

## Курс: Объектно-ориентированное программирование на C++

### Встреча №6

#### ТЕМА: ПЕРЕГРУЗКА ОПЕРАТОРОВ

Задания для самостоятельной работы:

#### Задание 1

Создайте класс **Date**, который будет содержать информацию о дате (день, месяц, год).

С помощью механизма перегрузки операторов, определите операцию разности двух дат (результат в виде количества дней между датами), а также операцию увеличения даты на определенное количество дней.

#### Задание 2

Добавить в строковый класс функцию, которая создает строку, содержащую пересечение двух строк, то есть общие символы для двух строк.

Например, результатом пересечения строк «*sdqcg*» «*rgfas34*» будет строка «*sg*». Для реализации функции перегрузить оператор \* (бинарное умножение).

**Курс: Объектно-ориентированное  
программирование на C++****Встреча №7**

ТЕМА: ПЕРЕГРУЗКА ИНКРЕМЕНТА И ДЕКРЕМЕНТА.  
ПЕРЕГРУЗКА ЛОГИЧЕСКИХ ОПЕРАТОРОВ.  
ВОЗВРАТ ПО ССЫЛКЕ. ПЕРЕГРУЗКА ОПЕРАТОРА  
ПРИСВАИВАНИЯ

Задания для самостоятельной работы:

**Задание 1**

Создайте класс *Circle* (окружность):

- Реализуйте через перегруженные операторы;
- Проверка на равенство радиусов двух окружностей (*операция* ==);
- Сравнения длин двух окружностей (*операция* >).

Пропорциональное изменение размеров окружности, путем изменения ее радиуса (*операция* += и -=)

**Задание 2**

Создать класс *Airplane* (самолет)

С помощью перегрузки операторов реализовать:

- Проверка на равенство типов самолетов (*операция* ==)
- Увеличение и уменьшение пассажиров в салоне самолета (*операции* ++ и -- в префиксной форме)
- Сравнение двух самолетов по максимально возможному количеству пассажиров на борту (*операция* >)

## Курс: Объектно-ориентированное программирование на C++

### Встреча №8

#### ТЕМА: КОНСТРУКТОР ПЕРЕНОСА

Задания для самостоятельной работы:

#### Задание 1

Добавить в уже существующий класс *Student* конструктор переноса. Память под поля класса выделять динамически.

#### Задание 2

Добавить в уже существующий проект многоквартирного дома с класса Дом, Квартира, Человек конструктор переноса в те классы, где это необходимо. Память под поля класса выделять динамически.

## Курс: Объектно-ориентированное программирование на C++

### Встреча №9

ТЕМА: ФУНКЦИИ-ЧЛЕНЫ, ГЕНЕРИРУЕМЫЕ АВТОМАТИЧЕСКИ, КЛЮЧЕВЫЕ СЛОВА `DEFAULT` И `DELETE` ПО ОТНОШЕНИЮ К КЛАССАМ, ПЕРЕГРУЗКА `[]` И `()`, ПЕРЕГРУЗКА ОПЕРАТОРОВ ПРЕОБРАЗОВАНИЯ ТИПА

Задания для самостоятельной работы:

#### Задание 1

В ранее созданный класс *Array* добавьте перегрузку `[]`, `()`, преобразования типов к *int* и *char\**:

- `[]` – возвращает элемент по указанному индексу
- `()` – увеличивает все элементы массива на указанную величину

Преобразование к *int* возвращает сумму элементов массива

Преобразование к *char\** возвращает значения элементов массива в виде строки

#### Задание 2

Реализовать класс *var* - «Универсальная переменная». Объекты экземпляры этого класса способны хранить в себе значения следующих типов: *int*, *double*, *String* (строка – уже готовый класс).

**Например:**

```
void main()
{
var a = 15;
var b = "Hello";
var c = 7.8;
var d = "50";
b = a + d;
b.Show();           // Выведет 65
if (a == b) cout << "Equal\n";
else cout << "Not Equal\n";
}
```

Для класса **var** необходимо:

- перегрузить арифметические операторы: +, -, \*, /, +=, -=, \*=, /=
- и операторы сравнения: <, >, <=, >=, ==, !=

(Вся перегрузка осуществляется как функции-члены класса **var**)

В переменной типа **var** в любой момент времени может храниться только одно значение (или **int**, или **double**, или **String**).

При перегрузке операторов учитывать следующее: если в операции типы значений двух операндов типа **var** разные, то необходимо приводить (конвертировать) значение из второго операнда к типу значения первого операнда и только после этого выполнять операцию. (При этом значение (и тип) второго операнда в самом объекте **var** не меняется)

**Например:**

```
var a = 10, b = "120", c;
c = a + b;
c.Show(); //Выведет 130
c = b + a;
c.Show(); //Выведет "12010"
```

Для типа **Строка операция** \* должна возвращать новую стро-

ку, состоящую только из символов первой строки, которые есть во второй строке, **например**:

```
var a = "Microsoft", b = "Windows", c;  
c = a * b;  
c.Show(); //Выведет "ioso"
```

Для типа **Строка операция** / должна возвращать новую строку, состоящую из символов первой строки, которых нет во второй строке, **например**:

```
var a = "Microsoft", b = "Windows", c;  
c = a / b;  
c.Show(); //Выведет "Mcrft"
```

Аналогично для Строки \*= и /=

Добавить в класс **var** преобразования к типам: *int, double, char\**

**Курс: Объектно-ориентированное  
программирование на C++****Встреча №10****ТЕМА: ПЕРЕГРУЗКА ОПЕРАТОРОВ ГЛОБАЛЬНЫМИ  
ФУНКЦИЯМИ, ДРУЖЕСТВЕННЫЕ ФУНКЦИИ,  
ДРУЖЕСТВЕННАЯ ПЕРЕГРУЗКА,  
ПЕРЕГРУЗКА ВВОДА-ВЫВОДА**

Задания для самостоятельной работы:

**Задание 1**

Создайте класс «Комплексное число» - *Complex*. Класс должен содержать несколько конструкторов.

Класс должен иметь возможность вводить и выводить комплексные числа посредством перегруженных операций  $>>$  и  $<<$ .

Перегрузите операции  $+$ ,  $-$ ,  $!=$ ,  $==$ ,  $()$ .

Используйте обычную и дружественную перегрузку.

**Задание 2**

Создайте класс с именем *Time* для хранения времени.

В классе должна быть функция-член, которая даёт приращение времени, хранящееся в объекте *Time*, равное 1 секунде.

Напишите соответствующие конструкторы и функции-члены. В классе должны быть перегружены операции  $++$ ,  $--$ ,  $!=$ ,  $==$ ,  $>$ ,  $<$ ,  $>>$ ,  $<<$ ,  $=$ ,  $+=$ ,  $-=$ ,  $()$ .

Используйте обычную и дружественную перегрузку.