Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 РЕФЕРАТ

По дисциплине «Основы алгоритмизации и программирования»

На тему «Отличие class,struct,union»

Выполнила:

Студентка1 курса 7 группы

Гриценко Анна Александровна

2023, Минск

Оглавление

[Класс(Class) 3](#_Toc158374318)

[Структура(Struct) 3](#_Toc158374319)

[Объединения(Union) 4](#_Toc158374320)

[Вывод: отличия class, struct, union 5](#_Toc158374321)

[Литература 6](#_Toc158374322)

# Класс(Class)

Классы предоставляют механизм для создания объектно-ориентированных программ. Класс является пользовательским типом данных, который мы можем создать для решения определенной задачи. В классе объявляются данные и функции, которые оперируют этими данными (выполняют их обработку). Каждая переменная типа «класс» является объектом. Те данные, которые характеризуют объекты класса называются полями, а функции, которые выполняют действия над данными, называются методами.

**Пример кода:**

#include <iostream>

using namespace std;

class Rectangle {

public: //данные с этим модификатором являются открытыми

// Конструктор класса

Rectangle(int w, int h) : width(w), height(h) {}

// Метод для вычисления площади прямоугольника

int calculateArea() {

return width \* height;

}

private: //данные с этим модификатором являются закрытыми

int width;

int height;

};

int main() {

setlocale(LC\_CTYPE, "rus");

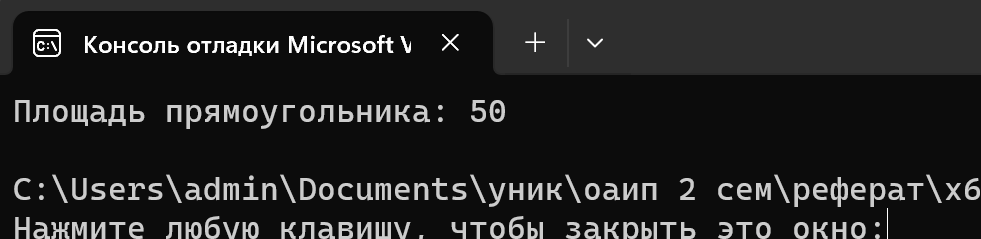
// Использование класса Rectangle

Rectangle rect(5, 10);

cout << "Площадь прямоугольника: " << rect.calculateArea() << endl;

return 0;

}

****

# Структура(Struct)

Структуры идентичны классам по своему синтаксису, но имеют ряд отличий, о которых говорится ниже. Чтобы определить структуру, применяют ключевое слово **struct**. Компоненты структуры - набор описаний объектов и функций, составляющих эту структуру.

**Пример кода:**

#include <iostream>

using namespace std;

struct Point {

int x;

int y;

};

int main() {

setlocale(LC\_CTYPE, "rus");

// Использование структуры Point

Point p1 = { 2, 3 };

cout << "Координаты точки: (" << p1.x << ", " << p1.y << ")" <<endl;

return 0;

}

**Пример кода:**

#include <iostream>

using namespace std;

struct Point {

int x;

int y;

};

int main() {

setlocale(LC\_CTYPE, "rus");

// Использование структуры Point

Point p1 = { 2, 3 };

cout << "Координаты точки: (" << p1.x << ", " << p1.y << ")" <<endl;

return 0;

}

# Объединения(Union)

Объединения — это тип класса, в котором все данные хранятся в одной и той же области памяти. В языке c++ объединение может включать как функции, так и данные. Все члены объединения являются открытыми по умолчанию.

**Пример кода:**

#include <iostream>

using namespace std;

union tag\_var {

char var\_ch;;

int var\_i;

double var\_d;

};

int main() {

setlocale(LC\_CTYPE, "rus");

// Использование объединения tag\_var

tag\_var var;

var.var\_ch='c';

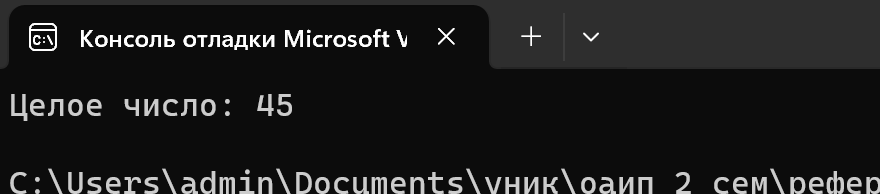
var.var\_i=45;

// Вывод значения целого числа

cout << "Целое число: " << var.var\_i << endl;

return 0;

}

****

В консоль будет выводится только последнее записанное значение, остальные значения затираются. Например, если мы попросим вывести значение var.var\_ch, то мы не получим значение с:

#include <iostream>

using namespace std;

union tag\_var {

char var\_ch;;

int var\_i;

double var\_d;

};

int main() {

setlocale(LC\_CTYPE, "rus");

// Использование объединения tag\_var

tag\_var var;

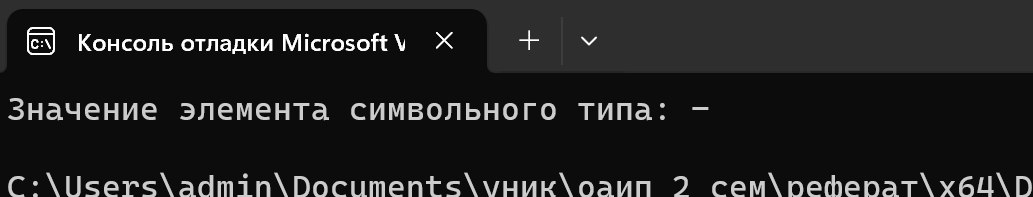
var.var\_ch='c';

var.var\_i=45;

// Вывод значения как целого числа

cout << "Значение элемента символьного типа: " << var.var\_ch << endl;

return 0;



# Вывод: отличия class, struct, union

1. В классах(**class)** все поля и методы по умолчанию являются закрытыми (private), в **struct**, наоборот, все члены по умолчанию открыты (public). Эти значения можно поменять, изменив модификатор доступа в коде.
2. При наследовании также дочерние элементы родительского класса(class) являются private, а для struct-public.
3. Отличается от предыдущих типов тем, что его используют для экономии памяти. Выделяется кол-во памяти, который занимает самый большой элемент в объединении(union).
4. Все поля в структуре(struct) в памяти располагаются последовательно, друг за другом. В объединении(union) все переменные хранятся в одном и том же начальном адресе. В этом случае значение одной переменной будет затирать значение оставшихся. Объединение не предназначено для хранения всех своих элементов, для этого предназначена структура (struct).

# Литература

* <https://codelessons.dev/ru/klassy-v-c-rukovodstvo-dlya-nachinayushhix/>
* <https://ravesli.com/urok-113-klassy-obekty-i-metody-klassov/>
* <http://profil.adu.by/mod/book/tool/print/index.php?id=3574>