

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

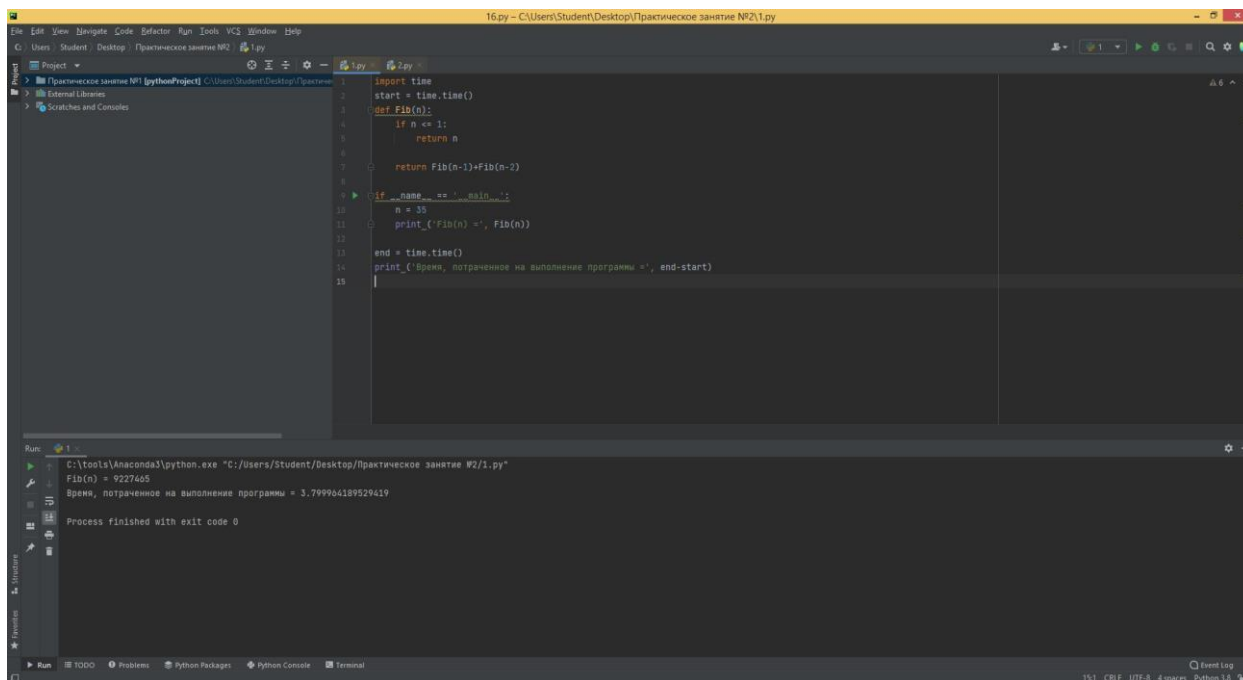
**Отчет по практическому занятию № 2
по дисциплине «Алгоритмизация»**

Выполнил студент группы ИВТ-б-о-22-1
Гончаров Даниил Ростиславович «23» октября 2023г.
Подпись студента _____
Работа защищена « » _____ 20__ г.
Преподаватель Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Ход работы:

1. Использование наивного алгоритма для вычисления числа Фибоначчи:

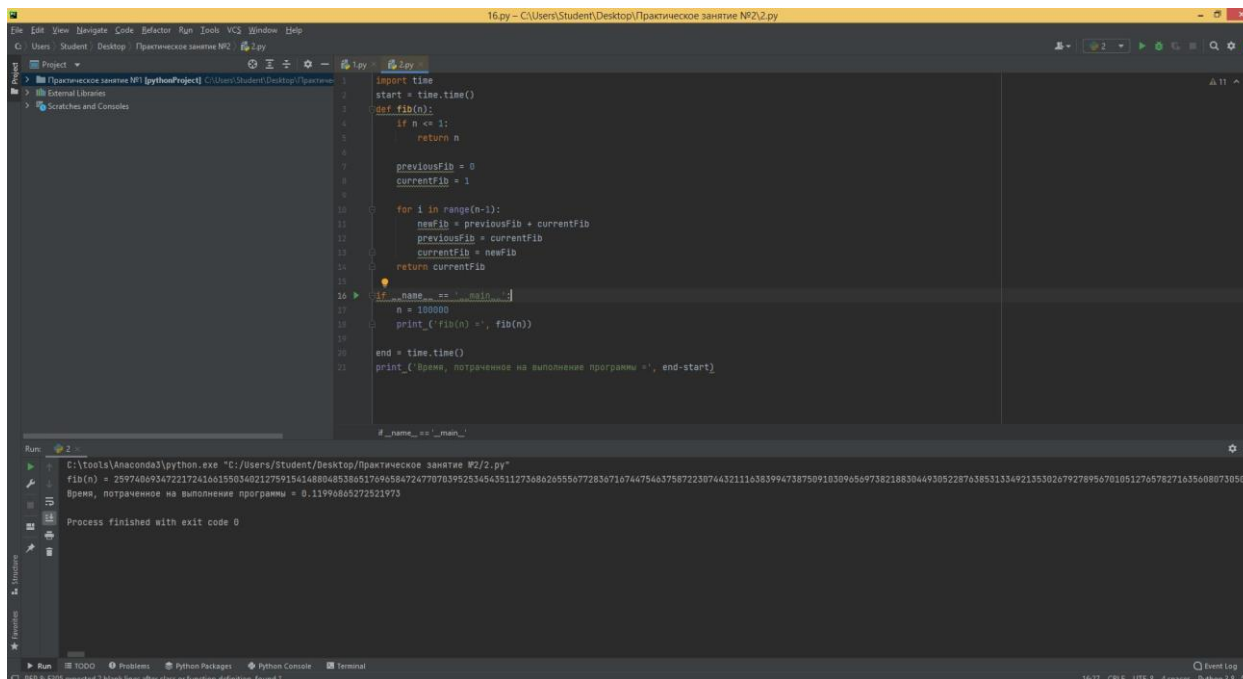


```
1 import time
2 start = time.time()
3 def Fib(n):
4     if n <= 1:
5         return n
6     return Fib(n-1)+Fib(n-2)
7
8
9 if __name__ == '__main__':
10     n = 35
11     print('Fib(n) = ', Fib(n))
12
13 end = time.time()
14 print('Время, потраченное на выполнение программы =', end-start)
15
```

Run: C:\Tools\Anaconda3\python.exe "C:/Users/Student/Desktop/Практическое занятие W2/1.py"
Fib(n) = 9227465
Время, потраченное на выполнение программы = 3.799964189529419
Process finished with exit code 0

Рисунок 1 – Код наивного алгоритма

2. Использование усовершенствованного алгоритма для вычисления числа Фибоначчи:



```
1 import time
2 start = time.time()
3 def fib(n):
4     if n <= 1:
5         return n
6     previousFib = 0
7     currentFib = 1
8
9     for i in range(n-1):
10         newFib = previousFib + currentFib
11         previousFib = currentFib
12         currentFib = newFib
13     return currentFib
14
15 if __name__ == '__main__':
16     n = 10000
17     print('fib(n) = ', fib(n))
18
19 end = time.time()
20 print('Время, потраченное на выполнение программы =', end-start)
21
22 # __name__ == '__main__'
```

Run: C:\Tools\Anaconda3\python.exe "C:/Users/Student/Desktop/Практическое занятие W2/2.py"
fib(n) = 25974069347221724166155034021275915414880485386517696584724770783952534535112736862695567728367167447546375872230744321116383994738750918309656973821883044930522876385313349213530267927895670105127657827163560807305053
Время, потраченное на выполнение программы = 0.11996865272521973
Process finished with exit code 0

Рисунок 2 – Код усовершенствованного алгоритма

3. Построение графиков зависимости времени выполнения программы от числа Фибоначчи:

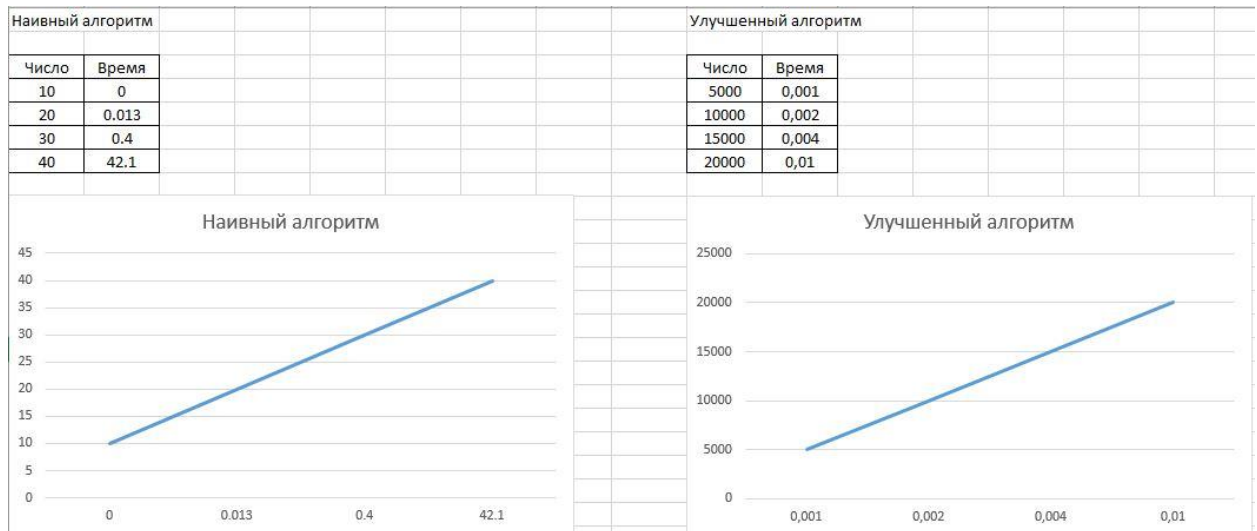


Рисунок 3 – Графики зависимости времени выполнения программы

Вывод: При выполнении работы я сделал вывод, что при использовании усовершенствованного алгоритма значительно сокращается время выполнения программы, что позволяет вычислить гораздо большие значения числа Фибоначчи.