Харківський національний університет імені В. Н. Каразіна Факультет комп'ютерних наук

Кафедра штучного інтелекту та програмного забезпечення

ЗВІТ

Практична робота №1

дисципліна: «Бази даних»

Виконав: студент групи КС-22

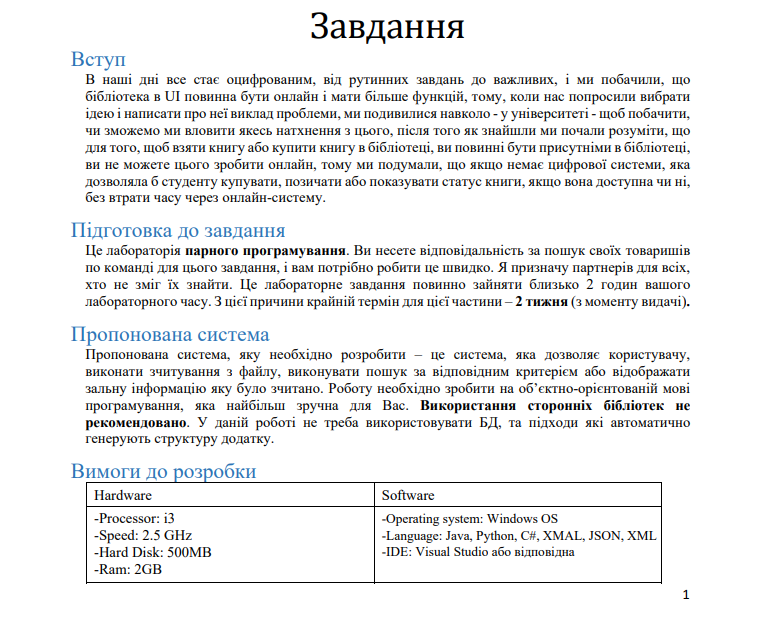
Ковальов Андрій

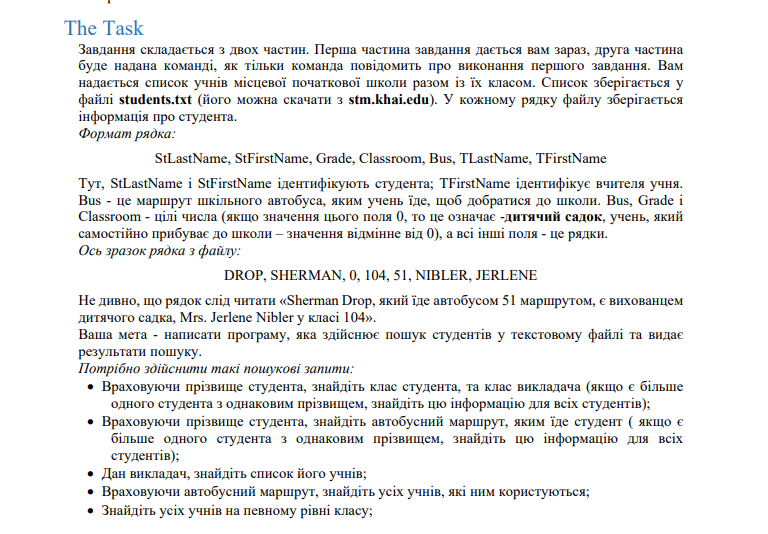
Перевірив: викладач

Бережний Артем

Харків

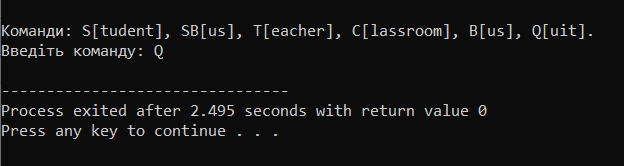
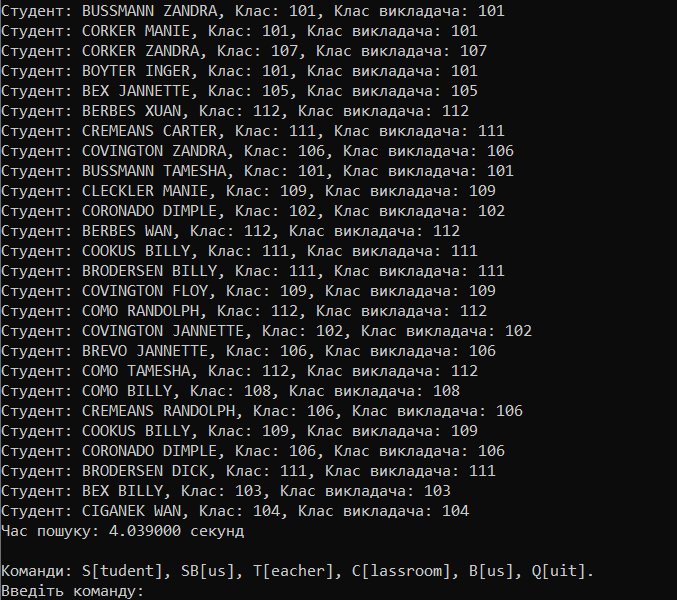
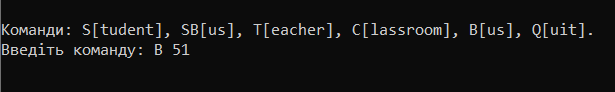
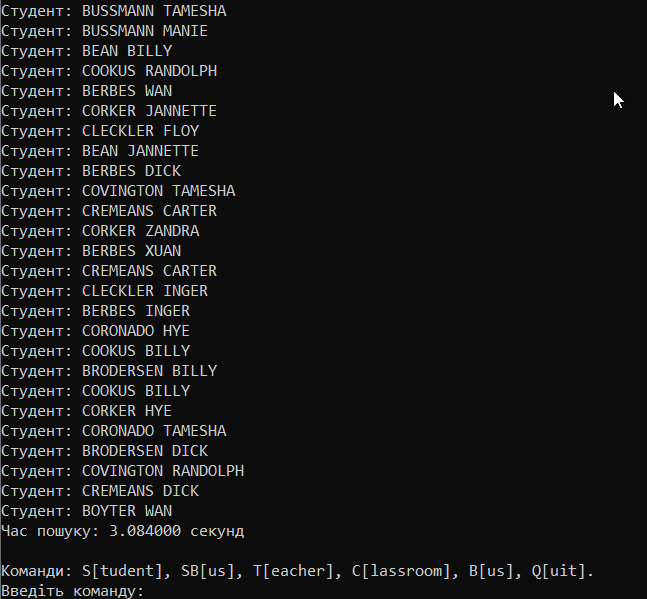
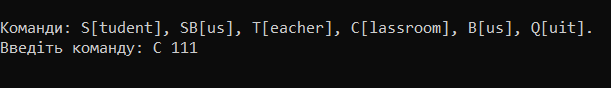
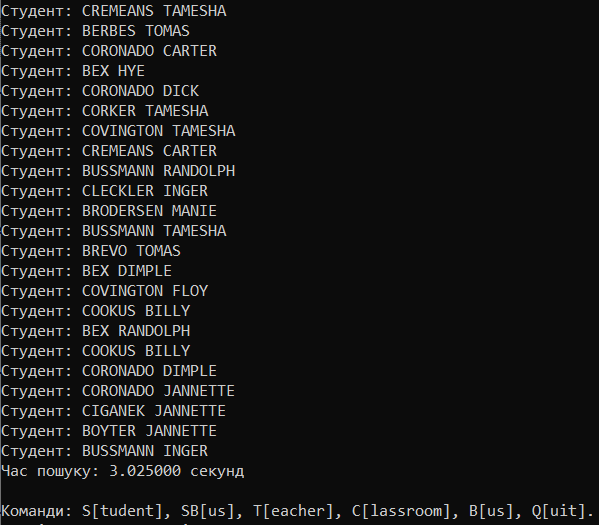
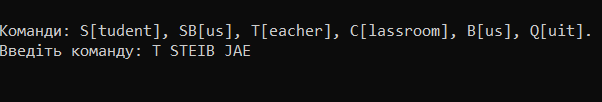
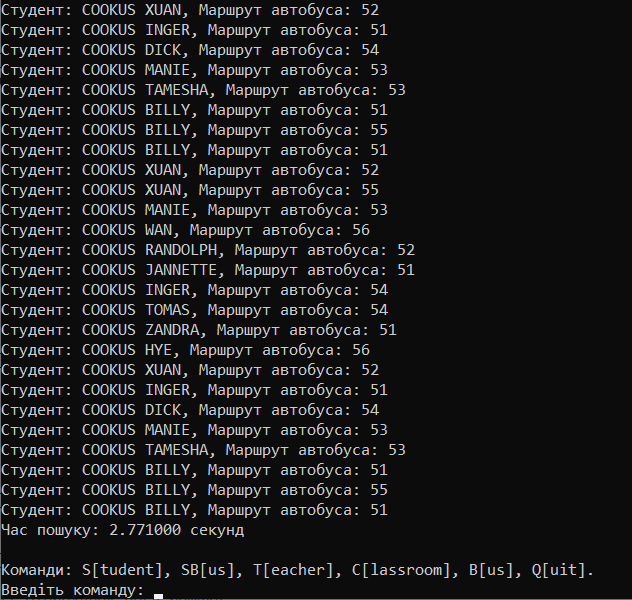
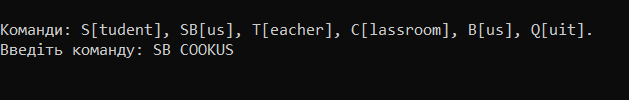
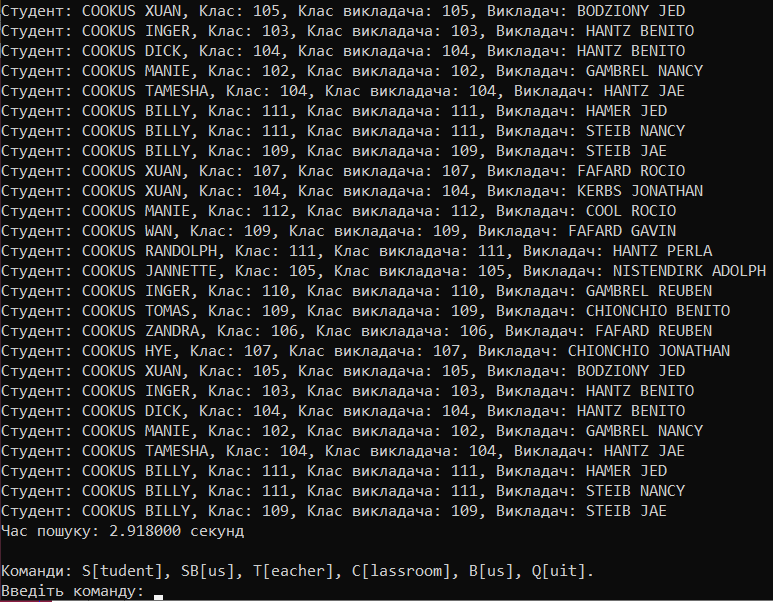
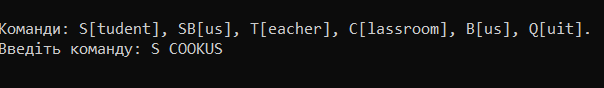
**Мета роботи**: Створити систему, яка дозволяє користувачу, виконати зчитування з файлу, виконувати пошук за відповідним критерієм або відображати зальну інформацію яку було зчитано.





|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <time.h>  #define MAX\_LINE\_LEN 100  #define MAX\_STUDENTS 10000000  typedef struct {  char lastName[30];  char firstName[30];  int grade;  int classroom;  int bus;  char teacherLastName[30];  char teacherFirstName[30];  } Student;  Student students[MAX\_STUDENTS];  int studentCount = 0;  void loadStudents(const char \*filename) {  FILE \*file = fopen(filename, "r");  if (file == NULL) {  printf("Не вдалося відкрити файл: %s\n", filename);  exit(1);  }    while (studentCount < MAX\_STUDENTS && fscanf(file, "%[^,], %[^,], %d, %d, %d, %[^,], %[^\n]\n",  students[studentCount].lastName,  students[studentCount].firstName,  &students[studentCount].grade,  &students[studentCount].classroom,  &students[studentCount].bus,  students[studentCount].teacherLastName,  students[studentCount].teacherFirstName) == 7) {  studentCount++;  }    fclose(file);  }  void searchStudentByLastName(char \*lastName) {  int found = 0;  for (int i = 0; i < studentCount; i++) {  if (strcmp(students[i].lastName, lastName) == 0) {  printf("Студент: %s %s, Клас: %d, Клас викладача: %d, Викладач: %s %s\n",  students[i].lastName, students[i].firstName,  students[i].classroom, students[i].classroom,  students[i].teacherLastName, students[i].teacherFirstName);  found = 1;  }  }  if (!found) {  printf("Студент з прізвищем %s не знайдений.\n", lastName);  }  }  void searchStudentBus(char \*lastName) {  int found = 0;  for (int i = 0; i < studentCount; i++) {  if (strcmp(students[i].lastName, lastName) == 0) {  printf("Студент: %s %s, Маршрут автобуса: %d\n",  students[i].lastName, students[i].firstName, students[i].bus);  found = 1;  }  }  if (!found) {  printf("Студент з прізвищем %s не знайдений.\n", lastName);  }  }  void searchTeacherStudents(char \*lastName) {  int found = 0;  for (int i = 0; i < studentCount; i++) {  if (strcmp(students[i].teacherLastName, lastName) == 0) {  printf("Студент: %s %s\n", students[i].lastName, students[i].firstName);  found = 1;  }  }  if (!found) {  printf("Учнів у викладача з прізвищем %s не знайдено.\n", lastName);  }  }  void searchByClassroom(int classroom) {  int found = 0;  for (int i = 0; i < studentCount; i++) {  if (students[i].classroom == classroom) {  printf("Студент: %s %s\n", students[i].lastName, students[i].firstName);  found = 1;  }  }  if (!found) {  printf("Студенти у класі номер %d не знайдені.\n", classroom);  }  }  void searchByBusRoute(int bus) {  int found = 0;  for (int i = 0; i < studentCount; i++) {  if (students[i].bus == bus) {  printf("Студент: %s %s, Клас: %d, Клас викладача: %d\n",  students[i].lastName, students[i].firstName, students[i].classroom, students[i].classroom);  found = 1;  }  }  if (!found) {  printf("Студенти на маршруті автобуса номер %d не знайдені.\n", bus);  }  }  int main() {  loadStudents("students.txt");  char command[20];  clock\_t start, end;  double cpu\_time\_used;  while (1) {  printf("\nКоманди: S[tudent], SB[us], T[eacher], C[lassroom], B[us], Q[uit]. ");  printf("\nВведіть команду: ");  scanf("%s", command);  if (strcmp(command, "Q") == 0) {  break;  } else if (strncmp(command, "S", 7) == 0) {  char lastName[30];  scanf("%s", lastName);    start = clock();  searchStudentByLastName(lastName);  end = clock();  } else if (strncmp(command,"SB",8) == 0){  char lastName[30];  scanf("%s", lastName);  start = clock();  searchStudentBus(lastName);  end = clock();  }  else if (strncmp(command, "T", 7) == 0) {  char lastName[30];  scanf("%s", lastName);    start = clock();  searchTeacherStudents(lastName);  end = clock();  } else if (strncmp(command, "C", 9) == 0) {  int classroom;  scanf("%d", &classroom);    start = clock();  searchByClassroom(classroom);  end = clock();  } else if (strncmp(command, "B", 3) == 0) {  int bus;  scanf("%d", &bus);    start = clock();  searchByBusRoute(bus);  end = clock();  } else {  printf("Невідома команда.\n");  continue;  }  // Підрахунок часу пошуку  cpu\_time\_used = ((double) (end - start)) / CLOCKS\_PER\_SEC;  printf("Час пошуку: %.6f секунд\n", cpu\_time\_used);  }  return 0;  } |

Лiстинг 1 – код програми



Скрiншот 1-9 – виконання програми

**Пояснення:**

Програма виконує пошук даних про студентів, вчителів, класи та автобуси, які знаходяться в текстовому файлі students.txt. Користувач може вводити команди в консоль для пошуку інформації. Програма продовжує працювати в інтерактивному режимі, доки користувач не введе команду виходу.

### ****Команди****

Програма підтримує кілька типів команд, кожна з яких виконує різний пошук:

* **S[tudent] <lastname>** – Пошук студента за прізвищем. Програма виведе клас і вчителя студента.
* **S[tudent]B[us] <lastname>** – Пошук студента за прізвищем і маршрут шкільного автобуса.
* **T[eacher] <lastname>** – Пошук студентів певного вчителя за прізвищем вчителя.
* **C[lassroom] <number>** – Пошук студентів у конкретному класі за номером класу.
* **B[us] <number>** – Пошук студентів, які користуються певним автобусним маршрутом.
* **Q[uit]** або **Q** – Вихід з програми.

### Підсумок:

* Програма дає можливість користувачу зручно вводити та шукати студентів, вчителів та іншої інформації, обробляючи задані команди.
* Програма працює доки користувач не вийде з програми.

**Висновок:**

В цій практичній роботі я отримав навички роботи з файлами, обробки текстових даних та використання структур для зберігання інформації про студентів. Я також використав базові алгоритми пошуку в масивах даних. За допомогою цього досвіду я зрозумів що являє собою даний вид роботи

**Посилання на GitHub:** **https://github.com/yunkaa-k/bd1**