**Лабораторна робота №9. Написання програм з використанням рядкових величин та структур**

**На період** карантину в дистанційній формі навчання потрібно використовувати замість **С++ Builder онлайн компілятор C++ Shell, який доступний за адресом** [**http://cpp.sh**](http://cpp.sh) (інструкція надана після тексту ЛР№3). При наявності на домашньому комп’ютері іншого завдання можна виконати на ньому. Результати надсилати на електронну адресу викладача [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)у вигляді URL-посилання для робіт, виконаних у **C++ Shell,** або для робіт, на компіляторі С++ домашнього комп’ютера cpp-файл з іменем у форматі

**АР<Номер групи><Номер практичної / лабораторної [літера позначення типу роботи P – практична, R – лабораторна]<Прізвище англійською>**. Наприклад, **АРPTBD-2107P**buts.срр.

При відсутності можливості використовувати компілятор або відсутності можливості доступу до Інтернету текст програми набрати в Блокноті або WordPad Windows, в крайньому випадку текст програми можна записати у зошити, сфотографувати, оформити відповідним чином (включити титульний лист, записати завдання, а потім включити файли з фотографіями) та надіслати на електронну адресу викладача.

Оформлені роботи повинні надсилатися у вигляді текстового документу з іменем файлу:

**АР<Номер групи><Номер лекції / практичної / лабораторної [літера позначення типу роботи L – лекція, P – практична, R – лабораторна]<Прізвище англійською>**. Наприклад, **АРPTBD-2104P**buts.doc.

**Тему в заголовку листа записати**

**АР <Номер групи>-><Номер лекції / практичної / лабораторної [літера позначення типу роботи L – лекція, P – практична, R – лабораторна]<Прізвище англійською>**

**Строк відсилки ЛР 10.04.2021.**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача, тему в заголовку листа записати

**АР<Номер групи>-Запитання-<Прізвище англійською>**.

**Мета**: навчитися писати програми з використанням рядкових величин та структур **на мові С++** в **консольному режимі.**

**Методичні вказівки щодо організації самостійної роботи студентів**

1. Прочитати теоретичні відомості, повторити Лекцію 9, відповісти на контрольні запитання для самоперевірки.
2. Запустити середовище програмування С++ .
3. Записати програму, що виконує завдання з пп.4, яке складається з 3-х частин. В першому рядку програми записати

*// ПТБД-21 Група № ЛР№ Прізвище*

вказавши номер своєї групи, номер ЛР та своє прізвище.

Вхідні дані ввести, а результати вивести, використовуючи потокове введення-виведення даних.

1. **Завдання**:

Написати програму, в яка містить структуру, що описує об'єкт будинок і складається зі змінних, в яких знаходиться інформація про місто (в якому будинок знаходиться), вулиця, номер будинку, кількість квартир в будинку, наявність інтернету в будинку (проведено чи ні, задати логічною/булевою змінною).

1. Сформувати масив структур (не менше 3 елементів), в якому номер будинку та кількість квартир заповнюється випадковими числами з діапазону від 1 до 20 та від 100 до 200, відповідно (див. ЛР№8), іншим елементам присвоюються значення в програмі.
2. В програмі запитати кількість об’єктів, про які потрібно вивести інформацію, та вивести перші із запитаної кількості елементи масиву структур з вказівкою індексу масиву, а після нього вся інформація про об’єкт. Значення, що виводяться по об’єкту, розділяються їх табуляцією.
3. Після цього підрахувати кількість будинків без інтернету та вивести їх кількість, після цього вивести заголовок "Будівлі з інтернетом" і вивести порядковий номер в списку, місто, вулиця, номер будинку.

**Контрольні запитання.**

1. Які оператори введення-виведення символьних масивів ви знаєте? В яких випадках вони використовуються?
2. Як визначається кінець рядка?

**2.** Як описуються дані типу структура?

**3.** Які типи полів може містити структура?

**4.** Які існують способи ініціювання полів структури?

**5.** Що таке «вкладеність» структур?

**6.** Як здійснюється робота з типом struct з використанням покажчиків?

**7.** Як реалізується доступ до елементів масиву типу структура?

**Теоретичні відомості.**

Структура – це об'єднання різних змінних (навіть з різними типами даних), якому можна присвоїти ім'я. Наприклад можна об'єднати дані про об'єкт Будинок: місто (в якому будинок знаходиться), вулиця, кількість квартир, інтернет(проведено чи ні) і т.д. в одній структурі. Синтаксис структури розглядався в лекції 9.

Оголошення структурної змінної здійснюється в 2 етапи:

* оголошення шаблону структури як нового типу даних. На цьому етапі пам‘ять не виділяється. Формується тільки інформація про вміст структури;
* оголошення самої змінної. На цьому етапі виділяється пам‘ять для кожного поля (змінної), що описується в шаблоні структури.

**Приклад.** Нехай задано шаблон структури, що описує книгу в бібліотеці.

**struct Book**

**{**

**title[70];**

**char author[50];**

**int year;**

**float price;**

**};**

Після такого опису пам‘ять не виділяється.

Якщо описати змінну тип Book, тоді для такої змінної виділиться пам’ять.

Book B; // під змінну B виділяється 128 байт

Розмір пам‘яті, що виділяється для змінної B становитиме: 70 + 50 + 4 + 4 = 128 байт. Для поля title виділиться 70 байт (тип char займає 1 байт). Для поля author виділиться 50 байт. Для поля year (в Win32) виділиться 4 байти (тип int). Для поля price виділиться 4 байти (тип float).

В залежності від розрядності операційної системи та конфігурації обладнання ця цифра може бути іншою.

**Приклад 1. Використання структури.**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main()**

**{**

**struct Book**

**{**

**string title;**

**string author;**

**int year;**

**float price;**

**};**

**Book PrgBook;**

**PrgBook.title="Вступ до програмування мовою С++";**

**PrgBook.author="Ю. А. Бєлов";**

**PrgBook.year=2012;**

**PrgBook.price=45.50;**

**cout << "Перелік книжок"<<endl;**

**cout << PrgBook.title << '\t' << PrgBook.author << '\t' << PrgBook.year << '\t' << PrgBook.price <<endl;**

**return 0;**

**}**

Шаблон будь-якої структури може містити в собі інші структури. Якщо в структурі описується інша структурна-змінна, тоді для цієї змінної пам‘ять виділяється згідно правил виділення пам‘яті для структурної змінної (див. вище).

**Приклад 2. Оголошення та використання вкладеної структури**

Нехай задано два шаблони структур: шаблон Point, що описує точку на координатній площині, та шаблон Triangle, що описує трикутник на площині:

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main()**

**{**

**// Шаблон структури, що описує точку на площині**

**struct Point**

**{**

**int x;**

**int y;**

**};**

**// Шаблон структури, що описує фігуру трикутник**

**struct Triangle**

**{**

**Point p1; // вкладена структура**

**Point p2; // вкладена структура**

**Point p3; // вкладена структура**

**string comment; // коментарій до фігури**

**};**

**Triangle T; // для змінної T виділяється пам'ять**

**// заповнення значень змінної T**

**T.p1.x = 25;**

**T.p1.y = 36;**

**T.p2.x = 100;**

**T.p2.y = 55;**

**T.p3.x = 60;**

**T.p3.y = 56;**

**T.comment = "Triangle #1";**

**cout << T.comment << " Formed"<<endl;**

**return 0;**

**}**

Як бачимо у структурі можна оголошувати іншу структуру.

**Приклад 3. Оголошення та використання масиву** **структур.**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main()**

**{**

**// Шаблон структури, що описує точку на площині**

**struct Point**

**{**

**int x;**

**int y;**

**};**

**// Масив структур - оголошення масиву з іменем M**

**Point M[10]; // оголошується 10 структур типу Point, для яких виділяється пам'ять**

**for (int i=0; i<10; i++)**

**{**

**M[i].x = i\*(i-2);**

**M[i].y = i\*3;**

**cout << i <<"\tx=" << M[i].x <<"\ty=" << M[i].y <<endl;**

**}**

**return 0;**

**}**

##### Приклад 4. Приклад оголошення та використання двовимірного масиву структур

Нехай задано структуру, що описує точку на площині. Тоді робота з двовимірним масивом таких структур розміром 5\*10 буде такою.

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main()**

**{**

**// Шаблон структури, що описує точку на площині**

**struct Point**

**{**

**int x;**

**int y;**

**int color;**

**};**

**// Оголошення масиву з іменем A розміром 5\*10**

**Point A[5][10];**

**// заповнення значеннями**

**for (int i=0; i<5; i++)**

**for (int j=0; j<10; j++)**

**{**

**A[i][j].x = i\*2 + j\*j;**

**A[i][j].y = i\*i + j;**

**A[i][j].color = i+j;**

**cout << i << '\t' << j << "\tx=" <<A[i][j].x <<"\ty=" << A[i][j].y<<"\tcolor=" <<A[i][j].color <<endl;**

**};**

**return 0;**

**}**

**Вкладені структури**.

Студентські списки містять дані: прізвище та ініціали, дата (день, місяць, рік) народження, група та середній бал успішності, а в списках викладачів присутні такі дані: прізвище, ініціали, дата народження, кафедра, посада. У процесі обробки списку студентів і списку викладачів можна оголосити відповідно такі структури:

**struct stud**

**{ char fio [25];**

**int den, god;**

**char mes [10];**

**char grup;**

**flout sb; }**

**struct prep  
{ char fio [25];  
int den, god;  
char mes [10];  
char kaf, dolg;  
}**

В оголошених типах однакові поля можна об’єднати в окрему структуру і застосовувати її при описі інших типів. Поетапно це виглядає так:

* загальна структура —

**struct spd**

**{ char fio [25];**

**int den, god;**

**char mas[10]; }**

* структура для опису інформації про студентів —

**struct stud**

**{ spd dr;**

**char grup;**

**float sb}**

**st1, st2;**

* структура для опису інформації про викладачів —

**struct prep**

**{ spd dr;**

**char kaf [10];**

**char dolg [15];**

**} pr1, pr2;**

У структурах **stud і prep** для оголошення поля, що містить дані про прізвище і дату народження, використовується раніше описаний тип **spd**. Тепер до поля **fio, den, god, mes**можна звернутися, використовуючи запис **st1.dr.fio**, наприклад, при зверненні до функції введення:

**gets (st1.dr.fio);** або    **gets (pr1.dr.fio);**

**Пояснення.**

Прототип функції gets: (б-ка cstdio) char \* gets( char \* string );

Отримати рядок зі стандартного потоку введення. Функція gets зчитує символи зі стандартного потоку введення до символу нового рядка n або доки, поки не буде досягнутий кінець файлу EOF, після чого зберігає лічені символи в рядок типу char.

Символ нового рядка n не копіюватиметься в рядок.

Нульовий символ автоматично додається після останнього копійованого символу в string, щоб сигналізувати про кінець рядка.

Функція gets дещо відрізняється від функції fgets, по-перше повертає рядок зі стандартного потоку введення в якості аргументу, а, по-друге, кінцевий символ нового рядка не входить до її складу . на відміну від fgets. А ще, gets не дозволяє встановити обмеження на кількість зчитувальних символів, тому потрібно бути обережними з розміром масиву, на який вказує параметр str, щоб уникнути переповнення буфера.

Параметри: **string**Покажчик на масив типу char, в якому зберігається рядок.

Значення, що повертається. У разі успіху, функція повертає той же параметр, string. Якщо відбувається помилка, повертається нульовий покажчик.

Використовуйте функції ferror або feof для перевірки внутрішніх станів показників помилки або кінця файлу – EOF.

Приклад: вихідний код програми

**#include <iostream>**

**#include <cstdio>**

**int main()**

**{**

**char string [256];**

**std::cout << "Введите свой полный адрес: ";**

**gets(string); // считать строку из стандартного потока ввода**

**std::cout << "Ваш адрес: " << string;**

**return 0;**

**}**

Розглянемо використання структурного типу на прикладах.

***Приклад*** Увести в комп’ютер відомість успішності групи сту­дентів, які здали сесію з дисциплін «Математика», «Фізика» і «Програмування», та обчислити:

* середній бал кожного студента;
* середній бал групи за кожним предметом;
* вивести на екран прізвища відмінників з програмування.

**// обработка ведомости успеваемости группы студентов**

***//-------------- использование данных типа структура***

**#include <iostream.h>**

**#include <string.h>**

**#include <conio.h>**

**void main()**

**{ const n = 5;     *//n — количество студентов***

**int і;**

**float sm, sf, sp;**

***/\* sm, sf.sp — сумма оценок группы соответственно по математике, физике, программированию \*/***

**struct stud**

**{ char fam[25];**

**int mat, fiz, prg;**

**float sb;**

**} ved[n];       //*ved[n] — массив студентов (ведомость)***

**sm = sf = sp = 0;**

**/\*---- *ввод и обработка информации о студентах и их успеваемости\*/***

**for (і = 0; і < n; i++)**

**{ cout <<"\*\*\*\*\* Введите информацию о "<< (i+1) << "студенте\n";**

**{ cout<<<<"Введите фамилию и инициалы\n";**

**gets(ved[i]. fam);**

**cout <<"Оценки по матем., физике и программир.\n";**

**cin >> ved[i].mat >> ved[i].fiz >> ved[i].prg;**

**//---------- *вычисление среднего балла студента за сессию***

**ved[i].sb = (float(ved[i].mat + ved[i].fiz + ved[i].prg))/3;**

**//---------- *суммирование оценок в группе по предметам***

**sm += ved[i].mat;**

**sf += ved[i].fiz;**

**sp += ved[i].prg;**

**}**

**}**

**//------------*вывод результатов вычислений***

**cout << ''\n\*\*\*\*\*\*\* Результаты сессии\n";**

**cout.precision(3);**

**for (і = 0; і < n; i++)**

**cout<<i+1<< " "<< ved[i].fam << " матем. = " << ved[i].mat << " физика = "<< ved[i].fiz << "програм. = " <<ved[i].prg <<**

**" ср. балл = " << ved[i].sb << "\n";**

**cout << "\n\nСредний балл группы по математике = "<< sm/n;**

**cout << "\nСредний балл группы по физике = "<< sf/n;**

**cout << "\nСредний балл группы по программированию = "<<sp/n;**

**cout << "\n\n\*\*\*\*\* Отличники по программированию: \n";**

**for (і = 0; і < n; i++)**

**if (ved[i].prg == 5) cout << ved[i].fam <<"\n";**

**getch ();**

**}**

Результати виконання програми:

**\*\*\*\*\* Введите информацию о 1 студенте**

**Введите фамилию и инициалы**

**Адаменко Р. Л.**

**Оценки по матем., физике и программир.**

**4 4 5**

**\*\*\*\*\* Введите информацию о 2 студенте**

**Введите фамилию и инициалы**

**Бочаренко П. П.**

**Оценки но матем., физике и программир.**

**5 5 4**

**\*\*\*\*\* Введите информацию о 3 студенте**

**Введите фамилию и инициалы**

**Волошко А. А.**

**Оценки но матем., физике и программир.**

**4 4 3**

**\*\*\*\*\* Введите информацию о 4 студенте**

**Введите фамилию и инициалы**

**Долбня В. Е.**

**Оценки по матем., физике и программир.**

**5 5 5**

**\*\*\*\*\* Введите информацию о 5 студенте**

**Введите фамилию и инициалы**

**Ильченко Г. Г.**

**Оценки по матем., физике и программир.**

**3 3 4**

**\*\*\*\*\*\*\* Результаты сессия**

**Адаменко Р. Л. матем. = 4 физика = 4 програм. = 5 ср. балл = 4.33**

**Бочаренко П. П. матем. = 5 физика = 5 програм. = 4 ср. балл = 4.67**

**Волошко А. А. матем. = 4 физика = 4 прогам. = 3 ср. балл = 3.67**

**Долбня В. Е. матем. = 5 физика = 5 програм. =5 ср. балл = 5**

**Ильченко Е. Г. матем. = 3 физика = 3 програм. = 4 ср. балл = 3.33**

**Средний балл группы по математике = 4.2**

**Средний балл группы по физике = 4.2**

**Средний балл группы по программированию = 4.2**

\*\*\*\*\*\*\***Отличники по программированию**:

**Адаменко Р.Л**.

**Долбня В.Е.**

Після оголошення структурного типу змінних для роботи з їхніми полями можна застосовувати і покажчики, тоді опис структури матиме вигляд:

**struct stud**

**{ char fam [25];**

**int mat, fiz, prg;**

**float sb;**

**} st1, \*pst;**

Доступ до полів може здійснюватися двома способами:

* з використанням операції розіменування «\*», тобто  
  **gets ((\*pst).fam);    (\*pst).fiz = 5;**
* **(\*pst).fam="абв";**
* з використанням покажчика **->**, наприклад,  
  **gets (pst -> fam);      pst-> fiz = 5;** тощо.

**a->b це Розкриття посилання на структуру ("вказує на член *b* об'єкту *a*")**

Крім того, до полів змінної **st1** можна звертатися, вказуючи поля через операцію **«.»**, як це робилося раніше.

Дані типу структура можна об’єднати в масиви, для попереднього прикладу з урахуванням кількості студентів, структуру можна записати так:

**struct stud**

**{ char fam [20];**

**int mat, fiz, prg;**

**float sb;**

**} spis[15], \*sp = &spis[0];.**

У випадку, коли масив описується десь у тексті програми, тобто не саме після опису структури, його можна оголосити у вигляді: **stud spis [15];** — масив типу структура з ім’ям **stud**, що містить відповідну інформацію про групу із **15** студентів.

Доступ до елементів масиву типу структура здійснюється із застосуванням індексу або через покажчик-константу, яким є ім’я масиву, тобто одним з таких способів:

**strcpy (spis[1].fam, ” “);**

**spis[1].fiz = 5;**

або

**strcpy ((sp + 1) -> fam, ” “);**

**(sp + 1) -> fiz = 5;**

або

**strcpy ((\*(spis + 1)).fam, ” “);**

**(\*(sp + 1)).fiz = 5;.**

У останньому виразі **(\*(spis + 1)).fiz = 5;** потрібна зовнішня пара дужок, тому що **операція «.» («крапка») має пріоритет вище, ніж операція розіменування «\*».**

Поля структури можуть також бути масивами. Наприклад, у розглянутій структурі **stud** можна оцінки з різних предметів об’єднати в масив. Тоді структуру слід описати у вигляді:

**struct stud1**

**{ char fam [25];**

**int pred [3];**

**float sb**

**} spis[15],** **\*ps = &spis[0];.**

Звернення до полів здійснюватиметься одним із способів:

**((\*ps).fam)**

**(ps->pred [0]),**

наприклад,

**gets ((\*ps).fam);**  
**сіn >> ps -> pred[0] >> ps -> pred[1] >> ps -> pred[2];**

або**сіn >> ps -> \*(pred + 0) >> ps -> \*(pred + 1) >> ps -> \*(pred + 2).**

Доцільно також зазначити, що **бібліотека stdlib.h містить спеціальні функції для пошуку та сортування структурних змінних.**