ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ

*Институт Принтмедиа и 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 181-722

Голиков Р.О.

(Фамилия И.О.)

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

(Дата) (Подпись)

Проверил: Ктн Доцент Арсентьев Д.А.*\_\_\_\_\_\_\_\_\_\_*

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

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

(Дата) (Подпись)

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва

2018

**Задание 1**

Номер телефона, например (212) 767—8900, можно условно разделить на три части: код города (212), номер телефонной станции (767) и номер абонента (8900). Напишите программу с использованием структуры, позволяющую раздельно хранить эти три части телефонного номера. Назовите структуру phone. Создайте две структурные переменные типа phone. Инициализацию одной из них произведите сами, а значения для другой запросите с клавиатуры. Затем выведите содержимое обеих переменных на экран.

**Код:**

#include <iostream>

using namespace std;

struct phone

{

int area; //area code (3 digits)

int exchange; //exchange (3 digits)

int number; //number (4 digits)

};

int main()

{

phone ph1 = { 212, 767, 8900 }; //initialize phone number

phone ph2; //define phone number

// get phone no from user

cout << "\nEnter your area code, exchange, and number";

cout << "\n(Don't use leading zeros): ";

cin >> ph2.area >> ph2.exchange >> ph2.number;

cout << "\nMy number is " //display numbers

<< '(' << ph1.area << ") "

<< ph1.exchange << '-' << ph1.number;

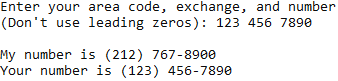
cout << "\nYour number is "

<< '(' << ph2.area << ") "

<< ph2.exchange << '-' << ph2.number << endl;

return 0;

}

**Вывод:**  


**Задание 2**

Расположение точки на плоскости можно задать с помощью двух координат: х и у. Например, если точка имеет координаты (4, 5), то это значит, что она отстоит на 4 единицы справа от вертикальной оси и на 5 единиц вверх от горизонтальной оси. Сумма двух точек определяется как точка, имеющая координаты, равные сумме соответствующих координат слагаемых.

Напишите программу, использующую для интерпретации точки на плоскости структуру с названием point. Определите три переменные типа point, и две из них инициализируйте с помощью значений, вводимых с клавиатуры. Затем присвойте третьей переменной значение суммы первых двух переменных и выведите результат на экран.

**Код:**

#include "pch.h"

#include <iostream>

using namespace std;

struct point

{

int xCo; //X coordinate

int yCo; //Y coordinate

};

int main()

{

point p1, p2, p3; //define 3 points

cout << "\nEnter coordinates for p1: "; //get 2 points

cin >> p1.xCo >> p1.yCo; //from user

cout << "Enter coordinates for p2: ";

cin >> p2.xCo >> p2.yCo;

p3.xCo = p1.xCo + p2.xCo; //find sum of

p3.yCo = p1.yCo + p2.yCo; //p1 and p2

cout << "Coordinates of p1+p2 are: " //display the sum

<< p3.xCo << ", " << p3.yCo << endl;

return 0;

}

**Вывод:**

****

**Задание 3**

Создайте структуру с именем Volume, содержащую три поля тина Distance из примера englstrc, для хранения трех измерений помещения. Определите переменную типа Volume, инициализируйте ее, вычислите объем, занимаемый помещением, и выведите результат на экран. Для подсчета объема переведите каждое из значений типа Distance в значение типа float, равное соответствующей длине в футах и хранимое в отдельной переменной. Затем для вычисления объема следует перемножить три полученные вещественные переменные.

**Код:**

#include <iostream>

using namespace std;

struct Distance

{

int feet;

float inches;

};

struct Volume

{

Distance length;

Distance width;

Distance height;

};

int main()

{

float l, w, h;

Volume room1 = { { 16, 3.5 }, { 12, 6.25 }, { 8, 1.75 } };

l = room1.length.feet + room1.length.inches / 12.0;

w = room1.width.feet + room1.width.inches / 12.0;

h = room1.height.feet + room1.height.inches / 12.0;

cout << "Volume = " << l \* w\*h << " cubic feet\n";

return 0;

}

**Вывод:**



**Задание 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.

**Код:**

#include <iostream>

using namespace std;

struct Employee

{

int Id;

float Salary;

};

int main()

{

setlocale(LC\_ALL, "");

Employee empl1, empl2, empl3;

cout << "Введите данные первого сотрудника: " << endl;

cin >> empl1.Id >> empl1.Salary;

cout << "Введите данные второго сотрудника: " << endl;

cin >> empl2.Id >> empl2.Salary;

cout << "Введите данные третьего сотрудника: " << endl;

cin >> empl3.Id >> empl3.Salary;

cout << "Сотрудник " << empl1.Id << ", пособие: $" << empl1.Salary << endl;

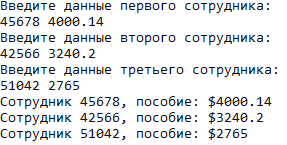
cout << "Сотрудник " << empl2.Id << ", пособие: $" << empl2.Salary << endl;

cout << "Сотрудник " << empl3.Id << ", пособие: $" << empl3.Salary << endl;

return 0;

}

**Вывод:**



**Задание 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.

**Код:**

#include <iostream>

using namespace std;

struct Date

{

int Day;

int Month;

int Year;

};

int main()

{

setlocale(LC\_ALL, "");

Date date;

char ch;

cout << "Введите дату: ";

cin >> date.Day >> ch >> date.Month >> ch >> date.Year;

cout << "Введённая дата: " << date.Day << ch << date.Month << ch << date.Year;

return 0;

}

**Вывод:**



**Задание 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.

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.

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

**Код:**

#include <iostream>

#include <string>

using namespace std;

enum etype

{

laborer,

secretary,

manager,

accountant,

executive,

researcher

};

int main()

{

setlocale(LC\_ALL, "");

etype pos;

char ch;

cout << "Введите первую букву должности\n"<<

"(laborer, secretary, manager, accountant, executive, researcher): ";

cin >> ch;

switch (ch)

{

case 'l': pos = laborer; break;

case 's': pos = secretary; break;

case 'm': pos = manager; break;

case 'a': pos = accountant; break;

case 'e': pos = executive; break;

case 'r': pos = researcher; break;

default: cout << "Ошибка ввода"; return -1;

}

string name;

switch (pos)

{

case 0: name = "laborer"; break;

case 1: name = "secretary"; break;

case 2: name = "manager"; break;

case 3: name = "accountant"; break;

case 4: name = "executive"; break;

case 5: name = "researcher"; break;

}

cout << "Полное название должности: " << name;

return 0;

}

**Вывод:**



**Задание 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.

**Код:**

#include <iostream>

#include <string>

using namespace std;

enum etype

{

laborer,

secretary,

manager,

accountant,

executive,

researcher

};

struct Date

{

int Day;

int Month;

int Year;

};

struct Employee

{

int Id;

float Salary;

Date Date;

etype Etype;

};

int main()

{

setlocale(LC\_ALL, "");

Employee empl;

char e, ch;

cout << "Введите даные сотрудника\n" <<

"(номер, величину зарплаты, должность(первую букву) и дату принятия на работу)\n";

cin >> empl.Id >> empl.Salary >> e >> empl.Date.Day >> ch >> empl.Date.Month >> ch >> empl.Date.Year;

switch (e)

{

case 'l': empl.Etype = laborer; break;

case 's': empl.Etype = secretary; break;

case 'm': empl.Etype = manager; break;

case 'a': empl.Etype = accountant; break;

case 'e': empl.Etype = executive; break;

case 'r': empl.Etype = researcher; break;

default: cout << "Ошибка ввода"; return -1;

}

string name;

switch (empl.Etype)

{

case 0: name = "laborer"; break;

case 1: name = "secretary"; break;

case 2: name = "manager"; break;

case 3: name = "accountant"; break;

case 4: name = "executive"; break;

case 5: name = "researcher"; break;

}

cout << "\nДанные сотрудника " << empl.Id << ":\n" <<

"зарплата $" << empl.Salary << endl <<

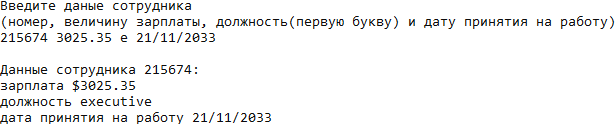
"должность " << name << endl <<

"дата принятия на работу " << empl.Date.Day << ch << empl.Date.Month << ch << empl.Date.Year;

return 0;

}

**Вывод:**



**Задание 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.

**Код:**

#include <iostream>

using namespace std;

struct fraction

{

int dividend;

int divider;

};

int main()

{

setlocale(LC\_ALL, "");

fraction a, b, res;

char dummychar;

cout << "Введите две дроби: \n";

cin >> a.dividend >> dummychar >> a.divider;

cin >> b.dividend >> dummychar >> b.divider;

res.dividend = a.dividend \* b.divider + b.dividend \* a.divider;

res.divider = a.divider \* b.divider;

cout << "Их сумма равна " << res.dividend << "/" << res.divider;

return 0;

}

**Вывод:**



**Задание 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.

**Код:**

#include <iostream>

using namespace std;

struct Time

{

int seconds;

int minutes;

int hours;

};

int main()

{

setlocale(LC\_ALL, "");

Time t1;

char ch;

cout << "Введите время: ";

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

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

cout << "Всего секунд: " << totalsecs;

return 0;

}

**Вывод:**



**Задание 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.

**Код:**

#include <iostream>

using namespace std;

struct Sterling

{

const float PoundsToOldShilling = 20.0f;

const float OldShillingToOldPennies = 12.0f;

int Pounds;

int Shillings;

int Pence;

Sterling(double newSum)

{

Pounds = (int)newSum;

double sh = (newSum - Pounds) \* PoundsToOldShilling;

Shillings = sh;

Pence = (int)((sh - Shillings) \* OldShillingToOldPennies);

}

};

int main()

{

setlocale(LC\_ALL, "");

double decimalPounds;

cout << "Введите число десятичных фунтов: ";

cin >> decimalPounds;

Sterling oldSum = Sterling(decimalPounds);

cout << "Эквивалентная сумма в старой форме записи: £" <<

oldSum.Pounds << "." << oldSum.Shillings << "." << oldSum.Pence;

return 0;

}

**Вывод:**



**Задание 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.

**Код:**

#include <iostream>

using namespace std;

struct Time

{

int seconds;

int minutes;

int hours;

int TotalSeconds()

{

return hours \* 3600 + minutes \* 60 + seconds;

}

Time(){}

Time(int totalSeconds)

{

hours = totalSeconds / 3600;

int buf = totalSeconds % 3600;

minutes = buf / 60;

seconds = buf % 60;

}

};

int main()

{

setlocale(LC\_ALL, "");

Time t1, t2;

char ch;

cout << "Введите два время:\n";

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

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

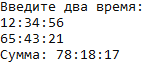
Time resTime = Time(t1.TotalSeconds() + t2.TotalSeconds());

cout << "Сумма: " << resTime.hours << ch << resTime.minutes << ch << resTime.seconds;

return 0;

}

**Вывод:**



**Задание 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>

using namespace std;

struct fraction

{

int dividend;

int divider;

};

int main()

{

setlocale(LC\_ALL, "");

fraction a, b, res;

char oper, ch;

do {

cout << "\nВведите выражение с дробями: ";

cin >> a.dividend >> ch >> a.divider >> oper >> b.dividend >> ch >> b.divider;

switch (oper)

{

case '+':

res.dividend = a.dividend \* b.divider + a.divider \* b.dividend;

res.divider = a.divider \* b.divider;

break;

case '-':

res.dividend = a.dividend \* b.divider - a.divider \* b.dividend;

res.divider = a.divider \* b.divider;

break;

case '\*':

res.dividend = a.dividend \* b.dividend;

res.divider = a.divider \* b.divider;

break;

default: res.dividend = res.divider = 0;

}

cout << "Answer = " << res.dividend << '/' << res.divider;

cout << "\nDo another (Enter 'y' or 'n')? ";

cin >> ch;

} while (ch != 'n');

return 0;

}

**Вывод:**

