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

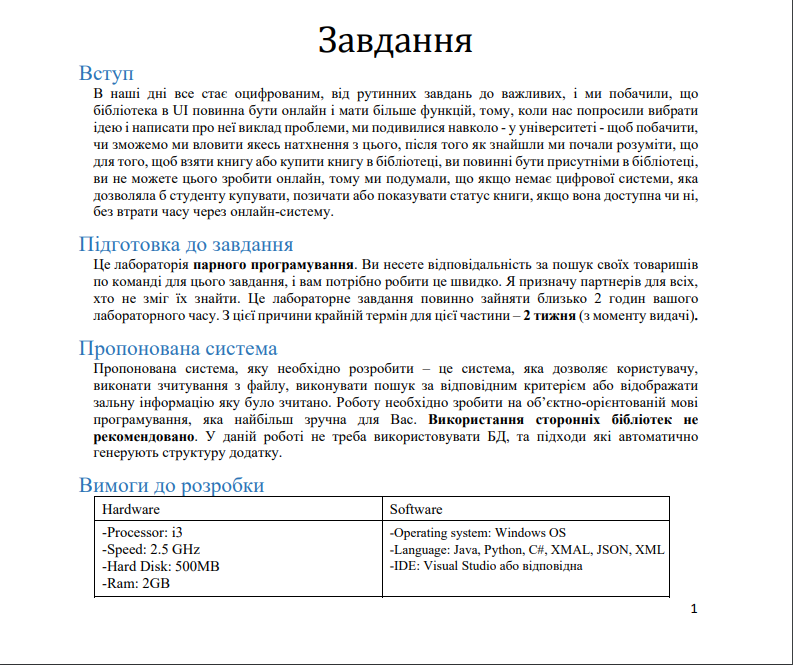
Малютiн Максим

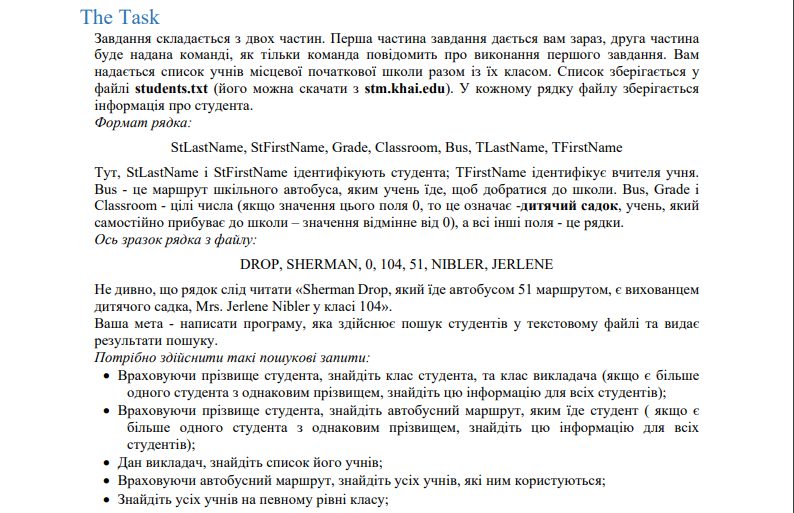
Перевірив: викладач

Бережний Артем

Харків

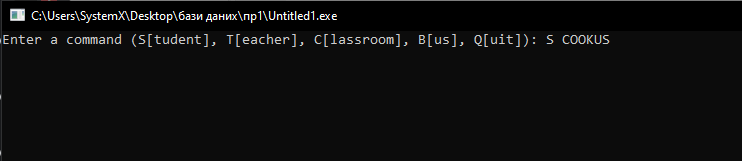
**Мета роботи**: Розробити систему, яка дозволяє користувачу, виконати зчитування з файлу, виконувати пошук за відповідним критерієм або відображати зальну інформацію яку було зчитано.

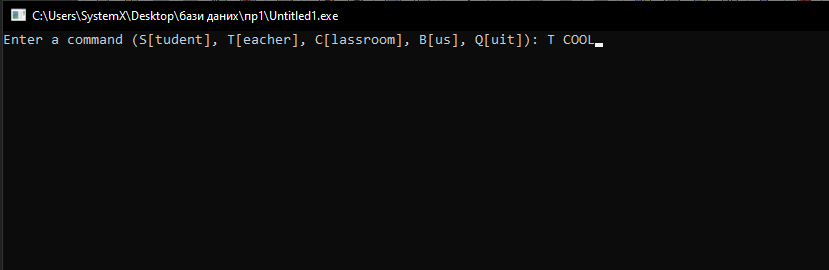
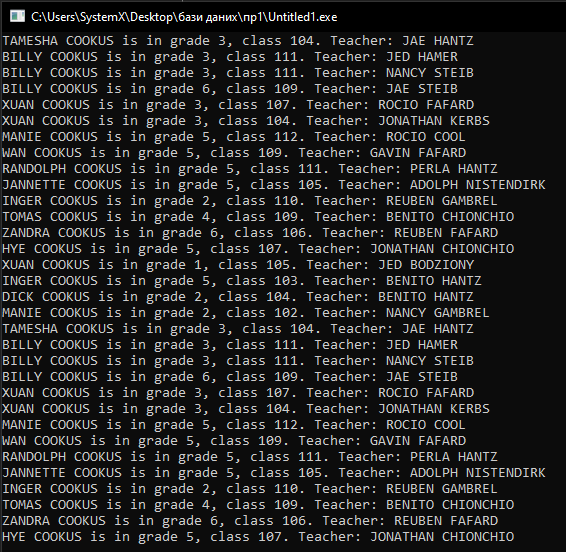


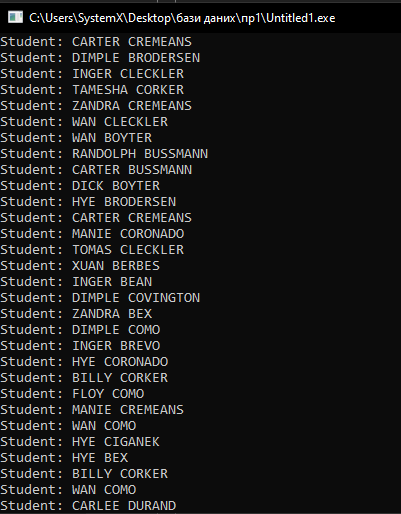


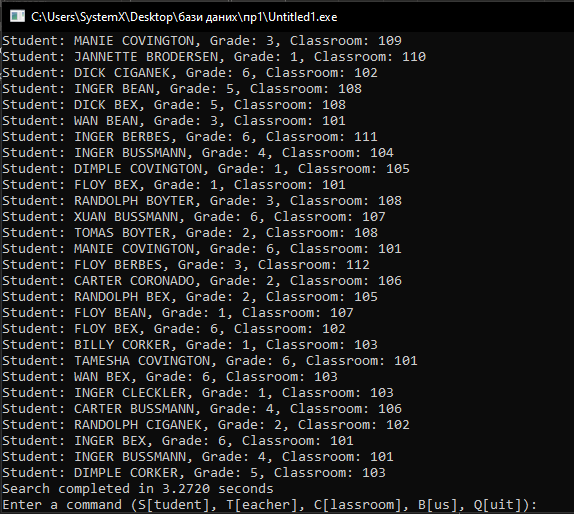
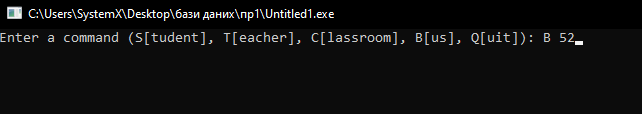
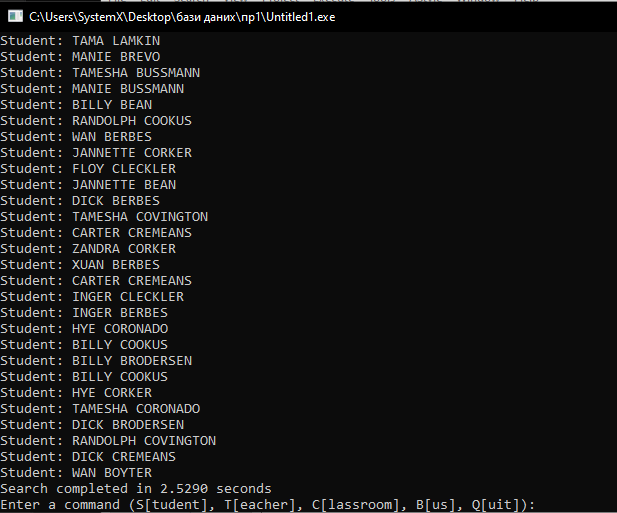
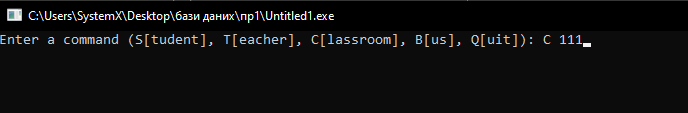
|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <time.h>  #define MAX\_STUDENTS 1000000000  #define MAX\_LINE\_LEN 256  typedef struct {  char lastName[50];  char firstName[50];  int grade;  int classroom;  int bus;  char teacherLastName[50];  char teacherFirstName[50];  } Student;  Student students[MAX\_STUDENTS];  int studentCount = 0;  // Функція для завантаження студентів з файлу  void loadStudents(const char \*filename) {  FILE \*file = fopen(filename, "r");  if (!file) {  perror("Could not open file");  exit(EXIT\_FAILURE);  }  char line[MAX\_LINE\_LEN];  while (fgets(line, sizeof(line), file)) {  Student s;  sscanf(line, "%[^,], %[^,], %d, %d, %d, %[^,], %[^\n]",  s.lastName, s.firstName, &s.grade, &s.classroom, &s.bus, s.teacherLastName, s.teacherFirstName);  students[studentCount++] = s;  }  fclose(file);  }  // Пошук за прізвищем студента (клас і викладач)  void searchStudentByLastName(const char \*lastName) {  for (int i = 0; i < studentCount; i++) {  if (strcmp(students[i].lastName, lastName) == 0) {  printf("%s %s is in grade %d, class %d. Teacher: %s %s\n",  students[i].firstName, students[i].lastName, students[i].grade, students[i].classroom,  students[i].teacherFirstName, students[i].teacherLastName);  }  }  }  // Пошук за прізвищем студента (автобус)  void searchStudentBusByLastName(const char \*lastName) {  for (int i = 0; i < studentCount; i++) {  if (strcmp(students[i].lastName, lastName) == 0) {  printf("%s %s rides bus number %d\n", students[i].firstName, students[i].lastName, students[i].bus);  }  }  }  // Пошук за прізвищем викладача  void searchStudentsByTeacher(const char \*teacherLastName) {  for (int i = 0; i < studentCount; i++) {  if (strcmp(students[i].teacherLastName, teacherLastName) == 0) {  printf("Student: %s %s\n", students[i].firstName, students[i].lastName);  }  }  }  // Пошук за номером класу  void searchStudentsByClassroom(int classroom) {  for (int i = 0; i < studentCount; i++) {  if (students[i].classroom == classroom) {  printf("Student: %s %s\n", students[i].firstName, students[i].lastName);  }  }  }  // Пошук за номером автобусу  void searchStudentsByBus(int busNumber) {  for (int i = 0; i < studentCount; i++) {  if (students[i].bus == busNumber) {  printf("Student: %s %s, Grade: %d, Classroom: %d\n",  students[i].firstName, students[i].lastName, students[i].grade, students[i].classroom);  }  }  }  // Головна функція  int main() {  loadStudents("students.txt");  char command[50];  while (1) {  printf("Enter a command (S[tudent], T[eacher], C[lassroom], B[us], Q[uit]): ");  fgets(command, sizeof(command), stdin);  command[strcspn(command, "\n")] = 0; // Видалення нової лінії  clock\_t start\_time = clock(); // Початок вимірювання часу  if (strncmp(command, "S ", 2) == 0) {  char lastName[50];  sscanf(command + 2, "%s", lastName);  searchStudentByLastName(lastName);  } else if (strncmp(command, "S", 1) == 0 && strstr(command, " B") != NULL) {  char lastName[50];  sscanf(command + 2, "%s", lastName);  searchStudentBusByLastName(lastName);  } else if (strncmp(command, "T ", 2) == 0) {  char teacherLastName[50];  sscanf(command + 2, "%s", teacherLastName);  searchStudentsByTeacher(teacherLastName);  } else if (strncmp(command, "C ", 2) == 0) {  int classroom;  sscanf(command + 2, "%d", &classroom);  searchStudentsByClassroom(classroom);  } else if (strncmp(command, "B ", 2) == 0) {  int busNumber;  sscanf(command + 2, "%d", &busNumber);  searchStudentsByBus(busNumber);  } else if (strcmp(command, "Q") == 0) {  break;  } else {  printf("Unknown command. Please try again.\n");  }  clock\_t end\_time = clock(); // Кінець вимірювання часу  double search\_time = (double)(end\_time - start\_time) / CLOCKS\_PER\_SEC;  printf("Search completed in %.4f seconds\n", search\_time);  }  return 0;  } |

Лiстинг 1 – код програми









Скрiншот 1-8 – 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 дозволяє користувачу виконувати пошук студентів, вчителів та іншої інформації, пов'язаної з навчальним процесом.
* Користувач вводить команди в консоль, а програма обробляє ці команди, виконує відповідні пошуки і виводить результати.
* Програма продовжує працювати в циклі, доки користувач не введе команду виходу.

**Висновок:**

У цій практичній роботі я навчився працювати з файлами, обробляти текстові дані та використовувати структури для зберігання інформації про студентів. Я також освоїв базові алгоритми пошуку в масивах даних та створив простий командний інтерфейс для взаємодії з користувачем. Ця робота допомогла мені краще зрозуміти, як організовувати програму та забезпечувати її продуктивність.