

Міністерство освіти і науки України
Національний авіаційний університет
Навчально-науковий інститут комп'ютерних інформаційних технологій
Кафедра комп'ютеризованих систем управління

Лабораторна робота №5
з дисципліни «Якість програмного забезпечення та тестування»
на тему «Автоматизація процесу тестування»

Виконав:
студент ННІКІТ СП-225
Клокун Владислав
Перевірила:
Апенько Н. В.

Київ 2017

Частина I

Програма та тести

1 Код програми

```
1 ADDRESS_BOOK_FILE = 'address-book'
2
3 import argparse
4
5 class Record:
6     def __init__(self):
7         self.name      = ''
8         self.surname   = ''
9         self.tel       = ''
10
11     def get_surname(self):
12         return self.surname
13
14     def set_surname(self, surname):
15         self.surname = surname
16
17     def get_name(self):
18         return self.name
19
20     def set_name(self, name):
21         self.name = name
22
23     def get_tel(self):
24         return self.tel
25
26     def set_tel(self, tel):
27         self.tel = tel
28
29     def set_record(self, surname, name, tel):
30         self.set_surname(surname)
31         self.set_name(name)
32         self.set_tel(tel)
33
34     def as_string(self):
35         return self.get_surname() + '\t' + self.get_name() + '\t' +
36             self.get_tel() + '\n'
37
38 class AddressBook:
39     def __init__(self):
```

```

39         self.records = []
40
41     def add_record(self, surname, name, tel):
42         record = Record()
43         record.set_record(surname, name, tel)
44         self.records.append(record)
45
46     def print_all(self):
47         for record in self.records:
48             print(record.as_string())
49
50
51     def read_from_file(self):
52         with open(ADDRESS_BOOK_FILE, 'r') as address_book:
53             for line in address_book:
54                 surname, name, tel = line.split(maxsplit = 3)
55                 self.add_record(surname, name, tel)
56
57     def find_person(self, s):
58         matches = []
59         for record in self.records:
60             if s in record.as_string():
61                 matches.append(record.as_string())
62
63         return matches
64
65     def write_to_file(self, f):
66         with open(f, 'w') as f:
67             for record in self.records:
68                 f.write(record.as_string())
69
70     def main(args):
71         address_book = AddressBook()
72         address_book.read_from_file()
73         if args.printall:
74             address_book.print_all()
75
76         if args.find:
77             if args.surname:
78                 print(address_book.find_person(args.surname))
79             else:
80                 print('Surname not specified. Ignoring.')
81
82         if args.insert:
83             if args.surname and args.name and args.telephone:

```

```

84         address_book.add_record(args.surname, args.name, args.
            telephone)
85         address_book.write_to_file(ADDRESS_BOOK_FILE)
86     else:
87         print('Params not specified. Ignoring.')
88
89     print('Done.')
90
91 if __name__ == '__main__':
92     parser = argparse.ArgumentParser()
93
94     # Parse modes
95     parser.add_argument('--insert', '-i', action='store_true')
96     parser.add_argument('--printall', '-p', action='store_true')
97     parser.add_argument('--delete', '-d', action='store_true')
98     parser.add_argument('--edit', '-e', action='store_true')
99     parser.add_argument('--find', '-f', action='store_true')
100
101     # Parse address book record fields
102     parser.add_argument('--name', '-n')
103     parser.add_argument('--surname', '-s')
104     parser.add_argument('--telephone', '-t')
105
106     args = parser.parse_args()
107
108     main(args)

```

2 Код тестів

```

1 import unittest
2 import address_book
3
4 class TestRecordMethods(unittest.TestCase):
5     """Test Record class methods"""
6
7     def test_name_setter(self):
8         a = address_book.Record()
9         a.set_name('Vasya')
10        self.assertEqual(a.name, 'Vasya')
11
12    def test_name_getter(self):
13        a = address_book.Record()
14        a.set_name('Vasya')
15        self.assertEqual(a.get_name(), 'Vasya')
16

```

```
17     def test_surname_setter(self):
18         a = address_book.Record()
19         a.set_surname('Pupkin')
20         self.assertEqual(a.surname, 'Pupkin')
21
22     def test_surname_getter(self):
23         a = address_book.Record()
24         a.set_surname('Pupkin')
25         self.assertEqual(a.get_surname(), 'Pupkin')
26
27 if __name__ == '__main__':
28     unittest.main()
```

Частина II

Звіт про помилки

1 Повідомлення про помилку під час спроби перегляду адресної книги, якщо вона не існує

1.1 Короткий опис

Якщо видалити файл адресної книги спробувати запустити програму для виведення змісту адресної книги, програма повідомляє про помилку.

1.2 Детальний опис

Якщо в директорії відсутній файл адресної книги `address-book`, то при запуску програми командою `address-book.py -p` виводиться така помилка: «OS error: [Errno 2] No such file or directory: 'address-book'».

1.3 Кроки для відтворення

1. Видалити, перемістити або перейменувати файл `address-book` у папці, що містить програму.
2. Запустити програму для перегляду адресної книги (`address-book.py -p`).
3. Спостерігати результат.

1.4 Відтворюваність

Проблема спостерігається завжди.

1.5 Важливість

S4 — Незначна.

1.6 Статус

Вирішений.

1.7 Резолюція

Не буде виправлений (WONTFIX): очікувана поведінка. У разі відсутності файла адресної книги користувачу виводиться повідомлення про помилку і програма нормально завершує роботу.

Перед тим, як переглядати адресну книгу, користувач повинен занести туди хоча б один запис.