

UBT-1 Разговор над 2 Вопросы

1. В C++ классами являются типы, объявленные с помощью `class`, `struct` и `union`.

Все правила для `class` и `struct` одинаковы, за исключением случаев, где типичны указатели явное

Стандарт C++:

Члены класса `class` являются `private`.

Члены класса `struct` являются `public`.

При отсутствии спецификатора доступа у члена класса при `class` он будет `private`, а в `struct` `public`.

Как правило структуры не имеют конструкторов и деструкторов.

2. Классы и объекты в C++ - различные понятия. Типичный класс в C++ - это тип данных, а объект - конкретный экземпляр класса, хранящийся в памяти компьютера. Как переменные относятся к типу, так и объекты относятся к классу.

Класс: фрукт Объект: яблоко

3. проект

_conf\ - файлы конфигурации
_src\ - основной источник
_src\module1 - для каждого внешнего модуля

_doc\ - документация

_include\ - для внешних проектов

_os\ - файлы ОС

_res\ - ресурсы, несвязанные с файлами

4. Include Guards в C++ - это директивы управления компилятором, которые помогают решить проблему, когда одна и та же функция или переменная по ошибке определяется несколько раз.

Include Guards решает проблему двойного подключения.

Также проблему двойной инициализации можно решить

```
Class(): Class(4, 20) {}
```


5. Инкапсуляция - упаковка данных и функций в один компонент (например, класс) и контролируемый доступ к этому компоненту.

private Члены класса могут использоваться только функциями членами и друзьями

public Члены класса могут использоваться любыми функциями.

protected Члены класса могут использоваться друзьями

6. Конструктор - особый метод класса, который выполняется автоматически в момент создания объекта класса.

Деструктор - особый метод, который выполняется во время уничтожения объектов класса.

Конструкторы бывают без параметров и с параметрами

```
Box () {
```

```
    int x = 5;
```

```
    y = 5;
```

```
}
```

```
BEZ
```

```
Box (int a, int b) {
```

```
    x = a;
```

```
    y = b;
```

```
}
```

```
C
```


7. Указатель `this` - это указатель, доступный только в нестатических функциях члена `class`, `struct` или `union` типа. Он указывает на объект, где находится функция-член. Статические ср-ции-члены не имеют указателя `this`.

8. Если элемент объявлен как `static`, где все объекты класса содержатся только одна копия данных. Статические данные-члены не входят в состав объектов указанного типа класса.

Статические функции-функции, не принадлежащие объекту класса и вызываются так же, но аналогично со статическими переменными.

9. Синтаксис переопределения операторов очень похож на обращение ср-ции с именем `operator@`, где

@ - это идентификатор оператора (например `+`, `-`)

~~`Operator + {`~~

```
class Integer {
private:
    int value;
public:
    Integer(int i): value(i) {}
    const Integer operator + (const Integer &r) const {
        return (value + r.value);
    }
}
```


42. Функция называется непрерывной, если она
создаётся в одной области видимости и имеет
одно и то же имя.

```
void Foo (int x);
```

```
void Foo (const int x);
```

10. Ключевое слово `const` указывает, что значение
переменной является константой, то есть представителем
её численного выражения.

Значения, определённые с помощью `const`, являются
проверки типов могут использоваться вместо
константных выражений. В C++ можно указать
размер массива с `const`.

`const` можно использовать в объявлениях указателей.
Объявление `const` функции указывает, что функцию
нельзя где-то изменить.

11. Можно использовать `delete` для всех переменных.

```
A() = delete;
```

```
A(const A&) = delete;
```

или же написать `constexpr` в `private`.

9.2. При шифровании исходный объект хранит копии значений объекта, тогда как при полнотекстовом шифровании исходный объект хранит ссылки на исходные данные.

При шифровании не требуется изменение, внесенное в исходный / исходный объект в исходном; в то время как полнотекстовые копии сохраняются