

Задание 2. Исследование методов сериализации данных

Задание 2. Исследование методов сериализации данных.....	1
1.1 Критерии оценивания.....	1
1.2 Методические указания	2
1.3 Примеры реализации	2
1.3.1 Пример реализации на языке Python	2
1.4 Задание на самостоятельную работу	4
1.5 Ссылки	4

Цель: на языке высокого уровня (Java, C#, Python и др. – на выбор обучающегося) реализовать приложение для тестирования эффективности работы с различными форматами сериализации данных (http://en.wikipedia.org/wiki/Comparison_of_data_serialization_formats). В процессе тестирования форматов сериализации необходимо учитывать следующие характеристики:

1. Размер сериализованной структуры данных;
2. Время сериализации/десериализации.

В сериализуемой структуре желательно представить несколько различных видов данных, включая:

- строковые данные,
- массивы данных,
- словари,
- целочисленные данные,
- данные с плавающей запятой.

Для отчета необходимо представить:

- 1) Отдельное приложение для тестирования форматов сериализации;
- 2) Отчет в формате таблицы Excel о форматах сериализации данных.

1.1 Критерии оценивания

№	Задача
1.	Протестировать нативный вариант сериализации (в зависимости от языка)
2.	Протестировать сериализацию в XML
3.	Протестировать сериализацию в JSON
5.	Протестировать сериализацию в Google Protocol Buffers
6.	Протестировать сериализацию в Apache Avro
7*	Протестировать сериализацию в YAML
8*	Протестировать сериализацию в MessagePack

1.2 Методические указания

Тестирование проводится на основе структуры данных, хранящейся в оперативной памяти в виде объектов или структур. Приложение должно выводить на консоль информацию о следующих характеристиках данных:

1. Процесс сохранения данных из структуры в сериализованную структуру в оперативной памяти:
 - 1.1. Полученный объем
 - 1.2. Время сериализации
2. Процесс загрузки данных из файла на жестком диске в оперативную память:
 - 2.1. Время десериализации

Обычно, тестирование времени сериализации и десериализации замеряется как среднее время при выполнении большого числа операций сериализации/десериализации (порядка 1000 циклов). Для учета времени выполнения процедуры используются различные механизмы профилирования времени выполнения кода, в зависимости от используемого языка:

- 1) C++: <http://esate.ru/blog/cpp/364.html>
- 2) C#: <http://msdn.microsoft.com/ru-ru/library/system.diagnostics.stopwatch.aspx>
- 3) Python: <http://john16blog.blogspot.ru/2012/02/python-timeit.html>
- 4) Java: <http://www.javaspecialist.ru/2012/04/system.html>

В сериализуемой структуре данных желательно представить несколько различных видов данных, включая: строковые данные, массивы данных, словари, целочисленные данные, данные с плавающей запятой.

1.3 Примеры реализации

1.3.1 Пример реализации на языке Python

Пример генерации структуры для сериализации (Python):

```
d = {
    'words': """
        Lorem ipsum dolor sit amet, consectetur adipiscing
        elit. Mauris adipiscing adipiscing placerat.
        Vestibulum augue augue,
        pellentesque quis sollicitudin id, adipiscing.
        """,
    'list': range(100),
    'dict': dict((str(i), 'a') for i in iter(range(100))),
    'int': 100,
    'float': 100.123456
}
```

```
}
```

Можно использовать данный вариант тестирования Native Serialization и JSON на Питоне как пример реализации таких тестов.

```
"""
Dependencies:

    pip install tabulate simplejson

"""

from timeit import timeit
from tabulate import tabulate
import sys

message = '''d = {
    'PackageID' : 1539,
    'PersonID' : 33,
    'Name' : ""MEGA_GAMER_2222"",
    'Inventory': dict((str(i),i) for i in iter(range(100))),
    'CurrentLocation': ""
        Pentos is a large port city, more populous than Astapor on Slaver Bay,
        and may be one of the most populous of the Free Cities.
        It lies on the bay of Pentos off the narrow sea, with the Flatlands
        plains and Velvet Hills to the east.
        The city has many square brick towers, controlled by the spice traders.
        Most of the roofing is done in tiles. There is a large red temple in
        Pentos, along with the manse of Illyrio Mopatis and the Sunrise Gate
        allows the traveler to exit the city to the east,
        in the direction of the Rhoyme.
    ""
}'''

setup_pickle    = '%s ; import pickle ; src = pickle.dumps(d, 2)' % message
setup_json      = '%s ; import json; src = json.dumps(d)' % message

tests = [
    # (title, setup, enc_test, dec_test)
    ('pickle (native serialization)', 'import pickle; %s' % setup_pickle, 'pickle.dump
s(d, 2)', 'pickle.loads(src)'),
    ('json', 'import json; %s' % setup_json, 'json.dumps(d)', 'json.loads(src)'),
]

loops = 5000
enc_table = []
dec_table = []

print ("Running tests (%d loops each)" % loops)
```

```

for title, mod, enc, dec in tests:
    print (title)

    print ("  [Encode]", enc)
    result = timeit(enc, mod, number=loops)
    exec (mod)
    enc_table.append([title, result, sys.getsizeof(src)])

    print ("  [Decode]", dec)
    result = timeit(dec, mod, number=loops)
    dec_table.append([title, result])

enc_table.sort(key=lambda x: x[1])
enc_table.insert(0, ['Package', 'Seconds', 'Size'])

dec_table.sort(key=lambda x: x[1])
dec_table.insert(0, ['Package', 'Seconds'])

print ("\nEncoding Test (%d loops)" % loops)
print (tabulate(enc_table, headers="firstrow"))

print ("\nDecoding Test (%d loops)" % loops)
print (tabulate(dec_table, headers="firstrow"))

```

1.4 Задание на самостоятельную работу

Далее, в зависимости от выбранной вами технологии и языка программирования, вам предлагается самостоятельно провести исследование методов сериализации данных и подготовить краткий отчет.

1.5 Ссылки

Для дальнейшего самостоятельного изучения данной темы можно воспользоваться следующими ресурсами:

1. Comparison of data serialization formats. http://en.wikipedia.org/wiki/Comparison_of_data_serialization_formats
2. Serialization. http://en.wikipedia.org/wiki/Data_serialization (в статье приводится информация об основных методах реализации сериализации в различных средах программирования).