

# Лабораторная работа №1

## Моделирование и алгоритмизация как средства проектирования программного обеспечения

1. Описать блок-схему алгоритма согласно варианта (таблица 1).
2. На основе описанной блок-схемы реализовать алгоритм средствами языка *Python*.
3. Протестировать разработанный алгоритм при помощи модульных тестов.
4. Создать консольное приложение для демонстрации работы алгоритма.
5. Составить отчет о проделанной работе.

Таблица 1 - Варианты задания

Вариант	Условие задачи
1	Быстрая сортировка
2	Блочная сортировка
3	Поразрядная сортировка
4	Троичный поиск
5	Двоичный поиск
6	Числа Фибоначчи
7	Алгоритм сортировки пузырьком
8	Сортировка выбором
9	Сортировка слиянием
10	Сортировка подсчетом
11	Экспоненциальный поиск
12	Двоичный поиск
13	Сортировка перемешиванием
14	Пирамидальная сортировка
15	Алгоритм КМП

## Дополнительно

Для составления блок-схемы можно использовать сервис [diagrams.net](https://diagrams.net).

Программа должна быть грамотно разбита на функции. В функции, реализующей заданный алгоритм, должен быть **ТОЛЬКО АЛГОРИТМ**. Т.е. в ней не должно быть ни пользовательского ввода, ни чтения из файла, ни вывода результатов на консоль. Дан

ные для обработки должны передаваться в функцию в качестве входного параметра и функция должна возвращать результат обработки.

Ввод и вывод данных должны быть реализованы в отдельных функциях.

Пример:

```
def input_data() -> list[int]:  
    # пользовательский ввод или чтение из файла  
    data = ...  
    return data  
  
def processing(data: list[int]) -> list[int]:  
    # обработка данных  
    return result  
  
def output_result(result: list[int]) -> None:  
    # вывод данных на консоль или в файл
```

Для реализации модульных тестов в *Python* можно использовать библиотеку [pytest](#). Сравнение фактических и ожидаемых результатов выполняется с помощью оператора *assert*:

```
def func_test():  
    actual_data = func(test_data)  
    assert actual_data == expect_data
```