

## Лабораторная работа №3

### Наследование, статические члены, перегрузка операторов

#### Контрольные вопросы:

1. Что такое перегрузка функций? Как компилятор определяет какая функция вызывается?
2. Зачем был введен nullptr и в чём его отличие от NULL?
3. Что такое конструктор копирования? В каких случаях он объявляется явно?
4. Что такое ссылки и в чём их отличие от указателей? В каких случаях надо применять ссылки, а в каких указатели?
5. Зачем используются ссылки как параметры функции? А в качестве возвращаемого значения?
6. Что такое статические поля и методы класса? Зачем они используются?
7. Что такое перегрузка операторов и какие её виды бывают? Как она объявляется? Какие операторы нельзя перегружать?
8. Что такое константные методы класса и для чего они нужны?
9. Что такое наследование и для чего оно нужно?
10. Что такое суперкласс? Подкласс? Базовый класс?
11. В каком порядке при наследовании вызываются конструкторы и деструкторы?
12. Какие типы наследования бывают?
13. Что такое переопределение метода и зачем оно используется?
14. Зачем нужно ключевое слово auto?

#### Задание

В данной лабораторной работе необходимо изменить класс, созданный в лабораторной работе №2 следующим образом:

1. Добавить для методов параметры-ссылки там, где это необходимо.
2. Добавить константные методы и параметры там, где это оправданно.
3. Заменить методы, реализующие операции над сущностью, перегруженными операторами (соответствующими смыслу операции). Например операция add() заменяется оператором +.
4. Добавить статический метод для класса, возвращающий новый объект, заполненный случайными значениями из заданного диапазона (который задаётся через параметры данного метода). Для бинарных значений диапазон не указывается.
5. Добавить наследование от базового класса Object, в котором должна быть реализация следующего функционала:
  - a. Подсчёт количества созданных объектов и количества активных объектов на данный момент через статические поля класса. Статический метод printTotalInfo для вывода этой информации в консоль.
  - b. Функционал для создания **динамического** массива – списка всех выполненных операций с объектом. При выполнении каждой операции её обозначение заносится в данный массив через метод addOp. Метод clearOp – очищает массив операций. Метод printOp – выводит полный список всех выполненных операций с объектом.
  - c. Конструктор копии и оператор присваивания (=) для класса Object.

6. Дополнить класс юнит-тестирования **минимум 5 тестами** для проверки работы базового класса Object и статического метода для случайной генерации класса сущности.

**Важно:** при перегрузке оператора присваивания (=) необходимо возвращать ссылку на текущий объект (\*this), чтобы можно было их использовать в цепочке вызовов. Также стоит всегда проверять на самоприсваивание (особенно при наличии выделения динамической памяти), и не выполнять лишних действий, сразу возвращая ссылку на текущий объект.