****

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №2  
**Технологія розроблення програмного забезпечення**

«ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ ВАРІАНТІВ ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ. КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ»

Варіант 18

Виконав Перевірив:

студент групи ІА-13 Драган Михайло

Окаянченко Давид Олександрович Сергійович

Київ 2023р.

**Тема:** ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ ВАРІАНТІВ ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ. КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ

**Завдання:**

1. Ознайомитися з короткими теоретичними відомостями.

2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторії.

3. Намалюйте діаграму класів для реалізованої частини системи.

4. Виберіть 3 прецеденти і напишіть на їх основі прецеденти.

5. Розробити основні класи і структуру системи баз даних.

6. Класи даних повинні реалізувати шаблон Репозиторію для взаємодії з базою даних.

7. Підготувати звіт про хід виконання лабораторних робіт. Звіт, що подається повинен містити: діаграму прецедентів, діаграму класів системи, вихідні коди класів системи, а також зображення структури бази даних.

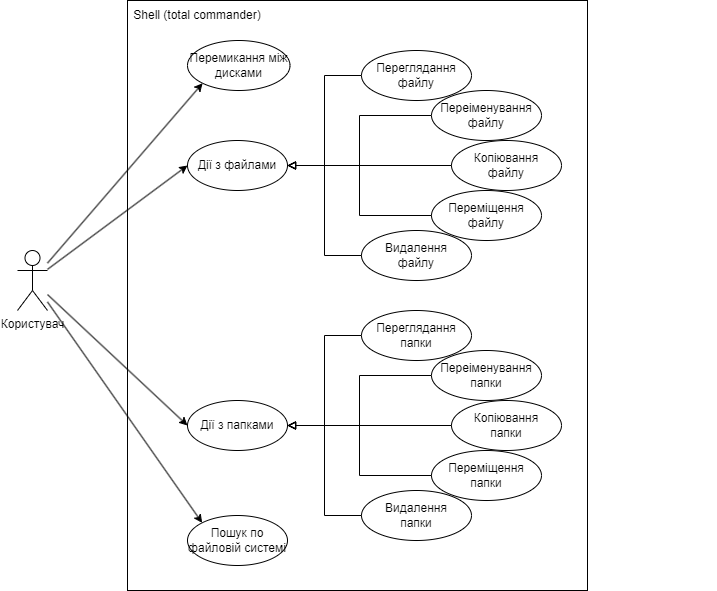
**Варіант:**

18. Shell (total commander) (state, prototype, factory method, template method, interpreter, client-server)

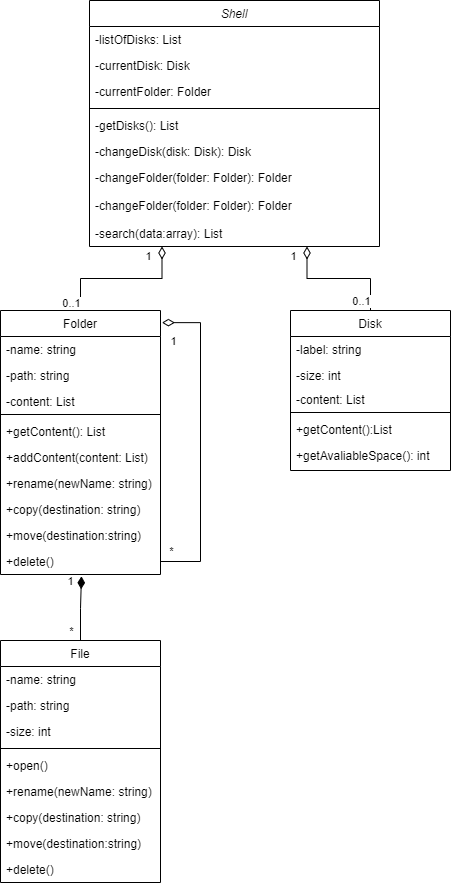
Оболонка повинна вміти виконувати основні дії в системі - перегляд файлів папок в файлової системі, перемикання між дисками, копіювання, видалення, переміщення об'єктів, пошук.

**Хід роботи**

Діаграма прецендентів:



Діаграма класів:



Розпишемо 3 сценарії використання на основі діаграми прецендентів:

Перемикання між дисками

Передумови: Користувач знаходиться в файловому менеджері.

Взаємодіючі сторони: Користувач, файловий менеджер, Диск.

Короткий опис: Користувач може перемикатися між різними дисками на своєму комп'ютері.

Основний перебіг подій:

* Користувач відкриває файловий менеджер.
* Користувач вибирає опцію "Перемикнути диск" або переходить до меню вибору диску.
* Система відображає список доступних дисків.
* Користувач вибирає потрібний диск.
* Файловий менеджер відображає вміст обраного диска.

Постумови: Користувач перейшов до вмісту іншого диска.

Виключення: відсутні.

Примітки: відсутні.

Пошук по файловій системі

Передумови: Користувач знаходиться в файловому менеджері.

Взаємодіючі сторони: Користувач, файловий менеджер, Файл, Папка.

Короткий опис: Користувач може використовувати пошукову функцію для знаходження файлів в файловій системі.

Основний перебіг подій:

* Користувач відкриває файловий менеджер.
* Користувач вибирає опцію "Пошук" або використовує відповідну комбінацію клавіш.
* Система відображає поле для введення пошукового запиту.
* Користувач вводить ключове слово або фразу для пошуку файлів.
* Файловий менеджер починає пошук і відображає результати пошуку, включаючи інформацію про знайдені файли, папки та їх розташування.

Постумови: Користувач отримав список знайдених файлів.

Виключення: відсутні.

Примітки: відсутні.

Видалення папки:

Передумови: Користувач перейшов до папки, яку бажає видалити.

Взаємодіючі сторони: Користувач, файловий менеджер, Папка, Файл.

Краткое описание: Користувач може видаляти папки та їх вміст з файлової системи.

Основний перебіг подій:

* Користувач вибирає цільову папку.
* Користувач вибирає опцію "Видалити" або використовує відповідну комбінацію клавіш.
* Система відображає підтвердження для видалення папки та інформацію про кількість файлів і підпапок, які будуть видалені.
* Користувач підтверджує дію видалення.
* Файловий менеджер видаляє обрану папку та її вміст і, за необхідності, переміщає їх у кошик.

Постумови: Папка та її вміст були видалені з файлової системи.

Виключення: Користувач відмовитися видаляти папку на етапі підтвердження.

Якщо під час виконання основного потоку виявиться, що користувач відмовився видаляти папку - використання варіанту видалення припиняється.

Примітки: відсутні.

Вихідні коди класів системи (Java):

Disk:

package com.example.os.model;  
  
public class Disk {  
 private final int id;  
 private final String label;  
 private final int size;  
  
 public Disk(int id, String label, int size) {  
 this.id = id;  
 this.label = label;  
 this.size = size;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getLabel() {  
 return label;  
 }  
  
 public int getSize() {  
 return size;  
 }  
}

File:

package com.example.os.model;  
  
public class File {  
 private final int id;  
 private final int folder\_id;  
 private final String name;  
 private final String path;  
 private final int size;  
  
 public File(int id, int folder\_id, String name, String path, int size) {  
 this.id = id;  
 this.folder\_id = folder\_id;  
 this.name = name;  
 this.path = path;  
 this.size = size;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public int getFolder\_id() {  
 return folder\_id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getPath() {  
 return path;  
 }  
  
 public int getSize() {  
 return size;  
 }  
}

Folder:

package com.example.os.model;  
  
public class Folder {  
 private final int id;  
 private final String name;  
 private final String path;  
  
 public Folder(int id, String name, String path) {  
 this.id = id;  
 this.name = name;  
 this.path = path;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getPath() {  
 return path;  
 }  
}

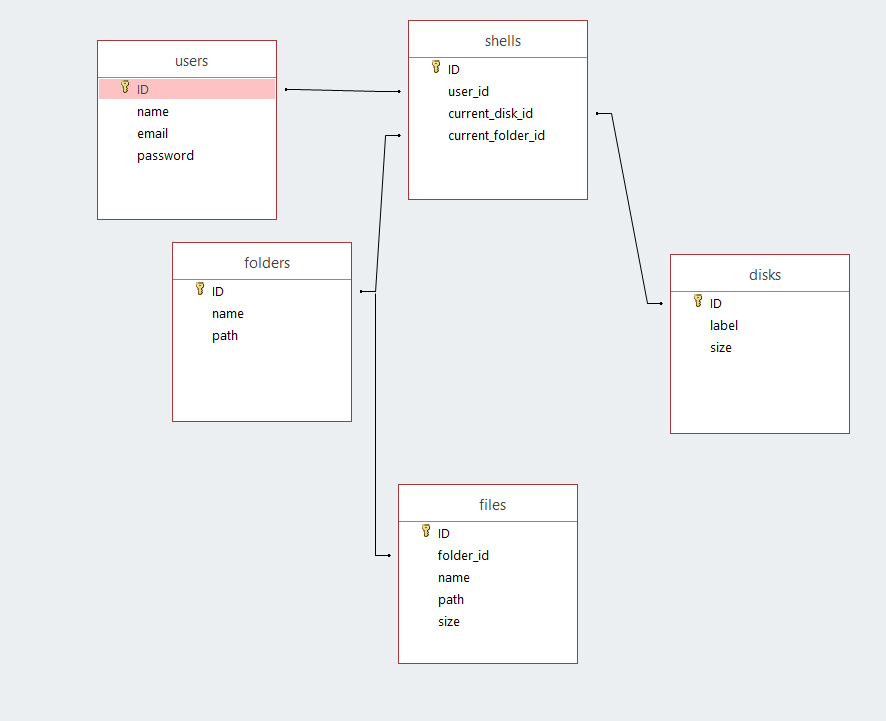
Shell:

package com.example.os.model;  
  
public class Shell {  
 private final int id;  
 private final int current\_disk\_id;  
 private final int current\_folder\_id;  
  
 public Shell(int id, int current\_disk\_id, int current\_folder\_id) {  
 this.id = id;  
 this.current\_disk\_id = current\_disk\_id;  
 this.current\_folder\_id = current\_folder\_id;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public int getCurrent\_disk\_id() {  
 return current\_disk\_id;  
 }  
  
 public int getCurrent\_folder\_id() {  
 return current\_folder\_id;  
 }  
}

User:

package com.example.os.model;  
  
public class User {  
 private final int id;  
 private final String name;  
 private final String email;  
 private final String password;  
  
 public User(int id, String name, String email, String password) {  
 this.id = id;  
 this.name = name;  
 this.email = email;  
 this.password = password;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
}

Структура бази даних:



**Висновок:** У ході виконання лабораторної роботи було проведено ознайомлення з теоретичними відомостями та розроблено діаграма прецедентів, діаграма класів. Окрім того, підготовлений звіт включає всі необхідні компоненти, що відображають початкову структуру розробленої системи.