**Практична робота 2**

**Мета роботи**

Ознайомитись зі створенням та застосуванням структур та отримати практичні навички роботи з їх полями.

**Хід роботи**

1) Ознайомитись з методичними вказівками до лабораторної роботи та темою "Складені типи даних";

2) сформувати структуру згідно з варіантом завдання;

3) розробити схему алгоритму програми, що демонструє роботу зі створеною структурою;

4) за схемою алгоритму написати програму на мові С++;

5) зробити висновки;

6) підготувати звіт про виконання лабораторної роботи, що включає наступні пункти: номер, тема, мета та хід лабораторної роботи, завдання, схема алгоритму програми, лістинг програми з коментуванням кожної інструкції, результат роботи програми (скріншот), висновки.

**Теоретичні відомості та рекомендації до виконання**

Як і масиви, структури дозволяють зберігати не одне значення, а декілька, проте на відміну від масивів у структурі для кожної змінної зазначено власний тип даних та унікальне ім'я. Таким чином, структури є групою змінних різного типа даних, що поєднані під одним ім'ям.

Як і масиви структури слугують для зберігання даних.

При виконанні завдання до лабораторної роботи студент повинен ознайомитися з особливостями створення структур та навчитись застосовувати їх в програмах на мові С++.

Виконання лабораторної роботи починається ознайомленням з теоретичними відомостями про оголошення та визначення структур і роботу з їх полями. Потім студент приступає до розробки схеми алгоритму програми згідно із завданням за варіантом. За схемою алгоритму виконується написання програми на мові С++.

У висновках до звіту студент повинен розкрити призначення теми "Створення та застосування структур". Висновок має бути поданий в формі пояснення сфери застосування вивченого матеріалу згідно із розумінням його студентом.

**Завдання для самостійного виконання**

Розробити схему алгоритму та написати програму, яка створює на основі структури за варіантом таблицю (табл. 9.1) та дозволяє заповнити її поля значеннями, що вводяться з консолі. За запитом користувача програма повинна виводити відповідний рядок або всю отриману таблицю.

**Варіанти завдань**

**Таблиця 9.1.**

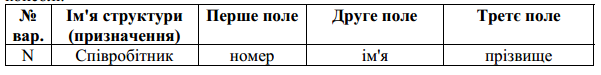


**Продовження таблиці 9.1**.



**Приклад виконання завдання**

Розробити схему алгоритму та написати програму, яка створює на основі структури таблицю та дозволяє заповнити її поля значеннями, що вводяться з консолі.



Розробимо схему алгоритму роботи програми створення та заповнення таблиці на основі структури (рис. 9.1).

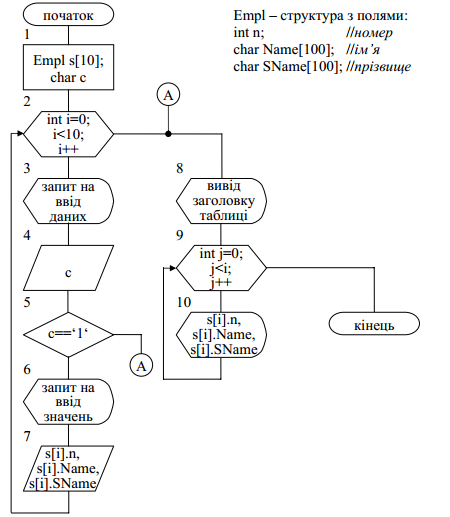


Рис. 9.1. Схема алгоритму програми роботи з таблицею

За схемою алгоритму напишемо програму на мові С++:

#include<stdio.h> //підключення бібліотеки вводу/виводу

struct Empl //оголошення та визначення структури співробітників

{int n; //цілочисельне поле номера

char Name[100]; //поле символьний масив для імені

char SName[100]; //поле символьний масив для прізвища

};

void main() //оголошення та визначення головної функції

{ Empl s[10]; //оголошення масиву екземплярів структури

char c; //оголошення змінної зупинки заповнення таблиці

printf("You start table filling!\r\n"); //вивід запиту до користувача

for(int i=0; i<10; i++) //цикл заповнення таблиці

{ //вивід запиту про закінчення заповнення

printf("Continue? 1 – No\r\n");

scanf("%s", &c); //отримання значення від користувача

if(c=='1') //якщо введено '1'

break; //вихід з оператора

//вивід запиту на ввід значення поля номера

printf("Enter first field value (Number): ");

scanf("%i", &s[i].n); //отримання значення від користувача

//вивід запиту на ввід значення поля імені

printf("Enter second field value (Name): ");

scanf("%s", s[i].Name); //отримання значення від користувача

//вивід запиту на ввід значення поля прізвища

printf("Enter third field value (Second Name): ");

scanf("%s", s[i].SName); //отримання значення від користувача

}

//вивід заповненої таблиці

printf("Table:\r\n"); //вивід константного рядка

printf("N\t\tName\t\tSName\r\n"); //вивід константного рядка

for (int j=0; j<i; j++) //цикл виводу полів таблиці

//вивід значень рядка таблиці

printf("%i\t\t%s\t\t%s\r\n", s[j].n, s[j].Name, s[j].SName);

}

Результат виконання програмного коду буде наступним:

