**Лабораторная работа №2 «**[**Объектно-ориентированное программирование**](file:///D:\diskd_Учеба\!_УМК\!_УМКД_бакал_2021_22уч.г\ВТ_Современные%20методы%20разработки%20ПО_5%20сем\Lr_классы.pdf)».

***Цель работы:*** приобретение практических навыков по созданию программных приложений с использованием объектно-оринтированной парадигмы.

Ход работы:

import time

class StringData:

def \_\_init\_\_(self, max\_length):

self.max\_length = max\_length

self.data = [0] \* (max\_length + 3) # Дополнительные 3 элемента для реальной длины и времени создания

self.data[0] = 3 # Начальное значение реальной длины

self.creation\_time = None

def set\_string(self, new\_string):

if len(new\_string) > self.max\_length:

print("Строка слишком длинная, обрезаем до максимальной длины.")

new\_string = new\_string[:self.max\_length]

real\_length = len(new\_string) # Получаем реальную длину строки

self.data[1:real\_length + 1] = map(ord, new\_string) # Записываем ASCII-коды символов

self.data[self.max\_length + 1] = real\_length # Устанавливаем реальную длину строки

self.data[self.max\_length + 2] = time.time() # Записываем время создания

def get\_string(self):

real\_length = self.data[self.max\_length + 1]

return ''.join(chr(code) for code in self.data[1:real\_length + 1])

def find\_substring(self, substring):

full\_string = self.get\_string()

if substring in full\_string:

return f"Подстрока найдена в позиции {full\_string.find(substring)}"

else:

return "Подстрока не найдена"

# Пример использования

my\_string = StringData(20)

my\_string.set\_string("Hello, World!")

print(my\_string.get\_string())

substring\_result = my\_string.find\_substring("World")

print(substring\_result)

