**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**РОМЕНСЬКИЙ КОЛЕДЖ**

# **ДЕРЖАВНОГО ВИЩОГО НАВЧАЛЬНОГО ЗАКЛАДУ**

**«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**

**імені Вадима ГЕТЬМАНА»**

**ПЛАН**

**відкритого заняття з дисципліни**

**«Системне програмування»**

**спеціальність «Обслуговування комп`ютерних систем і мереж»**

**Тема: *«Поняття структури. Шаблон структури. Структурна змінна»***

|  |  |
| --- | --- |
| *j0292982* | **Склав викладач спецдисциплін**  **Руденко А. П.**  **Розглянуто на засіданні циклової комісії**  **зі спеціальності «Обслуговування комп`ютерних систем і мереж»**  **Протокол №\_\_**  **від «\_\_» «\_\_\_\_\_\_\_» 2016 р.**  **Голова циклової комісії**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г. А. Хрокало** |

**Ромни – 2016 р.**

**НАВЧАЛЬНО-МЕТОДИЧНА КАРТКА (ПЛАН) ЗАНЯТТЯ № 29**

**Предмет** Системне програмування Група К-4-1 Дата\_\_\_\_\_\_\_\_

|  |
| --- |
| Тема заняття: **Поняття структури. Шаблон структури. Структурна змінна** |

Вид заняття (тип уроку): Лекція (з елементами бесіди)

**Мета заняття**

**Навчальна (дидактична):** ознайомити з модульним структурним програмуванням. Вивчити поняття структури. Навчити використовувати у програмуванні шаблони структур, структурних змінних, масивів структур

**Розвивальна:** розвиватианалітичне та логічне мислення**,**  пізнавальні можливості, технічні уміння

**Виховна:** виховувати зацікавленість дисципліною, прагнення отримувати нові знання самостійно. Формувати відповідне ставлення до обраної спеціальності

**Забезпечення заняття**

Дидактичне забезпечення:

* презентація «Системне програмування. Поняття структури. Шаблон структури. Структурна змінна»;
* індивідуальні картки з завданнями;
* Довідник «Перелік стандартних функцій С++».

Технічне забезпечення:

* ПК викладача;
* інтерактивна дошка;
* ЛКМ, мережа Інтернет

Програмне забезпечення:

* ОС Windows;
* ППЗ Microsoft Office;
* Visual C++ 6.0;
* <http://learningapps.org>;
* http://paslab.ru/blok-shema.php

Література:

1. О. І. Щедріна Алгоритмізація та програмування процедур обробки інформації. Навчальний посібник. - Київ: КНЕУ, 2001 – с. 176-181.
2. Т. А. Павловская С/С++. Программирование на языке высокого уровня - Питер, 2007 – с. 223-228.
3. А. Я. Архангельский Программирование в С++ Builder 5 - БИНОМ, 2002.
4. Н. Культин С/С++ в задачах и примерах .- БХВ-Петербург,-2004.- с.41-46.

**План заняття**

**І *Організаційна частина***

1. Перевірка присутності студентів.

**ІІ А*ктуалізація опорних знань***

1. Фронтальне опитуваннястудентів з теми *«Масиви. Використання масивів у функціях користувача»*
2. Перевірка домашнього завдання – виконання самостійної роботи відповідно до рівня знань (високий, середній, достатній).

|  |  |  |
| --- | --- | --- |
| **Високий рівень** | | |
| **№ завдання** | **Постановка** | **Відповідь** |
|  | Доповніть програму, яка обробляє динамічний одновимірний масив. | #include <iostream.h>  #include <conio.h>  double \* func(int i,double \* mas)  { double \* mas\_new = new double[i];  for (int z = 0; z<i; z++)  mas\_new[z] = pow( mas[z],2);  return mas\_new;  }  int main(int argc, char\* argv[])  { double mas[] = {1.0, 2.0, 3.0, 4.0};  double \* mas\_new;  for (int z = 0; z<4; z++)  cout << mas[z]<< endl;  mas\_new = func (4, mas);  for (int z = 0; z<4; z++)  cout << mas\_new[z]<< endl;  delete mas\_new;  return 0;  } |
|  | Дайте визначення функції користувача |  |

**ІІІ *Мотивація навчальної діяльності студентів***

Повідомлення теми, мети і завдання заняття. Обґрунтування значення структур та масивів структур у програмуванні.

**IV *Сприйняття і первинне усвідомлення студентами навчального матеріалу***

**План лекції**

1. ***Визначення структури. Шаблон структури***

* *Поняття структур*
* *Логічний зв`язок елементів структури*
* *Спільне використання елементівструктури*

***Додаток А***

1. ***Поля структури. Структурна змінна***

* *Символьні поля*
* *Числові поля*
* *Функція*

***Додаток Б***

1. ***Звернення до структурної змінної***

* *Оголошення змінної з ініціалізацією*
* *Оголошення змінної без ініціалізації*

***Додаток В***

1. ***Значення структурної змінної***

* *Ініціалізація структурної змінної*
* *Введення значень з клавіатури (файлу)*
* *Програмний спосіб*

***Додаток Г***

***Додаток Д***

* *Задача «Таблиця»* ***Додаток Е***
* *Задача «Спроектуйте результат виконання програми»*
* *Інтернет -задача «Структури»*
* *Інтернет - задача «Одномірний масив»*
* *Інтернет - задача «Матриця»*

1. ***Масив структур***

* *Відеоролик «Ви пам`ятаєте що таке масиви в С++?»*
* *Оголошення шаблону структур*
* *Оголошення масиву структур*
* *Звернення до елементів масиву. Присвоєння значень*
* *Задача «Пошук в масиві структур. Програма «Сесія»*
* *Побудова блок-схеми до програми «Сесія» (Інтернет)* ***Додаток Є***
* *Задача «Знайди помилку»*
* *Інтернет - задача «Масив структур»*

1. **Запитання на закріплення**

* Що таке структура? Дайте визначення.
* Якого типу можуть бути поля структури? Скільки може бути полів?
* Що таке структурна змінна?
* Як звернутись до поля структури?
* В чому відмінність типів char та struсt?
* Як оголосити в програмі масив типу struсt?
* Як звернутись до елементів масиву?

1. **Задачі на закріплення**

**V Підбиття заняття, оцінювання знань студентів**

**VІ Домашнє завдання**

* Вивчити тему «**Поняття структури. Шаблон структури. Структурна змінна»** Щедріна О. І. Алгоритмізація та програмування процедур обробки інформації. Навчальний посібник. - Київ: КНЕУ, 2001, – 176-181 с.;
* Розробити програму обробки масиву структур, який складається із 5 елементів. Кількість полів структури – 3. Виконати вибірку по першому полю структури. Результат вивести на екран монітора.
* Підготуватись до лабораторної роботи №19 «Складання програм з використанням структур».

**Лекція № 29**

Мова С++ володіє могутніми засобами представлення складних даних і надає користувачеві можливість конструювати з основних типів похідні типи даних: структури даних, бітові поля, об’єднання, перелічувальний тип і створення нових типів.

*Структура —* це тип складових (агрегатованих) даних, що являють собою сукупність різноманітних елементів, логічно пов’язаних між собою в процесі розв’язання задачі. Прикладом структури може служити особиста картка студента. Студент описується набором атрибутів, таких, як номер залікової книжки, ім’я, дата народження, стать, адреса, факультет, курс, група і т. д. В свою чергу, деякі з цих атрибутів самі можуть виявитися структурами, наприклад, ім’я, дата народження, адреса. Кожна структура включає в себе один чи кілька об’єктів (змінних, масивів, вказівників, структур), які називаються елементами структури. Елементи структури часто називають членами структури або полями. Структура може мати ім’я, і кожен елемент структури обов’язково повинен мати унікальне в межах структури ім’я. Елементи, об’єднані у структуру, можуть бути різного типу й обов’язково описуватися відповідними атрибутами.

Термін «структура» в мові відповідає двом різним поняттям:

1) позначення місця у пам’яті, де розміщується інформація (далі це місце називається *структурною змінною*);

2) правила формування структурної змінної, використовуваної компілятором для виділення їй місця в пам’яті та організації доступу до її полів. Далі такі правила називаються *шаблоном структури* або структурною змінною.

Як і будь-яка змінна, структурна змінна повинна описуватися в програмі. Опис структурної змінної у програмі відбувається в два етапи:

1. опис структурного шаблону;
2. означення структурної змінної.

Формат опису структурного шаблону є таким:

struct [<ім’я шаблону>]

{ <тип елемента><ім’я елемента>[розмірність];

...

};

де *struct —* ключове слово поняття «структура», яке повідомляє компілятору про оголошення структури;

*<ім’я шаблону>* інакше «тег», ім’я типу структури або ярлик структури є необов’язковим і може формуватися довільно за правилами побудови ідентифікаторів мови. Ім’я шаблону не є ім’ям змінної, воно являє собою мітку, яку можна використати для посилання на цю структуру. Ім’я шаблону є обов’язковим у тому разі, коли шаблон визначений в одному місці, а фактичні змінні в іншому. Для імені шаблону пам’ять не резервується, поки не буде оголошена відповідна структурна змінна. Імена шаблонів повинні бути унікальними в межах їх області визначення. У межах однієї функції може бути лише один шаблон без імені. Ім’я шаблону може збігатися з іменами елементів шаблону, іменами змінних або міток.

Далі перераховуються через крапку з комою описи елементів, що об’єднуються в структуру. Для кожного елемента задається тип, ім’я і, якщо елемент є масивом, то і його розмірність. Імена елементів в одному шаблоні повинні бути унікальними. У різних шаблонах можна використати імена елементів, що збігаються.

Оголошення завершується крапкою з комою, оскільки оголошення структури — це оператор.

Шаблон структури є лише описом схеми розміщення її елементів, яка повідомляє компілятору, але не викликає ніяких дій, тобто визначена тільки форма даних. Задання шаблону не пов’язано з резервуванням будь-якої пам’яті компілятора. Шаблон дає компілятору всю необхідну інформацію про поля структурної змінної для резервування пам’яті й організації доступу до цієї пам’яті під час опису структурної змінної і посилання на окремі поля структурної змінної.

Означення структурної змінної полягає у накладенні шаблону на реальну змінну пам’яті. Довжина змінної, що виділяється під структуру, дорівнює сумі довжин усіх її елементів.

Означення структурної змінної можна здійснювати двома способами. В першому випадку ім’я змінної записується безпосередньо після шаблону, тобто після дужки }, що закривається, вказується ім’я змінної і ставиться крапка з комою ; . При цьому ім’я типу структури вказувати не потрібно, бо опис шаблону суміщений з визначенням структурної змінної. Вигляд оголошення структури тоді є таким:

struct [<ім’я шаблону>]

{ <тип елемента><ім’я елемента>[розмірність];

...

}< список імен структурних змінних>;

У цьому випадку визначення структурного шаблону та структурної змінної суміщені. Імена структурних змінних у списку відокремлюються між собою комами. Структурні змінні або ім’я шаблону можуть бути відсутніми, але не одночасно. Наприклад,

struct anketa

{ int zalik\_nom;

char fio[20];

char pol;

} st1, st2;

Шаблон має ім’я anketa, структура містить три елементи: ціле, символьний масив і символьне. Імена елементів: zalik\_nom, fio, pol; оголошені дві структурні змінні st1, st2, для яких компілятор резервує пам’ять.

У іншому випадку потрібна обов’язкова наявність імені типу структури (імені шаблону), оскільки визначення структурної змінної виноситься за опис шаблону. Формат визначення змінної при цьому є таким:

struct <ім’я шаблону> <ім’я структурної змінної>;

Другий варіант визначення зручно використати в тому випадку, коли структурний шаблон використовується більше за один раз. Наприклад,

#include <stdio.h>

#define size1 30

#define size2 6

struct stt

{ char fio[size1]; // ПІБ студента

char nz[size2]; // номер залікової книжки

float sum\_st; // сума стипендії

};

void main( )

struct stt st; // визначення структурної змінної

Існує два способи ініціалізації елементів структури. Можна проініціалізувати елементи при оголошенні структури або виконати ініціалізацію в тілі програми.

Ініціалізація структурних змінних припускається лише в тому разі, коли вони описані як зовнішні чи статичні. Наприклад,

#include <stdio.h>

#define size1 30

#define size2 6

struct stt

{ char fio[size1];

char nz[size2];

float sum;

};

struct stt st = //оголошення й ініціалізація змінної st типу stt

{"Іваненко",

"608103",

25.00};

Дозволяється об’єднувати задання шаблону, опис структурних змінних і їх ініціалізацію в одному реченні мови С++, наприклад,

#include <stdio.h>

#define size1 30

#define size2 6

struct stt

{ char fio[size1];

char nz[size2];

float sum;

}st ={

"Іваненко",

"608103,

25.00};

Іншим способом є ініціалізація окремих елементів структурної змінної в тілі програми за допомогою операції крапка '.'. Наприклад, st.nz = "608103";. За допомогою даної операції можна вводити дані в структуру з клавіатури, використовуючи функції введення.

Масив, елементами якого є структури, називається *масивом структур*. Опис масиву структур відбувається за аналогією з описом окремої структури в два етапи, при цьому опис шаблону нічим не відрізняється, а при визначенні структурної змінної після її імені вказується розмірність у квадратних дужках, тобто

struct <ім’я шаблону> <ім’я структурної змінної>[розмірність];

Наприклад,

struct stt d[100];

визначений масив d, що складається зі 100 елементів типу stt. У цьому разі компілятор виділяє пам’ять для ста елементів d, довжина кожного з яких є сумою довжин елементів структури stt.

При зверненні до елементів структури необхідно зазначити в потрібному місці виразу чи оператора не лише ім’я елемента, а й ім’я структурної змінної, яка визначена раніше. Формат звернення є таким:

< ім’я структурної змінної >.< ім’я елемента>

Наприклад, для того, щоб звернутися до елемента *сума стипендії*, необхідно записати d.nz.

При звертанні до елементів масиву структур додатково до попереднього у форматі необхідно вказувати конкретний індекс структурної змінної, тобто

< ім’я структурної змінної >[індекс].< ім’я елемента>

Наприклад, d[5].nz.

Посилання на полі вкладеної структури формується з імені структурної змінної, імені структурного елемента, імені елемента вкладеної структури. Перелічені назви розділяються крапкою. Перелічені імена розділяються символом (.) операції крапка. Формат звернення до поля вкладеної структури є таким:

< ім’я структурної змінної >.<ім’я структурного елемента>.  
< ім’я елемента>

Наприклад, d.fio.i = "Олена"

Проілюструємо роботу з масивом структур на програмі, яка здійснює введення відомостей про стипендію студентів і їх виведення на екран.

// Введення даних про студентів

#include <stdio.h>

#define size 30

#define size1 7

#define kol 30

struct stt { // оголошення шаблону

char fio[size]; // ПІБ студента

char nz[size1]; // N залікової книжки

float sum; // сума стипендії

};

void main( )

struct stt d[kol]; // оголошення масиву структур

int i; float sum\_st;

for (i = 0; i<kol; i++)

{printf ("\nВведіть прізвище: ");

scanf ("%s", &d[i].fio);

printf ("\n Введіть номер залікової книжки: ");

scanf ("%s", &d[i].nz);

printf ("\nВведіть суму стипендії: ");

scanf ("%e", &sum\_st);

d[i].sum = sum\_st;

}

// Виведення даних

for (i = 0; i<kol; i++)

printf ("\n %29s %6s %4.2f", d[i]. fio, d[i].nz, d[i].sum);

}

**Додаток А**

**Формат опису структурного шаблону**

struct <ім’я шаблону>

{ <тип елемента><ім’я елемента1>[розмірність];

<тип елемента><ім’я елемента2>[розмірність];

<тип елемента><ім’я елемента3>[розмірність];

…

<тип елемента><ім’я елементаN>[розмірність];

};

де *struct —* ключове слово, яке повідомляє компілятору про оголошення структури;

*<ім’я шаблону>* ім’я типу структури, може формуватися довільно за правилами побудови ідентифікаторів мови.

**Приклад структурного шаблону**

struct anketa

{ int zalik\_nom;

char pib[20];

char pol;

};

**Додаток Б**

**Приклад полів шаблону**

#include <iostream.h>

struct tab

{ int nom;

char pib[20];

void vivod ()

{cout << "Table" << endl; }

};

void main()

{ tab student;

cout <<"Задайте номер " << endl;

cin >> student. nom;

cout <<" Задайте прізвище студента" << endl;

cin >> student. pib;

student.vivod ();

cout << student.nom;

cout << student.pib << endl;

}

**Додаток В**

**Приклад оголошення структурної змінної**

**без ініціалізації**

#include <iostream.h>

struct student

{ char pib[15]; // ПІБ студента

char nz[15]; // номер залікової книжки

float sum\_st; // сума стипендії

};

void main( )

{student st; // визначення структурної змінної

}

|  |  |  |
| --- | --- | --- |
| **pib** | **nz** | **sum\_st** |
|  |  |  |

**Приклад оголошення структурної змінної з ініціалізацією**

#include <iostream.h>

struct student

{ char pib[15]; // ПІБ студента

char nz[15]; // номер залікової книжки

float sum\_st; // сума стипендії

};

void main( )

{student st = {“Устименко”,

“СР 256256”,

802.50}; // визначення структурної змінної

}

|  |  |  |
| --- | --- | --- |
| **pib** | **nz** | **sum\_st** |
| Устименко | СР 256256 | 802.50 |

**Додаток Г**

**Приклад числового поля структури**

#include <iostream.h>

struct chislo

{ int pershe;

int druge;

int trete;

};

void main()

{ chislo zminna;

zminna.pershe = 100;

zminna.druge = 10;

zminna.trete = 15;

}

Приклад логічного поля структури

#include <iostream.h>

struct logika

{ char litera;

bool registr;

};

void main()

{ logika zminna;

zminna.litera = 'R';

zminna.registr = true;

}

Приклад символьного поля структури

#include <iostream.h>

struct ryadok

{ char pershiy;

char drugiy;

char tretiy[20];

};

void main()

{ ryadok zminna;

zminna.pershiy = 'R';

zminna.drugiy = 'F';

zminna.tretiy = "PROGRAMUVANNYA";}

**Додаток Д**

Приклад оголошення з ініціалізацією

#include <iostream.h>

struct tip

{ char pole\_1;

char pole\_2[10];

int pole\_3;

float pole\_4;

};

void main()

{ tip zminna = { '\*', "$$$$$$", 200, 100.55}

}

Приклад введення з клавіатури

#include <iostream.h>

struct tip

{ char pole\_1;

char pole\_2[10];

int pole\_3;

float pole\_4;

};

void main()

{ tip zminna;

cout << "Задайте перше поле" << endl;

cin >> zminna. pole\_1;

cout << "Задайте друге поле"<< endl;

cin >> zminna. pole\_2;

cout << "Задайте третє поле"<< endl;

cin >> zminna. pole\_3;

cout << "Задайте четверте поле"<< endl;

cin >> zminna. pole\_4;

}

**Приклад програмного способу**

#include <iostream.h>

struct tip

{ int pole\_1;

int pole\_2;

float pole\_3;

};

void main()

{ tip zminna;

zminna.pole\_1=100\*2;

zminna.pole\_2=50-25;

zminna.pole\_3= zminna.pole\_1/ zminna.pole\_2;}

**Додаток Е**

**Задача. Структура СТУДЕНТ**

//підключення бібліотек

#include "iostream.h"

#include "conio.h"

#include "windows.h"

#include "iomanip.h"

//оголошення структури

struct tab

{ int nom;

char pib[20];

void vivod()

{cout << "\t\t\tTABLE" << endl;

cout.width(80); cout.fill('-'); cout << '\n';cout.fill(' ');}

};

//прототип функції користувача

void output\_(tab student)

{ cout << setw(20)<<Rus("Номер ");

cout << setw(20)<< Rus("Прізвище студента\n");

cout.width(80); cout.fill('-'); cout << '\n'; cout.fill(' ');

cout << setw(20) << student.nom << setw(20) << student.pib <<setw(15) << endl;}

//головна функція

void main()

{ char vidp;

tab student;

do {

cout <<Rus("Задайте номер ") << endl;

cin >> student.nom;

cout <<Rus("Задайте прізвище студента") << endl;

cin >> student.pib;

student.vivod();

output\_(student);

cout << Rus ("Для продовження роботи натисніть (y)");

cin >> vidp;

} while (vidp == 'y' || vidp == 'Y');

getch();

}

**Спроектуйте результат виконання програми**

struct shablon

{int nom\_zayavu;

char pib[20];

char atestat[10];

int id\_kod;

};

void main ()

{ shablon a = {235689, “Вернигора”, “CР123456”, 1122334455};

if (a.pib[0] == `A` && a.id\_kod == 1122334455)

for (int i = 0; i < 3; i++)

cout << a.atestat;

}

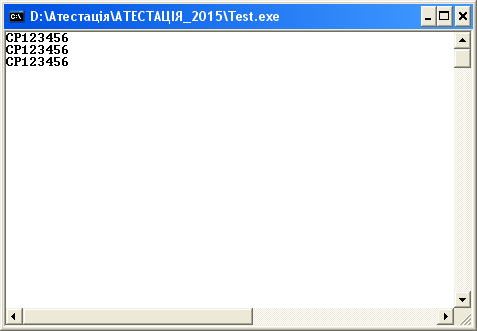


Рис.1 – Результат виконання програми

**Додаток Є**

**Приклад оголошення структури**

//шаблон структури

struct rozklad

{ char predmet[15];

int nomer\_aud[15];

char pib\_vikladach[20];

};

**Приклад оголошення масиву структур**

//шаблон структури

struct rozklad

{ char predmet[15];

int nomer\_aud[15];

char pib\_vikladach[20];

};

void main()

//масив структур

{ rozklad mas[3];

}

**predmet nomer\_aud pib\_vikladach**

|  |  |  |  |
| --- | --- | --- | --- |
| 0 |  |  |  |
| 1 |  |  |  |
| 2 |  |  |  |

Рис. 2 – Поля масиву структур

**Приклад звернення до елементів масиву структур**

#include iostream.h

// шаблон структури

struct rozklad

{ char predmet[15];

int nomer\_aud[15];

char pib\_vikladach[20];

};

//головна функція

void main()

//масив структур

{ rozklad mas[3];

//введення значень полів масиву структур

for (int i=0; i<3; i++)

{cout << "Задайте назву предмету " << endl;

cin >> mas. predmet[i];

cout << "Задайте номер аудиторії " << endl;

cin >> mas. aud[i];

cout << "Задайте прізвище викладача " << endl;

cin >> mas. pib\_vikladach[i];

}

}

**Задача «Знайди помилку»**

#include "iostream.h"

#include "windows.h "

struct sesiya

{ char pib\_stud[15];

int ocinka\_1;

int ocinka\_2;

int ocinka\_3;

};

void main()

{ int i;

double ser;

sesiya student[3];

for (i=0; i < 3; i++);

{ cout <<"Задайте прізвище студента;

cin >> student[i]. pib\_stud;

cout <<"Задайте першу оцінку ");

cin >> student[i]. ocinka\_1;

cout <<"Задайте другу оцінку");

cin >> student[i]. ocinka\_2;

cout <<"Задайте третю оцінку ");

cin >> student[i]. ocinka\_3;

}

for (i=0; i < 3; i++)

cout << student[i]. pib\_stud << "\n"<< student[i]. ocinka\_1<<"\n"<< student[i]. ocinka\_2<<"\n"<< student[i]. ocinka\_3;

for (i=0; i < 3; i++)

{ser = (student[i]. ocinka\_1+ student[i]. ocinka\_2+student[i]. ocinka\_3)/3;

if (ser>=4.00 && student[i]. ocinka\_1!=2 && student[i]. ocinka\_2!=2 && student[i].ocinka\_3!=2)

cout << "Студент отримує стипендію” ;

else

cout <<"Студент не отримує стипендію”;

} ser = 0;

}