**Мета:** Вивчити особливості роботи з класами та їх спадкуванням у Java.

**Обладнання:** ПК, програмне забезпечення Intellij IDEA, методичні вказівки та завдання до лабораторної роботи.

**Завдання.**

1. Розробити консольний застосунок для роботи з базою даних,  
   що зберігається у текстовому файлі (початковий масив не менше 5 записів).  
   Структура бази даних описується ієрархією класів згідно вашого варіанта.  
   Для ідентифікації спроби введення з клавіатури некоректних даних описати виключення.  
   Реалізувати методи у базовому класі для:  
   – додавання записів; редагування записів; знищення записів; виведення інформації з файла на екран; обчислення та виведення на екран результатів згідно свого варіанта індивідуального завдання.  
   Меню програми реалізувати по натисненню на певні клавіші: наприклад, Enter – вихід, п - пошук, р – редагування тощо.  
   Один з методів індивідуального завдання зробити віртуальним.  
   **Варіант 23.**  
   Батьківський (базовий) клас :  
   Сутність : Акаунт електронної пошти  
   Обов'язкові поля : E-mail, ПІБ володаря  
   Похідний клас :  
   Сутність : Спам  
   Обов'язкові поля : Дата, кількість спам повідомлень, загальна кількість повідомлень

Реалізувати з допомогою окремих методів обчислення та виведення на екран таких даних:  
1)Середня кількість спаму в день за період;  
2)Кількість днів, коли відсоток спам повідомлень був менший за задане значення;

3)Дні, коли кількість спаму збільшувалась

1. Модифікувати консольний застосунок з попереднього завдання таким чином, щоб:

А) Базовий клас був абстрактним.

Б) Реалізація методів індивідуального завдання була перенесена у похідний клас.

[**Код на GitHub**](https://github.com/pro100user15/Java/tree/master/Лабараторна%20робота%204(Ієрархія))

**Результат виконання :**

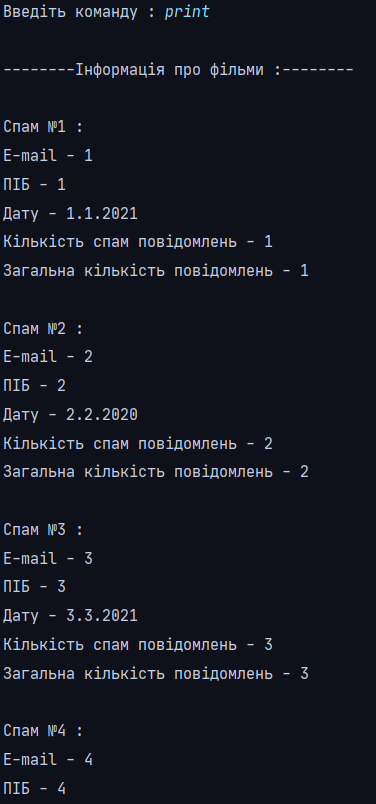
1. **Код :**

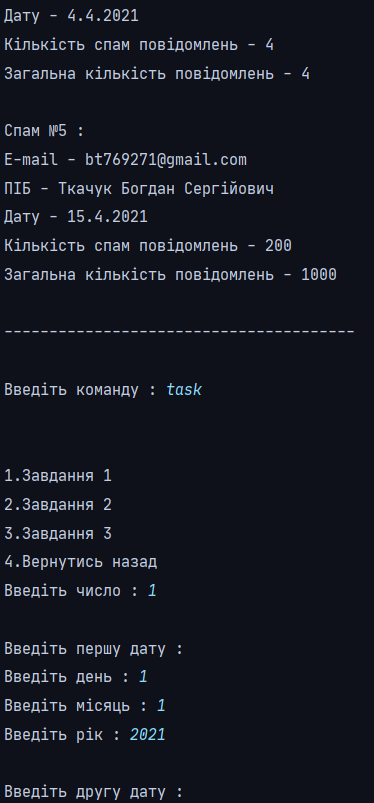
public class Email\_account {  
 String email; //E-mail  
 String fullname; //ПІБ власника  
  
 //конструктор без параметрів  
 public Email\_account() {  
 email = "";  
 fullname = "";  
 }  
  
 //конструктор з параметрами  
 public Email\_account(String email, String fullname) {  
 this.email = email;  
 this.fullname = fullname;  
 }  
  
 //метод для вводу даних  
 public void input() {  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("E-mail : ");  
 email = scanner.nextLine();  
 Pattern p = Pattern.compile("^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@" +  
 "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$");  
 Matcher m = p.matcher(email);  
 if(!m.matches()) {  
 System.err.println("Не вірно введений E-mail!");  
 this.input();  
 return;  
 }  
 System.out.print("ПІБ : ");  
 fullname = scanner.nextLine();  
 }  
  
 //метод для виводу даних  
 public void output() {  
 System.out.println("E-mail - " + email);  
 System.out.println("ПІБ - " + fullname);  
 }  
  
 //перевизначення методу toString()  
 @Override  
 public String toString() {  
 return "Email\_account{" +  
 "email='" + email + '\'' +  
 ", fullname='" + fullname + '\'' +  
 '}';  
 }  
  
 //гетери  
 public String getEmail() {return email;}  
 public String getFullname() {return fullname;}  
  
 //сетери  
 public void setEmail(String email) {this.email = email;}  
 public void setFullname(String fullname) {this.fullname = fullname;}  
}

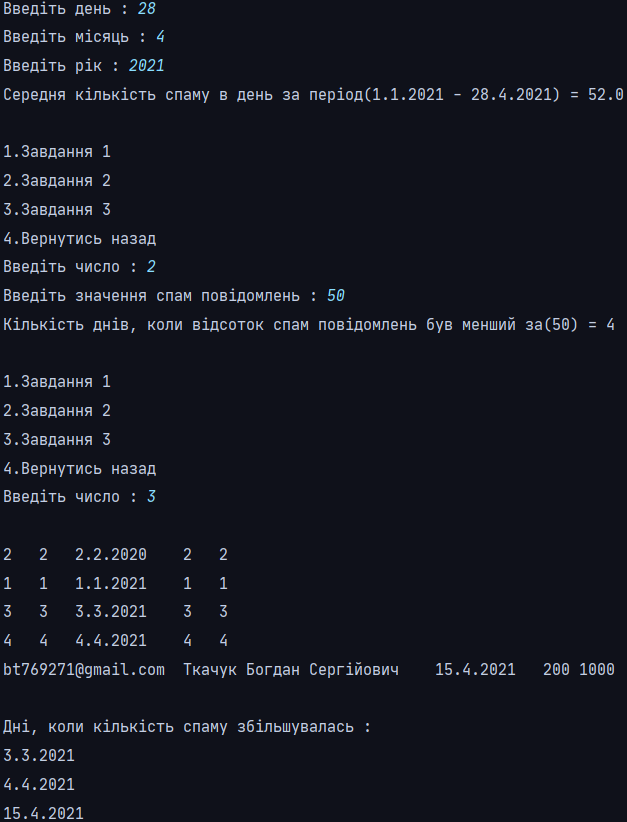
*public class* Spam *extends* Email\_account {  
 DateSpam date; *//Дата  
 int* amount; *//Кількість спам повідомлень  
 int* all\_amount\_spam; *//Загальна кількість повідомлень  
  
 //конструктор без параметрів  
 public* Spam() {  
 date = *new* DateSpam();  
 amount = 0;  
 all\_amount\_spam = 0;  
 }  
  
 *//конструктор з параметрами  
 public* Spam(*int* d, *int* m, *int* y, *int* amount, *int* all\_amount\_spam) {  
 *this*.date = *new* DateSpam(d,m,y);  
 *this*.amount = amount;  
 *this*.all\_amount\_spam = all\_amount\_spam;  
 }  
  
 *//конструктор з параметрами  
 public* Spam(String email, String fullname, *int* d, *int* m, *int* y, *int* amount, *int* all\_amount\_spam) {  
 *super*(email, fullname);  
 *this*(d, m, y, amount, all\_amount\_spam);  
 }  
  
 *//метод для вводу даних* @Override  
 *public void* input() {  
 Scanner scanner = *new* Scanner(System.*in*);  
 System.*out*.print("E-mail : ");  
 email = scanner.nextLine();  
 Pattern p = Pattern.*compile*("^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@" +  
 "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$");  
 Matcher m = p.matcher(email);  
 *if*(!m.matches()) {  
 System.*err*.println("Не вірно введений E-mail!");  
 *this*.input();  
 *return*;  
 }  
 System.*out*.print("ПІБ : ");  
 fullname = scanner.nextLine();  
 System.*out*.println("Дату : ");  
 date.inputDate();  
 System.*out*.print("Кількість спам повідомлень : ");  
 amount = scanner.nextInt();  
 System.*out*.print("Загальна кількість повідомлень : ");  
 all\_amount\_spam = scanner.nextInt();  
 }  
  
 *//метод для виводу даних* @Override  
 *public void* output() {  
 System.*out*.println("E-mail - " + email);  
 System.*out*.println("ПІБ - " + fullname);  
 System.*out*.println("Дату - " + date.toString());  
 System.*out*.println("Кількість спам повідомлень - " + amount);  
 System.*out*.println("Загальна кількість повідомлень - " + all\_amount\_spam);  
 }  
  
 *//перевизначення методу toString()* @Override  
 *public* String toString() {  
 *return* email + '\t' +  
 fullname + '\t' +  
 date + '\t' +  
 amount + '\t' +  
 all\_amount\_spam;  
 }  
  
 *public* String to\_string() {  
 *return* email + '\n' +  
 fullname + '\n' +  
 date.to\_string() + '\n' +  
 amount + '\n' +  
 all\_amount\_spam + '\n'  
 ;  
 }  
  
 *//гетери  
 public* DateSpam getDate() {  
 *return* date;  
 }  
 *public int* getAmount() {  
 *return* amount;  
 }  
 *public int* getAll\_amount\_spam() {  
 *return* all\_amount\_spam;  
 }  
  
 *//сетери  
 public void* setDate(DateSpam date) {  
 *this*.date = date;  
 }  
 *public void* setAmount(*int* amount) {  
 *this*.amount = amount;  
 }  
 *public void* setAll\_amount\_spam(*int* all\_amount\_spam) {  
 *this*.all\_amount\_spam = all\_amount\_spam;  
 }  
}

*public void* task() *throws* IOException {  
 count\_note();  
  
 *for* (; ; ) {  
 Scanner scanner = *new* Scanner(System.*in*);  
 System.*out*.print("\n1.Завдання 1" + "\n" +  
 "2.Завдання 2" + "\n" +  
 "3.Завдання 3" + "\n" +  
 "4.Вернутись назад" + "\n" +  
 "Введіть число : ");  
 *int* nomer = scanner.nextInt();  
  
 *if* (nomer < 1 || nomer > 4) {  
 System.*out*.println("Не вірно введене число!! Введіть ще раз :");  
 *continue*;  
 }  
  
 *switch* (nomer) {  
 *case* 1:  
 task\_1();  
 *break*;  
 *case* 2:  
 task\_2();  
 *break*;  
 *case* 3:  
 task\_3();  
 *break*;  
 *case* 4:  
 *return*;  
 }  
 }  
}  
  
*private void* task\_1() {  
 *try* {  
 DateSpam beg\_date = *new* DateSpam();  
 DateSpam end\_date = *new* DateSpam();  
 System.*out*.println("\nВведіть першу дату :");  
 beg\_date.inputDate();  
 System.*out*.println("\nВведіть другу дату :");  
 end\_date.inputDate();  
  
 *if* (beg\_date.better(end\_date))  
 *throw new* IOException("Перша дата не може бути бульшою за другу!");  
 *int* count\_spam = 0;  
 *int* count\_day = 0;  
 *for* (Spam spam : spam) {  
 *if* (!beg\_date.better(spam.date) || beg\_date.equally(spam.date))  
 *if* (end\_date.better(spam.date) || end\_date.equally(spam.date)) {  
 count\_spam += spam.amount;  
 count\_day++;  
 }  
 }  
  
 *if* (count\_day == 0)  
 *throw new* IOException("В заданий період не було найдено записів!");  
  
 *double* result = (*double*) count\_spam / count\_day;  
  
 System.*out*.println("Середня кількість спаму в день за період(" + beg\_date.toString() + " - " + end\_date.toString() + ") = " + result);  
 } *catch* (IOException ex) {  
 System.*out*.println(ex.getMessage());  
 }  
}  
  
*private void* task\_2() {  
 Scanner scanner = *new* Scanner(System.*in*);  
 System.*out*.print("Введіть значення спам повідомлень : ");  
 *int* amount\_spam = scanner.nextInt();  
  
 *int* count\_day = 0;  
 *for* (Spam spam : spam) {  
 *if* (spam.amount < amount\_spam)  
 count\_day++;  
 }  
  
 System.*out*.println("Кількість днів, коли відсоток спам повідомлень був менший за(" + amount\_spam + ") = " + count\_day);  
}  
  
*private void* task\_3() {  
 Spam[] arr = sort\_date();  
  
 System.*out*.println();  
 *for* (Spam p : arr) {  
 System.*out*.println(p.toString());  
 }  
  
 System.*out*.println("\nДні, коли кількість спаму збільшувалась :");  
 *for* (*int* i = 1; i < arr.length; i++)  
 *if* (arr[i].amount > arr[i - 1].amount) {  
 System.*out*.println(arr[i].getDate().toString());  
 }  
}

**Результат :**







**2)Код :**

public abstract class Email\_account {  
 String email; //E-mail  
 String fullname; //ПІБ власника  
  
 //конструктор без параметрів  
 public Email\_account() {  
 email = "";  
 fullname = "";  
 }  
  
 //конструктор з параметрами  
 public Email\_account(String email, String fullname) {  
 this.email = email;  
 this.fullname = fullname;  
 }  
  
 //метод для вводу даних  
 public abstract void input();  
  
 //метод для виводу даних  
 public abstract void output();  
  
 //гетери  
 public String getEmail() {return email;}  
 public String getFullname() {return fullname;}  
  
 //сетери  
 public void setEmail(String email) {this.email = email;}  
 public void setFullname(String fullname) {this.fullname = fullname;}  
}

public final class Spam extends Email\_account {  
 DateSpam date; //Дата  
 int amount; //Кількість спам повідомлень  
 int all\_amount\_spam; //Загальна кількість повідомлень  
  
 //конструктор без параметрів  
 public Spam() {  
 date = new DateSpam();  
 amount = 0;  
 all\_amount\_spam = 0;  
 }  
  
 //конструктор з параметрами  
 public Spam(int d, int m, int y, int amount, int all\_amount\_spam) {  
 this.date = new DateSpam(d,m,y);  
 this.amount = amount;  
 this.all\_amount\_spam = all\_amount\_spam;  
 }  
  
 //конструктор з параметрами  
 public Spam(String email, String fullname, int d, int m, int y, int amount, int all\_amount\_spam) {  
 super(email, fullname);  
 this.date = new DateSpam(d,m,y);  
 this.amount = amount;  
 this.all\_amount\_spam = all\_amount\_spam;  
 }  
  
 //метод для вводу даних  
 @Override  
 public void input() {  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("E-mail : ");  
 email = scanner.nextLine();  
 System.out.print("ПІБ : ");  
 fullname = scanner.nextLine();  
 System.out.println("Дату : ");  
 date.inputDate();  
 System.out.print("Кількість спам повідомлень : ");  
 amount = scanner.nextInt();  
 System.out.print("Загальна кількість повідомлень : ");  
 all\_amount\_spam = scanner.nextInt();  
 }  
  
 //метод для виводу даних  
 @Override  
 public void output() {  
 System.out.println("E-mail - " + email);  
 System.out.println("ПІБ - " + fullname);  
 System.out.println("Дату - " + date.toString());  
 System.out.println("Кількість спам повідомлень - " + amount);  
 System.out.println("Загальна кількість повідомлень - " + all\_amount\_spam);  
 }  
  
 //перевизначення методу toString()  
 @Override  
 public String toString() {  
 return "Spam{" +  
 email + '\t' +  
 fullname + '\t' +  
 date + '\t' +  
 amount + '\t' +  
 all\_amount\_spam +  
 '}';  
 }  
  
 public String to\_string() {  
 return email + '\n' +  
 fullname + '\n' +  
 date.to\_string() + '\n' +  
 amount + '\n' +  
 all\_amount\_spam + '\n'  
 ;  
 }  
  
 //гетери  
 public DateSpam getDate() {  
 return date;  
 }  
 public int getAmount() {  
 return amount;  
 }  
 public int getAll\_amount\_spam() {  
 return all\_amount\_spam;  
 }  
  
 //сетери  
 public void setDate(DateSpam date) {  
 this.date = date;  
 }  
 public void setAmount(int amount) {  
 this.amount = amount;  
 }  
 public void setAll\_amount\_spam(int all\_amount\_spam) {  
 this.all\_amount\_spam = all\_amount\_spam;  
 }  
}

**Висновок:** я вивчив особливості роботи з класами та їх спадкуванням у Java.