Перегрузка операторов:

<http://en.cppreference.com/w/cpp/language/operators>

Подробнее:

1. Перегрузка обычного оператора  
   operator *op*

Что из операторов может быть перегружено:

+ - \* / % ˆ & | ~ ! = < > += -= \*= /= %= ˆ= &= |= << >> >>= <<= == != <= >= && || ++ -- , ->\* -> ( ) [ ]

Что не может быть перегружено:

:: .\* . ?:

1. Переопределение функции конвертации (user-defined conversion function)  
   operator *type*

Может быть использован любой тип, в том числе класс

1. Аллокация  
   operator new  
   operator new []
2. Деаллокация  
   operator delete  
   operator delete []
3. Переопределение литерала (user-defined literal)  
   operator “”

Специальные члены класса, или члены по умолчанию:

<http://www.cplusplus.com/doc/tutorial/classes2/>

Интерфейсы по умолчанию:

|  |  |
| --- | --- |
| [Default constructor](http://www.cplusplus.com/doc/tutorial/classes2/#default_constructor) | C::C(); |
| [Destructor](http://www.cplusplus.com/doc/tutorial/classes2/#destructor) | C::~C(); |
| [Copy constructor](http://www.cplusplus.com/doc/tutorial/classes2/#copy_constructor) | C::C (const C&); |
| [Copy assignment](http://www.cplusplus.com/doc/tutorial/classes2/#copy_assignment) | C& operator= (const C&); |
| [Move constructor](http://www.cplusplus.com/doc/tutorial/classes2/#move) | C::C (C&&); |
| [Move assignment](http://www.cplusplus.com/doc/tutorial/classes2/#move) | C& operator= (C&&); |

Условия, при которых специальные члены создаются:

|  |  |  |
| --- | --- | --- |
| **Member function** | **implicitly defined:** | **default definition:** |
| [Default constructor](http://www.cplusplus.com/doc/tutorial/classes2/#default_constructor) | if no other constructors | does nothing |
| [Destructor](http://www.cplusplus.com/doc/tutorial/classes2/#destructor) | if no destructor | does nothing |
| [Copy constructor](http://www.cplusplus.com/doc/tutorial/classes2/#copy_constructor) | if no move constructor and no move assignment | copies all members |
| [Copy assignment](http://www.cplusplus.com/doc/tutorial/classes2/#copy_assignment) | if no move constructor and no move assignment | copies all members |
| [Move constructor](http://www.cplusplus.com/doc/tutorial/classes2/#move) | if no destructor, no copy constructor and no copy nor move assignment | moves all members |
| [Move assignment](http://www.cplusplus.com/doc/tutorial/classes2/#move) | if no destructor, no copy constructor and no copy nor move assignment | moves all members |

Что делать если всё же хочется, чтобы был специальный член (например, хочется пустой конструктор, который не создался по причине невыполнения условий выше)? Использовать = default:

C::C(const C& other) = default; // Конструктор копирования

Что делать если хочется, чтобы у класса не было специального члена, когда условия выше выполняются (например, необходимо запретить копирование)?  
Использовать = delete:

C::C(const C& other) = delete; // Конструктор копирования

Задание "Дата"

* Создать класс Date, который представляет собой дату и содержит поля year, month, day. Для простоты считаем, что в каждом месяце 30 дней.
* Перегрузить операторы ==, <, >, <=, >=
* Перегрузить оператор =
* Перегрузить операторы +, - (и для Date, и для int)
* Перегрузить операторы ++, --
* Перегрузить операторы +=, -= (и для Date, и для int)
* Создать функцию interval, которая возвращает сколько дней разницы между двумя датами.

Опциональное задание:

* Создать класс, содержащий строку (можно скопировать из моего репозитория)
* Добавить перегрузку оператора &&, которая возвращает первое пересечение строк (например для «Heisenberg» и «Nurnburg» будет «nb», для «Aabccc» и «Aaxccc» будет «Aa»)
* Перегрузку оператора \*, которая возвращает символы, общие для строк, например, для «sure» и «absolute» будет «sue»