Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Реферат

По дисциплине «Основы алгоритмизации и программирования»

На тему «Отличие struct, class, union»

Выполнила:

Студент(ка) 1 курса 7 группы

Подшиваленко Диана Игоревна

Проверил:

Белодед Николай Иванович

2023, Минск

**Содержание**

1. Предисловие … 3
2. Основное отличие1 … 4
3. Основное отличие2 … 4
4. Ещё отличия … 5

**Предисловие**

В данном реферате будут рассмотрены основные отличия struct, class, union.

**Основное отличие 1**

Члены класса, определенного с помощью ключевого слова class, по умолчанию являются private (закрытыми). Члены класса, определенного с помощью ключевого слова struct или union, по умолчанию являются public(открытыми).

#include <iostream>

#include<Windows.h>

#include <conio.h>

using namespace std;

class Person

{

string name;

};

struct person {

string name;

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Person people1;

cout << people1.name; // компилятор выдает ошибку, так как private

person people2;

cout << people2.name; // все работает, так как public

}

**Основное отличие 2**

При отсутствии спецификатора доступа (т.е. private/protected/public) у базового класса, базовый класс будет public если класс определен с помощью struct и private если класс определен с помощью class.

#include <iostream>

#include<Windows.h>

#include <conio.h>

using namespace std;

class Basis {

public:

int x;

};

class New : Basis { };

struct New2 : Basis { };

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

New d1;

New2 d2;

d1.x = 13; // компилятор выдает ошибку, потому что private

d2.x = 13; // все хорошо, потому что public

}

**Ещё отличия**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Класс** | **Структура** | **Объединение** |
| видимость члена по умолчанию | private | public | public |
| режим наследования по умолчанию | private | public | Не наследуется |
| Конструктор | есть | есть | есть |
| Деструктор | есть | есть | есть |
| Функция-член | есть | есть | есть |
| Перегрузка функции | есть | есть | есть |
| Перегрузка оператора | есть | есть | есть |
| Наследование | есть | есть | нет |
| Полиморфизм | есть | есть | нет |
| Переопределение функции | есть | есть | нет |
| ограничение использования | нет | нет | доступна только 1 переменная в одно и то же время |