МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
 федеральное государственное бюджетное образовательное учреждение

высшего образования

АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ЦИФРОВЫХ ТЕХНОЛОГИЙ, ЭЛЕКТРОНИКИ И ФИЗИКИ

Кафедра информационной безопасности

«РЕШЕНИЕ ЗАДАНИЯ № 6»

Отчет по лабораторной работе по дисциплине

«Технологии и методы программирования»

Выполнил студент

2 курса, 5.107-1 группы

Кулеба Никита Александрович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Проверил к.ф.-м.н., доцент,

Лепендин Андрей Александрович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Барнаул 2023

Вариант 11

В данной работе вам нужно будет реализовать простую последовательную структуру данных, в которой возможен быстрый поиск элементов.

**Цель работы:** реализация алгоритма из лабораторного задания № 6

Код программы

from uuid import uuid1

class ListContainer:

array: list

def \_\_init\_\_(self, array: list=[]) -> None:

self.array = list(map(lambda item: (uuid1().int, item), array.copy()))

def \_\_repr\_\_(self) -> str:

return f"{self.array}"

def \_\_str\_\_(self) -> str:

return f"{self.array}"

def \_\_len\_\_(self) -> int:

return len(self.array)

def \_\_getitem\_\_(self, index: int):

return self.array[index]

def \_\_iter\_\_(self):

return iter(self.array)

def insert(self, item):

self.array.append((uuid1().int, item))

def \_\_delitem\_\_(self, index: int):

del self.array[index]

def \_\_contains\_\_(self, item\_id):

st = 0

fin = len(self.array) - 1

while st <= fin:

mid = (st + fin) // 2

mid\_val = self.array[mid][0]

if mid\_val == item\_id:

return True

if mid\_val > item\_id:

fin = mid - 1

else:

st = mid + 1

return False

def find(self, item\_id: int) -> int:

st = 0

fin = len(self.array) - 1

while st <= fin:

mid = (st + fin) // 2

mid\_val = self.array[mid][0]

if mid\_val == item\_id:

return mid

if mid\_val > item\_id:

fin = mid - 1

else:

st = mid + 1

return False

def \_\_delitem2\_\_(self, index: int) -> int:

st = 0

fin = len(self.array) - 1

IDD = self.array[index][0]

while st <= fin:

mid = (st + fin) // 2

mid\_val = self.array[mid][0]

if mid\_val == IDD:

del self.array[index]

if mid\_val > IDD:

fin = mid - 1

else:

st = mid + 1

return False

Мы реализовали простую последовательную структуру данных, в которой возможен быстрый поиск элементов.