

Курс
«Объектно-ориентированное
программирование на C++»

Встреча №6

***Тема:** Перегрузка операторов*

Задание 1.

Создайте класс Дробь (или используйте уже ранее созданный вами). Используя перегрузку операторов реализуйте для него арифметические операции для работы с дробями (операции $+$, $-$, $*$, $/$).

Задание 2.

Создайте класс Complex (комплексное число) или используйте уже созданный вами класс. Создайте перегруженные операторы для реализации арифметических операций для по работе с комплексными числами (операции $+$, $-$, $*$, $/$).

Встреча №7

Тема: Перегрузка инкремента и декремента. Перегрузка логических операторов. Возврат по ссылке. Перегрузка оператора присваивания

Задание 1.

Создать класс *Overcoat* (верхняя одежда).

Реализовать перегруженные операторы:

1. Проверка на равенство типов одежды (операция `=`).
2. Операцию присваивания одного объекта в другой (операция `=`).
3. Сравнение по цене двух пальто одного типа (операция `>`).

Задание 2.

Создать класс *Flat* (квартира).

Реализовать перегруженные операторы:

1. Проверка на равенство площадей квартир (операция `==`).
2. Операцию присваивания одного объекта в другой (операция `=`).
3. Сравнение двух квартир по цене (операция `>`).

Встреча №8

Тема: Конструктор переноса

Задание 1.

Добавить в уже существующий класс *String* конструктор переноса.

Задание 2.

Добавить в уже существующий класс *Array* конструктор переноса.

Встреча №9

Тема: *Функции-члены, генерируемые автоматически, ключевые слова `default` и `delete` по отношению к классам, перегрузка `[]` и `()`, перегрузка операторов преобразования типа*

Задание.

В ранее созданный класс *String* добавьте перегрузку `[]`, `()`, преобразования типа к *int*:

- `[]` — возвращает элемент по указанному индексу;
- `()` — ищет символ в строке, если символ есть возвращает индекс, если нет `-1`.

Преобразование к *int* возвращает длину строки.

Встреча №10

Тема: Перегрузка операторов глобальными функциями, дружественные функции, дружественная перегрузка, перегрузка ввода-вывода

Задание.

Создайте класс с именем *Date* для хранения даты (или используйте ранее созданный).

В классе должна быть функция-член, которая увеличивает день на 1.

Напишите соответствующие конструкторы и функции-члены.

В классе должны быть перегружены операции ++, --, !=, ==, >, <, >>, <<, =, +=, -=, ().

Используйте обычную и дружественную перегрузку.