізація функції-меню класу RGZ

{

system("cls");

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << " Меню 'RGZ' " << endl;

std::cout << " \n 1. Вивести на екран список\n";

system("ping 1.1.1.1 -n 0.001 -w 0.001 > nul");

std::cout << " 2. Додати елемент в список\n";

std::cout << " 3. Видалити елемент з списку за номереом\n";

std::cout << " 4. Відобразити елемент списку за номером\n";

std::cout << " 5. Очистити список \n\n";

std::cout << " 6. Перейти до додаткового меню \n";

std::cout << " 7. Перейти до попереднього меню \n\n\n";

std::cout << " 13. Завершити роботу \n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << " \nваш вибiр: ";

std::cin >> e;

system("cls");

return e;

}

const void RGZ::printTableHead()///<Реалізація виводу шапки таблиці

{

std::cout << "\n\n-----------------------------------------------------------------------------------" << std::endl;

std::cout << "| | Предмет | Темa | Кiлькiсть балiв |Строк здачi| Кiлькiсть |" << std::endl;

std::cout << "| | | |за виконане RGZ | RGZ | завдань |" << std::endl;

std::cout << "------------------------------------------------------------------------------------" << std::endl;

}

/\*\*

Реалізація функцій-текст класу Dialog\_menu

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 4.1.0

\* @file Dialog\_menu.cpp

\*/

#include "prototaype.h"

#include "SR.h"

int Dialog\_menu::Start\_Menu(number n)///Реалізація функцій

{

List\_SR<SR> List;

RGZ A;

Scientific\_SR B;

kolektsiya\_SR C;

while (n != 0)

{

cout << "\nВиберіть список, яким ви хочете керувати: \n\n";

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << " список RGZ ------> 1\n список Наслідник наукові робот ---- > 2\n список Наслідник збирники завдань ---- > 3\n ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

cout << " \nваш вибiр: ";

cin >> n;

switch (n)

{

case 1: n = A.vstup\_info(n, List);

break;

case 2: n = B.vstup\_info(n, List);

break;

case 3: n = C.vstup\_info(n, List);

break;

}

}

return n;

}

const int Dialog\_menu::main\_menu(number e)///Реалізація функцій

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

system("cls");

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << " Додаткове загальне меню " << endl;

std::cout << " \n 1. Вивести на екран список\n";

system("ping 1.1.1.1 -n 0.001 -w 0.001 > nul");

std::cout << " 2. Сортування\n";

std::cout << " 3. очистити список\n";

std::cout << " 5. Видалити елемент з списку за номереом\n";

std::cout << " 6. Додати з файлу\n";

std::cout << " 7. Зберегти дані у файл\n";

std::cout << " 8. Визначити кiлькiсть РГЗ, що виконує студент за весь\n перiод навчання в iнститутi у вiдповiдностi до\n навчального плану\n\n\n";

std::cout << " 12. Перейти до попереднього меню \n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << " \nваш вибiр: ";

std::cin >> e;

system("cls");

return e;

}

int Dialog\_menu::yes\_and\_no(number& t, string p)///Реалізація функцій

{

std::cout << "\n\n" << p << "\n ";

std::cout << "так- 1\n ";

std::cout << "нi- 2\n ";

std::cout << "вибрана робота :";

std::cin >> t;

return t;

}

int Dialog\_menu::kriter(number n)///Реалізація функцій

{

cout << "\nЗа яким крiтерiєм хочете сортувати список \n\n";

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << " 1.Адреса\n";

cout << " 2.Предмет\n";

cout << " 3.Тип\n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

cout << " \nваш вибiр: ";

cin >> n;

return n;

}

const void Dialog\_menu::error(number& a, number& d)///Реалізація функцій

{

std::cout << "\nНажаль неможливо вiдобразити неiснуючий елемент\n";

std::cout << "Введiть iнше значення ";

std::cin >> d;

a = 1;

} /\*\*

Реалізація функцій класу koleksiya\_SR

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 3.3.3

\* @file kolektsiya\_SR.cpp

\*/

#include "prototaype.h"

#include "SR.h"

kolektsiya\_SR& kolektsiya\_SR::operator = (const kolektsiya\_SR& SR)

{

//cout << "Перегрузка оператора '='" << endl;

if (this == &SR)

return \*this;

predmet = SR.predmet;

collection = SR.collection;

author = SR.author;

Vydavnytstvo = SR.Vydavnytstvo;

data = SR.data;

return \*this;

}

kolektsiya\_SR& kolektsiya\_SR::operator = (const kolektsiya\_SR\* SR)

{

//cout << "Перегрузка оператора '='" << endl;

if (this == SR)

return \*this;

if (typeid(\*this).name() != typeid(\*SR).name()) throw std::invalid\_argument("помилка зберігання");;

predmet = SR->predmet;

collection = SR->collection;

author = SR->author;

Vydavnytstvo = SR->Vydavnytstvo;

data = SR->data;

return \*this;

}

bool operator < (kolektsiya\_SR& p, kolektsiya\_SR& p2) {

int i = 0;

if (p.Get\_collection() < p2.Get\_collection())i++;

if (p.Get\_predmet() < p2.Get\_predmet())i++;

if (p.Get\_author() < p2.Get\_author())i++;

if (p.Get\_Vydavnytstvo() < p2.Get\_Vydavnytstvo())i++;

if (p.Get\_data() < p2.Get\_data())i++;

if (i < 3)return 0;

else return 1;

}

bool operator > (kolektsiya\_SR& p, kolektsiya\_SR& p2)

{

int i = 0;

if (p.Get\_collection() > p2.Get\_collection())i++;

if (p.Get\_predmet() > p2.Get\_predmet())i++;

if (p.Get\_author() > p2.Get\_author())i++;

if (p.Get\_Vydavnytstvo() > p2.Get\_Vydavnytstvo())i++;

if (p.Get\_data() > p2.Get\_data())i++;

if (i < 3)return 0;

else return 1;

}

bool operator== (kolektsiya\_SR& p, kolektsiya\_SR& p2) { return !(p < p2) && !(p > p2); }

bool operator!= (kolektsiya\_SR& p, kolektsiya\_SR& p2) { return !(p == p2); }

bool operator <= (kolektsiya\_SR& p, kolektsiya\_SR& p2) { return (p == p2) || !(p2 < p); }

bool operator >= (kolektsiya\_SR& p, kolektsiya\_SR& p2) { return (p == p2) || !(p2 > p); }

///вивод на екран

ostream& operator<< (ostream& out, kolektsiya\_SR& pr)

{

out << "|" << std::setw(14) << pr.Get\_predmet() << "|" << std::setw(14) << pr.Get\_collection() << "|" << std::setw(18) << pr.Get\_author()

<< "|" << std::setw(11) << pr.Get\_Vydavnytstvo() << "|" << std::setw(11) << pr.Get\_data() << "|"

<< "\n------------------------------------------------------------------------------------\n" ;

return out;

}

///читання з файлу

ifstream& operator >> (ifstream& stream, kolektsiya\_SR& add)

{

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9іІїЇ -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2("^[A-za-zА-Яа-яіІЇї][A-za-zА-Яа-я0-9іІїЇ]{1,20}([\\.,]?[-\_ ]?[А-Яа-яA-za-zіІЇї]{1,20}){0,2}$");//перевірка на правильний ввод

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

std::ifstream fOpen;

std::string text;

stream >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

SetConsoleTextAttribute(hConsole, (WORD)(12));

std::cout << "1Критична помилка в написаному" << std::endl;

text = "Eror(No name)";

SetConsoleTextAttribute(hConsole, (WORD)(10));

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << "Програма виявила помилку в написаному та успішно випрвила її" << std::endl;

SetConsoleTextAttribute(hConsole, (WORD)(10));

} add.Set\_predmet(text);////туту

stream >> text;

add.Set\_collection(text);////туту

stream >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

SetConsoleTextAttribute(hConsole, (WORD)(12));

std::cout << "3Критична помилка в написаному" << std::endl;

text = "Eror(No name)";

SetConsoleTextAttribute(hConsole, (WORD)(10));

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << "Програма виявила помилку в написаному та успішно випрвила її" << std::endl;

SetConsoleTextAttribute(hConsole, (WORD)(10));

} add.Set\_author(text);////туту

stream >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

text = "Eror(No name)";

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

std::cout << "4Програма виявила помилку в написаному та успішно випрвила її" << std::endl;

}add.Set\_Vydavnytstvo(text);

/\*fOpen >> text;\*/

int value;

stream >> value;

if (stream.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(15));

value = NULL;

std::cout << "Критична помилка в напиcаннi" << std::endl;

stream.clear();

stream >> text;

SetConsoleTextAttribute(hConsole, (WORD)(12));

}

add.Set\_data(value);

return stream;

}

/////////////////////////////

/\*\*

@brief

@details

@param

@return

\*/

void kolektsiya\_SR::EL(kolektsiya\_SR& add)

{

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9іІїЇ -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2("^[A-za-zА-Яа-яіІЇї][A-za-zА-Яа-я0-9іІїЇ]{1,20}([\\.,]?[-\_ ]?[А-Яа-яA-za-zіІЇї]{1,20}){0,2}$");//перевірка на правильний ввод

int value5;

std::string value1, value2, value3 , value4,s2;

bool condition = true;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть назву предмета : ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> value1;

while (cin.peek() != '\n') cin >> s2, value1 += " " + s2;

if ((regex\_match(value1, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

continue;

}

if ((regex\_match(value1, regex\_integer)) == 0)

{

value1[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}std::cout << "додано" << std::endl;

condition = true;

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть назву збирника завдань : ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> value2;

while (cin.peek() != '\n') cin >> s2, value2 += " " + s2;

if ((regex\_match(value2, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value2, regex\_integer)) == 0)

{

value2[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}std::cout << "додано" << std::endl;

condition = true;

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть призвище автора : ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> value3;

while (cin.peek() != '\n') cin >> s2, value3 += " " + s2;

if ((regex\_match(value3, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value3, regex\_integer)) == 0)

{

value3[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}std::cout << "додано" << std::endl;

condition = true; condition = true;

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть назву видаництва : ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> value4;

while (cin.peek() != '\n') cin >> s2, value4 += " " + s2;

if ((regex\_match(value4, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value3, regex\_integer)) == 0)

{

value3[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}std::cout << "додано" << std::endl;

condition = true;

while (condition) {///////регулярка дати

SetConsoleTextAttribute(hConsole, (WORD)(14));

cout << "Введiть рік видання : ";

std::cin >> value5;

if (std::cin.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << "Спробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if (value5 > 0)

condition = false;

}SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << " додано" << std::endl;

add.Set\_predmet(value1);

add.Set\_collection(value2);

add.Set\_author(value3);

add.Set\_Vydavnytstvo(value4);

add.Set\_data(value5);

}

/\*\*

@brief

@details

@param

@return

\*/

void kolektsiya\_SR::Riad\_FL(List\_SR<SR>& A)

{

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9 -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2(("^[A-za-zА-Яа-я][A-za-zА-Яа-я0-9]{1,20}[\\.,]?[-\_ ]?[А-Яа-яA-za-z]{1,20}$"));//перевірка на правильний ввод

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

kolektsiya\_SR add;

std::ifstream fOpen;

int temp = 0;

std::string text;

fOpen.open("zbyrnk\_zavdan.txt", std::ios\_base::in);

if (!fOpen)

{

SetConsoleTextAttribute(hConsole, (WORD)(5));

std::cout << "File is not open\n\n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

return;

}SetConsoleTextAttribute(hConsole, (WORD)(10));

while (!fOpen.eof())

{

fOpen >> add;

A.add(&add);

}

fOpen.close();

}

/\*\*

@brief

@details

@param

@return

\*/ const void kolektsiya\_SR::poshuk(List\_SR<SR>& A)

{

if (A.Get\_kol\_Index() < 1)

{

cout << "!!!!Пустий масив!!!" << endl;

system("pause&cls");

return;

}

system("cls");

kolektsiya\_SR p ;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9 -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2(("^[A-za-zА-Яа-я][A-za-zА-Яа-я0-9]{1,20}[\\.,]?[-\_ ]?[А-Яа-яA-za-z]{1,20}$"));//перевірка на правильний ввод

string autor;

int perevirka=0;

bool condition = true;

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

cout << "Введіть прізвище автора :";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> autor;

if ((regex\_match(autor, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(autor, regex\_integer)) == 0)

{

autor[0] -= 32;

}

condition = false;

}

cout << endl << endl;

int kol = A.Get\_kol\_Index();

for (int i = 0; i < kol; i++)

try{

p = (kolektsiya\_SR\*)A[i];

if (autor == p.Get\_author())

{

perevirka++;

cout << "|RGZ-" <<i+1 << p<< std::endl;

}

}

catch (std::invalid\_argument e) { ; }

if (perevirka == 0)

cout << "\nТакого автора незнайдено\n";

system("pause");

}

/\*\*

Реалізація функцій-тексту класу koleksiya\_SR

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 2.1.4

\* @file kolektsiya\_SR\_menu.cpp

\*/

#include "prototaype.h"

#include "SR.h"

/\*\*

@brief

@details

@param

@return

\*/

const void kolektsiya\_SR::printTableHead()

{

std::cout << "\n\n-----------------------------------------------------------------------------------" << std::endl;

std::cout << "| | Предмет | Збірник | Автор |Видавництво| рік |" << std::endl;

std::cout << "| | | завдань | збіника | RGZ |видавництва |" << std::endl;

std::cout << "-------------------------------------------------------------------------------------" << std::endl;

}

/\*\*

@brief

@details

@param

@return

\*/

const int kolektsiya\_SR::main\_menu(number e)

{

system("cls");

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << " Меню 'Збірник завдань' " << endl;

std::cout << " \n 1. Вивести на екран список збірників РГЗ\n";

system("ping 1.1.1.1 -n 0.001 -w 0.001 > nul");

std::cout << " 2. Додати збірник в список\n";

std::cout << " 3. Видалити збірник з списку за номереом\n";

std::cout << " 4. Відобразити збірник списку за номером\n";

std::cout << " 5. Очистити список \n";

std::cout << " 6. Пошук зiбірника ргз за автором \n\n";

std::cout << " 7. Перейти до додаткового меню \n";

std::cout << " 8. Перейти до попереднього меню \n\n";

std::cout << " 10. Завершити роботу \n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << " \nваш вибiр: ";

std::cin >> e;

system("cls");

return e;

}

/\*\*

@brief

@details

@param

@return

\*/

int kolektsiya\_SR::vstup\_info(number nov, List\_SR<SR>& List)

{

kolektsiya\_SR h = kolektsiya\_SR();

Dialog\_menu D;

system("cls");

number a = 1, u = 0;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

cout << "\n\nВи знаходитесь у класi наслiднику 'збірники завдань' ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << "\n\n Почнемо роботу\n\n Ввести данi\n ";

while (a != 0)

{

SetConsoleTextAttribute(hConsole, (WORD)(12));

cout << "\n 1: З клавiатири\n 2: 3 Файлу\n Ваш вiбiр: ";

cin >> a;

switch (a)

{

case 1:SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << "Скiльки елементiв ви хочете додати? :";

std::cin >> a;

for (int i = 0; i < a; i++)

List.add\_from\_keyboard(h);

a = 0;

break;

case 2:SetConsoleTextAttribute(hConsole, (WORD)(10));

h.Riad\_FL(List), a = 0;

break;

}

}

a = h.main\_menu(a);

while (a)

{

try {

switch (a)

{

case 1:

system("cls");

h.printTableHead();

List.Print\_run(List, h);;

break;

case 2: system("cls"), List.dop\_INFO(List, h);

break;

case 3: system("cls"), List.delet\_el(List, h);

break;

case 4: system("cls"), List.Print\_el(List, h);

break;

case 5:

system("cls");

List.delete\_list\_one\_Class(h);

std::cout << "Успішно";

break;

case 6:system("cls"), h.poshuk(List);

break;

case 7:system("cls"), List.vstup\_info(nov, List);

break;

case 8: system("cls");

return 123;

break;

case 10:List.delete\_list(List);

system("cls");

nov = 0;

return 0;

break;

}

}

catch (std::exception & aErr)

{

cout << " спійманов виняток у Kolektsiya\_SR:" << aErr.what() << endl;

}

catch (std::logic\_error & aErr)

{

cout << "Сталася помилка: " << aErr.what() << endl;

}

catch (int aMsg)

{

cout << "Перехоплено int:" << aMsg << endl;

}

catch (std::out\_of\_range e) {

cout << e.what() << endl;

}

catch (std::ios\_base::failure e) {

cout << e.what() << endl;

}

catch (std::invalid\_argument e) {

cout << e.what() << endl;

}

if(a!=7)

{string pit = "Бажаєте ще щось зробити?";

a = D.yes\_and\_no(a, pit);

if (a == 1)

a = h.main\_menu(a);

else a = 10;

}

else a = h.main\_menu(a);

}

return 0;

}

/\*\*

Реалізація функція класу RGZ та перевантажень

\* \file RGZ.cpp

\* \author Георгій

\* \version 5.1.4

\* \date 2020.04.22

\*/

#include "prototaype.h"

#include "SR.h"

template<class TS> class List\_SR;

/////перегрузки операторів

RGZ& RGZ::operator = (const RGZ& SR)

{

// cout << "Перегрузка оператора '='" << endl;

if (this == &SR)

return \*this;

predmet = SR.predmet;

tema = SR.tema;

termin = SR.termin;

bal = SR.bal;

zavdanj = SR.zavdanj;

cathedra = SR.cathedra;

return \*this;

}

RGZ& RGZ::operator = (const RGZ\* SR)

{

//cout << "Перегрузка оператора '='" << endl;

if (this == SR)

return \*this;

if (typeid(\*this).name() != typeid(\*SR).name()) throw std::invalid\_argument("помилка зберігання");;

predmet = SR->predmet;

tema = SR->tema;

termin = SR->termin;

bal = SR->bal;

zavdanj = SR->zavdanj;

cathedra = SR->cathedra;

return \*this;

}

bool operator < (RGZ& p, RGZ& p2) {

int i=0;

if (p.Get\_bal() < p2.Get\_bal())i++;

if (p.Get\_predmet() < p2.Get\_predmet())i++;

if (p.Get\_teme() < p2.Get\_teme())i++;

if (p.Get\_zavdanj() < p2.Get\_zavdanj())i++;

if (p.Get\_termin() < p2.Get\_termin())i++;

if (i < 3)return 0;

else return 1;

}

bool operator > (RGZ& p, RGZ& p2)

{

int i=0;

if (p.Get\_bal() > p2.Get\_bal())i++;

if (p.Get\_predmet()>p2.Get\_predmet())i++;

if (p.Get\_teme() > p2.Get\_teme())i++;

if (p.Get\_zavdanj()> p2.Get\_zavdanj())i++;

if (p.Get\_termin() > p2.Get\_termin())i++;

if (i < 3)return 0;

else return 1;

}

bool operator== (RGZ& p, RGZ& p2) { return !(p < p2) && !(p > p2); }

bool operator!= (RGZ& p, RGZ& p2) { return !(p == p2) ; }

bool operator <= (RGZ& p, RGZ& p2) { return (p == p2) || !(p2 < p); }

bool operator >= (RGZ& p, RGZ& p2) { return (p == p2) || !(p2 > p); }

//вивод на екран

ostream& operator<< (ostream& out, RGZ& pr)

{

out << "|" << std::setw(14) << pr.Get\_predmet() << "|" << std::setw(14) << pr.Get\_teme()

<< "|" << std::setw(18) << pr.Get\_bal() << "|" << std::setw(11) << pr.Get\_termin() << "|" << std::setw(11)

<< pr.Get\_zavdanj() << "|"

<< "\n------------------------------------------------------------------------------------\n";

return out;

}

//читання з файлу

ifstream& operator >> (ifstream& stream, RGZ& add)

{

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9іІїЇ -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2("^[A-za-zА-Яа-яіІЇї][A-za-zА-Яа-я0-9іІїЇ]{1,20}([\\.,]?[-\_ ]?[А-Яа-яA-za-zіІЇї]{1,20}){0,2}$");//перевірка на правильний ввод

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

std::ifstream fOpen;

std::string text;

int value;

stream >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

SetConsoleTextAttribute(hConsole, (WORD)(12));

std::cout << "Критична помилка в напиcаннi" << std::endl;

text = "Eror(No name)";

SetConsoleTextAttribute(hConsole, (WORD)(10));

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

SetConsoleTextAttribute(hConsole, (WORD)(10));

}

add.Set\_predmet(text);

stream >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

text = "Eror(No name)";

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

add.Set\_teme(text);

stream >> value;

if (stream.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(15));

value = NULL;

std::cout << "Критична помилка в напиcаннi" << std::endl;

stream.clear();

stream >> text;

SetConsoleTextAttribute(hConsole, (WORD)(12));

} add.Set\_bal(value);

stream >> value;

if (stream.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(15));

value = NULL;

std::cout << "Критична помилка в напиcаннi" << std::endl;

stream >> text;

std::cin.ignore(std::cin.rdbuf()->in\_avail());

SetConsoleTextAttribute(hConsole, (WORD)(12));

}add.Set\_termin(value);

stream >> value;

if (stream.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(15));

value = NULL;

std::cout << "Критична помилка в напиcаннi" << std::endl;

stream.clear();

stream >> text;

SetConsoleTextAttribute(hConsole, (WORD)(12));

}

add.Set\_zavdanj(value);

stream >> value;

if (stream.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(15));

value = NULL;

std::cout << "Критична помилка в напиcаннi" << std::endl;

stream.clear();

stream >> text;

SetConsoleTextAttribute(hConsole, (WORD)(12));

}

add.Set\_cathedra(value);

return stream;

}

/////////////////////////////

/\*\*

@brief

@details

@param

@return

\*/

int RGZ::vstup\_info(number nov, List\_SR<SR>& List)

{

RGZ h;

Dialog\_menu D;

system("cls");

number a = 1;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

cout << "\n\nВи знаходитесь у класі RGZ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << "\n\n Почнемо роботу\n\n Ввести данi\n ";

while (a != 0)

{

SetConsoleTextAttribute(hConsole, (WORD)(12));

cout << "\n 1: З клавiатири\n 2: 3 Файлу\n Ваш вiбiр: ";

cin >> a;

switch (a)

{

case 1:SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << "Скiльки елементiв ви хочете додати? :";

std::cin >> a;

for (int i = 0; i < a; i++)

List.add\_from\_keyboard(h);

a = 0;

break;

case 2:SetConsoleTextAttribute(hConsole, (WORD)(10));

h.Riad\_FL(List), a = 0;

break;

}

}

//List.show();

a = h.main\_menu(a);

while (a)

{

try{switch (a)

{

case 1:

h.printTableHead();

List.Print\_run(List, h);

break;

case 2:List.dop\_INFO(List, h);

break;

case 3:List.delet\_el(List, h);

break;

case 4: List.Print\_el(List, h);

break;

case 5: List.delete\_list\_one\_Class(h);

system("cls");

std::cout << "Успішно";

break;

case 6:

system("cls");

List.vstup\_info(nov, List);

break;

case 7: system("cls");

return 123;

break;

case 13:

system("cls");

nov = 0;

return 0;

break;

}

}

catch (std::exception & aErr)

{

cout << " спійманов виняток у RGZ:" << aErr.what() << endl;

}

catch (std::logic\_error & aErr)

{

cout << "Сталася помилка: " << aErr.what() << endl;

}

catch (int aMsg)

{

cout << "Перехоплено int:" << aMsg << endl;

}

catch (std::out\_of\_range e) {

cout << e.what() << endl;

}

catch (std::ios\_base::failure e) {

cout << e.what() << endl;

}

catch (std::invalid\_argument e) {

cout << e.what() << endl;

}

if(a!=10)

{string pit = "Бажаєте ще щось зробити?";

a =D.yes\_and\_no(a, pit);

if (a == 1)

a = h.main\_menu(a);

else a = 13; }

else a = h.main\_menu(a);

}

return 0;

}

/\*\*

@brief

@details

@param

@return

\*/ void RGZ::Riad\_FL(List\_SR<SR>& A)

{

RGZ add;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

std::ifstream fOpen;

int temp = 0;

std::string text, adresa;

std::cout << "\nЯкий файл відкрити?\n\n 1. RGZ.txt\n 2. add\_RGZ.txt\n Ваш вiбiр:";

std::cin >> temp;

switch (temp)

{

case 1:adresa = "RGZ.txt";

break;

case 2:adresa = "add\_RGZ.txt";

break;

}

fOpen.open(adresa, std::ios\_base::in);

if (!fOpen)

{

SetConsoleTextAttribute(hConsole, (WORD)(5));

std::cout << "File is not open\n\n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

return;

}SetConsoleTextAttribute(hConsole, (WORD)(10));

while (!fOpen.eof()) {

fOpen >> add;

A.add(&add);

}

fOpen.close();

}

/\*\*

@brief

@details

@param

@return

\*/

void RGZ::EL(RGZ& add)

{

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9іІїЇ -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2("^[A-za-zА-Яа-яіІЇї][A-za-zА-Яа-я0-9іІїЇ]{1,20}([\\.,]?[-\_ ]?[А-Яа-яA-za-zіІЇї]{1,20}){0,2}$");//перевірка на правильний ввод

int value3, value4, value5, value6;

std::string value1, value2,s2;

bool condition = true;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть назву предмету : ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> value1;

while (cin.peek() != '\n') cin >> s2, value1 += " " + s2;

if ((regex\_match(value1, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value1, regex\_integer)) == 0)

{

value1[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}

SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << " додано" << std::endl;

condition = true;

while (condition) {

SetConsoleTextAttribute(hConsole, (WORD)(10));

/\*if ((std::cin.rdbuf()->in\_avail()) != 1);\*/

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введіть тему :";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> value2;

while (cin.peek() != '\n') cin >> s2, value2 += " " + s2;

if ((regex\_match(value2, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value2, regex\_integer)) == 0)

{

value2[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

if (value2[0] >= 'a' && value2[0] <= 'z' || value2[0] >= 'а' && value2[0] <= 'я' || value2[0] == 'ё')

value2[0] -= 32;

}

SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << "додано" << std::endl;

condition = true;

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

/\*if ((std::cin.rdbuf()->in\_avail()) != 1);\*/

std::cout << "Введiть Кiлькiсть балiв за завдання : ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> value3;

if (std::cin.fail()) {//Если есть ошибка, выводим сообщение

SetConsoleTextAttribute(hConsole, (WORD)(10));

cout << "Спробуйте ще раз\n";

std::cin.clear();//Почистили поток

std::cin.ignore(std::cin.rdbuf()->in\_avail());//Восстановили поток

continue;

}

if (value3 > 0)

condition = false;

else

cout << "\nВи сильно недооцiнюєте складнiсть цього RGZ\n";

}

SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << "додано" << std::endl;

condition = true;

while (condition) {

SetConsoleTextAttribute(hConsole, (WORD)(14));

/\*if ((std::cin.rdbuf()->in\_avail()) != 1);\*/

cout << "Введiть Перiод (у днях) : ";

std::cin >> value4;

if (std::cin.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if (value4 > 0)

condition = false;

else

cout << "Навiть професiоналу на це потребується бiльше часу";

}SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << "додано" << std::endl;

condition = true;

while (condition) {

SetConsoleTextAttribute(hConsole, (WORD)(14));

cout << "Введiть кiлькiсть завдань : ";

std::cin >> value5;

if (std::cin.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << "Спробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if (value5 > 0)

condition = false;

else

cout << " Навiщо взагалi його робити?\n";

}SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << " додано" << std::endl;

condition = true;

while (condition) {

SetConsoleTextAttribute(hConsole, (WORD)(14));

cout << "Введіть факультет/кафедру : ";

std::cin >> value6;

if (std::cin.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << " Спробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

condition = false;

if (value6 > 0)

condition = false;

else

cout << " Ви впевненi, що таке iснує? " << std::endl;

}SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << " додано" << std::endl;

SetConsoleTextAttribute(hConsole, (WORD)(5));

add.Set\_predmet(value1);

add.Set\_teme(value2);

add.Set\_bal(value3);

add.Set\_termin(value4);

add.Set\_zavdanj(value5);

add.Set\_cathedra(value6);

}

/\*\*

Реалізація функцій класу Scientific\_SR та перевантажень

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 3.2.4

\* @file Scientific\_SR.cpp

\*/

#include "prototaype.h"

#include "SR.h"

Scientific\_SR& Scientific\_SR::operator = (const Scientific\_SR& SR)

{

//cout << "Перегрузка оператора '='" << endl;

if (this == &SR)

return \*this;

surname = SR.surname;

cientific\_topic = SR.cientific\_topic;

predmet = SR.predmet;

zavdanj = SR.zavdanj;

termin =SR.termin;

return \*this;

}

Scientific\_SR& Scientific\_SR::operator = (const Scientific\_SR\* SR)

{

//cout << "Перегрузка оператора '='" << endl;

if (this == SR)

return \*this;

if (typeid(\*this).name() != typeid(\*SR).name()) throw std::invalid\_argument("помилка зберігання");;

surname = SR->surname;

cientific\_topic = SR->cientific\_topic;

predmet = SR->predmet;

zavdanj = SR->zavdanj;

termin = SR->termin;

return \*this;

}

bool operator < (Scientific\_SR& p, Scientific\_SR& p2) {

int i = 0;

if (p.Get\_surname() < p2.Get\_surname())i++;

if (p.Get\_predmet() < p2.Get\_predmet())i++;

if (p.Get\_cientific\_topic() < p2.Get\_cientific\_topic())i++;

if (p.Get\_zavdanj() < p2.Get\_zavdanj())i++;

if (p.Get\_termin() < p2.Get\_termin())i++;

if (i < 3)return 0;

else return 1;

}

bool operator > (Scientific\_SR& p, Scientific\_SR& p2)

{

int i = 0;

if (p.Get\_surname() > p2.Get\_surname())i++;

if (p.Get\_predmet() > p2.Get\_predmet())i++;

if (p.Get\_cientific\_topic() > p2.Get\_cientific\_topic())i++;

if (p.Get\_zavdanj() > p2.Get\_zavdanj())i++;

if (p.Get\_termin() > p2.Get\_termin())i++;

if (i < 3)return 0;

else return 1;

}

bool operator== (Scientific\_SR& p, Scientific\_SR& p2) { return !(p < p2) && !(p > p2); }

bool operator!= (Scientific\_SR& p, Scientific\_SR& p2) { return !(p == p2); }

bool operator <= (Scientific\_SR& p, Scientific\_SR& p2) { return (p == p2) || !(p2 < p); }

bool operator >= (Scientific\_SR& p, Scientific\_SR& p2) { return (p == p2) || !(p2 > p); }

//вивод на екран

ostream& operator<< (ostream& out, Scientific\_SR& pr)

{

out << "|" << std::setw(14) <<pr.Get\_predmet() << "|" << std::setw(14) << pr.Get\_surname()

<< "|" << std::setw(18) << pr.Get\_cientific\_topic()

<< "|" << std::setw(11) << pr.Get\_termin() << "|" << std::setw(11) << pr.Get\_zavdanj() << "|"

<< "\n------------------------------------------------------------------------------------\n";

return out;

}

//читання з файлу

ifstream& operator >> (ifstream& stream, Scientific\_SR& add)

{

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9іІїЇ -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2("^[A-za-zА-Яа-яіІЇї][A-za-zА-Яа-я0-9іІїЇ]{1,20}([\\.,]?[-\_ ]?[А-Яа-яA-za-zіІЇї]{1,20}){0,2}$");//перевірка на правильний ввод

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

std::ifstream fOpen;

int value;

std::string text;

stream >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

SetConsoleTextAttribute(hConsole, (WORD)(12));

std::cout << "Критична помилка в написаному" << std::endl;

text = "Eror(No name)";

SetConsoleTextAttribute(hConsole, (WORD)(10));

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << "Програма виявила помилку в написаному та успішно випрвила її" << std::endl;

SetConsoleTextAttribute(hConsole, (WORD)(10));

} add.Set\_surname(text);////туту

stream >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

text = "Eror(No name)";

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

std::cout << "Програма виявила помилку в написаному та успішно випрвила її" << std::endl;

}add.Set\_predmet(text);

stream >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

text = "Eror(No name)";

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

std::cout << "Програма виявила помилку в написаному та успішно випрвила її" << std::endl;

}add.Set\_cientific\_topic(text);

stream >> value;

if (stream.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(15));

value = NULL;

std::cout << "Критична помилка в напиcаннi" << std::endl;

stream.clear();

stream >> text;

SetConsoleTextAttribute(hConsole, (WORD)(12));

}

add.Set\_termin(value);

stream >> value;

if (stream.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(15));

value = NULL;

std::cout << "Критична помилка в напиcаннi" << std::endl;

stream.clear();

stream >> text;

SetConsoleTextAttribute(hConsole, (WORD)(12));

}

add.Set\_zavdanj(value);

return stream;

}

/////////////////////////////

/\*\*

@brief

@details

@param

@return

\*/

void Scientific\_SR::EL(Scientific\_SR& add)

{

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9іІїЇ -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2("^[A-za-zА-Яа-яіІЇї][A-za-zА-Яа-я0-9іІїЇ]{1,20}([\\.,]?[-\_ ]?[А-Яа-яA-za-zіІЇї]{1,20}){0,2}$");//перевірка на правильний ввод

int value4, value5;

std::string value1, value2, value3,s2;

bool condition = true;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть ПРIЗВИЩЕ ВИКЛАДАЧА предмету : ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> value1;

while (cin.peek() != '\n') cin >> s2, value1 += " " + s2;

if ((regex\_match(value1, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value1, regex\_integer)) == 0)

{

value1[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}std::cout << "додано" << std::endl;

condition = true;

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть назву предмету : ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> value3;

while (cin.peek() != '\n') cin >> s2, value3 += " " + s2;

if ((regex\_match(value3, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value3, regex\_integer)) == 0)

{

value1[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}std::cout << "додано" << std::endl;

condition = true;

while (condition) {

SetConsoleTextAttribute(hConsole, (WORD)(10));

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введіть Наукову теме\n(Примiтка: при введенi слiв бiльше одного в можна використовувати пробіл''\_''):";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cin >> value2;

while (cin.peek() != '\n') cin >> s2, value2 += " " + s2;

if ((regex\_match(value2, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value2, regex\_integer)) == 0)

{

value2[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}

std::cout << "додано" << std::endl;

condition = true;

while (condition) {

SetConsoleTextAttribute(hConsole, (WORD)(14));

cout << "Введiть Перiод (у днях) : ";

std::cin >> value4;

if (std::cin.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if (value4 > 0)

condition = false;

else

cout << "Навiть професiоналу на це потребується бiльше часу";

}SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << "додано" << std::endl;

condition = true;

while (condition) {

SetConsoleTextAttribute(hConsole, (WORD)(14));

cout << "Введiть кiлькiсть завдань : ";

std::cin >> value5;

if (std::cin.fail()) {

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << "Спробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if (value5 > 0)

condition = false;

else

cout << " Навiщо взагалi його робити?\n";

}SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << " додано" << std::endl;

add.Set\_zavdanj(value5);

add.Set\_surname(value1);

add.Set\_cientific\_topic(value2);

add.Set\_predmet(value3);

add.Set\_termin(value4);

}

/\*\*

@brief

@details

@param

@return

\*/

void Scientific\_SR::Riad\_FL(List\_SR<SR>& A)

{

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9 -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2(("^[A-za-zА-Яа-я][A-za-zА-Яа-я0-9]{1,20}[\\.,]?[-\_ ]?[А-Яа-яA-za-z]{1,20}$"));//перевірка на правильний ввод

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

Scientific\_SR add;

std::ifstream fOpen;

int temp = 0;

std::string text, adresa;

std::cout << "\nЯкий файл відкрити?\n\n 1. follower.txt\n 2. follower.txt\n Ваш вiбiр:";

std::cin >> temp;

switch (temp)

{

case 1:adresa = "follower.txt";///

break;

case 2:adresa = "follower.txt";//

break;

}

fOpen.open(adresa, std::ios\_base::in);

if (!fOpen)

{

SetConsoleTextAttribute(hConsole, (WORD)(5));

std::cout << "File is not open\n\n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

return;

}SetConsoleTextAttribute(hConsole, (WORD)(10));

while (!fOpen.eof())

{

fOpen >> add;

A.add(&add);

}

fOpen.close();

}

/\*\*

Реалізація функцій-тексту класу Scientific\_SR

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 3.2.4

\* @file Scientific\_SR\_menu.cpp

\*/

#include "prototaype.h"

#include "SR.h"

/\*\*

@brief

@details

@param

@return

\*/

int Scientific\_SR::vstup\_info(number nov, List\_SR<SR>& List)

{

Scientific\_SR h = Scientific\_SR();

Dialog\_menu D;

system("cls");

number a = 1, u = 0;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

std::cout << "Доброго дня користувач\n ";

cout << "\n\nВи знаходитесь у класi наслiднику 'наукові ргз' ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << "\n\n Почнемо роботу\n\n Ввести данi\n ";

while (a != 0)

{

SetConsoleTextAttribute(hConsole, (WORD)(12));

cout << "\n 1: З клавiатири\n 2: 3 Файлу\n Ваш вiбiр: ";

cin >> a;

switch (a)

{

case 1:SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << "Скiльки елементiв ви хочете додати? :";

std::cin >> a;

for (int i = 0; i < a; i++)

List.add\_from\_keyboard(h);

a = 0;

break;

case 2:SetConsoleTextAttribute(hConsole, (WORD)(10));

h.Riad\_FL(List) , a = 0;

break;

}

}

a = h.main\_menu(a);

while (a)

{

try{switch (a)

{

case 1:

system("cls");

h.printTableHead();

List.Print\_run(List, h);

break;

case 2: system("cls"), List.dop\_INFO(List, h);

break;

case 3: system("cls"), List.delet\_el(List, h);

break;

case 4: system("cls"), List.Print\_el(List, h);

break;

case 5:

system("cls");

List.delete\_list\_one\_Class(h);

std::cout << "Успішно";

break;

case 6: List.vstup\_info(nov, List);

break;

case 7: nov = 123;

system("cls");

return 123;

break;

case 9:List.delete\_list(List);

system("cls");

nov = 0;

return 0;

break;

}

}

catch (std::exception & aErr)

{

cout << " спійманов виняток у Scientific SR:" << aErr.what() << endl;

}

catch (std::logic\_error & aErr)

{

cout << "Сталася помилка: " << aErr.what() << endl;

}

catch (int aMsg)

{

cout << "Перехоплено int:" << aMsg << endl;

}

catch (std::out\_of\_range e) {

cout << e.what() << endl;

}

catch (std::ios\_base::failure e) {

cout << e.what() << endl;

}

catch (std::invalid\_argument e) {

cout << e.what() << endl;

}

if (a != 6)

{

string pit = "Бажаєте ще щось зробити?";

a = D.yes\_and\_no(a, pit);

if (a == 1)

a = h.main\_menu(a);

else a = 9;

}

else a = h.main\_menu(a);

}

return 0;

}

/\*\*

@brief

@details

@param

@return

\*/

const int Scientific\_SR::main\_menu(number e)

{

system("cls");

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << " Меню 'наукові роботи' " << endl;

std::cout << " \n 1. Вивести на екран список\n";

system("ping 1.1.1.1 -n 0.001 -w 0.001 > nul");

std::cout << " 2. Додати елемент в список\n";

std::cout << " 3. Видалити елемент з списку за номереом\n";

std::cout << " 4. Відобразити елемент списку за номером\n";

std::cout << " 5. Очистити список \n\n";

std::cout << " 6. Перейти до додаткового меню \n";

std::cout << " 7. Перейти до попереднього меню \n\n";

std::cout << " 9. Завершити роботу \n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << " \nваш вибiр: ";

std::cin >> e;

system("cls");

return e;

}

/\*\*

@brief

@details

@param

@return

\*/

const void Scientific\_SR::printTableHead()

{

std::cout << "\n\n--------------------------------------------------------------------------------------" << std::endl;

std::cout << "| | Предмет | Прізвище | Наукова |Строк здачi| Кiлькiсть |" << std::endl;

std::cout << "| | викладача | викладач | темаматика | RGZ | завдань |" << std::endl;

std::cout << "---------------------------------------------------------------------------------------" << std::endl;

}

/\*\*

Опис Каласів

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 10.0.4

\* @file SR.h

\*/

#include "prototaype.h"

#define ARRAY\_SIZE 255///розмір контейнера

template<class TS> class List\_SR;

//class Dialog\_menu;

//Dialog\_menu D;

/\*\*

Клас - меню

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 1.0.1

\* @file SR.h

\*/

///Клас - меню

class Dialog\_menu

{

public:

int Start\_Menu(number n);

const int main\_menu(number e);

int yes\_and\_no(number& t, string p);

int kriter(number n);

const void error(number& a, number& d);

};

/\*\*

Абстрактний клас

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 2.0.1

\* @file SR.h

\*/

///Абстрактний клас

class SR

{

public:

virtual SR\* clone() = 0;

virtual void Riad\_FL(List\_SR<SR>&) = 0;///< віртуальна функція для додавання данних із файлу

virtual const int main\_menu(number) = 0;///< віртуальна функція для реалізацію головного меню для класу

virtual const void printTableHead() = 0;///< віртуальна функція для реалізації вивода шапки таблиці

virtual const void show() = 0;///< вивод елеменів

virtual int vstup\_info(number, List\_SR<SR>&) = 0;///< віртуальна функція для реалізації початку програми

virtual const string Get\_predmet() = 0;

virtual const string Get\_subject() = 0;

void\* operator new(size\_t size)

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(5));

void\* ptr = malloc(size);

if (ptr == NULL)

cout << "Помилка вилілення памяті" << endl;////trow std::bad\_allic()

SetConsoleTextAttribute(hConsole, (WORD)(10));

return ptr;

}

void\* operator new[](size\_t size)

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(5));

void\* ptr = malloc(size);

if (ptr == NULL)

cout << "Помилка вилілення памяті" << endl;////trow std::bad\_allic()

SetConsoleTextAttribute(hConsole, (WORD)(10));

return ptr;

}

void operator delete(void\* del)

{

free(del);

}

void operator delete[](void\* del)

{

free(del);

}

};

/// Клас - RGZ

class RGZ : public SR

{

private:

string subject;

string predmet;///< предмет

string tema;///< тема

int bal;///< кількість балів за завдання

int zavdanj;///< кількість завдань

int termin;///< строк здачи РГЗ у днях

int cathedra;///< кафедра

public :

/\*\*

@brief Функція конструктор з параметрами

\*/

RGZ()///конструктор з параметрами

{

subject = "RGZ";

predmet = "NO name";

tema = "No name";

bal = NULL;

zavdanj = NULL;

termin = NULL;

cathedra = NULL;

}

/\*\*

@brief Функція деструктор

\*/

~RGZ() { ; }///деструктор

const void show()//// вивод на екран

{

cout << " |" << std::setw(14) << Get\_predmet() << "|" << std::setw(14) << Get\_teme()

<< "|" << std::setw(18) << Get\_bal() << "|" << std::setw(11) << Get\_termin() << "|" << std::setw(11)

<< Get\_zavdanj() << "|"

<< "\n------------------------------------------------------------------------------------\n";

}

/\*\*

@Опис і реалізація Get

\*/

const string Get\_subject() { return subject; }

/\*\*

@brief Опис і реалізація Get предмету

@return predmet

\*/

const string Get\_predmet() { return predmet; }

/\*\*

@brief Опис і реалізація Get тема

@return tema

\*/

string Get\_teme() { return tema; }

/\*\*

@brief Опис і реалізація Get кількість баліа

@return bal

\*/

int Get\_bal() { return bal; }

/\*\*

@brief Опис і реалізація Get кількість завдань

@return zavdanj

\*/

int Get\_zavdanj() { return zavdanj; }

/\*\*

@brief Опис і реалізація Get терміну виконання завдання

@return termin

\*/

int Get\_termin() { return termin; }

int Get\_cathedra() { return cathedra; }

/\*\*

@brief Опис і реалізація Set предмет

@param[in] predmet

@return Nothing returns

\*/

void Set\_predmet(string value1) { this->predmet = value1; }

/\*\*

@brief Опис і реалізація Set тема

@param[in] tema

@return Nothing returns

\*/

void Set\_teme(string value1) { this->tema = value1; }

/\*\*

@brief Опис і реалізація Set кількість балів

@param[in] bal

@return Nothing returns

\*/

void Set\_bal(int value) { this->bal = value; }

/\*\*

@brief Опис і реалізація Set кількість завдань

@param[in] zavdanj

@return Nothing returns

\*/

void Set\_zavdanj(int value) { this->zavdanj = value; }

/\*\*

@brief Опис і реалізація Set термін здачі РГЗ

@param[in] termin

@return Nothing returns

\*/

void Set\_termin(int value) { this->termin = value; }

void Set\_cathedra(int value) { this->cathedra = value; }

void Riad\_FL(List\_SR<SR>&) override;

const int main\_menu(number) override;

const void printTableHead() override;

int vstup\_info(number, List\_SR<SR>&) override;

void EL(RGZ& add);

SR\* clone() { return (SR\*)new RGZ(\*this);}

/////Перегрузки//////

RGZ& operator = (const RGZ&);/// Перевантаження оператора "=" для класу RGZ

RGZ& operator = (const RGZ\*);////// Перевантаження оператора "=" для класу RGZ\*

friend bool operator > (RGZ& p, RGZ&);////// Перевантаження оператора ">" для класу RGZ

friend bool operator < (RGZ& p, RGZ&);/// Перевантаження оператора "<" для класу RGZ

friend bool operator == (RGZ& p, RGZ&);/// Перевантаження оператора "==" для класу RGZ

friend bool operator != (RGZ& p, RGZ& p2);/// Перевантаження оператора "!=" для класу RGZ

friend bool operator <= (RGZ& p, RGZ& p2);/// Перевантаження оператора "<=" для класу RGZ

friend bool operator >= (RGZ& p, RGZ& p2);/// Перевантаження оператора ">=" для класу RGZ

friend ostream& operator << (ostream&, RGZ&);/// Перевантаження оператора "<<" для класу RGZ

friend ifstream& operator >> (ifstream&,RGZ&);/// Перевантаження оператора ">>" для класу RGZ

};

/\*\*

Клас - наукових RGZ

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 3.1.1

\* @file SR.h

\*/

///Клас - наукових завдань

class Scientific\_SR final : public RGZ

{

private:

string subject;

string surname;///< прізвище викладача

string cientific\_topic;///< наукова тема

string predmet;///<

int zavdanj;///<

int termin;///<

public:

Scientific\_SR()///@brief конструктор

{

subject = "Scientific SR";

surname = "NO name";

cientific\_topic = "No name";

predmet = "No name";

zavdanj=NULL;

termin=NULL;

}

~Scientific\_SR()/// деструктор

{}

const string Get\_subject() { return subject; }

const string Get\_surname() { return surname; }/// @return surname of program creation.

const string Get\_cientific\_topic() { return cientific\_topic; }/// @return cientific\_topic of program creation.

const string Get\_predmet() { return predmet; }/// @return Предмет вивчення

const int Get\_zavdanj() { return zavdanj; }/// @return Кількіcть завдань

const int Get\_termin() { return termin; }/// @return Термін здачі РГЗ

void Set\_surname(string value1) { surname = value1; }/// @brief Опис і реалізація Set прізвищя викладача @param[in] surname @return Nothing returns

void Set\_cientific\_topic(string value) { cientific\_topic = value; } /// @brie fОпис і реалізація Set Тема РГЗ @param[in] cientific\_topic @return Nothing returns

void Set\_predmet(string value1) { this->predmet = value1; }/// @brie fОпис і реалізація Set Предмет @param[in] predmet @return Nothing returns

void Set\_zavdanj(int value) { this->zavdanj = value; }/// @brie fОпис і реалізація Set кількість завдань @param[in] zavdanj @return Nothing returns

void Set\_termin(int value) { this->termin = value; }/// @brie fОпис і реалізація Set терміну здачі РГЗ @param[in] termin @return Nothing returns

SR\* clone() {return (SR\*)new Scientific\_SR(\*this); }

const void show() {

cout << " |" << std::setw(14) << Get\_predmet()<< "|" << std::setw(14) << Get\_surname()

<< "|" << std::setw(18) << Get\_cientific\_topic()

<< "|" << std::setw(11) << Get\_termin() << "|" << std::setw(11) << Get\_zavdanj() << "|"

<< "\n------------------------------------------------------------------------------------\n";

}

const int main\_menu(number) override;

const void printTableHead() override;

void Riad\_FL(List\_SR<SR>&) override;

int vstup\_info(number, List\_SR<SR>&) override;

void EL(Scientific\_SR& add);

Scientific\_SR& operator = (const Scientific\_SR&);/// Перевантаження оператора "=" для класу Scientific\_SR

Scientific\_SR& operator = (const Scientific\_SR\*);/// Перевантаження оператора "=" для класу Scientific\_SR

friend bool operator > (Scientific\_SR& p, Scientific\_SR&);/// Перевантаження оператора ">" для класу Scientific\_SR

friend bool operator < (Scientific\_SR& p, Scientific\_SR&);/// Перевантаження оператора "<" для класу Scientific\_SR

friend bool operator == (Scientific\_SR& p, Scientific\_SR&);/// Перевантаження оператора "==" для класу Scientific\_SR

friend bool operator != (Scientific\_SR& p, Scientific\_SR&);/// Перевантаження оператора "!=" для класу Scientific\_SR

friend bool operator <= (Scientific\_SR& p, Scientific\_SR&);/// Перевантаження оператора "<=" для класу Scientific\_SR

friend bool operator >= (Scientific\_SR& p, Scientific\_SR&);/// Перевантаження оператора ">=" для класу Scientific\_SR

friend ostream& operator << (ostream&, Scientific\_SR&);/// Перевантаження оператора "<<" для класу Scientific\_SR

friend ifstream& operator >> (ifstream&, Scientific\_SR&);/// Перевантаження оператора ">>" для класу Scientific\_SR

};

\*\*

Клас - збірник РГЗ

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 3.1.1

\* @file SR.h

\*/

///Клас - збірник РГЗ

class kolektsiya\_SR :public SR

{

private:

string subject;

string predmet;///< Предмет

string author;///< Автор збірника

string collection;///< Назва збірника

string Vydavnytstvo;///< Видавництво

int data;///< Dата публікації

public:

kolektsiya\_SR()////Конструтор з параметрами

{

subject = "kolektsiya SR";

predmet = "No name";

author = "No name";

collection = "No name";

Vydavnytstvo = "No name";

data = 0;

}

~kolektsiya\_SR() { ; }////Деструктор

const string Get\_subject() { return subject; }

const string Get\_predmet() { return predmet; }/// @return Предмет

const string Get\_author() { return author; }/// @return Автор

const string Get\_collection() { return collection; }/// @return Назва збірника

const string Get\_Vydavnytstvo() { return Vydavnytstvo; }/// @return Назва видавництва

const int Get\_data() { return data; }/// @return Дата

void Set\_predmet(string value) { predmet = value; }/// @brief Опис і реалізація Set Предмету @param[in] surname @return Nothing returns

void Set\_data(int value1) { data = value1; }/// @brief Опис і реалізація Set Дата випуску збірника @param[in] surname @return Nothing returns

void Set\_author(string value1) { author = value1; }/// @brief Опис і реалізація Set Автор збірника @param[in] surname @return Nothing returns

void Set\_collection(string value) { collection = value; }/// @brief Опис і реалізація Set Назва збірника @param[in] surname @return Nothing returns

void Set\_Vydavnytstvo(string value1) { Vydavnytstvo = value1; }/// @brief Опис і реалізація Set Назва видавництва @param[in] surname @return Nothing returns

SR\* clone() { return (SR\*)new kolektsiya\_SR(\*this); }

const void show() {

cout << " |" << std::setw(14) << predmet << "|" << std::setw(14) <<collection << "|" << std::setw(18) << author

<< "|" << std::setw(11) << Vydavnytstvo << "|" << std::setw(11) << data << "|"

<< "\n------------------------------------------------------------------------------------\n";

}

const int main\_menu(number) override;

const void printTableHead() override;

void Riad\_FL(List\_SR<SR>&) override;

int vstup\_info(number, List\_SR<SR>&) override;

const void poshuk (List\_SR<SR>&);

void EL (kolektsiya\_SR& add);

kolektsiya\_SR& operator = (const kolektsiya\_SR&);/// Перевантаження оператора "=" для класу kolektsiya\_SR

kolektsiya\_SR& operator = (const kolektsiya\_SR\*);/// Перевантаження оператора "=" для класу kolektsiya\_SR

friend bool operator > (kolektsiya\_SR& p, kolektsiya\_SR&);/// Перевантаження оператора ">" для класу kolektsiya\_SR

friend bool operator < (kolektsiya\_SR& p, kolektsiya\_SR&);/// Перевантаження оператора "<" для класу kolektsiya\_SR

friend bool operator == (kolektsiya\_SR& p, kolektsiya\_SR&);/// Перевантаження оператора "==" для класу kolektsiya\_SR

friend bool operator != (kolektsiya\_SR& p, kolektsiya\_SR&);/// Перевантаження оператора "!=" для класу kolektsiya\_SR

friend bool operator <= (kolektsiya\_SR& p, kolektsiya\_SR&);/// Перевантаження оператора "<=" для класу kolektsiya\_SR

friend bool operator >= (kolektsiya\_SR& p, kolektsiya\_SR&);/// Перевантаження оператора ">=" для класу kolektsiya\_SR

friend ostream& operator << (ostream&, kolektsiya\_SR&);/// Перевантаження оператора "<<" для класу kolektsiya\_SR

friend ifstream& operator >> (ifstream&, kolektsiya\_SR&);/// Перевантаження оператора ">>" для класу kolektsiya\_SR

};

/\*\*

Клас - контролер

\* @author Гряник Георгій

\* @date 2020.05.22

\* @version 2.1.1

\* @file SR.h

\*/

/// Клас - контролер

template<class TS>

class List\_SR {

protected:

int index = -1;/// кількість елементів-1

TS\* data[ARRAY\_SIZE];/// масив контейнерзберіганн елементів

public:

List\_SR() { ; }/// Конструктор

List\_SR(List\_SR& rhs)/// Конструктор

{

for (int i = 0; i < rhs.index; i++) {

data[i] = rhs.data[i]->clone();

}

index = rhs.index;

}

~List\_SR()/// Деструктор

{

while (index >= 0)

delete data[index--];

}

int vstup\_info(number a, List\_SR<TS>& List)/// Навігації

{

system("cls");

Dialog\_menu D;

a = D.main\_menu(a);

while (a)

{

try {

switch (a)

{

case 1:

system("cls");

cout << List << endl;

break;

case 2: system("cls"), List.sort();

break;

case 3: system("cls"), List.delete\_list(List);

break;

case 5:

int i;

cout << "Який елемент видалити :";

cin >> i;

if (i > 0 && i - 1 < List.Get\_kol\_Index())

List.del\_el(i);

break;

case 6:

cin >> List;

break;

case 7:List.save\_in\_Fl(List);

break;

case 8:List.curriculum(List);

break;

case 12:

return a;

break;

}

}

catch (std::exception & aErr)

{

cout << " спійманов виняток у д.меню:" << aErr.what() << endl;

}

catch (std::logic\_error & aErr)

{

cout << "Сталася помилка: " << aErr.what() << endl;

}

catch (int aMsg)

{

cout << "Перехоплено int:" << aMsg << endl;

}

catch (std::out\_of\_range e) {

cout << e.what() << endl;

}

catch (std::ios\_base::failure e) {

cout << e.what() << endl;

}

catch (std::invalid\_argument e) {

cout << e.what() << endl;

}

string pit = "Бажаєте ще щось зробити?";

a = D.yes\_and\_no(a, pit);

if (a == 1)

a = D.main\_menu(a);

else a = 12;

}

return 0;

}

void sort()/// Сортування

{

int a = 0, k = 0;

Dialog\_menu D;

a = D.kriter(a);

cout << "\n В якому напрямі сортувати?" << endl << " 1. Від початку до кінція '>'\n" << " 2. Від кінця до початку '<'" << "\n Ваш вибір :"; cin >> k;

bool pr; /// для проверки отсортирован ли массив

do

{

pr = false;

switch (a)

{

case 1:

for (int i = 0; i < index; i++) /// проходим слева направо

switch (k)

{

case 1:

if (data[i] > data[i + 1]) { swap(i); pr = true; } break;

case 2:

if (data[i] < data[i + 1]) { swap(i); pr = true; }break;

} break;

case 2:

for (int i = 0; i < index; i++) /// проходим слева направо

switch (k)

{

case 1:

if (data[i]->Get\_predmet() > data[i + 1]->Get\_predmet()) { swap(i); pr = true; } break;

case 2:

if (data[i]->Get\_predmet() < data[i + 1]->Get\_predmet()) { swap(i); pr = true; }break;

} break;

case 3:

for (int i = 0; i < index; i++) /// проходим слева направо

switch (k)

{

case 1:

if (data[i]->Get\_subject() > data[i + 1]->Get\_subject()) { swap(i); pr = true; } break;

case 2:

if (data[i]->Get\_subject() < data[i + 1]->Get\_subject()) { swap(i); pr = true; }break;

} break;

}

} while (pr);

cout << " \*сортування завершено " << endl;

}

template <typename T > const void Print\_el(List\_SR<TS>& A, T pr) /// Вивод одного елементу

{

Dialog\_menu D;

if (A.Get\_kol\_Index() < 1)

{

cout << "!!!!Пустий масив!!!" << endl;

system("pause&cls");

return;

}

number d, a = 1;

std::cout << "Який елемент хочете вiдобразити: ";

std::cin >> d;

while (a)

{

a = 0;

if (d > 0 && d < A.Get\_kol\_Index())

{

try {

pr = (T\*)A[d - 1];

}

catch (std::invalid\_argument e) {

cout << "помилка виводу елементу";

D.error(a, d);

}

}

else { D.error(a, d); }

if (d == -1) a = 0;

}

pr.printTableHead();

cout << "|RGZ-" << d << pr << std::endl;

}

void save\_in\_Fl(List\_SR<TS>& A) /// зберегти все у файл

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

RGZ\* p = NULL;

std::string adresa;

int temp = 0;

std::fstream fOpen;

std::cout << "\nЯкий файл відкрити?\n\n 1. txt.txt(текстовий файл)\n 2. bib.bin(бінарний файл)\n Ваш вiбiр:";

std::cin >> temp;

switch (temp)

{

case 1:adresa = "txt.txt";

break;

case 2:adresa = "bin.bin";

break;

}

fOpen.open(adresa, std::ios::out);

if (!fOpen)

{

SetConsoleTextAttribute(hConsole, (WORD)(5));

std::cout << "File is not open\n\n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

return;

}SetConsoleTextAttribute(hConsole, (WORD)(10));

for (int i = 0; i < A.Get\_kol\_Index(); i++) {

try {

fOpen << p << std::endl;

}

catch (std::invalid\_argument e) { /\*cout << e.what() << endl;\*/ }

}

fOpen.close();

std::cout << "\nЗбереження успішно завершено" << std::endl;

}

TS\* SGet\_Elem(int Index) { return data[Index]; }/// Звернення до обєкту

const int Get\_kol\_Index() { return index + 1; }/// Кількість елементів

void add(TS\* a)/// Додати елемент в контейнер

{

if (index + 1 >= 255) return;

data[++index] = a->clone();

}

template <typename T >void add\_from\_keyboard(T& add\_from\_keyboard)

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

add\_from\_keyboard.EL(add\_from\_keyboard);

add(&add\_from\_keyboard);

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

std::cout << "\nВашi данi успiшно записанi\n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

}

template <typename T > const void Print\_run(List\_SR<TS>& A, T pr) /// спадок вивести

{

int kol = A.Get\_kol\_Index();

for (int i = 0; i < kol; i++)

try {

pr = (T\*)A[i];

cout << "|RGZ-" << i + 1 << pr << std::endl;

}

catch (std::invalid\_argument e) { /\*cout << e.what() << endl;\*/ }

}

void swap(int i) /// Swap елемент

{

try{TS\* temp;

temp = data[i];

data[i] = data[i + 1];

data[i + 1] = temp;

}

catch (std::invalid\_argument e) { cout << e.what() << endl; }

}

const void curriculum(List\_SR<TS>& A) /// Визначити кількість РГЗ, що виконує студент за весь період навчання в інституті відповідно до навчального плану

{

if (A.Get\_kol\_Index() < 1)

{

cout << "!!!!Пустий масив!!!" << endl;

system("pause&cls");

return;

}

RGZ pr;

int cu = 0, i = 0;

std::cout << "\n\nВведіть спецiальнiсть/кафедру(номер) :";

int size = A.Get\_kol\_Index();

std::cin >> cu;

for (int j = 0; j < size; j++)

try {

pr = (RGZ\*)A[j];

if (cu == pr.Get\_cathedra())

i++;

}

catch (std::invalid\_argument e) { ; }

if (i > 0)

{

std::cout << "Вашому студенту у цьому навчальному році треба зробити: " << i << " РГЗ" << std::endl;

std::cout << "Вам цiкаво з яких предметiв вам потрiбно робити \n 1:да\n 2:нет \n Ваш вибiр:";

int king;

std::cin >> king;

if (king == 1)

{

std::cout << "\n З таких предметiв :";

for (int j = 0; j < size; j++)

try {

pr = (RGZ\*)A[j];

if (cu == pr.Get\_cathedra())

if ((i++) % 4 != 0)

std::cout << " | " << pr.Get\_predmet();

else std::cout << "\n | " << pr.Get\_predmet();

}

catch (std::invalid\_argument e) { ; }

}

else return;

}

else

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(11));

std::cout << "\n\nВiтаю у цьому роцi ваи непотрiбно робити РГЗ з жодних предметiв!!!\n\n ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

}

}

template <typename T >

const void dop\_INFO(List\_SR<SR>& A,T h) /// Спосб додавання елементів

{

number a;

std::cout << "\n 1: З клавiатири\n 2: 3 Файлу\n Ваш вiбiр: ";

std::cin >> a;

switch (a)

{

case 1:

A.add\_from\_keyboard(h);

break;

case 2:

h.Riad\_FL(A), a = 0;

break;

}

}

void del()/// видалення

{

if (index - 1 < -1) return;

delete data[index--];

}

void del\_el(int i)/// допоміжна функція видалення

{

for (i--; i <= index; i++)

{

try {

data[i] = data[i + 1];

}

catch (std::invalid\_argument e) { cout << e.what() << endl; }

}

index--;

//del();

}

void delete\_list(List\_SR& A)/// Видалити список

{

while (A.Get\_kol\_Index() > 0)

A.del();

}

template <typename T >void delet\_el(List\_SR<TS>& A,T pr)/// Видалити елемент

{

if (A.Get\_kol\_Index() < 1)

{

cout << "!!!!Пустий масив!!!" << endl;

system("pause&cls");

return;

}

number d, a = 1;

std::cout << "Який елемент хочете видалити: ";

std::cin >> d;

while (a)

{

a = 0;

if (d > 0 && d <= A.Get\_kol\_Index())

{

try {

pr = (T\*)A[d - 1];

A.del\_el(d);

}

catch (std::invalid\_argument e) {

std::cout << "\nНажаль неможливо видалити неiснуючий елемент\n";

std::cout << "Введiть iнше значення ";

std::cin >> d;

a = 1;

}

}

else

{

std::cout << "\nНажаль неможливо видалити неiснуючий елемент\n";

std::cout << "Введiть iнше значення ";

std::cin >> d;

if (d == -1)

a = 0;

else a = 1;

}

}

}

template <class T2 > void delete\_list\_one\_Class(T2& pr)/// Видалення одного конкретного Класу

{

int kol = Get\_kol\_Index() - 1;

for (int i = kol; i > -1; i--)

try {

pr = (T2\*)SGet\_Elem(i);

del\_el(i + 1);

}

catch (std::invalid\_argument e) { /\*cout << e.what() << endl;\*/ }

}

friend ostream& operator << (ostream& out, List\_SR& a)/// Перевантаження <<

{

if (a.Get\_kol\_Index() == 0)

out << "Масив пустий " << endl;

else

{

out << "Ваш масив :" << endl;

for (int i = 0; i < a.Get\_kol\_Index(); i++)

{

if (typeid(\*a[i]).name() == typeid(RGZ).name())

{

RGZ temp;

temp = (RGZ\*)a[i];

out <<"|" << temp.Get\_subject()<<std::setw(12) <<temp << endl;

}

else if (typeid(\*a[i]).name() == typeid(Scientific\_SR).name())

{

Scientific\_SR temp;

temp = (Scientific\_SR\*)a[i];

out << "|" << temp.Get\_subject() << temp << endl;

}

else if (typeid(\*a[i]).name() == typeid(kolektsiya\_SR).name())

{

kolektsiya\_SR temp;

temp = (kolektsiya\_SR\*)a[i];

out << "|" << temp.Get\_subject() << temp << endl;

}

}

out << endl;

}

return out;

}

friend istream& operator >> (std::istream& is, List\_SR& obj)/// Перевантаження >>

{

RGZ x;

Scientific\_SR y;

kolektsiya\_SR z;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

std::ifstream fOpen;

int a;

string adressa;

cout << " список RGZ ------> 1\n список Наслідник наукові робот ---- > 2\n список Наслідник збирники завдань ---- > 3\n ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

cout << " \nваш вибiр: ";

cin >> a;

switch (a)

{

case 1:adressa = "add\_RGZ";

break;

case 2:adressa = "nauka";

break;

case 3:adressa = "zbyrnk";

break;

}

fOpen.open(adressa+".txt", std::ios\_base::in);

if (!fOpen)

{

std::cout << "File is not open\n\n";

return is;

}

while (!fOpen.eof()) {

switch (a)

{

case 1:fOpen >> x;

obj.add(&x);

break;

case 2:fOpen >> y;

obj.add(&y);

break;

case 3: fOpen >> z;

obj.add(&z);

break;

}

}

fOpen.close();

return is;

}

List\_SR& operator=(const List\_SR& rhs)/// Перевантаження =

{

if (this == &rhs) // a = a;

return \*this;

while (index >= 0)

delete data[index--];

for (int i = 0; i < rhs.index; i++) {

data[i] = rhs.data[i]->clone();

}

index = rhs.index;

}

SR\* operator[] (const int indx) /// Перевантаження []

{

if (indx > index || indx < 0)

throw std::out\_of\_range("Вихід за рамки діапазона масива");

return data[indx]->clone();

}

};

Додаток Б.

Результати роботи програми

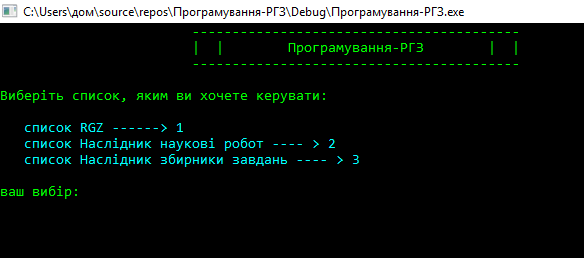


Рисунок 3 – початкове меню програми

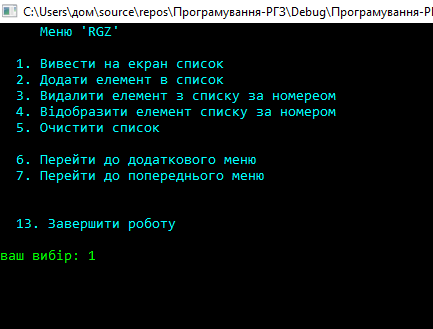


Рисунок 4 – Меню РГЗ

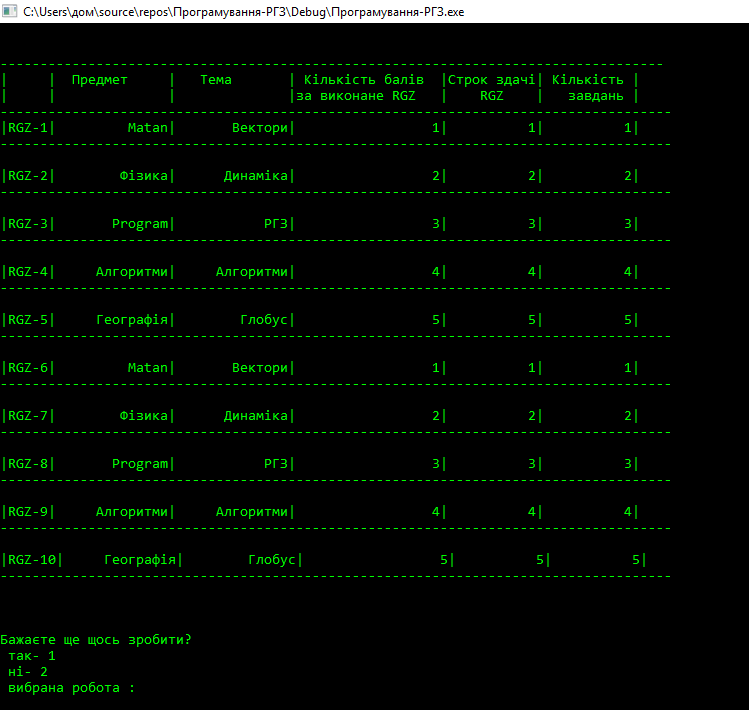


Рисунок 5 – Результат виводу даних на екран

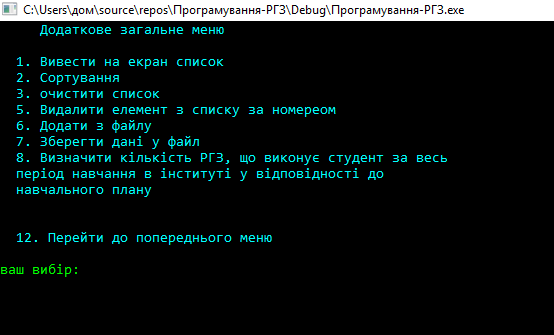


Рисунок 6 – Меню загальне меню

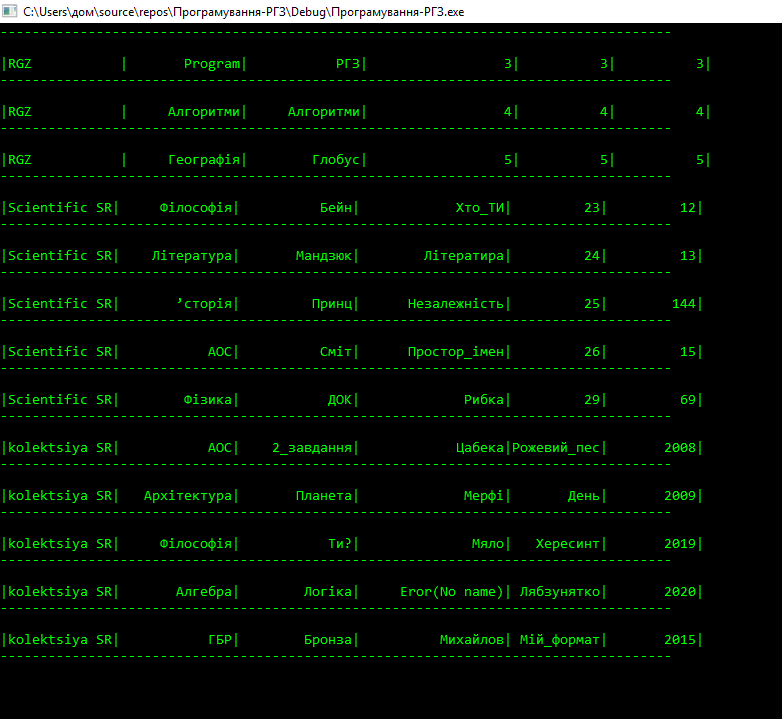


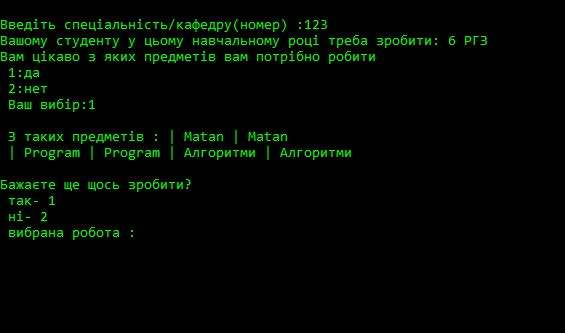
Рисунок 7 – Вивод всіх елементів на екран

Рисунок 8 – Результат визначення кількості РГЗ

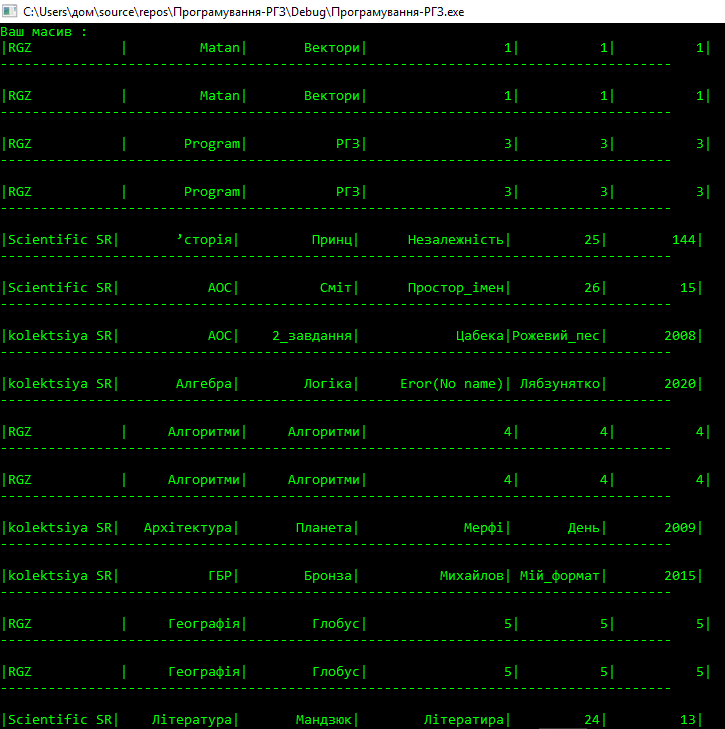


Рисунок 9 – Результат сортування даних за полем предмет

Додаток В.

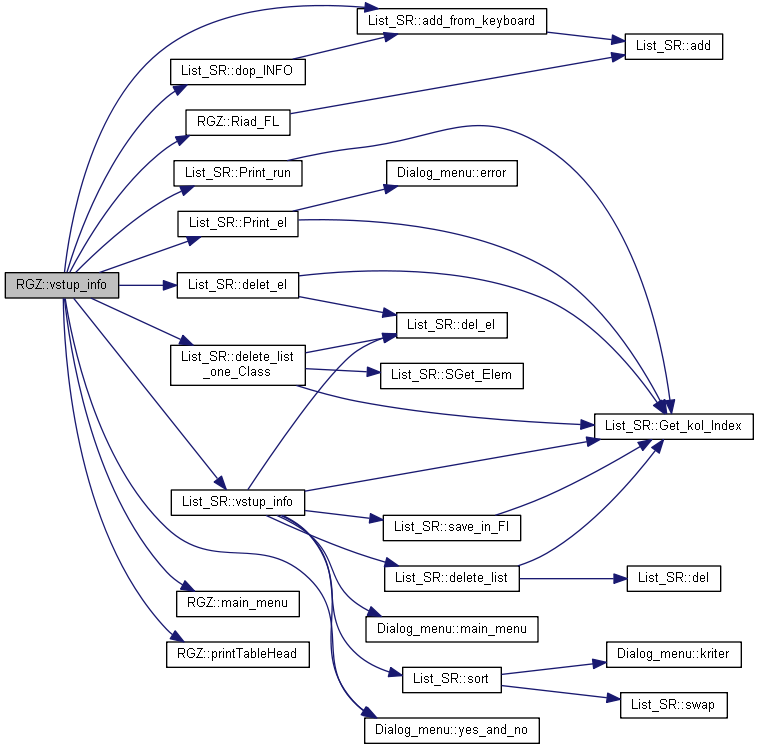
Алгоритми роботи класів

Рисунок 10 - Схема класу RGZ

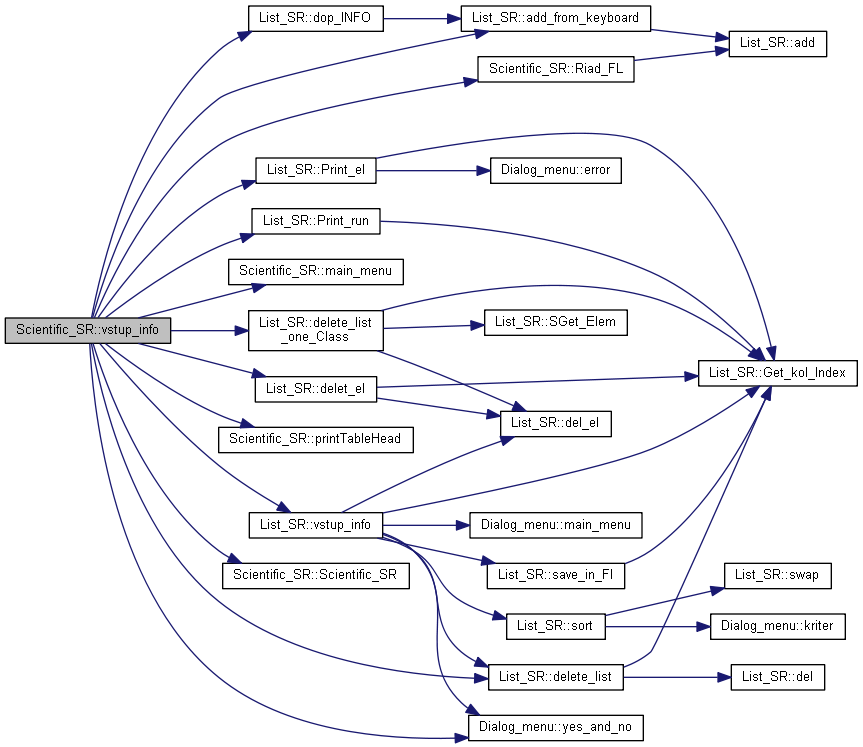


Рисунок 11 - Схема класу Scientific\_SR

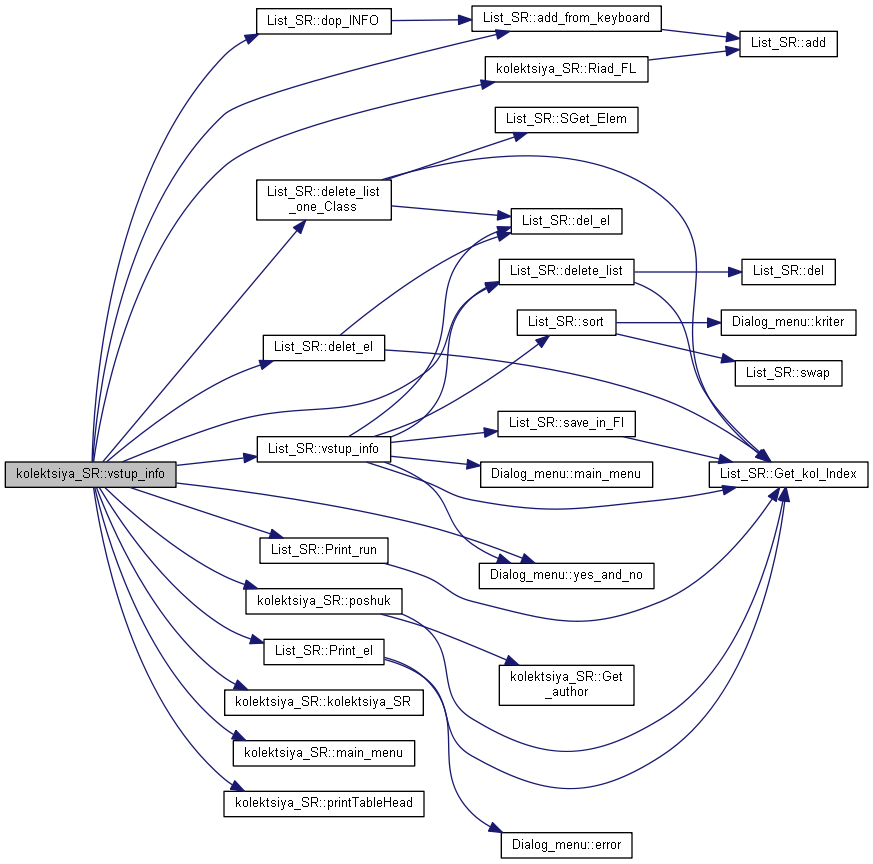


Рисунок 12 - Схема класу kolektsiy\_SR