# ПРАКТИЧЕСКАЯ РАБОТА №5
# ПЕРЕГРУЗКА ФУНКЦИЙ

## ЦЕЛЬ ПРАКТИЧЕСКОЙРАБОТЫ:

Целью данной практической работы является приобретение практических навыков по программированию перегрузки функций на языке программирования C++.

## ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ:

Под перегрузкой функции понимается, определение нескольких функций (две или больше) с одинаковым именем, но различными параметрами. Наборы параметров перегруженных функций могут отличаться порядком следования, количеством, типом. Таким образом перегрузка функций нужна для того, чтобы избежать дублирования имён функций, выполняющих сходные действия, но с различной программной логикой.

Пример 1.

Создадим простую функцию max, которая будет определять максимальное из двух целых чисел.

```
1)/* Функция max для целых чисел */
2)int max(int num1, int num2)
3){
4)if (num1 > num2)
       a.return num1;
5)return num2;
6)}
```

В эту функцию мы можем передавать только целочисленные параметры. Для того, чтобы сделать аналог этой функции для чисел с плавающей запятой, выполним перегрузку этой функции:

```
1)/* Функция max для дробных чисел */
2)double max(double num1, double num2)
3){
4)if (num1 > num2)
       a.return num1;
5)return num2;
```

```
6) }
```

Теперь, когда мы будет вызывать функцию max с целыми параметрами, то вызовется первая функция. А если с дробными — то вторая. Например:

```
1) //Здесь будет использоваться первый вариант функции
   max
2) int imax = max(1, 10);
3) // А здесь - второй
4) double dmax = max(1.0, 20.0);
```

Пример 2.

Рассмотрим функцию areaRectangle(), которая вычисляет площадь прямоугольника.

```
1) float areaRectangle(float, float)
2) //функция, вычисляющая площадь прямоугольника с двумя
   параметрами a(см) и b(см)
3) {
4) return a * b;
5) // умножаем длинны сторон прямоугольника и возвращаем
   полученное произведение
6) }
```

Итак, это функция с двумя параметрами типа float, причём аргументы передаваемые в функцию должны быть в сантиметрах, возвращаемое значение типа float — тоже в сантиметрах.

Предположим, что наши исходные данные (стороны прямоугольника) заданы в метрах и сантиметрах, например такие: a = 2м 35 см; b = 1м 86 см. В таком случае, удобно было бы использовать функцию с четырьмя параметрами. То есть, каждая длинна сторон прямоугольника передаётся в функцию по двум параметрам: метры и сантиметры.

```
1) float areaRectangle(int a_m, float a_sm, int
   b_m, float b_sm)
2) // функция, вычисляющая площадь прямоугольника с 4-мя
   параметрами a(м) a(см); b(м) b(см)
3) {
```

```
4) return (a_m * 100 + a_sm) * (b_m * 100 + b_sm);
5) }
```

В теле функции значения, которые передавались в метрах (a_m и b_m) переводятся в сантиметры и суммируются с значениями a_sm b_sm, после чего перемножаем суммы и получаем площадь прямоугольника в см. Конечно же можно было перевести исходные данные в сантиметры и пользоваться первой функцией, но сейчас не об этом.

Теперь, самое главное – у нас есть две функции, с разной сигнатурой, но одинаковыми именами (перегруженные функции). Сигнатура – это комбинация имени функции с её параметрами. Как же вызывать эти функции? А вызов перегруженных функций ничем не отличается от вызова обычных функций, например:

```
1) areaRectangle( 32, 43);
2) // будет вызвана функция, вычисляющая площадь
   прямоугольника с двумя параметрами a(см) и b(см)
3) areaRectangle( 4, 43, 2, 12);
4) // будет вызвана функция, вычисляющая площадь
   прямоугольника с 4-мя параметрами a(м) a(см); b(м)
   b(см)
```

Как видите, компилятор самостоятельно выберет нужную функцию, анализируя только лишь сигнатуры перегруженных функций. Минуя перегрузку функций, можно было бы просто объявить функцию с другим именем, и она бы хорошо справлялась со своей задачей. Но представьте, что будет, если таких функций надо больше, чем две, например 10. И для каждой нужно придумать осмысленное имя, а сложнее всего их запомнить. Вот именно поэтому проще и лучше перегружать функции, если конечно в этом есть необходимость.

## ВАРИАНТЫ ЗАДАНИЙ

1) Реализовать сортировку пузырьком для целых чисел, а затем перегрузить её для дробных.
2) Реализовать сортировку выбором для целых чисел, а затем перегрузить её для дробных.

3) Реализовать сортировку вставками для целых чисел, а затем перегрузить её для дробных.
4) Реализовать программу-калькулятор, работающий с различными типами данных