

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
По учебной практике
Тема: Проект Moodle

Студент гр. 0303

Калмак Д.А.

Руководитель

Шевская Н.В.

Санкт-Петербург

2022

ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Калмак Д.А.

Группа 0303

Тема практики: Проект Moodle

Задание на практику:

Ознакомиться с библиотеками pandas и scikit-learn. Составить задачу на Moodle с использованием библиотеки pandas и типом данных DataFrame, в которой считывается набор данных и выполняется сортировка с заданным условием этого набора данных.

Протестировать задачи других студентов на наличие багов.

При необходимости исправить собственную задачу.

Сроки прохождения практики: 29.06.2022 – 12.07.2022

Дата сдачи отчета: 10.07.2022

Дата защиты отчета: 10.07.2022

Студент

Калмак Д.А.

Руководитель

Шевская Н.В.

АННОТАЦИЯ

Во время учебной практики были получены умения работать с Moodle, составлять, редактировать задачи с правильным условием, а также с правильной системой проверки. Была создана задача, где используется библиотека pandas, pandas-datareader и тип данных DataFrame. Условием задачи была сортировка считанных данных. Были протестированы работы двух студентов. В одной работе была найдена недоработка. Во второй работе замечаний не было. Исправление собственной работы не потребовалось.

СОДЕРЖАНИЕ

	Введение	5
1.	Реализация собственной задачи	6
1.1.	Условие задачи	6
1.2.	Реализация задачи	6
2.	Тестирование задачи другого человека	8
2.1.	Тестирование задачи первого человека	8
2.2.	Тестирование задачи второго человека	9
3.	Исправление недочетов в собственной задаче	10
3.1.	Недочеты и ошибки в собственной задаче.	10
	Заключение	11
	Список использованных источников	12
	Приложение А. Код системы проверки.	13

ВВЕДЕНИЕ

Цель.

Познакомиться с Moodle, библиотекой pandas, библиотекой scikit-learn. Составить и реализовать задачу на Moodle. Протестировать задачи других пользователей.

Задачи.

- Ознакомиться с Moodle, узнать, как устроена система проверки кода ученика.
- Ознакомиться с библиотеками pandas и scikit-learn.
- Составить условие задачи на Moodle и реализовать ее с верной системой проверки кода.
- Протестировать задачи других пользователей.

1. РЕАЛИЗАЦИЯ СОБСТВЕННОЙ ЗАДАЧИ

1.1. Условие задачи

Вы стали акционером британской страховой компании Aviva plc (AV.UK) и для мониторинга решили использовать финансовый сайт Stooq (stooq). Реализуйте функцию solve(), которой на вход подано начало периода рассмотрения данных по вашим акциям, и вам необходимо прочитать набор данных с заданного начала периода по сегодняшний день, произвести быструю сортировку по убыванию по столбцам цены открытия интервала (Open) и цены закрытия интервала (Close).

([https://analyzingalpha.com/open-high-low-close-stocks#Open and Close Prices](https://analyzingalpha.com/open-high-low-close-stocks#Open_and_Close_Prices))

Отсортированный набор данных необходимо вернуть.

Замечание: обозначение Aviva plc на сайте Stooq: AV.UK

Для удобства используйте datetime. Подробнее здесь:

<https://docs.python.org/3/library/datetime.html>

Ознакомиться с используемой функцией и библиотекой для получения данных можно по ссылкам: [Сортировка](#) и [Библиотека](#)

1.2. Реализация задачи

```
import pandas_datareader.data as web
from datetime import datetime

def solve(start):
    end = datetime.today()
    f = web.DataReader('AV.UK', 'stooq', start=start, end=end)
    f = f.sort_values(by=['Open', 'Close'], ascending=False)
    return f
```

Решения студентов сверяются с результатами функции solve(). Система проверки учитывает правильное использование: datetime, нужной функции сортировки, аргументов сортировки. Если студент все учел, но получил

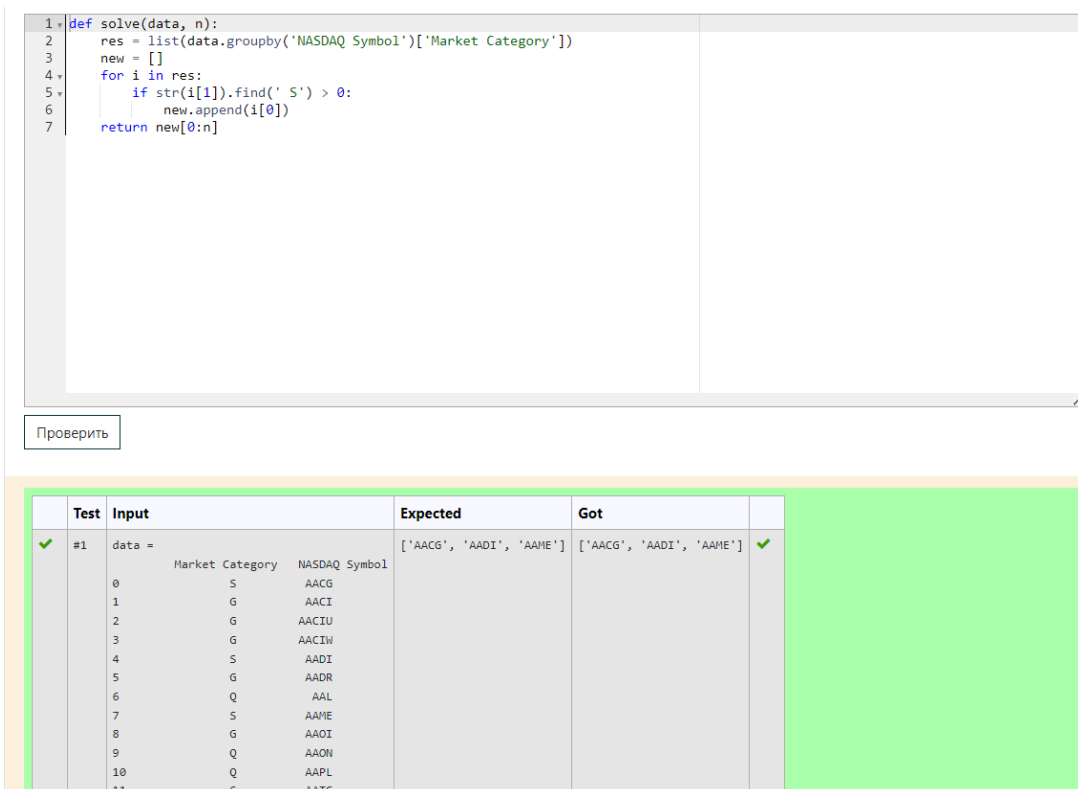
неправильный ответ, то система проверки выведет сообщение с тестом и правильным ответом. Если студент не импортирует библиотеки, то система проверки об этом сообщит.

2. ТЕСТИРОВАНИЕ ЗАДАЧИ ДРУГОГО ЧЕЛОВЕКА

2.1. Тестирование задачи первого человека

Тестирование работы студента группы 0303 Архипова Вадима.

1. По условию требовалась группировка по параметру 'Market Category', но это можно обойти и группировать по параметру 'NASDAQ Symbol'. (см. рис. 1)



```
1 def solve(data, n):
2     res = list(data.groupby('NASDAQ Symbol')['Market Category'])
3     new = []
4     for i in res:
5         if str(i[1]).find('S') > 0:
6             new.append(i[0])
7     return new[0:n]
```

Test	Input	Expected	Got	
✓ #1	data = Market Category NASDAQ Symbol 0 S AACG 1 G AACI 2 G AACIU 3 G AACIN 4 S AADI 5 G AADR 6 Q AAL 7 S AAME 8 G AAOI 9 Q AAOH 10 Q AAPL 11 S AATC	['AACG', 'AADI', 'AAME']	['AACG', 'AADI', 'AAME']	✓

Рисунок 1 – Тест

2. Попытка решить задачу путем подборов тестов удалась в силу условия составления системы проверки, которая должна выводить правильные ответы. Однако у студента есть проверка использования метода группировки. Простой перебор тестов сделать не получится.

3. Подсказки, какие методы необходимо использовать для правильного решения в условии задания есть, поэтому изучив всю информацию, решить задание предоставляется возможным. Условие понятное. Больше обходных решений не было найдено.

4. Решение задания:


```
def solve(data, n):  
    res = list(data.groupby('Market  
Category').get_group('S')['NASDAQ Symbol'])[0:n]  
    return res
```

2.2. Тестирование задачи второго человека

Тестирование работы студента группы 0303 Бодунова Петра.

1. Понятное условие, в котором сразу рассказано, что надо использовать, и приложена справочная информация.
2. Свобода импортирования библиотек, можно импортировать разными способами.
3. При тестировании ошибок не обнаружено.
4. Решение задания:

```
import numpy  
  
def solve(arr, ord):  
    return numpy.sort(arr, order=ord)
```

3. ИСПРАВЛЕНИЕ НЕДОЧЕТОВ В СОБСТВЕННОЙ ЗАДАЧЕ

3.1. Недочеты и ошибки в собственной задаче.

Недочетов и ошибок в собственной задаче не обнаружили.

ЗАКЛЮЧЕНИЕ

Таким образом, были изучены библиотеки pandas и scikit-learn. Проведена работа с Moodle, в которой были получены умения взаимодействовать с системой Moodle, составлять и редактировать задания, работать с настройкой задач, их системой тестирования. Была создана задача, где используется библиотека pandas, pandas-datareader и тип данных DataFrame. Проведена работа над грамотной системой тестирования. Были протестированы работы двух студентов. Исправление собственной работы не потребовалось.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Библиотека pandas <https://pandas.pydata.org/>
2. Библиотека scikit-learn <https://scikit-learn.org/>
3. Методы pandas.DataFrame <https://pandas.pydata.org/docs/reference/frame.html>
4. Библиотека pandas_datareader https://pandas-datareader.readthedocs.io/en/latest/remote_data.html

ПРИЛОЖЕНИЕ А

КОД СИСТЕМЫ ПРОВЕРКИ

```
import subprocess, sys
import pandas_datareader.data as web
from datetime import datetime

code_to_add = ''
a = input()
print(solve(a))
'''

def generate():
    return ['2018-05-05', '2022-02-05', '2016-05-05', '2022-07-06',
'2021-07-01']

def check(reply, clue):
    return str(reply).strip() == str(clue).strip()

def solve(start):
    end = datetime.today()
    f = web.DataReader('AV.UK', 'stooq', start=start, end=end)
    f = f.sort_values(by=['Open', 'Close'], ascending=False)
    return f

lines = """{{ STUDENT_ANSWER | e('py') }}"""
# checking with inspect
if ('datetime' not in lines):
    raise Exception('Для удобства используйте datetime!')
if ('.today()' not in lines):
    raise Exception('Используйте datetime правильно!')
if ('sort_values(' not in lines):
    raise Exception('Изучите сортировку!')
```

```

if ('Close' not in lines):
    raise Exception('Внимательно читайте условие сортировки!')
if ('mergesort' in lines):
    raise Exception('Внимательно читайте условие сортировки!')
if ('heapsort' in lines):
    raise Exception('Внимательно читайте условие сортировки!')
if ('stable' in lines):
    raise Exception('Внимательно читайте условие сортировки!')

with open("prog.py", "w") as src:
    print(lines, file=src)
    print('\n', file=src)
    print(code_to_add, file=src)

tests = generate()
len_tests = len(tests)
incorrect_count = 0
correct_count = 0
COUNT_OPEN_TESTS = 2

try:
    for test in tests:
        output = subprocess.check_output(["python3", "prog.py"],
input=test, universal_newlines=True)
        correct_output = solve(test)
        if not check(output, correct_output):
            incorrect_count += 1
            if incorrect_count < COUNT_OPEN_TESTS:
                result = 'Test: {}\n'.format(test)
                result += 'Your answer: {}\n'.format(output)
                result += 'Correct: {}\n'.format(correct_output)
                print(result)
        else:
            correct_count += 1
except subprocess.CalledProcessError as e:
    if e.returncode > 0:
        # Ignore non-zero positive return codes

```

```

        if e.output:
            print(e.output)
    else:
        # But negative return codes are signals - abort
        if e.output:
            print(e.output, file=sys.stderr)
        if e.returncode < 0:
            print("Task failed with signal", -e.returncode,
file=sys.stderr)
            print("** Further testing aborted **", file=sys.stderr)
    if correct_count == len(tests):
        print('OK')
    else:
        print('Wrong answer')

```