

Санкт-Петербургский Национальный Исследовательский Университет  
Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий и систем связи

**Лабораторная работа №2**

**Вариант №1**

Выполнил(и:)

Алексеев Т.Ю.

Оншин Д.Н.

Проверил

Мусаев А.А.

Санкт-Петербург,

2022

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1. ЗАДАНИЕ 1.....	4
2. ЗАДАНИЕ 2.....	5
3. ЗАДАНИЕ 3.....	6
ЗАКЛЮЧЕНИЕ.....	7
СПИСОК ЛИТЕРАТУРЫ.....	8

## **ВВЕДЕНИЕ**

Целью данной работы являлось знакомство с алгоритмами бинарного поиска и работа со словарями.

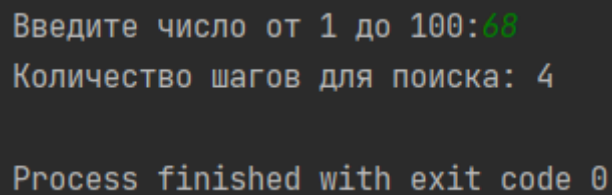
Для достижения данной цели необходимо было выполнить следующие задания:

1. Задание 1: создание программы для бинарного поиска, результатом которой должно быть количество шагов необходимых для нахождения числа.
2. Задание 2: создание словаря, содержащего описание каждого студента, а также создание программы, считывающей ответы на вопросы и отгадывающей студентов.
3. Задание 3: составление графа для Задания 2.

Решения данных задач будут находиться на GitHub по ссылке:  
[https://github.com/NorthPole0499/Algoritms\\_task\\_2](https://github.com/NorthPole0499/Algoritms_task_2)

## 1. ЗАДАНИЕ 1

В данной программе реализуется алгоритм бинарного поиска. Пользователь подаёт на вход любое число от 1 до 100, которое программа будет угадывать. Запускается цикл `while`, работающий до тех пор, пока число не будет отгадано. Рассмотрим каждую итерацию. В переменную-счётчик прибавляется 1. После этого высчитывается переменная `center`, текущий центр диапазона. Если `center` совпадает с загаданным числом, то программа завершается. Если число больше, чем `center`, то левая граница становится равна `center + 1`. В противном случае правая граница становится `center - 1`. После завершения цикла выводится значение переменной `count`, количество шагов, необходимых для нахождения числа.



```
Введите число от 1 до 100: 68
Количество шагов для поиска: 4

Process finished with exit code 0
```

Приложение 1 – Результат алгоритма бинарного поиска

## 2. ЗАДАНИЕ 2

В задании 2 нужно реализовать программу, которая по определённым характеристикам будет угадывать студента. В начале программы были созданы списки, являющиеся теми или иными критериями, с именами студентов. Далее был создан словарь, в котором ключами являются характеристики студентов, а значениями являются списки с именами студентов, подходящих под эту характеристику. После описания всех студентов с помощью словаря программа начинает задавать пользователю различные вопросы, ответы на которые определяют нужные ключи для отгадывания студентов. Создается список, куда добавляются все нужные ключи. После чего находится ключ, значением которого является список (с именами студентов) минимальной длины. Это сделано для того, чтобы в дальнейшем сократить время поиска загаданного студента. Затем программа «бежит» по списку минимальной длины и списку собранных ранее ключей и осуществляет проверку, есть ли элемент первого списка (т.е. студент) в значении словаря по выбранному ключу или нет. Если этот студент присутствует во всех значениях по заданным ключам, то это и есть искомый студент.

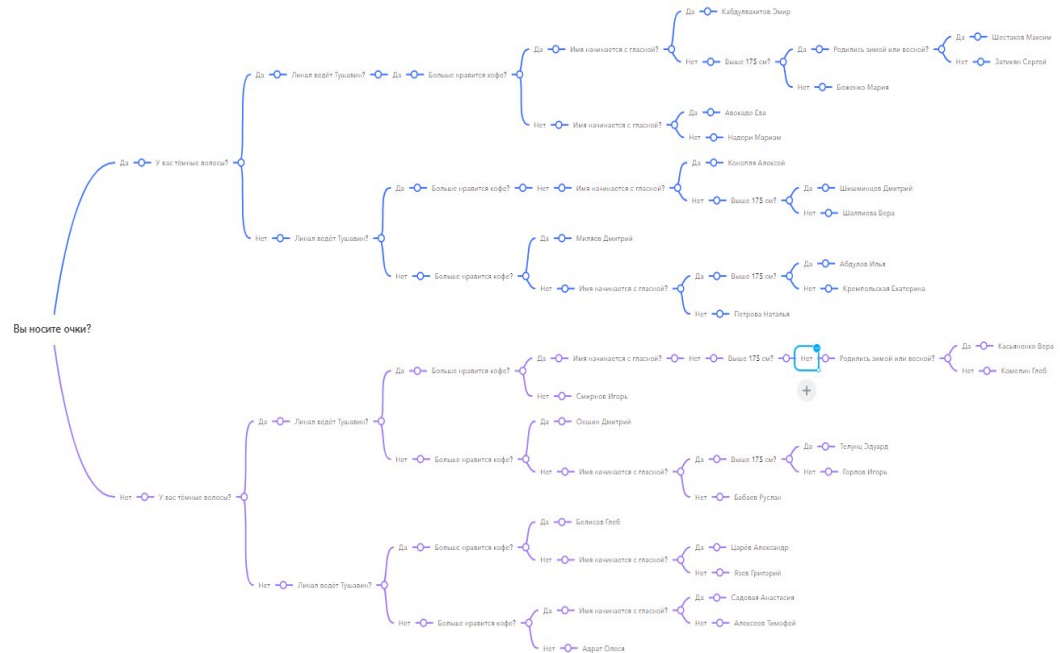
```
Студент носит очки?нет
У студента темные волосы?нет
У студента Тушавин?нет
Студент любит кофе?да
Полное имя студента начинается с гласной?нет
Студент выше 175 см?да
Др студента зимой/весной?нет
Алексеев Тимофей

Process finished with exit code 0
```

Приложение 2 – Пример работы программы из второго задания

### 3. ЗАДАНИЕ 3

В данном задании было необходимо составить граф для задания 2. В нашем случае для описания логики программы из задания 2 мы решили построить бинарное дерево.



Приложение 3 – Бинарное дерево, описывающее логику решения задания 2

## **ЗАКЛЮЧЕНИЕ**

Таким образом, в ходе выполнения данной лабораторной работы были достигнуты все поставленные цели. Произошло знакомство с алгоритмами бинарного поиска и работа со словарями и их методами. А также был построен граф, описывающий логику программы.

## **СПИСОК ЛИТЕРАТУРЫ**

- 1) Словари (dict) и работа с ними. Методы словарей // URL:  
<https://pythonworld.ru/tipy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovarej.html> (дата обращения: 19.10.2022)