

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«Основи програмної інженерії»

Лабораторна робота № 10
«Тестування та управління змінами програмного забезпечення»

на тему:
«Автоматизована система «Виклик таксі»»

Виконав:	Богатько Олександр Геннадійович	Перевірила:	Юрчук Ірина Аркадіївна
Група	ІПЗ-12	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2020			

Завдання 1

1. Створити сценарій тестування мікрохвильової печі (використовуючи State diagram).
2. Скласти план тестування свого варіанту.
3. На основі варіантів використання системи та інтерфейсу, розробленому в роботі 5 скласти тестові варіанти для тестування інтерфейсу користувача.

1.



2.

1) Цілі тестування:

- забезпечення виконання всіх системних вимог і критеріїв, встановлених до програмного продукту;
- підвищення ймовірності того, що додаток при будь-яких обставинах буде функціонувати належним чином і відповідати встановленим вимогам за рахунок виявлення максимально можливого числа дефектів;
- забезпечення працездатності всієї системи в цілому та її компонентів.

2)

- № п/п	Вид тестування	Ресурси	Дата/термін
1.	Інтегроване тестування	Тестувальник, менеджер, комп'ютер	23.04 – 25.04
2.	Модульне тестування	Тестувальник, комп'ютер	По закінченню розробки кожного модуля
3.	Тестування інтерфейсу	Тестувальник, комп'ютер	26.04
4.	Функціональне тестування	Тестувальник, менеджер, комп'ютер	2.05 – 4.05

- Інтегроване тестування: перевірка на співвідношення класів, розроблених окремо один від одного, їхню взаємодію.
- Модульне тестування: перевірка роботи окремого модуля, його працездатності.
- Тестування інтерфейсу: перевірка клієнтських і адміністративних інтерфейсів на можливість виконання з їх допомогою сценарію використання.
- Функціональне тестування: розглядає продукт, складений із багатьох класів та модулів, як єдине ціле, перевіряє його загальну працездатність та відповідність вимогам.

3)

Специфікація тестового варіанту	
Назва взаємодіючих класів: Адміністратор, Центр підтримки	Назва тесту: AdmToServ
Опис тесту: Тест перевіряє можливість адміністратора увійти до центру підтримки клієнтів, провести операції.	
Передумови: Код працівника, ім'я працівника	

Очікуваний результат: Адміністратор може увійти до центру, відповісти на заявку клієнта, відповісти в онлайн чаті, отримати оцінку своїх дій від клієнта.
Отриманий результат: Дані співпадають, операції відбуваються правильно.

Специфікація тестового варіанту	
Назва взаємодіючих класів: Користувач, Особистий кабінет	Назва тесту: UserToCabinet
Опис тесту: Тест перевіряє можливість клієнта увійти до особистого кабінету та виконати дії.	
Передумови: Наявність особистого кабінету, логіну та паролю.	
Очікуваний результат: Вводячи правильні дані для входу, клієнт може увійти до свого особистого кабінету та виконати дії.	
Отриманий результат: Дані співпадають, операції відбуваються правильно.	

Специфікація тестового варіанту	
Назва взаємодіючих класів: Користувач, поїздка	Назва тесту: UserToRide
Опис тесту: Тест перевіряє можливість обрати поїздку, замовити її та оплатити.	
Передумови: Наявність потрібних для оформлення поїздки.	
Очікуваний результат: Клієнт, вводячи правильні бажані дані, знаходить потрібний білет, має можливість його обрати та оплатити.	
Отриманий результат: Дані співпадають, дії відбуваються правильно.	

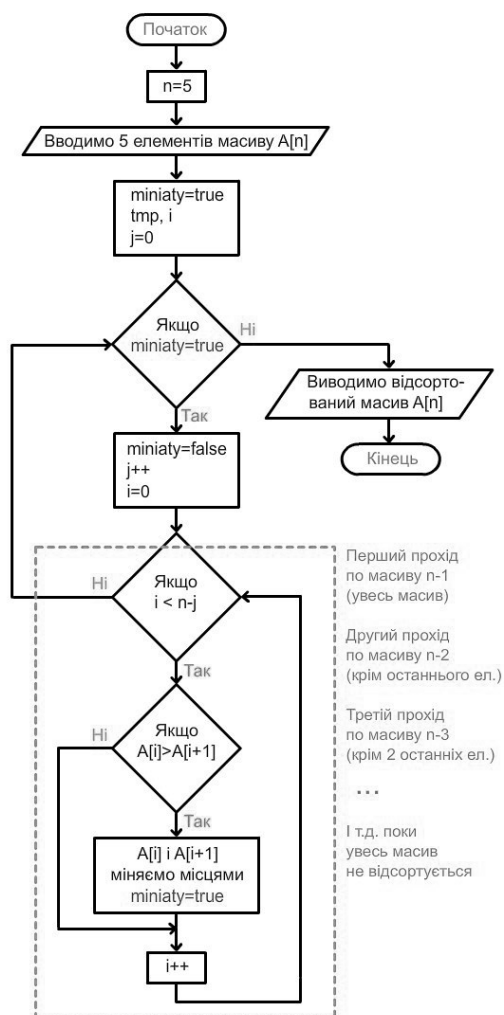
Специфікація тестового варіанту	
Назва взаємодіючих класів: Account, Cabinet	Назва тесту: AccountToCabinet
Опис тесту: Тест перевіряє правильність відповідності між кабінетом та рахунком	
Передумови: Наявність рахунку	
Очікуваний результат: Номер та стан рахунку, до якого прив'язаний кабінет співпадають з номером та станом наявного рахунку, платежі, що проводяться в кабінеті, відбуваються з використанням наявного рахунку	
Отриманий результат: Дані співпадають, операції відбуваються правильно.	

Завдання 2

1. Розробити набір тестових даних для наступних компонентів (використовуючи Activity diagram):

- програма сортування масивів цілих чисел;
- програма, яка вираховує кількість символів (відмінних від пропусків) в текстових строках;
- програма, яка перевіряє текстові строки і замінює послідовності пропусків одним пропуском, а там де один пропуск – змінює його на символ %.

2. Написати код для вищеперерахованих програм, використовуючи будь-яку мову програмування.



```

using namespace std;

int main()
{
    setlocale(LC_ALL, "rus");

    bool change = true;
    int j = 0;
    int i = 0;
    int tmp;
    const int size = 5;
    int a[size];

    cout << "Введите массив: ";
    for (int i = 0; i < size; ++i)
    {
        cin >> a[i];
    }

    cout << "\n" << "Ваш массив: " << endl;
    for (int i = 0; i < size; ++i)
    {
        cout << a[i] << " ";
    }
    cout << "\n";
    while (change)
    {
        change = false;
        j++;
        for (i = 0; i < size - j; i++)
        {
            if (a[i] > a[i + 1])
            {
                tmp = a[i];
                a[i] = a[i + 1];
                a[i + 1] = tmp;
                change = true;
            }
            cout << "\n";
        }
    }
    cout << "\n" << "Ваш массив после сортировки: " << endl;
    for (int i = 0; i < size; ++i)
    {
        cout << a[i] << " ";
    }
    cout << "\n";
}

```

```

Введите массив: 23
-700
0
56
987

Ваш массив:
23 -700 0 56 987

```

```

Ваш массив после сортировки:
-700 0 23 56 987

```

Введите массив: 2

2
0
13
13
86
-54
-54
23
2

Ваш массив:

2 2 0 13 13 86 -54 -54 23 2

Ваш массив после сортировки:

-54 -54 0 2 2 2 13 13 23 86

Введите массив: -900

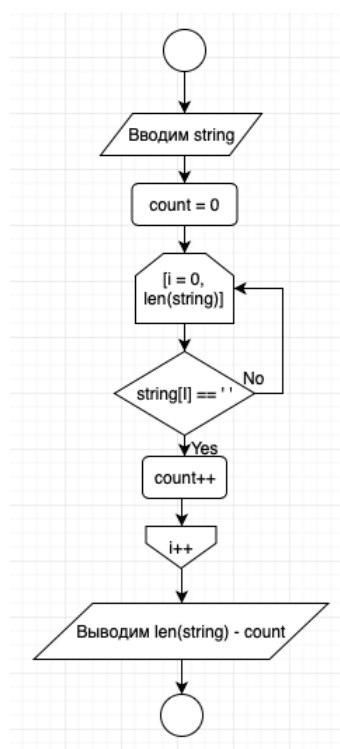
843
-765
654
-587
421
-380
217
-155
41

Ваш массив:

-900 843 -765 654 -587 421 -380 217 -155 41

Ваш массив после сортировки:

-900 -765 -587 -380 -155 41 217 421 654 843



```
string = input('Введите строку, в которой хотите посчитать символы:\n')
count = 0

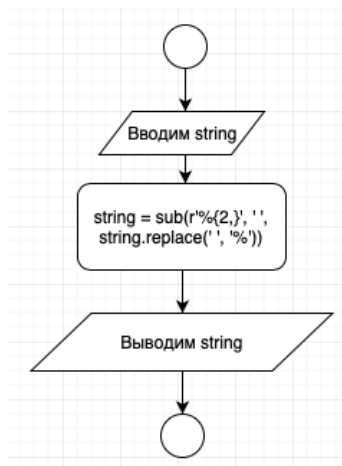
for i in range(0, len(string)):
    if (string[i] == ' '):
        count += 1

print('В вашей строке ' + str(len(string) - count) + ' символов')
```

Введите строку, в которой хотите посчитать символы:
1 2 3 4 5 6 7 8 9 0
В вашей строке 10 символов

Введите строку, в которой хотите посчитать символы:
посчитать сколько символов
В вашей строке 24 символов

Введите строку, в которой хотите посчитать символы:
один 1 два 2 три 3
В вашей строке 13 символов



```
from re import sub

string = input('Введите строку:\n')
string = sub(r'%(2,}', ' ', string.replace(' ', '%'))

print('Строка после преобразования:\n')
print(string)
```

Введите строку: 123 fgh 123 ghj 123	Введите строку: 123 123 123 123 123
Строка после преобразования: 123%fgh%123%ghj%123	Строка после преобразования: 123 123 123 123 123
Введите строку: апр 123 апр 123 апр	
Строка после преобразования: апр 123%апр 123%апр	

Висновок:

В цій лабораторній роботі було розроблено план та сценарій тестування для системи, що розробляється. Було описано кожен етап тестування, специфіковано всі тестові варіанти, визначено необхідні ресурси та призначено дату для кожного тестування. Було розглянуто можливі варіанти управління змінами ПЗ. Було розроблено тести для приладу мікрохвильова піч та трьох написаних кодів.