МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

кафедра програмних засобів

Звіт

з лабораторної роботи № 3

з дисципліни «Професійна практика програмної інженерії»

на тему:

«Реалізація базових функцій системи»

Виконав

Ст. гр. КНТ-137 В.В. Козлов

Прийняв

Старший викладач І.В. Левада

1. Мета роботи

Навчитися аналізувати технічне завдання, предметну область, на основі проведеного аналізу й дослідження виявляти першочергові для реалізації функції програмного забезпечення та реалізовувати їх.

1. Завдання на лабораторну роботу

Ознайомитися з літературою та основними теоретичними відомостями, необхідними для виконання роботи.

Виконати аналіз ТЗ і розробленої архітектури системи та виділити першочергові функції, які має виконувати система.

Розробити програму, що реалізує першочергові функції, які має виконувати система. На даному етапі потрібно розробити консольний додаток (або декілька консольних додатків за необхідності). Реалізація програми виконується за допомогою обраної на попередньому етапі мови програмування. Реалізована програма має відповідати нормам програмування, тобто має бути виключена надлишковість програмного коду тощо, що має забезпечуватися за допомогою використання принципів об’єктно-орієнтованого програмування: поліморфізм, інкапсуляція, спадкування.

Виконати тестування розробленого програмного забезпечення. У процесі тестування має обов’язково застосовуватись модульне тестування. Тестування має виконуватися за різноманітних умов роботи програми, тобто шляхом введення різних даних, шляхом виконання на пристроях з різними апаратними характеристиками, а також під керуванням різних операційних систем або версійопераційних систем. Результатами тестування є швидкість роботи програми, вимоги до ресурсів, а також коректність отримуваних результатів.

Виконати аналіз отриманих результатів тестування. У процесі аналізу отриманих результатів має бути порівняно результати, отримані під керування різних операційних систем (або їх версій) та на різних пристроях.

Оформити звіт з роботи.

Відповісти на контрольні запитання.

1. Результати виконання роботи
   1. Аналіз предметної області

В результаті аналізу предметної області та складання технічного завдання було віднесено до базових функцій системи такі, що забезпечують логіку роботи програмного забезпечення. Згідно з цим, в ході цієї роботи було реалізовано бібліотеку класів, описаних в попередній роботі.

Другорядними було визначено такі функції, що забезпечують взаємодію додатку з базою даних.

* 1. Лістинг програми

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CargoDelivery.Model

{

class User

{

public int ID { get; } = -1;

public string UserName { get; set; } = "";

public string Password { get; set; } = "";

}

class Consumer : User

{

public string FirstName { get; set; }

public string SecondName { get; set; }

public string Email { get; set; }

public Consumer(string userName, string firstName, string secondName, string email)

{

UserName = userName;

FirstName = firstName;

SecondName = secondName;

Email = email;

}

}

class Operator : User

{

public Operator(string userName)

{

UserName = userName;

}

}

class Truck

{

public int ID { get; } = -1;

public string TruckName { get; set; }

public string Departure { get; set; }

public string Destination { get; set; }

public int Spaciousness { get; set; }

public string DepartureDate { get; set; }

public string ArrivalDate { get; set; }

public Truck(int iD, string truckName, string departure, string destination, int spaciousness, string departureDate, string arrivalDate)

{

ID = iD;

TruckName = truckName;

Departure = departure;

Destination = destination;

Spaciousness = spaciousness;

DepartureDate = departureDate;

ArrivalDate = arrivalDate;

}

}

public class Order

{

public int ID { get; } = -1;

public string SenderFirstName { get; set; }

public string SenderLastName { get; set; }

public string SenderPatronymic { get; set; }

public string SenderCity { get; set; }

public string RecieverFirstName { get; set; }

public string RecieverLastName { get; set; }

public string RecieverPatronymic { get; set; }

public string RecieverCity { get; set; }

public string Type { get; set; }

public int Weight { get; set; }

public int Size\_X { get; set; }

public int Size\_Y { get; set; }

public int Size\_Z { get; set; }

public int DeclaredValue { get; set; }

public int Price { get; set; }

public bool IsPaid { get; set; } = false;

public Order(string senderFirstName, string senderLastName, string senderPatronymic, string senderCity,

string recieverFirstName, string recieverLastName, string recieverPatronymic, string recieverCity,

string type, int weight, int size\_X, int size\_Y, int size\_Z, int declaredValue)

{

SenderFirstName = senderFirstName;

SenderLastName = senderLastName;

SenderPatronymic = senderPatronymic;

SenderCity = senderCity;

RecieverFirstName = recieverFirstName;

RecieverLastName = recieverLastName;

RecieverPatronymic = recieverPatronymic;

RecieverCity = recieverCity;

Type = type;

Weight = weight;

Size\_X = size\_X;

Size\_Y = size\_Y;

Size\_Z = size\_Z;

DeclaredValue = declaredValue;

Price = get\_price(type, weight, size\_X, size\_Y, size\_Z, declaredValue);

}

public Order(int id, string senderFirstName, string senderLastName, string senderPatronymic, string senderCity,

string recieverFirstName, string recieverLastName, string recieverPatronymic, string recieverCity,

string type, int weight, int size\_X, int size\_Y, int size\_Z, int declaredValue, int price, bool isPaid)

{

ID = id;

SenderFirstName = senderFirstName;

SenderLastName = senderLastName;

SenderPatronymic = senderPatronymic;

SenderCity = senderCity;

RecieverFirstName = recieverFirstName;

RecieverLastName = recieverLastName;

RecieverPatronymic = recieverPatronymic;

RecieverCity = recieverCity;

Type = type;

Weight = weight;

Size\_X = size\_X;

Size\_Y = size\_Y;

Size\_Z = size\_Z;

DeclaredValue = declaredValue;

Price = price;

IsPaid = isPaid;

}

int get\_price(string type, int weight, int size\_X, int size\_Y, int size\_Z, int declaredValue)

{

int ResultPrice = 50 + Convert.ToInt32(declaredValue \* 0.01);

if (type == "Documents") ResultPrice += 20;

ResultPrice += (Convert.ToInt32(size\_X \* size\_Y \* size\_Z \* 0.000001) \* 10);

ResultPrice += (weight \* 10);

return ResultPrice;

}

}

}

* 1. Модульне тестування

Модульне тестування – це метод тестування програмного забезпечення, який полягає в окремому тестуванні кожного модуля коду програми. Модулем називають найменшу частину програми, яка може бути протестованою. У процедурному програмуванні модулем вважають окрему функцію або процедуру. В об'єктно-орієнтованому програмуванні — метод.

Метою модульного тестування є ізоляція кожної частини програми та впевненість у тому, що кожна окрема частина є коректною. Зазвичай unit-тести застосовують для того, щоб упевнитися, що код відповідає вимогам архітектури та має очікувану поведінку.

Для тестування було обрано реалізовану першочергову функцію, яка розраховує вартість замовлення (Order.get\_price).

* 1. Лістинг модульного тесту

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using CargoDelivery.Model;

namespace CargoDelivery.Tests

{

[TestClass]

public class CargoDeliveryTests

{

[TestMethod]

public void IsRightPrice()

{

Order TestOrder = new Order("Kozlov", "Valeriy", "Valentynovych", "Kyiv", "Levada", "Iryna", "Vasylivna", "Zaporizhzhya", "Documents", 1, 100, 100, 100, 100);

Assert.AreEqual(91, TestOrder.Price);

}

}

}

* 1. Результати тесту

Результати модульного тесту наведено на рисунку 3.1.

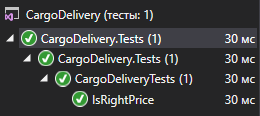


Рисунок 3.1 – Результати модульного тесту

Висновки

Протягом виконання цієї лабораторної роботи навчився аналізувати технічне завдання, предметну область.

На основі проведеного аналізу й дослідження виявив першочергові для реалізації функції програмного забезпечення та реалізовував їх.

Реалізував бібліотеку класів за допомогою мови програмування C#. Отримані результати задовольняють поставленим задачам та виконуються з достатньою швидкістю.