**Мета:** Вивчити особливості роботи з інтерфейсами у Java.

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

**Завдання.**

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.

**1.** Застосунок організувати таким чином, щоб в ньому існував інтерфейсний клас, якому належать усі методи (реалізація методів повинна залишитись у класі, що успадковує інтерфейс).

**Варіант 23.** Описати клас для бази зданих з інформацією про успішність групи студентів з полями: прізвище та ім’я, № залікової книжки, оцінки за 100 бальною шкалою з п’яти предметів. Впорядкувати записи у порядку зростання зростання середнього балу і вивести їх на екран у формі таблиці. Визначити відсоток студентів, що мають незадовільні оцінки.

**2.** Для консольного застосунку організувати інтерфейсний клас з двома абстрактними методами (які з створених у проекті методів робити абстрактними – обирайте самі).

**Варіант 23.**  
Батьківський (базовий) клас :  
Сутність : Поет  
Обов'язкові поля : Прізвище, мова, кількість збірок  
Похідний клас :  
Сутність : Виступ  
Обов'язкові поля : Дата, місце, кількість слухачів

Реалізувати з допомогою окремих методів обчислення та виведення на екран таких даних:  
1) Сумарна кількість слухачів;  
2) День з найбільшою кількістю слухачів;

3) Довжина прізвища;

[**Код на GitHub**](https://github.com/pro100user15/Java/tree/master/%D0%9B%D0%B0%D0%B1%D0%B0%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%20%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B0%205(%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81%D0%B8))

**Результат виконання :**

1. **Код :**

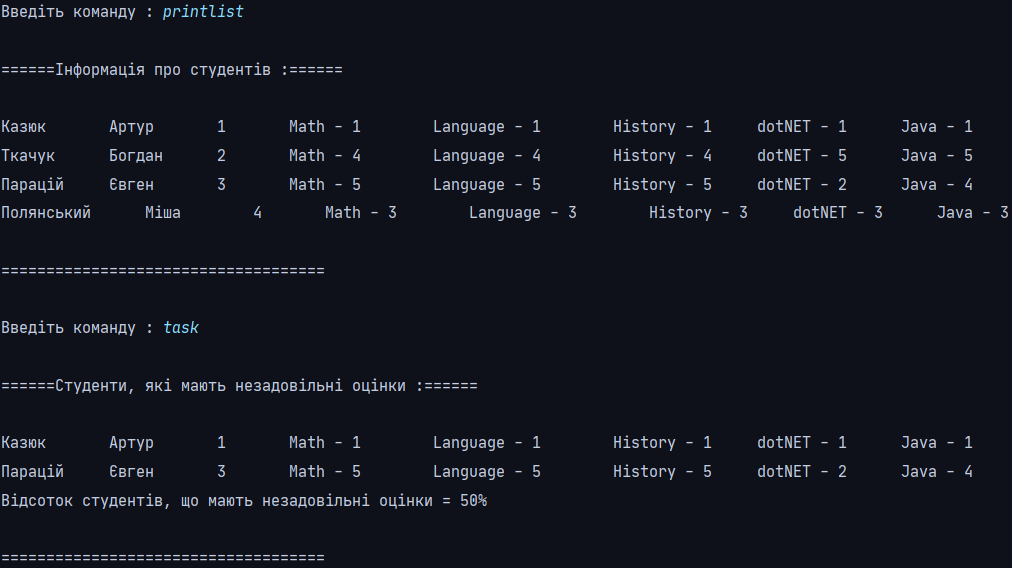
**Клас :**

*public class* Student *implements Serializable* {  
 String lastname; *//прізвище* String name; *//ім'я  
 int* nomer; *//№ залікової книжки* Subject subjects; *//предмети  
  
 public* Student() {  
 lastname = "";  
 name = "";  
 nomer = 0;  
 subjects = *new* Subject();  
 }  
  
 *public* Student(String lastname, String name, *int* nomer, *byte* Math, *byte* Language, *byte* History, *byte* dotNET, *byte* Java) {  
 *this*.lastname = lastname;  
 *this*.name = name;  
 *this*.nomer = nomer;  
 *this*.subjects = *new* Subject(Math, Language, History, dotNET, Java);  
 }  
  
  
 *void* input() {  
 Scanner scanner = *new* Scanner(System.*in*);  
 System.*out*.print("Прізвище : ");  
 lastname = scanner.nextLine();  
 System.*out*.print("Ім'я : ");  
 name = scanner.nextLine();  
 System.*out*.print("№ залікової книжки : ");  
 nomer = scanner.nextInt();  
 subjects.input();  
 }  
  
 *void* output() {  
 System.*out*.println("Прізвище - " + lastname);  
 System.*out*.println("Ім'я - " + name);  
 System.*out*.println("№ залікової книжки - " + nomer);  
 subjects.output();  
 }  
  
  
 @Override  
 *public* String toString() {  
 *return* lastname + '\t' + '\t' +  
 name + '\t' + '\t' +  
 nomer + '\t' + '\t' +  
 subjects.toString();  
 }  
  
  
 *//гетери  
 public* String getLastname() {*return* lastname;}  
 *public* String getName() {*return* name;}  
 *public int* getNomer() {*return* nomer;}  
 *public* Subject getSubject() {*return* subjects;}  
 *public double* getAverage() {  
 *int* average = 0;  
 *for*(*byte* rating : subjects.rating) {  
 average += rating;  
 }  
 *return* (*double*) average/subjects.rating.length;  
 }  
  
 *//сетери  
 public void* setLastname(String lastname) {*this*.lastname = lastname;}  
 *public void* setName(String name) {*this*.name = name;}  
 *public void* setNomer(*int* nomer) {*this*.nomer = nomer;}  
 *public void* setSubjects(Subject subjects) {*this*.subjects = subjects;}  
  
 *public boolean* isUnsatisfactory() {  
 *for*(*int* i=0;i<subjects.rating.length;i++)  
 *if*(subjects.rating[i] <= 2)  
 *return true*;  
 *return false*;  
 }  
}

**Завдання :**

*public void* task() *throws* IOException {  
 count\_note();  
  
 System.*out*.println("\n======Cтуденти, які мають незадовільні оцінки :======\n");  
 *int* count = 0;  
 *for* (*var* p :  
 students) {  
 *if* (p.isUnsatisfactory()) {  
 System.*out*.println(p.toString());  
 count++;  
 }  
 }  
 System.*out*.println("Відсоток студентів, що мають незадовільні оцінки = " + 100\*count/students.size() + "%");  
 System.*out*.println("\n====================================");  
}

**Результат :**



1. **Код :**

**Батьківський клас :**

*public abstract class* Poet *implements Serializable* {  
 String lastname; *//прізвище* String language; *//мова  
 int* count\_collections; *//кількість збірок  
  
 public* Poet() {  
 lastname = "";  
 language = "";  
 count\_collections = 0;  
 }  
 *public* Poet(String lastname, String language, *int* count\_collections) {  
 *this*.lastname = lastname;  
 *this*.language = language;  
 *this*.count\_collections = count\_collections;  
 }  
  
  
 *public abstract void* input();  
  
 *public abstract void* output();  
  
  
 @Override  
 *public* String toString() {  
 *return* "Poet{" +  
 "lastname='" + lastname + '\'' +  
 ", language='" + language + '\'' +  
 ", count\_collections=" + count\_collections +  
 '}';  
 }  
  
 *//гетери  
 public* String getLastname() { *return* lastname; }  
 *public* String getLanguage() { *return* language; }  
 *public int* getCount\_collections() { *return* count\_collections; }  
  
 *//сетери  
 public void* setLastname(String lastname) { *this*.lastname = lastname; }  
 *public void* setLanguage(String language) { *this*.language = language; }  
 *public void* setCount\_collections(*int* count\_collections) { *this*.count\_collections = count\_collections; }  
}

**Похідний клас :**

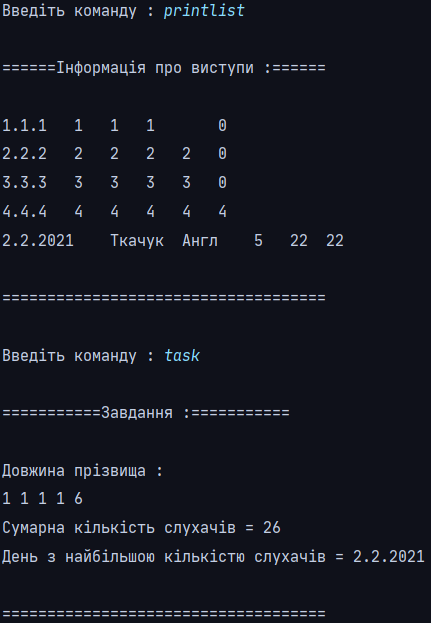
*final public class* Speech *extends* Poet *implements Serializable* {  
 DataSpeech date; *//дата виступу* String place; *//місце виступу  
 int* count\_listeners; *//кількість слухачів  
  
 public* Speech() {  
 date = *new* DataSpeech(0, 0 , 0);  
 place = "";  
 count\_listeners = 0;  
 }  
  
 *public* Speech(DataSpeech date, String place, *int* count\_listeners) {  
 *this*.date = date;  
 *this*.place = place;  
 *this*.count\_listeners = count\_listeners;  
 }  
  
 *public* Speech(String lastname, String language, *int* count\_collections, DataSpeech date, String place, *int* count\_listeners) {  
 *super*(lastname, language, count\_collections);  
 *this*.date = date;  
 *this*.place = place;  
 *this*.count\_listeners = count\_listeners;  
 }  
  
  
 @Override  
 *public void* input() {  
 Scanner scanner = *new* Scanner(System.*in*);  
 System.*out*.print("Прізвище : ");  
 lastname = scanner.nextLine();  
 System.*out*.print("Мова : ");  
 language = scanner.nextLine();  
 System.*out*.print("Кількість збірок : ");  
 count\_collections = scanner.nextInt();  
 System.*out*.println("Дата :");  
 date.enterDate();  
 System.*out*.print("Місце виступу : ");  
 scanner.nextLine();  
 place = scanner.nextLine();  
 System.*out*.print("Кількість слухачів : ");  
 count\_listeners = scanner.nextInt();  
 }  
  
 @Override  
 *public void* output() {  
 System.*out*.println("Прізвище - " + lastname);  
 System.*out*.println("Мова - " + language);  
 System.*out*.println("Кількість збірок - " + count\_collections);  
 System.*out*.println("Дата - " + date.toString());  
 System.*out*.println("Місце виступу - " + place);

System.*out*.println("Кількість слухачів - " + count\_listeners);  
 }  
  
  
 @Override  
 *public* String toString() {  
 *return* date.toString() + '\t' +  
 lastname + '\t' +  
 language + '\t' +  
 count\_collections + '\t' +  
 place + '\t' +  
 count\_listeners;  
 }  
  
 *//гетери  
 public* DataSpeech getDate() { *return* date; }  
 *public* String getPlace() { *return* place; }  
 *public int* getCount\_listeners() { *return* count\_listeners; }  
  
 *//сетери  
 public void* setDate(DataSpeech date) { *this*.date = date; }  
 *public void* setPlace(String place) { *this*.place = place; }  
 *public void* setCount\_listeners(*int* count\_listeners) { *this*.count\_listeners = count\_listeners; }  
  
 *public static boolean* compareTo(Speech a, Speech b) {  
 *if* (a.getDate().years > b.getDate().years ||  
 (a.getDate().years == b.getDate().years && a.getDate().month > b.getDate().month) ||  
 (a.getDate().years == b.getDate().years && a.getDate().month == b.getDate().month && a.getDate().day > b.getDate().day))  
 *return true*;  
 *return false*;  
 }  
}

**Завдання :**

*public void* task() *throws* IOException {  
 count\_note();  
  
 System.*out*.println("\n===========Завдання :===========\n");  
 *int* suma = 0, index = 0, max = speeches.get(index).count\_listeners;  
 *int* i = 0;  
 System.*out*.println("Довжина прізвища :");  
 *for* (*var* p :  
 speeches) {  
 System.*out*.print(p.getLastname().length() + " ");  
 *if*(p.getCount\_listeners() > max){  
 max = p.getCount\_listeners();  
 index = i;  
 }  
 suma += p.getCount\_listeners();  
 i++;  
 }  
 System.*out*.println();  
  
 System.*out*.println("Сумарна кількість слухачів = " + suma);  
 System.*out*.println("День з найбільшою кількістю слухачів = " + speeches.get(index).getDate().toString());  
 System.*out*.println("\n====================================");  
}

**Результат :**



**Висновок:** я вивчив особливості роботи з інтерфейсами у Java.