Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

*Факультет программной инженерии и компьютерной техники*

Лабораторная работа №4

*Исследование протоколов, форматов обмена информацией и языков разметки документов*

По дисциплине

*Информатика*

Вариант №15

Выполнил студент группы P3112:

Степутенко Илья Сергеевич

Преподаватель:

Белозубов Александр Владимирович

Оглавление

**Элементы оглавления не найдены.**

Вариант заданий

|  |  |  |
| --- | --- | --- |
| Исходный формат | Результирующий формат | День недели |
| YAML | JSON | Среда |

Задание №1

Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый.

Выполнение задания №1

def yml\_parser(file: str) -> None:

with open(file) as inputfile:

d = {}

lkey = ""

while (s := inputfile.readline()):

s = s.strip()

if s[0] == "-":

d[lkey] += [s[2:]]

elif s[-1] == ":":

lkey = s[:-1]

d[lkey] = []

else:

lkey = ""

k = s.split(":")

d[k[0]] = k[1][1:]

with open(f"{file.split('.')[0]}\_1.json", "w+") as output:

output.write(str(d))

yml\_parser("schedule.yml")

Задание №2

1. Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
2. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.
3. Сравнить полученные результаты и объяснить их сходство/различие.

Выполнение задания №2

import yaml, json

def yml\_parser(file: str) -> None:

with open(file, 'r') as inputstream:

try:

tempdict = yaml.safe\_load(inputstream)

except yaml.YAMLError as exc:

tempdict = {}

with open(f"{file.split('.')[0]}\_2.json", "w", encoding="utf-8") as output:

json.dump(tempdict, output, ensure\_ascii=False, indent=4)

Различия:

1. Код больше не содержит циклы.
2. Исходный код не индентирует JSON-файл.
3. Исходный код не имеет зависимостей, так как не подключает дополнительные библиотеки.

Сходства:

1. Как исходный, так и переписанный код успешно выполняют конвертацию.
2. Оба кода требуют открытия файла.

Задание №3

1. Переписать исходный код, добавив в него использование регулярных выражений.
2. Сравнить полученные результаты и объяснить их сходство/различие.

Выполнение задания №3

import re

def yml\_parser(file: str) -> None:

with open(file) as inputfile:

d = {}

lkey = ""

while (s := inputfile.readline().strip()):

if re.match("^\s\*-.\*", s):

d[lkey] += [re.findall("^\s\*-\s\*(.\*)", s)[0]]

elif re.match("^.\*:\s\*$", s):

lkey = s

d[lkey] = []

else:

key, data = re.findall("^(.\*): (.\*)", s)[0]

d[key] = data

with open(f"{file.split('.')[0]}\_3.json", "w+") as output:

output.write(str(d))

yml\_parser("schedule.yml")

Различия и сходства:

1. В отличие от кода из задания №2 и подобно коду из задания №1, код из задания №3 не индентирует JSON-файл.
2. В отличие от кода из задания №1 и подобно коду из задания №2, код из этого задания содержит зависимости в связи с использованием дополнительных библиотек.

Задание №4

1. Используя свою исходную программу из обязательного задания, программу из дополнительного задания №1 и программу из дополнительного задания №2, сравнить стократное время выполнения парсинга + конвертации в цикле.
2. Проанализировать полученные результаты и объяснить их сходство/различие.

Выполнение задания №4

from task1 import yml\_parser as parser\_1

from task2 import yml\_parser as parser\_2

from task3 import yml\_parser as parser\_3

from timeit import timeit

class Tests:

f = "schedule.yml"

def test\_1():

parser\_1(Tests.f)

def test\_2():

parser\_2(Tests.f)

def test\_3():

parser\_3(Tests.f)

print(timeit(Tests.test\_1, number=100))

print(timeit(Tests.test\_2, number=100))

print(timeit(Tests.test\_3, number=100))

Пример вывода программы из задания №4