**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет Информационных технологий**

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 3**

**Дисциплина:**

Объектно-ориентированное программирование

**Тема:**

Структуры и перечисления.

**Выполнил(а): студент(ка) группы 211-7210**

**Салов Д.К.**

(Фамилия И.О.)

**Дата, подпись** 12.03.22

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2021**

**Цель:**  Получить практические навыки в создании программ, использующих структуры и перечисления.

**Задания:**

4. Создайте структуру с именем employee, содержащую два поля: номер сотрудника типа int и величину его пособия в долларах типа float. Запросите с клавиатуры данные о трех сотрудниках, сохраните их в трех структурных переменных типа employee и выведите информацию о каждом из сотрудников на экран.

5. Создайте структуру типа date, содержащую три поля типа int: месяц, день и год. Попросите пользователя ввести день, месяц и год в формате 31/12/2002, сохраните введенное значение в структурной переменной, а затем извлеките данные из этой переменной и выведите их на экран в том же формате, в каком они вводились.

6. Как мы говорили, стандартные средства ввода/вывода C++ вместо значений перечисляемых типов данных выводят их внутреннее представление в виде целых чисел. Для того чтобы преодолеть это ограничение, вы можете использовать конструкцию switch, с помощью которой устанавливается соответствие между значением переменной перечисляемого типа и се внутренним представлением. Пусть, например, в программе определен перечисляемый тип данных etype, отражающий должность сотрудника: enum etype { laborer, secretary, manager, accountant, executive, researcher };

Напишите программу, которая сначала по первой букве должности, введенной пользователем, определяет соответствующее значение переменной, помещает это значение в переменную типа etype, а затем выводит полностью название должности, первую букву которой ввел пользователь. Взаимодействие программы с пользователем может выглядеть следующим образом: Введите первую букву должности

(laborer, secretary, manager, accountant, executive, researcher): a

Полное название должности; accountant

Возможно, вам понадобится два ветвления switch: одно — для ввода значения, другое — для вывода.

7. Добавьте поля типа enum etype (см. упражнение 6) и struct date (см. упражнение 5) в структуру employee из упражнения 4. Организуйте программу таким образом, чтобы пользователь вводил 4 пункта данных о каждом из трех сотрудников: его номер, величину зарплаты, его должность и дату принятия на работу. Программа должна хранить введенные значения в трех переменных типа employee и выводить их содержимое на экран.

8. Вернитесь к упражнению 9 комплекта заданий 2. В этом упражнении требуется написать программу, которая хранит значения двух дробей в виде числителя и знаменателя, а затем складывает эти дроби согласно арифметическому правилу. Измените эту программу так, чтобы значения дробей хранились в структуре fraction, состоящей из двух полей типа int. предназначенных для хранения числителя и знаменателя. Все значения дробей должны храниться в переменных типа fraction.

9. Создайте структуру с именем time. Три ее поля, имеющие тип int. будут называться hours, minutes и seconds. Напишите программу, которая просит пользователя ввести время в формате часы, минуты, секунды. Можно запрашивать на ввод как три значения сразу, так и выводить для каждой величины отдельное приглашение. Программа должна хранить время в структурной переменной типа time и выводить количество секунд в введенном времени, определяемое следующим образом: long totalsecs = t1.hours\*3600 + t1 minutes\*60 + t1.seconds

10. Создайте структуру с именем sterling, хранящую денежные суммы в старой английской системе, описанной в упражнениях 8 и 11 комплекса заданий 3. Поля структуры могут быть названы pounds, shillings и pence и иметь тип int. Программа должна запрашивать у пользователя значение денежной суммы в новых десятичных фунтах (значение должно храниться в переменной типа double), затем переводить эту сумму в старую систему, сохранять переведенное значение в переменной типа sterling и выводить на экран полученную сумму в фунтах, шиллингах и пенсах.

11. Используя структуру time из упражнения 9, напишите программу, которая получает от пользователя два значения времени в формате 12:59:59, сохраняет их в переменных типа struct time, затем переводит оба значения в секунды, складывает их, переводит сумму в исходный формат, сохраняет его в переменной типа time и выводит полученный результат на экран в формате 12:59:59.

12. Переработайте программу-калькулятор для дробей, описанную в упражнении 12 комплекта упражнений 3 так, чтобы каждая из дробей хранилась как значение переменной типа struct fraction, по аналогии с упражнением 8 этого комплекса упражнений.

**Код:**

#include <iostream>

#include <iomanip>

using namespace std;

void fun4(){

struct employee

{

int number;

int salary;

};

employee employee1, employee2, employee3;

employee1 = { 1, 28457 };

employee2 = { 2, 14852 };

employee3 = { 3, 123456 };

cout << "сотрудник №" << employee1.number << endl << "\nПособие:" << employee1.salary;

cout << "сотрудник №" << employee2.number << endl << "\nПособие:" << employee2.salary;

cout << "сотрудник №" << employee3.number << endl << "\nПособие:" << employee3.salary;

}

void fun5(){

typedef struct{

int month;

int day;

int year;

}date;

//Вот зачем тут псевдонимы?

date dt;

cin >> dt.day;

std::cin.ignore(2, '/');

std::cin >> dt.month;

std::cin.ignore(2, '/');

std::cin >> dt.year;

std::cout << dt.day << '/' << dt.month << '/' << dt.year << std::endl;

}

void fun6() {

enum etype { laborer, secretary, manager, accountant, executive, researcher };

char smb;

cout << "БУКАВА! НУЖНА БУКАВА.......";

cin >> smb;

cout << "\nвозможно вы имели ввиду ";

etype name;

switch (smb) {

case 'l': name = laborer; break;

case 's': name = secretary; break;

case 'm': name = manager; break;

case 'a': name = accountant; break;

case 'e': name = executive; break;

case 'r': name = researcher; break;

}

switch (name) {

case laborer: cout << "labrory"; break;

case secretary: cout << "secretary"; break;

case manager: cout << "manager"; break;

case accountant: cout << "accountant"; break;

case executive: cout << "executive"; break;

case researcher: cout << "researcher"; break;

}

}

void fun7() {

struct date

{

int day, month, year;

};

enum etype {laborer, secretary, manager, accountant, executive, researcher};

struct employee

{

string EmpName;

double salary;

etype post;

date data;

};

employee employee1, employee2, employee3;

char slash = '/', s;

//первый

cout << "name of the employee 1, salary, date of employment (xx/xx/xxxx), post (1-st letter)" << endl;

cin >> employee1.EmpName >> employee1.salary >> employee1.data.day >> slash >> employee1.data.month >>

slash >> employee1.data.year >> s;

switch (s)

{

case 'l': {employee1.post = laborer; break; }

case 's': {employee1.post = secretary; break; }

case 'm': {employee1.post = manager; break; }

case 'a': {employee1.post = accountant; break; }

case 'e': {employee1.post = executive; break; }

case 'r': {employee1.post = researcher; break; }

default:cout << "ОШИБКА ВВОДА!";

}

//второй

cout << "name of the employee 2, salary, date of employment, post (1-st letter)" << endl;

cin >> employee2.EmpName >> employee2.salary >> employee2.data.day

>> slash >> employee2.data.month >> slash >> employee2.data.year >> s;

switch (s)

{

case 'l': {employee2.post = laborer; break; }

case 's': {employee2.post = secretary; break; }

case 'm': {employee2.post = manager; break; }

case 'a': {employee2.post = accountant; break; }

case 'e': {employee2.post = executive; break; }

case 'r': {employee2.post = researcher; break; }

default:cout << "ОШИБКА ВВОДА!";

}

//третий

cout << "name of the employee 3, salary, date of employment, post (1-st letter)" << endl;

cin >> employee3.EmpName >> employee3.salary >> employee3.data.day

>> slash >> employee3.data.month >> slash >> employee3.data.year >> s;

switch (s)

{

case 'l': {employee3.post = laborer; break; }

case 's': {employee3.post = secretary; break; }

case 'm': {employee3.post = manager; break; }

case 'a': {employee3.post = accountant; break; }

case 'e': {employee3.post = executive; break; }

case 'r': {employee3.post = researcher; break; }

default:cout << "ОШИБКА ВВОДА!";

}

//экипаж галеры погружен - выгружаем--------------------------------------------------------------

//первый пошёл

cout << "\nПервый раб " << employee1.EmpName << ", " << employee1.salary << "$, " << employee1.data.day

<< slash << employee1.data.month << slash << employee1.data.year << " date of employment, " << endl;

cout << "post" << endl;

switch (employee1.post){

case laborer: {cout << "laborer\n"; break; }

case secretary: {cout << "secretary\n"; break; }

case manager: {cout << "manager\n"; break; }

case accountant: {cout << "accountant\n"; break; }

case executive: {cout << "executive\n"; break; }

case researcher: {cout << "researcher\n"; break; }

};

//второй пошёл

cout << "Второй раб " << employee2.EmpName << ", " << employee2.salary << "$, " << employee2.data.day

<< slash << employee2.data.month << slash << employee2.data.year << " date of employment, " << endl;

cout << "post" << endl;

switch (employee2.post)

{

case laborer: {cout << "laborer\n"; break; }

case secretary: {cout << "secretary\n"; break; }

case manager: {cout << "manager\n"; break; }

case accountant: {cout << "accountant\n"; break; }

case executive: {cout << "executive\n"; break; }

case researcher: {cout << "researcher\n"; break; }

};

//третий пошёл

cout << "Третий раб " << employee3.EmpName << ", " << employee3.salary << "$, " << employee3.data.day <<

slash << employee3.data.month << slash << employee3.data.year << " date of employment, " << endl;

cout << "post" << endl;

switch (employee3.post)

{

case laborer: {cout << "laborer\n"; break; }

case secretary: {cout << "secretary\n"; break; }

case manager: {cout << "manager\n"; break; }

case accountant: {cout << "accountant\n"; break; }

case executive: {cout << "executive\n"; break; }

case researcher: {cout << "researcher\n"; break; }

};

}

void fun8() {

struct Fraction {

int chisl;

int znam;

};

int a, b, c, d;

cout << "Введите первую дробь: \n";

cin >> a;

cout << "/\n";

cin >> b;

cout << "Введите вторую дробь:\n";

cin >> c;

cout << "/\n";

cin >> d;

Fraction fraction;

fraction.chisl = a \* d + b \* c;

fraction.znam = b \* d;

cout << "Сумма: " << fraction.chisl << '/' << b \* d;

}

void fun9() {

struct Time

{

int hours;

int minutes;

int seconds;

};

Time t1;

cout << "Введите часы, затем минуты, затем секунды " << endl;

cin >> t1.hours >> t1.minutes >> t1.seconds;

long totalsecs = (t1.hours \* 3600) + (t1.minutes \* 60) + t1.seconds;

cout << "Время в секундах " << totalsecs << endl;

}

void fun10() {

struct sterling

{

int pounds;

int shilling;

int pens;

};

sterling newp, oldp;

char ch = '.';

int sumpens;

cout << "Введите есятичные фунты: " << endl;

cin >> newp.pounds >> ch >> newp.pens;

sumpens = ((newp.pounds \* 100) + newp.pens) \* 2.4;

newp.pounds = sumpens / (20 \* 12);

newp.shilling = sumpens % (20 \* 12) / 12;

newp.pens = sumpens % (20 \* 12) % 12;

cout << "Старые фунты фунты: " << newp.pounds << ch << newp.shilling << ch << newp.pens << endl;

}

void fun11() {

struct Time

{

int hours;

int minutes;

int seconds;

};

Time t1, t2, tsum;

char ch;

cout << "Ввод по шаблону (часы:минуты:секунды) первого времени" << endl;

cin >> t1.hours >> ch >> t1.minutes >> ch >> t1.seconds;

cout << "Ввод по шаблону (часы:минуты:секунды) второго времени)" << endl;

cin >> t2.hours >> ch >> t2.minutes >> ch >> t2.seconds;

long totalsecs = ((t1.hours \* 3600) + (t1.minutes \* 60) + t1.seconds) + ((t2.hours \* 3600) + (t2.minutes \* 60) + t2.seconds);

tsum.hours = totalsecs / (60 \* 60);

tsum.minutes = totalsecs % (60 \* 60) / 60;

tsum.seconds = totalsecs % (60 \* 60) % 60;

cout << "Общее время: " << tsum.hours << ch << tsum.minutes << ch << tsum.seconds << endl;

}

void fun12() {

struct fraction

{

int numerator;

int denominator;

};

fraction f1, f2, sumf;

char slash = '/', signal, yesorno;

do

{

cout << "Введите первый операнд: ";

cin >> f1.numerator >> slash >> f1.denominator;

cout << "Знак операции: ";

cin >> signal;

cout << "Второй операнд: ";

cin >> f2.numerator >> slash >> f2.denominator;

switch (signal){

case '+':

sumf.numerator = f1.numerator \* f2.denominator + f1.denominator \* f2.numerator;

sumf.denominator = f1.denominator \* f2.denominator;

cout << "\nРезультат: " << sumf.numerator << slash << sumf.denominator << endl;

break;

case '-':

sumf.numerator = f1.numerator \* f2.denominator - f1.denominator \* f2.numerator;

sumf.denominator = f1.denominator \* f2.denominator;

cout << "\nРезультат: " << sumf.numerator << slash << sumf.denominator << endl;

break;

case '\*':

sumf.numerator = f1.numerator \* f2.numerator;

sumf.denominator = f1.denominator \* f2.denominator;

cout << "\nРезультат: " << sumf.numerator << slash << sumf.denominator << endl;

break;

case '/':

sumf.numerator = f1.numerator \* f2.denominator;

sumf.denominator = f1.denominator \* f2.numerator;

cout << "\nРезультат: " << sumf.numerator << slash << sumf.denominator << endl;

break;

default:

cout << "Убери кота с клавиатуры!" << endl;

}

cout << "\nМожем повторить (y / n)? ";

cin >> yesorno;

} while (yesorno == 'y');

}

int main(){

setlocale(LC\_ALL, "Russian");

int nomer;

cout << "\nВведите номер задачи\n";

cin >> nomer;

switch (nomer) {

case 5:

fun5();

break;

case 6:

fun6();

break;

case 7:

fun7();

break;

case 8:

fun8();

break;

case 9:

fun9();

break;

case 10:

fun10();

break;

case 11:

fun11();

break;

case 12:

fun12();

break;

}

return 0;

}