

ЗМІСТ

Вступ	6
1. Постановка завдання	
1.1. Уточнення постановки завдання	8
1.2. Аналіз предметної області та вибір програмних засобів для вирішення поставленого завдання.....	9
2. Конструкторська частина	
2.1. Проектування та створення бази даних і її опис	10
2.2. Побудова UML-моделі класів.....	13
2.3. Проектування та розроблення графічного інтерфейсу програмного забезпечення	15
3. Програмна частина	
3.1. Опис та розроблення класів	22
3.2. Опис та розроблення методів доступу до даних	32
4. Експериментальна частина	
4.1. Приклад виконання програмного додатку з тестовими даними та отримання результатів	34
Висновки.....	39
Список використаних джерел	41
Додаток А Код програми	42

					КП.ПІ-18-01.10.03.00.000 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Косак А.В			Розробка програмного додатку для роботи з БД предметної області «Туристичне агентство» засобами ООП			Літ.	Арк.	Аркушів	
Перевір.		Разметанюк О.Т								5	127
Н. контр.					Пояснювальна записка			ФКЕП ІФНТУНГ ПІ-18-01			

ВСТУП

Об'єктно-орієнтоване програмування (ООП) – одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. В ній використано декілька технологій від попередніх парадигм, зокрема успадкування, модульність, поліморфізм та інкапсуляцію.

На відміну від традиційних поглядів, коли програму розглядали як набір підпрограм або як перелік інструкцій комп'ютеру, ООП програми можна вважати сукупністю об'єктів. Відповідно до парадигми об'єктно-орієнтованого програмування, кожний об'єкт здатний отримувати повідомлення, обробляти дані та надсилати повідомлення іншим об'єктам. Кожен об'єкт – своєрідний незалежний автомат з окремим призначенням та відповідальністю.

Об'єктно-орієнтоване програмування – це спосіб програмування, оснований на поданні програми у вигляді сукупності взаємодіючих об'єктів, кожен з яких є екземпляром певного класу, а класи є членами певної ієрархії наслідування. [3]

Клас представляє собою шаблон, за яким визначається форма об'єкту. Клас є подібним до типів в традиційних мовах програмування. Відмінність полягає в тому, що користувач може створювати власні типи (класи) в об'єктно-орієнтованому програмуванні.

Якщо клас є подібним до типу в традиційних мовах програмування, тоді об'єкт є подібним до змінної. Інколи об'єкти називаються екземплярами класу. Об'єкт класу реалізує один з можливих варіантів опису класу. При описі об'єкту класу виділяється оперативна пам'ять.

При описі класу оголошуються дані, які він містить, та програмний код, що оперує цими даними.

Також клас може містити тільки дані або тільки програмний код. Але при роботі з класами більшість класів містить і дані, і програмний код.

Члени-дані ще називаються полями класу. До членів-даних відносяться змінні екземпляру класу та статичні змінні. Поля класу можуть мати будь-який допустимий в програмі тип. [4]

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

Актуальність теми курсового проєкту зумовлена практичною затребуваністю технології, яка надасть змогу автоматизувати облік здійснених замовлень та повернень квитків, структуровано зберігати дані про клієнтів і працівників туристичного агентства, а також спростити доступ адміністрації до цих даних. Програма, створена в рамках курсового проєкту, була розроблена з урахуванням специфіки діяльності туристичного агентства і застосування баз даних при веденні обліку.

Виконання курсового проєкту з об'єктно-орієнтованого програмування формує фахові компетентності, які визначають здатність володіти основами моделювання програмного забезпечення, типами моделей, основними концепціями уніфікованої мови моделювання UML, основами методів та технологій об'єктно-орієнтованого програмування, здійснювати аналіз вимог, розробляти специфікацію програмних вимог, виконувати їхню верифікацію та атестацію.

Темою курсового проєкту є розроблення графічного додатку для роботи з базою даних, що містить інформацію про предметну область «Туристичне агентство».

Метою курсового проєктування є закріплення навиків створення програмних застосунків для зберігання, видобування та опрацювання даних із використанням об'єктно-орієнтованого підходу до проєктування та розроблення програмного забезпечення, забезпечуючи захищеність програм і даних від несанкціонованих дій.

Предметом дослідження курсового проєкту є програмні засоби Visual C++ для надання користувачу можливості взаємодії з БД шляхом створення графічного інтерфейсу. Об'єктом дослідження є методи та засоби розроблення комп'ютерних систем управління з використанням принципів об'єктно-орієнтованого програмування.

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ПОСТАНОВКА ЗАВДАННЯ

1.1 Уточнення постановки завдання

До розроблювального курсового проєкту були поставлені нижченаведені вимоги.

Розробити проєкт зі зручним графічним інтерфейсом користувача для роботи з базою даних, який повинен містити наступні візуальні форми: стартову форму, форму для відображення даних та пошуку за вказаним критерієм, форму для вводу/редагування даних. Окрім цього, програма повинна мати головне меню, панель інструментів, кнопки для виклику відповідних функцій. Програмний застосунок повинен виконувати наступні дії:

- зчитувати дані з бази даних і відображати їх у таблицях;
- додавати записи;
- редагувати записи;
- видаляти записи;
- виконувати пошук за вказаними критеріями;
- забезпечувати контроль помилок при введенні користувачем даних;
- виводити інформацію на друк.

Підготувати дані, що містять інформацію про предметну область «Туристичне агентство». Кількість записів довільна.

Для розробки програмного застосунку засобами мови програмування C++ було використано середовище MS Visual Studio 2019, яке дозволяє використовувати синтаксичні конструкції та функціональні можливості мови для розробки об'єктно-орієнтованих програм і готові бібліотеки класів та засоби ООП Windows-застосунків, надаючи при цьому стандартний інтерфейс візуальних компонентів для створення GUI.

Для збереження даних вказаної інформаційної системи було використано вбудований у середовище Sql Server, який полегшує створення бази даних, надаючи для цього візуальні засоби розробки БД та не потребує встановлення системи керування базами даних.

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз предметної області та вибір програмних засобів для вирішення поставленого завдання

При дослідженні організаційної структури туристичного агентства було досліджено бізнес-правила в даній галузі. Кожна з більшості туристичних агентств надає широкий спектр послуг. Зокрема, різноманітні тури в країни світу з різними типами відпочинку, бронювання номерів у готелях різного класу через популярні світові сервіси, бронювання квитків на автобусні та авіарейси, страхування мандрівок та відпочинку, швидка та надійна допомога у підготовці документів для візового оформлення, необхідна інформація про знижки, «гарячі» путівки та спеціальні пропозиції тощо.

Спроектований програмний продукт дозволяє адміністрації турагентства зберігати, оновлювати та видаляти дані всіх клієнтів, персоналу, турів, готелів, здійснених замовлень та повернень путівок. Окрім цього, забезпечено можливість пошуку даних за різними критеріями, вибраними користувачем, та швидкого пошуку з вказаними умовами. Для зручності в користуванні програмним застосунком створено панель інструментів, на якій розміщено «швидкі» піктограми для виклику необхідних функцій, а також піктограми для виклику окремих Windows-форм, які зберігають інформацію про призначення та автора програми.

Окрім цього, забезпечено можливість програмним методом формувати звіт щодо всіх зроблених замовлень, вказуючи всі необхідні дані та виводячи при цьому суму коштів, сплачених за путівки. Даний звіт формується завдяки таблиці, яка зберігає потрібні дані, роблячи скріншот і роздруковуючи зображення за допомогою принтера, перед тим виводячи аркуш паперу, який йде на друк.

Виконання проєкту було полегшено завдяки використанню візуальних засобів системи керування базами даних SQL Server Compact 4, яка забезпечує можливість створення бази даних та маніпулювання даними, що в ній зберігаються, завдяки використанню різноманітних операцій.

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

2 КОНСТРУКТОРСЬКА ЧАСТИНА

2.1 Проектування та створення бази даних та її опис

Дослідивши предметну область «Туристичне агентство» можна виділити наступні сутності:

- «Працівники» містить інформації про персонал агентства – код, ПІБ, дата народження, електронна пошта, домашня адреса, номер телефону, посада, фото;
- «Клієнти» містить інформацію про клієнтів – код, ПІБ, дата народження, номер телефону, номер паспорта;
- «Готелі» містить інформацію про готелі, з якими співпрацює агентство – код, назва, тип, клас, номер телефону, фото;
- «Тури» містить основну інформацію про тури, які пропонує агентство – код, назва, тип, сезон, ціна, тривалість, кількість туристів, країна, код готелю (зовнішній ключ), дата туру;
- «Замовлення» містить розширену інформацію про замовлення – код, дата замовлення, кількість путівок, загальна ціна, код працівника (зовнішній ключ), код клієнта (зовнішній ключ), код туру(зовнішній ключ);
- «Повернення» містить інформацію про повернені путівки – код, дата повернення, кількість путівок, повернена сума, код замовлення.

Дані таблиці було створено в СКБД SQL Server Compact 4:

- таблиця «Workers», яка містить інформацію про працівників туристичного агентства. Структура таблиці «Workers» зображена на рисунку 2.1.

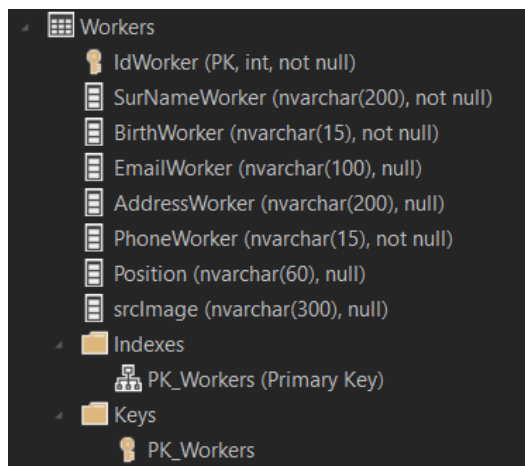


Рисунок 2.1 – Структура таблиці «Workers»

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

- таблиця «Clients», яка містить загальну інформацію про клієнтів. Структура таблиці «Clients» зображена на рисунку 2.2.

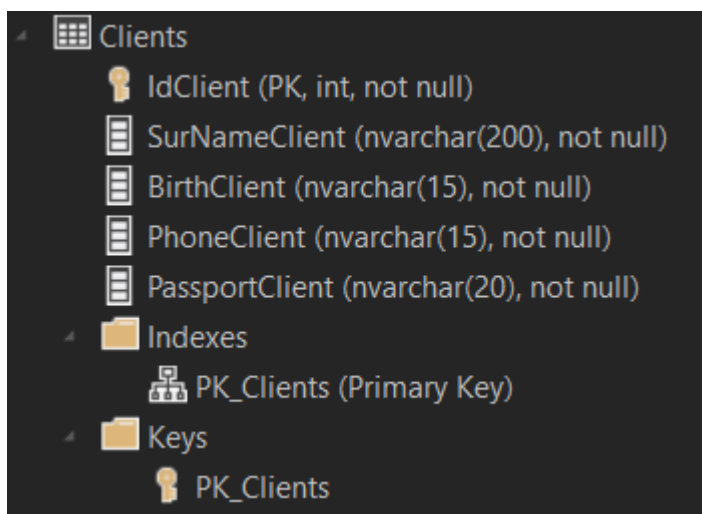


Рисунок 2.2 – Структура таблиці «Clients»

- таблиця «Hotel», яка містить узагальнену інформацію про готелі. Структура таблиці «Hotel» зображена на рисунку 2.3.

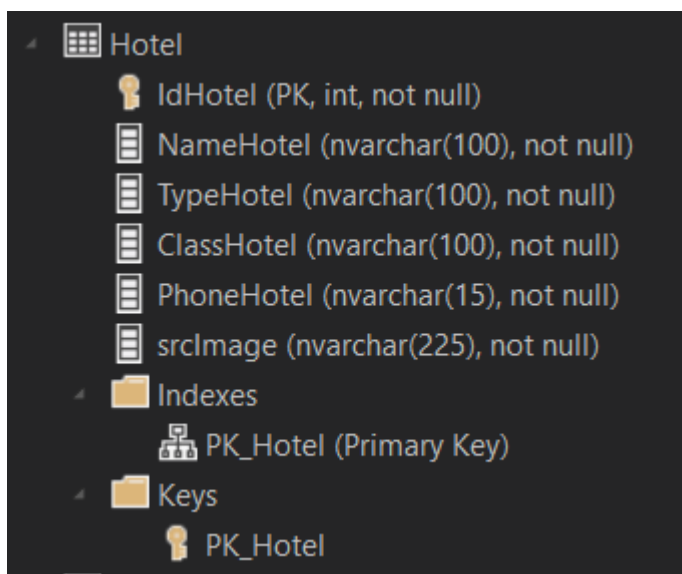


Рисунок 2.3 – Структура таблиці «Hotel»

- таблиця «Tours», яка містить коротку інформацію про тури, які пропонує агентство. Структура таблиці «Tours» зображена на рисунку 2.4.

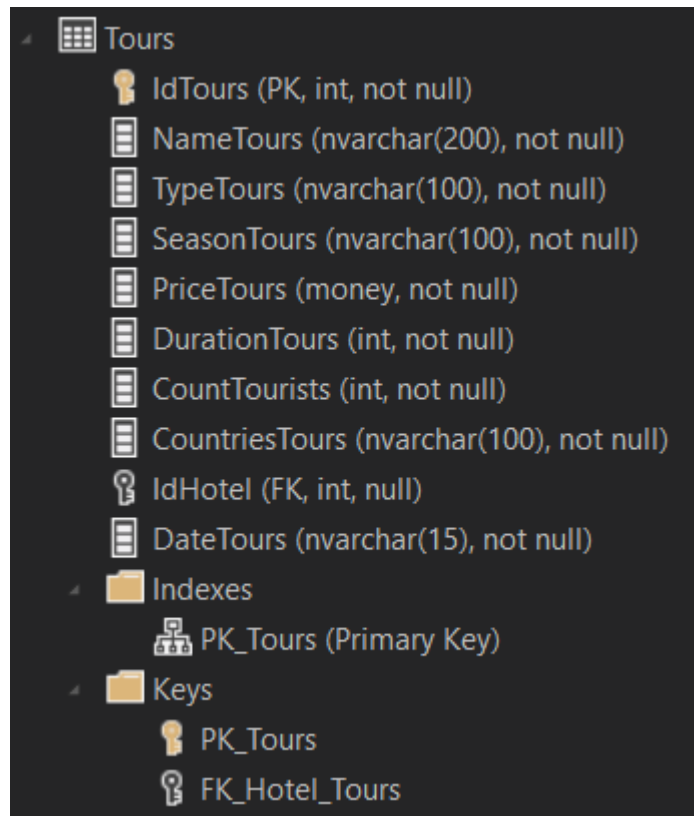


Рисунок 2.4 – Структура таблиці «Tours»

- таблиця «Orders», яка містить розгорнуту інформацію про замовлення. Структура таблиці «Orders» зображена на рисунку 2.5.

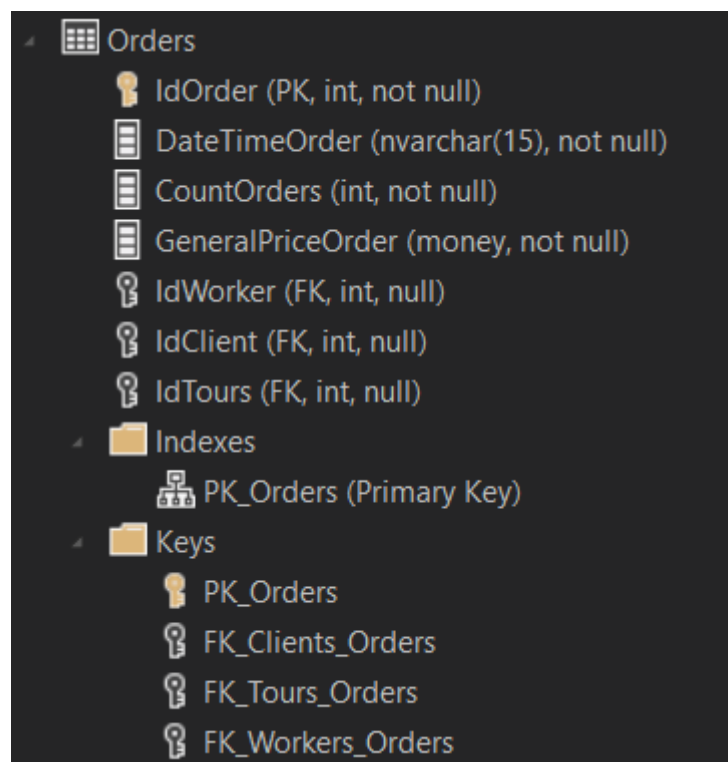


Рисунок 2.5 – Структура таблиці «Orders»

- таблиця «Comeback», яка містить розгорнуту інформацію про повернення путівок. Структура таблиці «Comeback» зображена на рисунку 2.6.

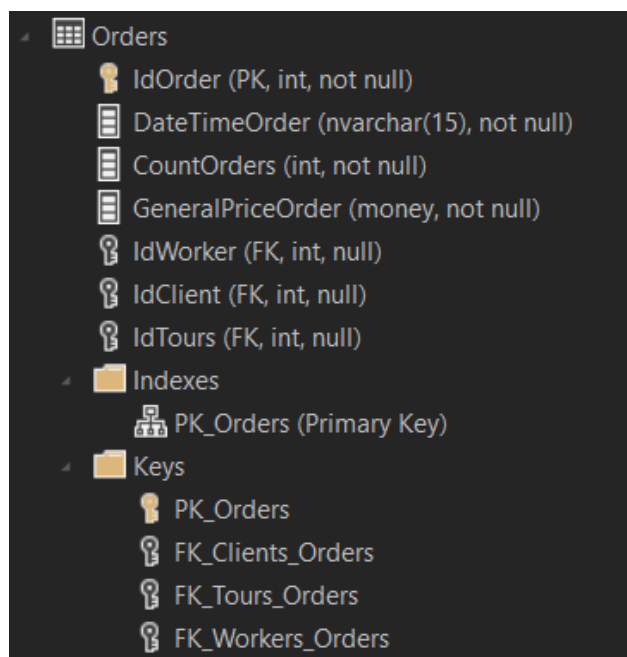


Рисунок 2.6 – Структура таблиці «Comeback»

2.2 Побудова UML-моделі класів

Unified Modeling Language (UML) – уніфікована мова моделювання. Розшифруємо: modeling передбачає створення моделі, що описує об'єкт. Unified (універсальний, єдиний) – підходить для широкого класу проєктованих програмних систем, різних областей додатків, типів організацій, рівнів компетентності, розмірів проєктів. UML описує об'єкт в єдиному заданому синтаксисі, тому де б ви не намалювали діаграму, її правила будуть зрозумілими для всіх, хто знайомий з цією графічною мовою – навіть в іншій країні.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML чудово зарекомендувала себе в багатьох успішних програмних проєктах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки.

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Як будь-яка інша мова, UML має власні правила оформлення моделей і синтаксис. За допомогою графічної нотації UML можна візуалізувати систему, об'єднати всі компоненти в єдину структуру, уточнювати і покращувати модель у процесі роботи. [5]

Створено відношення між класами (рисунок 2.7):

- класи Workers і Orders – відношення асоціації, бо два класи просто пов'язані між собою, без додаткових умов. Кратність 1:1...n, відповідно Workers : Orders.
- класи Clients і Orders – відношення асоціації, бо два класи просто пов'язані між собою, без додаткових умов. Кратність 1:1...n, відповідно Clients : Orders.
- класи Tours і Orders – відношення асоціації, бо два класи просто пов'язані між собою, без додаткових умов. Кратність 1:1...n, відповідно Tours : Orders.
- класи Hotel і Tours – відношення композиції, бо клас Hotel залежний від класу Tours, а саме якщо не існуватиме класу Tours, то клас Hotel не матиме основної ланки. Кратність 1:1...n, відповідно Hotel : Tours.
- класи Comeback і Orders – відношення асоціації, бо два класи просто пов'язані між собою, без додаткових умов. Кратність 1:1, відповідно Comeback : Orders.

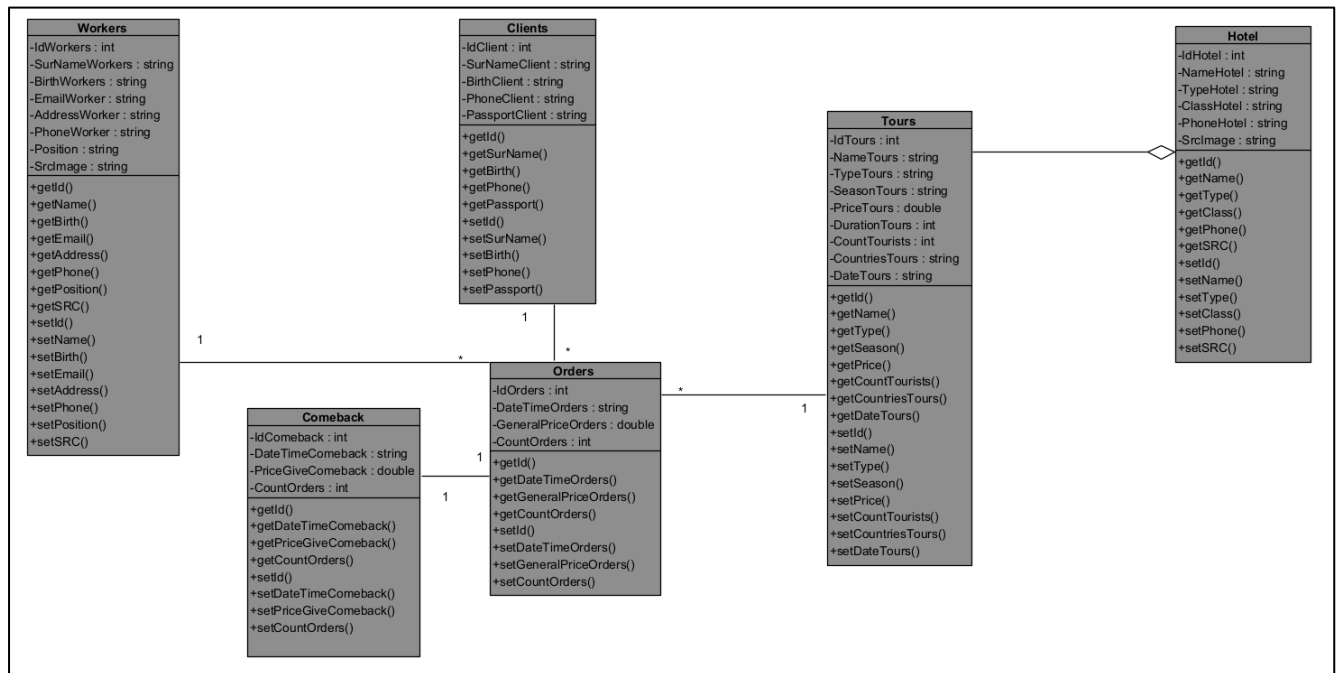


Рисунок 2.7 – UML-модель класів

2.3 Проектування та розроблення графічного інтерфейсу програмного забезпечення

Графічний інтерфейс користувача (ГІК) – тип інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки, на відміну від текстових інтерфейсів, заснованих на використанні тексту, текстовому наборі команд та текстовій навігації.

Виконання дій в ГІК – це безпосередня маніпуляція з графічними елементами. [6]

Можна виокремити такі види ГІК:

- простий: типові екранні форми та стандартні елементи інтерфейсу, що забезпечуються самою підсистемою GUI;
- істинно-графічний, двовимірний: нестандартні елементи інтерфейсу та оригінальні метафори, що реалізовані власними засобами програми або сторонньою бібліотекою;
- тривимірний.

Згідно структури програми, програма складається з 6 модулів, кожен з яких має свою візуальну частину.

Головна форма з меню програми зображена на рисунку 2.8.

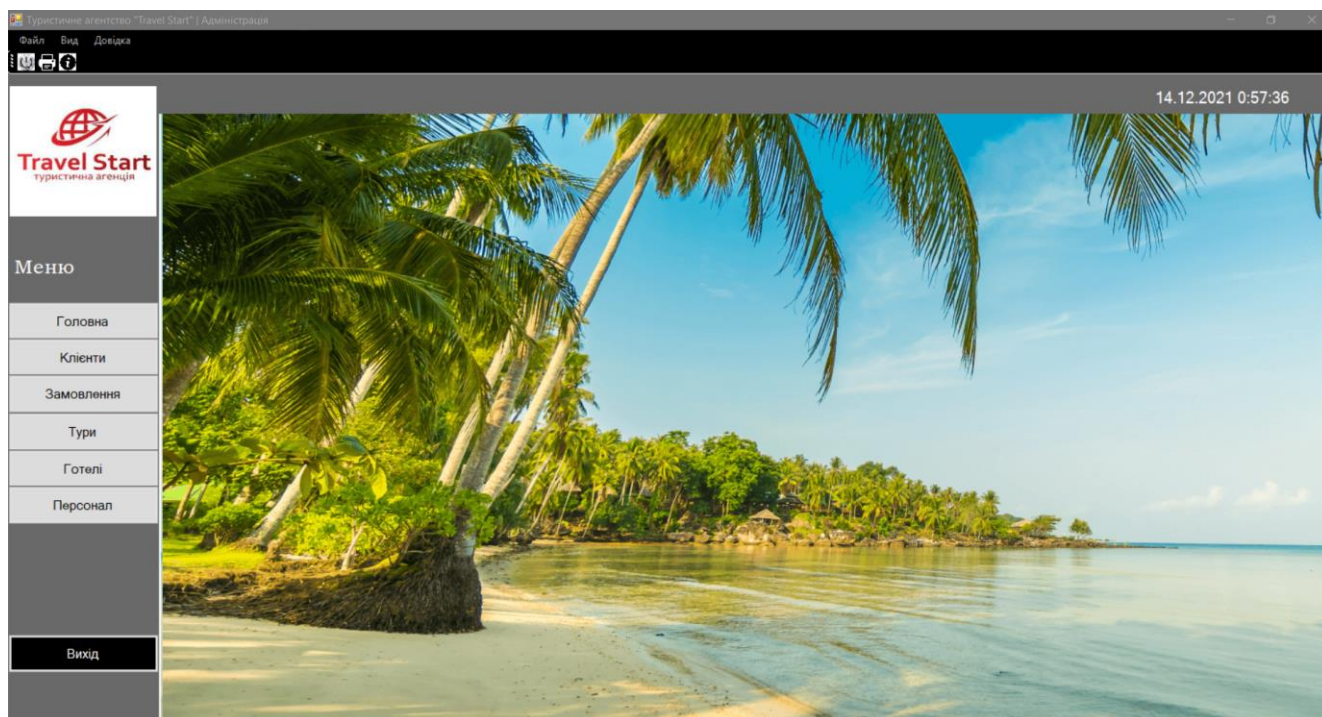


Рисунок 2.8 – Головне меню програми

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

Для відображення та обробки даних про клієнтів створено форму «Клієнти». Її зображено на рисунку 2.9.

Номер	ПІБ клієнта	Дата народження	Номер телефону	Паспорт
35	Григорук М.Р.	2002-09-09	(240) 395-0349	2495949654
33	Паранен М.Р.	13.07.2002	0995738923	094895478
36	Андрашук В.І.	20.01.2002	0978475934	090087626
37	Войцук Ю.В.	11.26.2003	0993575934	085492758
38	Гаргат Я.В.	21.06.2003	0689747823	938593858
39	Горбата В.А.	08.01.2004	0667842175	857937257
40	Гринюк В.Р.	16.02.2003	0975938502	974037193
41	Савчук А.А.	26.06.2003	0993751274	284728724
42	Іванов Д.О.	18.05.2003	0969852738	947279429
43	Кельбас Б.Р.	16.06.2003	0679383825	078357248
44	Косак А.В.	21.04.2003	0976976346	398572985
45	Кусий М.В.	09.06.2003	0948375927	094758274
46	Лішук М.Т.	16.11.2002	0993575478	088982758
47	Морош Р.Т.	15.07.2003	0983751846	938590184
48	Мансичев В.І.	26.02.2003	0938921947	984184959
49	Николаєв Д.М.	14.12.2003	0984391937	048628508
50	Паранен М.Р.	13.07.2002	0995738923	094895478

Рисунок 2.9 – Форма «Клієнти»

Форма для редагування і додавання записів про клієнтів. Її зображено на рисунку 2.10.

Рисунок 2.10 – Форма «Форма редагування»

Форма повідомлення зображена на рисунку 2.11.

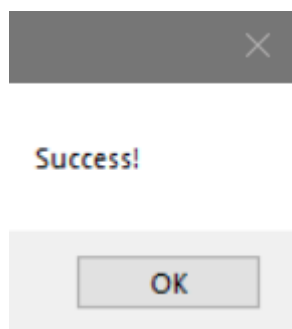


Рисунок 2.11 – Форма для повідомлення

Для відображення та обробки даних про працівників створено форму «Працівники». Її зображено на рисунку 2.12.

Туристичне агентство "Travel Start" | Адміністрація

14.12.2021 1:06:18

Працівники агентства

Персональна інформація

Посада:

Дата народження:

Електронна пошта:

Номер телефону:

Адреса:

Вибір ПІБ працівника:

Список працівників агентства

Номер	ПІБ працівника	Дата народження	Пошта	Телефон	Адреса	Посада
21	Захарук Р.В.	29.09.1998	romanchello@gmail.com	IF Незалежності, 150	0947981284	Директор
22	Однотел В.В.	15.12.2000	opiniumV@gmail.com	IF Тролейбусна, 79	0997843929	Адміністратор
23	Савошнік Ю.І.	13.05.2001	julaS@gmail.com	IF Симоненка, 28	0687832948	Фінансист
24	Дорожки З.М.	15.03.2001	zakharD@gmail.com	IF Стуса, 76	0994782974	Консультант
25	Ткачук Х.Т.	03.11.2002	khyys@gmail.com	IF Незалежності, 12...	0997872928	Консультант
26	Бондар І.М.	24.05.1987	bondarVan@gmail.com	IF Симоненка, 58	0984782829	Консультант
27	Грицак Б.Р.	19.02.2000	boGden@gmail.com	IF Карпатська, 68	0997843759	Сторінцевий
28	Захарук Р.П.	29.04.1996	neznagomka@gmail.com	IF Незалежності, 183	0997478393	Менеджер
29	Аріпова Ю.А.	12.10.1995	strategyAkhp@gmail.com	Київ Шевченка, 220	0993785483	Парменеджер
30	Соловій І.З.	30.06.1994	solovii@gmail.com	IF Мазепи, 63	0673984902	Юрист
34	Арап І.В.	11.12.1982	iarapen@gmail.com	IF Вовчицька, 23	(093) 848-7534	Власник

Додати Редагувати Видалити

Рисунок 2.12 – Форма «Працівники»

Форма для редагування і додавання записів про працівників. Її зображено на рисунку 2.13.

Редагування даних персоналу

ПІБ працівника

Дата народження
14.12.2021

Електронна пошта

Адреса

Номер телефону
() - -

Посада

Вибір фото

Вибрати

Зберегти Скасувати

Рисунок 2.13 – Форма «Форма редагування»

Для відображення та обробки даних про готелі створено форму «Готелі». Її зображено на рисунку 2.14.

Туристичне агентство "Travel Start" | Адміністрація

14.12.2021 1:08:52

Готелі, з якими співпрацює агентство

Код готелю	Назва готелю	Тип готелю	Клас готелю	Номер телефону
31	Grand Oasis Resort	Пляжний	перший (4*)	0489352973
32	Tui Blue Palm Garden	Пляжний	вищий (5*)	0484592174
33	Martin Jolie Ville Resort & Cas...	Пляжний	вищий (5*)	8984124909
34	Delphin De Luxe Resort	Пляжний	вищий (5*)	0485728419
35	Falmont Ajman	Пляжний	вищий (5*)	0478721275
36	Continental 3 Sunny Beach	Пляжний	середній (3*)	9485128593
37	El Malkia Abu Dabbab	Пляжний	вищий (5*)	9450981275
38	Sentido Marlowe Palace Res...	Пляжний	вищий (5*)	0953812839
39	Sandies Baobab Beach	Пляжний	перший (4*)	9308912940
40	Varadero	Пляжний	середній (3*)	0937912837
41	Titanic Mardian Palace Execut...	Пляжний	вищий (5*)	0274912746
42	Vista Sol Punta Cana	Пляжний	перший (4*)	0841682393
43	Hawaii Caesar Dreams Aquap...	Пляжний	перший (4*)	0493812739
44	Tununc Dream Hotel	Пляжний	перший (4*)	0379487610
45	Arcanus Side Resort	Пляжний	вищий (5*)	0358012548

Детальна інформація

Тип готелю: Клас готелю:

Номер телефону:

Назва готелю:

Додати Редагувати Видалити

Рисунок 2.14 – Форма «Готелі»

Форма для редагування і додавання записів про готелі. Її зображено на рисунку 2.15.

Рисунок 2.15 – Форма «Форма редагування»

Для відображення та обробки даних про тури створено форму «Тури». Її зображено на рисунку 2.16.

Код туру	Назва туру	Тип туру	Сезон туру	Ціна	Тривалість	Кількість туристів	Країна	Код готелю	Дата туру
62	Akka Alinda	Пляжний	Осінь	13175	11	1	Єгипет	32	30.11.2021
63	Delphin Botanik	Пляжний	Осінь	67697	9	2	Туреччина	35	30.11.2021
64	The S Hotel Al Barsha	Пляжний	Осінь	26144	7	2	Дубай	38	29.11.2021
65	Samaki Lodge	Екзотичний	Зима	51214	7	2	Занзібар	42	25.12.2021
66	Karon Princess	Весільний	Зима	65579	7	2	Таїланд	36	14.12.2021
67	Catalonia Bavaro Beach	Екзотичний	Зима	77902	8	2	Домінікана	40	23.01.2022
68	Reveries Diving Village	Екзотичний	Зима	58433	11	2	Мальдіви	43	14.02.2022
69	Grand Liza	Ранні бронювання	Зима	14295	5	2	Туреччина	39	21.12.2021
70	Ufuk Pension	Екзотичний	Зима	13301	4	2	Капладокія	44	28.12.2021
71	Xanadu Island	Екзотичний	Зима	54836	9	2	Туреччина	36	23.02.2022
72	Beachwood	Пляжний	Зима	54711	8	2	Мальдіви	42	11.01.2022
73	Velana Blu	Екзотичний	Зима	56802	11	2	Мальдіви	36	17.12.2021
74	Viva Wyndham Dominica	Екзотичний	Зима	79224	8	2	Домінікана	41	12.02.2022
75	Sea Cleopatra Napa	Екзотичний	Зима	20680	7	2	Кіпр	44	21.01.2022
76	Holiday Cottage Guest Ho	Пляжний	Зима	55480	8	2	Мальдіви	39	12.12.2021

Рисунок 2.16 – Форма «Тури»

Форма для редагування і додавання записів про тури. Її зображено на рисунку 2.17.

Рисунок 2.17 – Форма редагування

Для відображення та обробки даних про замовлення та повернення створено форму «Замовлення/Повернення». Її зображено на рисунку 2.18.

Рисунок 2.18 – Форма «Замовлення/Повернення»

Форма для редагування і додавання записів про замовлення. Її зображено на рисунку 2.19.

Редагування замовлень

Дата замовлення 14.12.2021

Кількість путівок

Код працівника

Код клієнта

Код туру

Зберегти Скасувати

Рисунок 2.19 – Форма «Форма редагування»

Форма для редагування і додавання записів про повернення. Її зображено на рисунку 2.20.

Редагування даних ...

Дата повернення 14.12.2021

Кількість путівок

Причина повернення

Код замовлення

Зберегти Скасувати

Рисунок 2.20 – Форма «Форма редагування»

3 ПРОГРАМНА ЧАСТИНА

3.1 Опис та розроблення класів

З метою обміну даними між БД і програмним застосунком у вигляді цілих об'єктів, а не окремих значень, створено шість класів.

Клас `ClassWorkers` описаний у файлі `ClassWorkers.h` має наступний вигляд:

```
#pragma once
#include <stdexcept>
#include <String.h>

using namespace std;
using namespace System;

ref class ClassWorkers
{
    int IdWorker;
    String^ SurNameWorker;
    String^ BirthWorker;
    String^ EmailWorker;
    String^ AddressWorker;
    String^ PhoneWorker;
    String^ PositionWorker;
    String^ srcImage;

public:
    ClassWorkers() {}

    ClassWorkers(int i, String^ sn, String^ b, String^ e, String^ addr, String^
ph, String^ pos, String^ src)
    {
        setId(i);
        setSurName(sn);
        setBirth(b);
        setEmail(e);
        setAddress(addr);
        setPhone(ph);
        setPosition(pos);
        setSRC(src);
    }

    void setId(int id)
    {
        if (id > 0)
            this->IdWorker = id;
        else throw invalid_argument("Некоректне id!");
    }

    int getId()
    {
        return this->IdWorker;
    }

    void setSurName(String^ doc)
    {
        if (doc->Length != 0)
            this->SurNameWorker = doc;
        else throw invalid_argument("Неправильно введені дані ПІВ!");
    }
}
```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

```

String^ getName()
{
    return this->SurNameWorker;
}

void setBirth(String^ birth)
{
    if (birth->Length != 0)
        this->BirthWorker = birth;
    else throw invalid_argument("Некоректна дата народження!");
}
String^ getBirth()
{
    return this->BirthWorker;
}

void setEmail(String^ email)
{
    this->EmailWorker = email;
}
String^ getEmail()
{
    return this->EmailWorker;
}

void setAddress(String^ address)
{
    this->AddressWorker = address;
}
String^ getAddress()
{
    return this->AddressWorker;
}

void setPhone(String^ phone)
{
    if (phone->Length != 0)
        this->PhoneWorker = phone;
    else throw invalid_argument("Некоректний ввід номера телефону!");
}
String^ getPhone()
{
    return this->PhoneWorker;
}

void setPosition(String^ pos)
{
    if (pos->Length != 0)
        this->PositionWorker = pos;
    else throw invalid_argument("Некоректний ввід назви позиції!");
}
String^ getPosition()
{
    return this->PositionWorker;
}

void setSRC(String^ src)
{
    if (src->Length != 0)
        this->srcImage = src;
    else throw invalid_argument("Некоректний ввід шляху!");
}
String^ getSRC()
{

```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return this->srcImage;
    }
};

```

Клас ClassClients описаний у файлі ClassClients.h має наступний вигляд:

```

#pragma once
#include <stdexcept>
#include <String.h>

using namespace std;
using namespace System;

ref class ClassClients
{
    int IdClient;
    String^ SurNameClient;
    String^ BirthClient;
    String^ PhoneClient;
    String^ PassportClient;

public:
    ClassClients() {}

    ClassClients(int i, String^ sn, String^ b, String^ ph, String^ pass)
    {
        setId(i);
        setSurName(sn);
        setBirth(b);
        setPhone(ph);
        setPassport(pass);
    }

    void setId(int id)
    {
        if (id > 0)
            this->IdClient = id;
        else throw invalid_argument("Некоректне id!");
    }

    int getId()
    {
        return this->IdClient;
    }

    void setSurName(String^ doc)
    {
        if (doc->Length != 0)
            this->SurNameClient = doc;
        else throw invalid_argument("Неправильно введені дані ПІБ!");
    }

    String^ getName()
    {
        return this->SurNameClient;
    }

    void setBirth(String^ birth)
    {
        if (birth->Length != 0)
            this->BirthClient = birth;
        else throw invalid_argument("Некоректна дата народження!");
    }

    String^ getBirth()
    {
        return this->BirthClient;
    }
}

```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

```

void setPhone(String^ phone)
{
    if (phone->Length != 0)
        this->PhoneClient = phone;
    else throw invalid_argument("Некоректний ввід номера телефону!");
}
String^ getPhone()
{
    return this->PhoneClient;
}

void setPassport(String^ pass)
{
    if (pass->Length != 0)
        this->PassportClient = pass;
    else throw invalid_argument("Некоректний ввід номера паспорта!");
}
String^ getPassport()
{
    return this->PassportClient;
}
};

```

Клас ClassHotel описаний у файлі ClassHotel.h має наступний вигляд:

```

#pragma once
#include <stdexcept>
#include <String.h>

using namespace std;
using namespace System;

ref class ClassHotel
{
    int IdHotel;
    String^ NameHotel;
    String^ TypeHotel;
    String^ ClassOfHotel;
    String^ PhoneHotel;
    String^ srcImage;

public:
    ClassHotel() {}

    ClassHotel(int i, String^ n, String^ t, String^ cl, String^ ph, String^ src)
    {
        setId(i);
        setName(n);
        setType(t);
        setClassOfHotel(cl);
        setPhone(ph);
        setSRC(src);
    }

    void setId(int id)
    {
        if (id > 0)
            this->IdHotel = id;
        else throw invalid_argument("Некоректне id!");
    }

    int getId()
    {
        return this->IdHotel;
    }
}

```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

```

void setName(String^ doc)
{
    if (doc->Length != 0)
        this->NameHotel = doc;
    else throw invalid_argument("Неправильно введена назва готелю!");
}
String^ getName()
{
    return this->NameHotel;
}

void setType(String^ type)
{
    if (type->Length != 0)
        this->TypeHotel = type;
    else throw invalid_argument("Некоректний тип готелю!");
}
String^ getType()
{
    return this->TypeHotel;
}

void setClassOfHotel(String^ coh)
{
    if (coh->Length != 0)
        this->ClassOfHotel = coh;
    else throw invalid_argument("Некоректний клас готелю!");
}
String^ getClassOfHotel()
{
    return this->ClassOfHotel;
}

void setPhone(String^ phone)
{
    if (phone->Length != 0)
        this->PhoneHotel = phone;
    else throw invalid_argument("Некоректний ввід номера телефону!");
}
String^ getPhone()
{
    return this->PhoneHotel;
}

void setSRC(String^ src)
{
    if (src->Length != 0)
        this->srcImage = src;
    else throw invalid_argument("Некоректний ввід шляху!");
}
String^ getSRC()
{
    return this->srcImage;
}
};

```

Клас ClassTours описаний у файлі ClassTours.h має наступний вигляд:

```

#pragma once
#include <stdexcept>
#include <String.h>

using namespace std;
using namespace System;

```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

```

ref class ClassTours
{
    int IdTours;
    String^ NameTours;
    String^ TypeTours;
    String^ SeasonTours;
    double PriceTours;
    int DurationTours;
    int CountTourists;
    String^ CountriesTours;
    int IdHotel;
    String^ DateTours;

public:
    ClassTours() {}

    ClassTours(int i, String^ n, String^ t, String^ s, double pr, int d, int c,
String^ ct, int idh, String^ dt)
    {
        setId(i);
        setName(n);
        setType(t);
        setSeason(s);
        setPrice(pr);
        setDuration(d);
        setCountTourists(c);
        setCountries(ct);
        setHotel(idh);
        setDate(dt);
    }

    void setId(int id)
    {
        if (id > 0)
            this->IdTours = id;
        else throw invalid_argument("Некоректне id!");
    }
    int getId()
    {
        return this->IdTours;
    }

    void setName(String^ doc)
    {
        if (doc->Length != 0)
            this->NameTours = doc;
        else throw invalid_argument("Неправильно введена назва туру!");
    }
    String^ getName()
    {
        return this->NameTours;
    }

    void setType(String^ type)
    {
        if (type->Length != 0)
            this->TypeTours = type;
        else throw invalid_argument("Некоректний тип туру!");
    }
    String^ getType()
    {
        return this->TypeTours;
    }
}

```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

```

void setSeason(String^ coh)
{
    if (coh->Length != 0)
        this->SeasonTours = coh;
    else throw invalid_argument("Некоректна назва сезону туру!");
}
String^ getSeason()
{
    return this->SeasonTours;
}

void setPrice(double pr)
{
    if (pr > 0)
        this->PriceTours = pr;
    else throw invalid_argument("Некоректне значення ціни!");
}
double getPrice()
{
    return this->PriceTours;
}

void setDuration(int d)
{
    if (d > 0)
        this->DurationTours = d;
    else throw invalid_argument("Некоректне значення тривалості туру!");
}
int getDuration()
{
    return this->DurationTours;
}

void setCountTourists(int c)
{
    if (c > 0)
        this->CountTourists = c;
    else throw invalid_argument("Некоректне значення кількості
туристів!");
}
int getCountTourists()
{
    return this->CountTourists;
}

void setCountries(String^ ct)
{
    if (ct->Length != 0)
        this->CountriesTours = ct;
    else throw invalid_argument("Некоректна назва країни!");
}
String^ getCountries()
{
    return this->CountriesTours;
}

void setHotel(int id)
{
    if (id > 0)
        this->IdHotel = id;
    else throw invalid_argument("Некоректне значення id готелю!");
}
int getHotel()

```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		


```

{
    return this->IdHotel;
}

void setDate(String^ dt)
{
    this->DateTours = dt;
}
String^ getDate()
{
    return this->DateTours;
}
};

```

Клас ClassOrders описаний у файлі ClassOrders.h має наступний вигляд:

```

#pragma once
#include <stdexcept>
#include <String.h>

using namespace std;
using namespace System;

ref class ClassOrders
{
    int IdOrder;
    String^ DateTimeOrder;
    int CountOrders;
    double GeneralPriceOrder;
    int IdWorker;
    int IdClient;
    int IdTours;

public:
    ClassOrders() {}

    ClassOrders(int i, String^ dto, int co, double gpo, int idw, int idc, int
idt)
    {
        setId(i);
        setDateTime(dto);
        setCountOrders(co);
        setGeneralPriceOrder(gpo);
        setWorker(idw);
        setClient(idc);
        setTours(idt);
    }

    void setId(int id)
    {
        if (id > 0)
            this->IdOrder = id;
        else throw invalid_argument("Некоректне id!");
    }
    int getId()
    {
        return this->IdOrder;
    }

    void setDateTime(String^ dt)
    {
        this->DateTimeOrder = dt;
    }
    String^ getDateTime()
    {

```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return this->DateTimeOrder;
    }

void setCountOrders(int co)
{
    if (co > 0)
        this->CountOrders = co;
    else throw invalid_argument("Некоректна кількість замовлень!");
}
int getCountOrders()
{
    return this->CountOrders;
}

void setGeneralPriceOrder(double gpo)
{
    if (gpo > 0)
        this->GeneralPriceOrder = gpo;
    else throw invalid_argument("Некоректна сума замовлення!");
}
double getGeneralPriceOrder()
{
    return this->GeneralPriceOrder;
}

void setWorker(int idw)
{
    if (idw > 0)
        this->IdWorker = idw;
    else throw invalid_argument("Некоректне значення id працівника!");
}
int getWorker()
{
    return this->IdWorker;
}

void setClient(int idc)
{
    if (idc > 0)
        this->IdClient = idc;
    else throw invalid_argument("Некоректне значення id клієнта!");
}
int getClient()
{
    return this->IdClient;
}

void setTours(int idt)
{
    if (idt > 0)
        this->IdTours = idt;
    else throw invalid_argument("Некоректне значення id туру!");
}
int getTours()
{
    return this->IdTours;
}
};

```

Клас ClassComeback описаний у файлі ClassComeback.h має наступний вигляд:

```

#pragma once
#include <stdexcept>
#include <String.h>

```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using namespace std;
using namespace System;

ref class ClassComeback
{
    int IdComeback;
    String^ DateTimeComeback;
    int CountComeback;
    double PriceGiveComeback;
    String^ ReasonComeback;
    int IdOrder;

public:
    ClassComeback() {}

    ClassComeback(int i, String^ dto, int cc, double pgc, String^ rc, int ido)
    {
        setId(i);
        setDateTime(dto);
        setCountComeback(cc);
        setPriceGiveComeback(pgc);
        setReasonComeback(rc);
        setOrder(ido);
    }

    void setId(int id)
    {
        if (id > 0)
            this->IdComeback = id;
        else throw invalid_argument("Некоректне id!");
    }
    int getId()
    {
        return this->IdComeback;
    }

    void setDateTime(String^ dt)
    {
        this->DateTimeComeback = dt;
    }
    String^ getDateTime()
    {
        return this->DateTimeComeback;
    }

    void setCountComeback(int co)
    {
        if (co > 0)
            this->CountComeback = co;
        else throw invalid_argument("Некоректна кількість повернень!");
    }
    int getCountComeback()
    {
        return this->CountComeback;
    }

    void setPriceGiveComeback(double gpo)
    {
        if (gpo > 0)
            this->PriceGiveComeback = gpo;
        else throw invalid_argument("Некоректна сума повернення!");
    }
    double getPriceGiveComeback()
    {

```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return this->PriceGiveComeback;
    }

    void setReasonComeback(String^ rc)
    {
        this->ReasonComeback = rc;
    }
    String^ getReasonComeback()
    {
        return this->ReasonComeback;
    }

    void setOrder(int idc)
    {
        if (idc > 0)
            this->IdOrder = idc;
        else throw invalid_argument("Некоректне значення id замовлення!");
    }
    int getOrder()
    {
        return this->IdOrder;
    }
};

```

3.2 Опис та розроблення методів доступу до даних

Для отримання доступу до даних, що зберігаються в БД, використано технологію ADO.NET. Вона являє собою набір класів, через які можна відправляти запити до бази даних, встановлювати підключення, отримувати результат з БД і виконувати ряд інших операцій.

Створення зв'язку з базою даних реалізовано у конструкторах форм для забезпечення з'єднання з БД відразу після відкриття форми, наступним чином:

```

SqlCeEngine^ dbEngine = gcnew SqlCeEngine();
dbEngine->LocalConnectionString = "Data Source = TourAgency.sdf";
connection = gcnew SqlCeConnection(dbEngine->LocalConnectionString);
clients = gcnew List<ClassClients^>();
workers = gcnew List<ClassWorkers^>();
hotels = gcnew List<ClassHotel^>();
tours = gcnew List<ClassTours^>();
orders = gcnew List<ClassOrders^>();
comebacks = gcnew List<ClassComeback^>();

```

Для роботи з базою реалізовано наступні методи:

– метод оновлення таблиці dataGridView, призначених для відображення даних про клієнтів:

```

void updateTable() {
    int i = 0;
    dataGridView1->Rows->Clear();
    for each (ClassClients ^ dep in clients) {
        this->dataGridView1->Rows->Add();
        this->dataGridView1->Rows[i]->Cells[0]->Value = dep -> getId();
        this->dataGridView1->Rows[i]->Cells[1]->Value = dep ->
        getName();
    }
}

```

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        this->dataGridView1->Rows[i]->Cells[2]->Value = dep ->
getBirth();
        this->dataGridView1->Rows[i]->Cells[3]->Value = dep ->
getPhone();
        this->dataGridView1->Rows[i]->Cells[4]->Value = dep ->
getPassport();
        ++i;
    }
    if (dataGridView1->Rows->Count > 0)
    {
        buttonEditClient->Enabled = true;
        buttonRemoveClient->Enabled = true;
    }
    else
    {
        buttonEditClient->Enabled = false;
        buttonRemoveClient->Enabled = false;
    }
}

```

— метод повернення результатів виконання запитів з таблиці з даними про клієнтів:

```

void getQueryClients(String^ query) {
    connection->Open();
    SqlCeCommand^ command = gcnew SqlCeCommand(query, connection);
    SqlCeDataReader^ sqlRead = command->ExecuteReader();
    clients->Clear();
    while (sqlRead->Read()) {
        clients->Add(gcnew ClassClients(
            System::Convert::ToInt32(sqlRead["IdClient"]),
            System::Convert::ToString(sqlRead["SurNameClient"]),
            System::Convert::ToString(sqlRead["BirthClient"]),
            System::Convert::ToString(sqlRead["PhoneClient"]),
            System::Convert::ToString(sqlRead["PassportClient"])
        ));
    }
    sqlRead->Close();
    connection->Close();
    this->updateTable();
}

```

— метод повернення записів з даними клієнтів з БД:

```

void getDataClient() {
    connection->Open();
    SqlCeCommand^ command = connection->CreateCommand();
    command->CommandText = "SELECT * FROM Clients";
    SqlCeDataReader^ sqlRead = command->ExecuteReader();
    clients->Clear();
    while (sqlRead->Read()) {
        clients->Add(gcnew ClassClients(
            System::Convert::ToInt32(sqlRead["IdClient"]),
            System::Convert::ToString(sqlRead["SurNameClient"]),
            System::Convert::ToString(sqlRead["BirthClient"]),
            System::Convert::ToString(sqlRead["PhoneClient"]),
            System::Convert::ToString(sqlRead["PassportClient"])
        ));
    }
    connection->Close();
    this->updateTable();
}

```

Аналогічний програмний код до інших таблиць наведено у додатку А.

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

4.1 Приклад виконання програмного додатку з тестовими даними та отримання результатів

Після запуску програми «Travel Start» на екран викликається стартова форма з головним меню та панеллю інструментів, що зображена на рисунку 4.1. Дана форма дозволяє за допомогою кнопок в меню переходити на інші вбудовані форми з відображенням даних і здійснювати маніпулювання ними.

При запуску програми відкривається вкладка Головна, де немає жодних даних, а тільки стартовий вигляд.

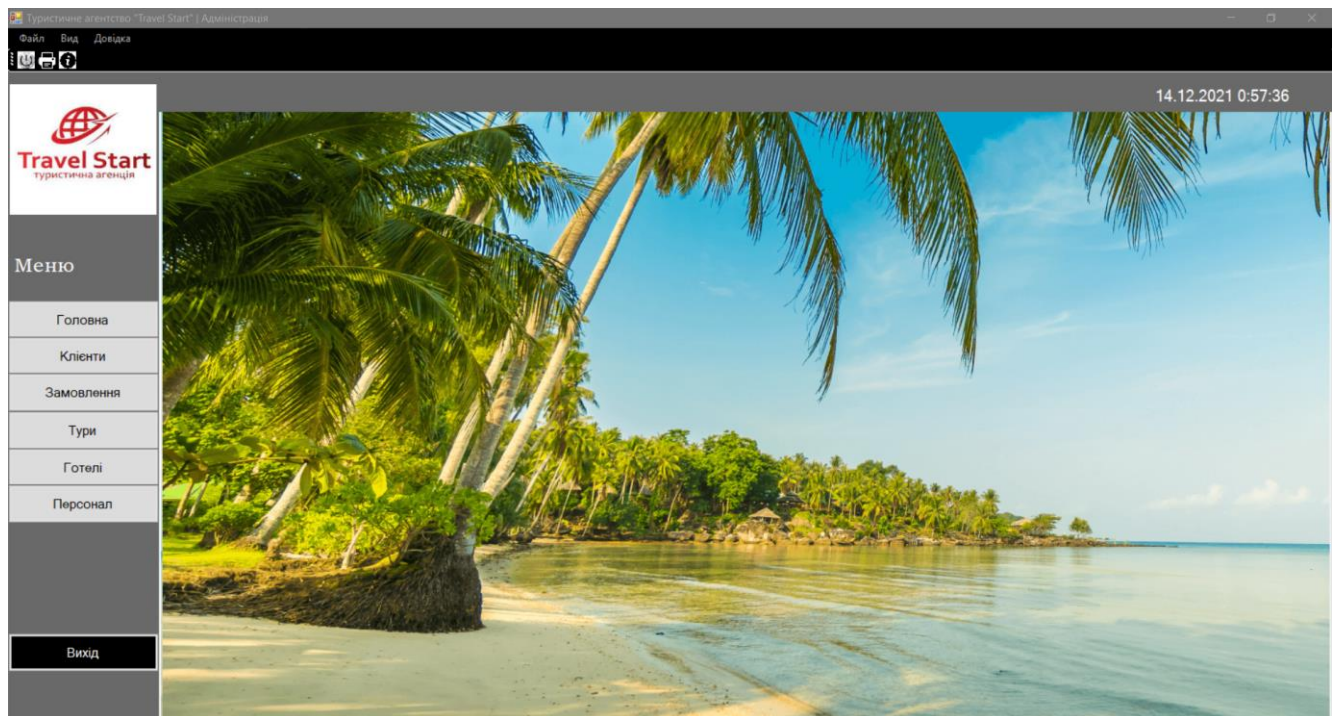


Рисунок 4.1 – Стартова форма

При виборі пункту меню Клієнти відображається таблиця з даними, де є також можливість додавати, редагувати та видаляти дані з неї. Окрім цього, реалізовано різнотипні пошуки (рисунок 4.2).

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

Туристичне агентство "Travel Start" | Адміністрація

14.12.2021 0:58:49

Travel Start
туристична агенція

Клієнти агентства

Список клієнтів

Номер	ПІБ клієнта	Дата народження	Номер телефону	Паспорт
35	Виткоther	2002-09-09	(240) 395-0349	2495849654
33	Параненч М.Р.	13.07.2002	0995738923	094895478
36	Анарацук В.І.	20.01.2002	0978475934	090087626
37	Войцех Ю.В.	11.26.2003	0993575934	085492758
38	Гаргат Я.В.	21.06.2003	0689747823	938593858
39	Горбата В.А.	08.01.2004	0667842175	857937257
40	Гринюк В.Р.	16.02.2003	0975938502	974037193
41	Савук А.А.	26.06.2003	0993751274	284728724
42	Іванюк Д.О.	18.05.2003	0969852738	947279429
43	Кельбас Б.Р.	16.06.2003	0679383825	078357248
44	Косак А.В.	21.04.2003	0976976346	398572985
45	Кусий М.В.	09.06.2003	0948375927	094758274
46	Лілик М.Т.	16.11.2002	0993575478	088982758
47	Морова Р.Т.	15.07.2003	0983751846	938590184
48	Максимен В.І.	26.02.2003	0938921947	984184869
49	Николає Д.М.	14.12.2003	0984391937	048628508
50	Параненч М.Р.	13.07.2002	0995738923	094895478

Редагувати Видалити

Додавання

ПІБ клієнта Дата народження

Номер телефону Номер паспорта

Додати

Пошук

Постійні клієнти ☐ Очистити фільтр

ПІБ клієнта Номер телефону

Рисунок 4.2 – Форма «Клієнти»

При виборі пункту меню Замовлення відображаються дві таблиці Замовлення та Повернення, в яких збережені відповідні дані. На формі реалізовано додавання замовлення, оформлення повернення, а також пошук за критеріями (рисунок 4.3).

Туристичне агентство "Travel Start" | Адміністрація

14.12.2021 1:16:00

Travel Start
туристична агенція

Облік замовлень та повернень путівок

Очистити фільтр

Пошук за критеріями

Тип оформлення Дата: від до Загальна ціна: мінімум максимум Знайти

Оформлені замовлення

Код замовлення	Дата замовлення	Кількість путівок	Загальна ціна	Код працівника	Код клієнта	Код туру
37	01.12.2021	2	23904	23	36	63
38	28.11.2021	1	46802	26	41	75
39	02.12.2021	3	97284	25	38	71
40	11.29.2021	1	23280	21	45	79
41	01.12.2021	2	32849	34	44	65
42	03.12.2021	3	94892	28	46	64
43	26.11.2021	3	98492	30	50	80
44	02.12.2021	2	30482	25	41	77
45	02.12.2021	1	39539	27	45	68
46	23.11.2021	2	38592	27	50	62
47	01.12.2021	3	69382	28	44	75
48	04.12.2021	1	27904	23	48	62
49	01.12.2021	2	63284	27	50	77
50	27.11.2021	1	28904	22	40	67
51	01.12.2021	2	63985	23	46	63
53	12.12.2021	3	62040	25	45	75

Оформити замовлення

Оформлені повернення

Код повернення	Дата повернення	Кількість путівок	Загальна ціна	Причина повернення	Код замовлення
12	10.12.2021	2	23948	особисте	46
13	10.12.2021	2	30482	особисте	38
14	08.12.2021	3	97284	особисте	43
15	06.12.2021	1	28904	особисте	44
16	12.12.2021	3	94892		51

Оформити повернення путівок

Рисунок 4.3 – Форма «Замовлення»

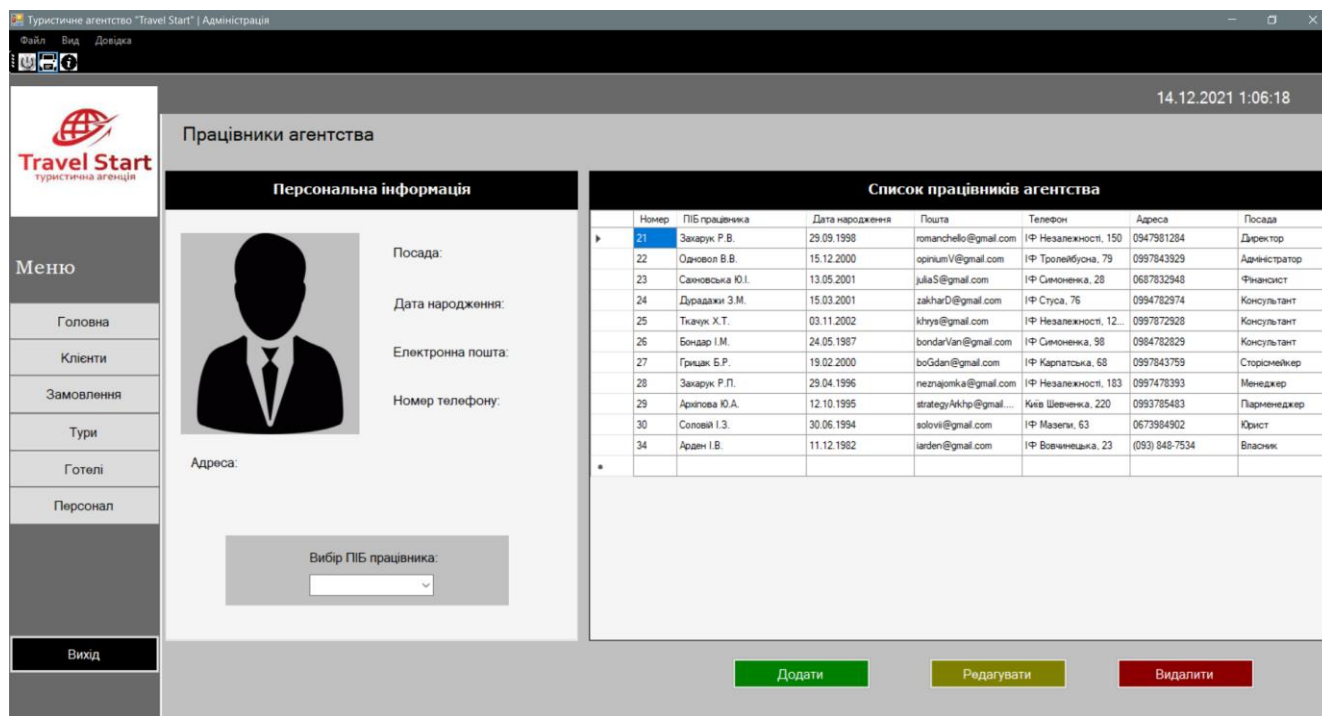
При виборі пункту меню Тури відображаються дані з відповідної таблиці. На формі реалізовано додавання, редагування та видалення необхідних даних, а також загальний пошук за різнотипними критеріями та швидкий пошук з передбаченими умовами. Окрім цього, забезпечено можливість швидкого переходу на офіційний веб-сайт туристичного агентства (рисунок 4.4).

Рисунок 4.4 – Форма «Тури»

При виборі пункту меню Готелі відображаються дані з таблиці, в якій збережена інформація про готелі. На формі реалізовано додавання, редагування та видалення необхідних даних, а також пошук за назвою готелю з виведенням повної інформації та зображенням (рисунок 4.5).

Рисунок 4.5 – Форма «Готелі»

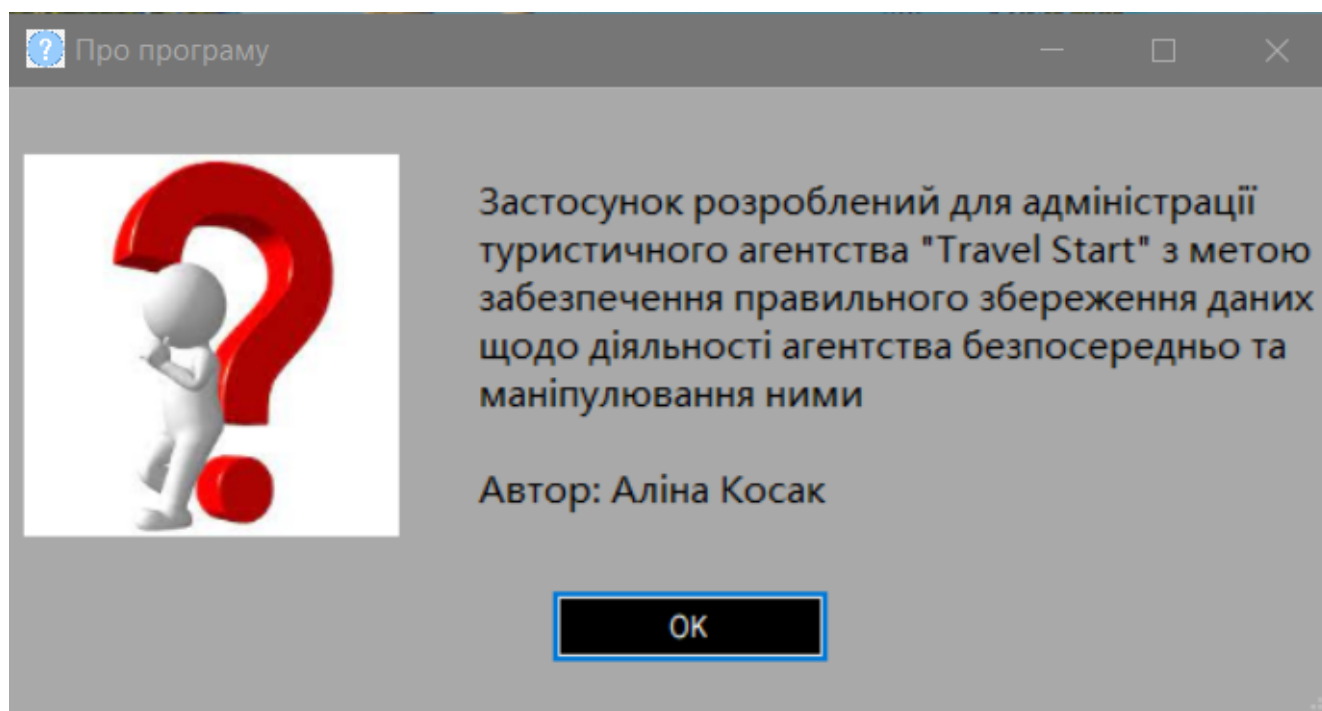
При виборі пункту меню Персонал відображаються дані з відповідної таблиці. На формі реалізовано додавання, редагування та видалення необхідних даних, а також пошук детальної інформації за ПІБ працівника (рисуюнок 4.6).



Номер	ПІБ працівника	Дата народження	Пошта	Телефон	Адреса	Посада
21	Закарук Р.В.	29.09.1998	romanchello@gmail.com	ІФ Незалежності, 150	0947981284	Директор
22	Оановей В.В.	15.12.2000	opinium.V@gmail.com	ІФ Тролейбуса, 79	0997843929	Адміністратор
23	Савенюська Ю.І.	13.05.2001	julia5@gmail.com	ІФ Симоненка, 28	0687832948	Фінансист
24	Дурадажи З.М.	15.03.2001	zakharD@gmail.com	ІФ Стуса, 76	0994782974	Консультант
25	Ткачук Х.Т.	03.11.2002	khyyt@gmail.com	ІФ Незалежності, 12...	0997872928	Консультант
26	Бондар І.М.	24.05.1987	bondarIvan@gmail.com	ІФ Симоненка, 98	0984782829	Консультант
27	Гришак Б.Р.	19.02.2000	boGdan@gmail.com	ІФ Карпатська, 68	0997843759	Сторонній
28	Закарук Р.П.	29.04.1996	neznajomka@gmail.com	ІФ Незалежності, 183	0997478393	Менеджер
29	Аріпова Ю.А.	12.10.1995	strategiAkh@gmail.com	Київ Шевченка, 220	0993785483	Парменеджер
30	Соловей І.З.	30.06.1994	solovii@gmail.com	ІФ Мазепи, 63	0673904902	Юрист
34	Ардан І.В.	11.12.1982	iarden@gmail.com	ІФ Вовчинецька, 23	(093) 848-7534	Власник

Рисуюнок 4.6 – Форма «Персонал»

При виборі пункту меню Довідка відображається нова форма з інформацією про програму та її автора (рисуюнок 4.7).



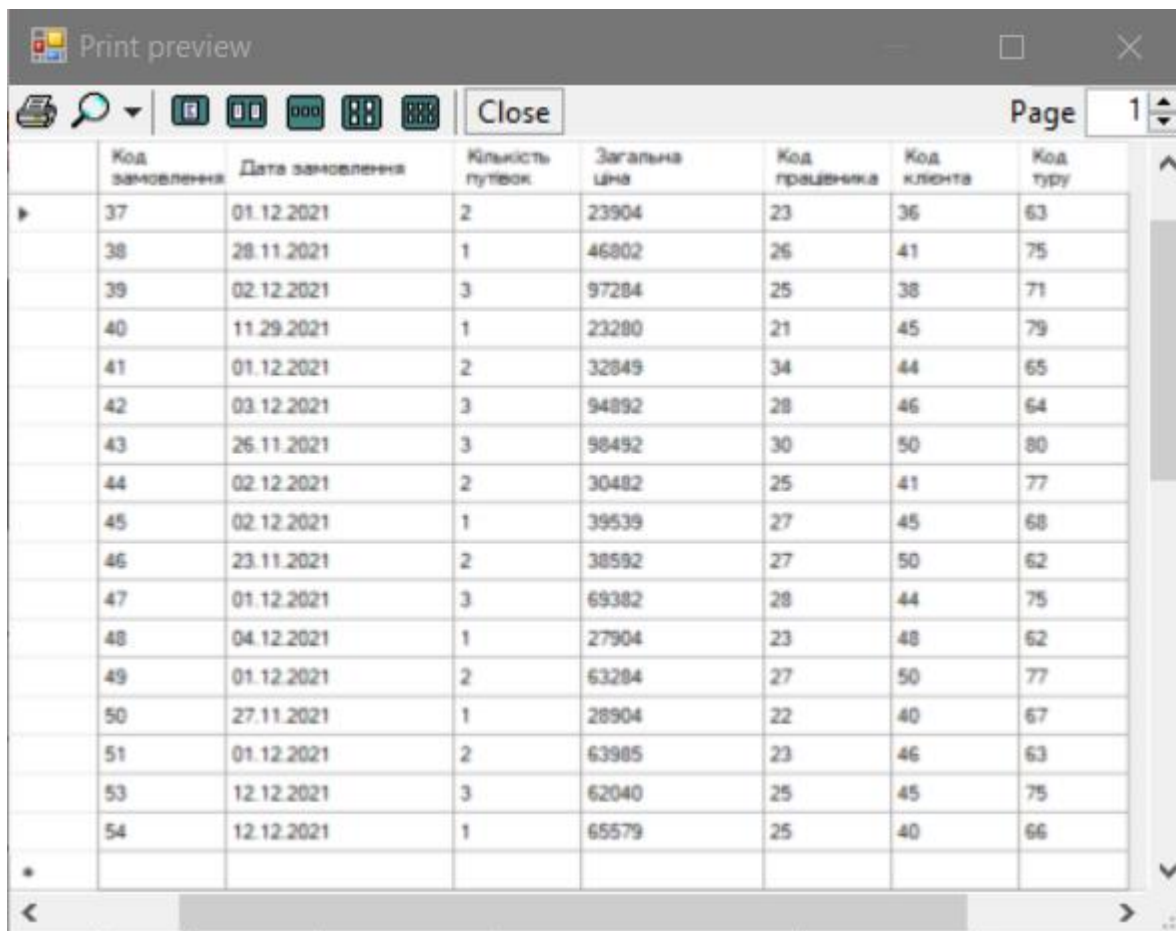
Застосунок розроблений для адміністрації туристичного агентства "Travel Start" з метою забезпечення правильного збереження даних щодо діяльності агентства безпосередньо та маніпулювання ними

Автор: Аліна Косак

OK

Рисуюнок 4.7 – Форма «Про програму»

При виборі пункту меню Друк відображається діалогове вікно з попереднім переглядом сторінки, яка має бути роздрукована. В даному програмному застосунку реалізовано друк звіту з таблиці, яка зберігає дані щодо замовлень (рисунок 4.8).



	Код замовлення	Дата замовлення	Кількість півток	Загальна ціна	Код працівника	Код клієнта	Код туру
▶	37	01.12.2021	2	23904	23	36	63
	38	28.11.2021	1	46802	26	41	75
	39	02.12.2021	3	97284	25	38	71
	40	11.29.2021	1	23280	21	45	79
	41	01.12.2021	2	32849	34	44	65
	42	03.12.2021	3	94892	28	46	64
	43	26.11.2021	3	98492	30	50	80
	44	02.12.2021	2	30482	25	41	77
	45	02.12.2021	1	39539	27	45	68
	46	23.11.2021	2	38592	27	50	62
	47	01.12.2021	3	69382	28	44	75
	48	04.12.2021	1	27904	23	48	62
	49	01.12.2021	2	63284	27	50	77
	50	27.11.2021	1	28904	22	40	67
	51	01.12.2021	2	63985	23	46	63
	53	12.12.2021	3	62040	25	45	75
	54	12.12.2021	1	65579	25	40	66
*							

Рисунок 4.8 – Форма «Друк»

ВИСНОВКИ

Під час розробки курсового проєкту було закріплено знання з дисципліни «Об'єктно-орієнтоване програмування».

У процесі виконання курсового проєктування було закріплено уявлення про особливості сучасних технологій проєктування і розробки баз даних при розробленні застосунку у середовищі Microsoft Visual Studio 2019. Набуто практичні навички розробки та описання програм з класами, написання конструкторів, деструкторів, методів (функцій-членів). Окрім цього, набуто умінь та навички успадкування класів та їх використання для побудови об'єктно-орієнтованих програм. Закріплено умінь опису структурних сутностей, можливих стосунків між ними, структурних аспектів архітектури, що використовуються для опису.

Крім набутих умінь, згаданих вище, було закріплено практичні навички у використанні класів, поліморфізму, зв'язувань, множинному та одинарному успадкуванні тощо. Сформовано практичні вміння у створенні інтерфейсів, опрацюванні записів з бази даних та створенні повноцінного Windows-додатку.

Окрім технічних вмінь у розробці коду, закріплено основи створення діаграм класів та їх опрацювання, використовуючи Visual Paradigm. Під час виконання практичних робіт було засвоєно: основи об'єктно-орієнтованого аналізу та проєктування систем, створення і реалізація класів, синтаксис, семантичні і функціональні властивості C++ для розроблення об'єктно-орієнтованих програм, засоби об'єктно-орієнтованого програмування Windows – застосунків, засоби створення інтерфейсу користувача для роботи з таблицями бази даних

Реалізація проєкту включала проходження ряду етапів, серед яких: уточнення постановки завдання, аналіз предметної області, проєктування структури БД, виділення сутностей, опис класів та їх реалізація відповідно, програмне проєктування, створення і розроблення графічного інтерфейсу, створення методів для взаємодії з БД.

При розробленні БД було виділено шість сутностей, які в подальшому були відображені як класи, а саме: Workers, Clients, Hotel, Tours, Orders, Comeback.

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтерфейс розроблявся зрозумілим для користувача. На головній формі розміщується панель навігації з кнопками, за допомогою яких можна переходити на інші форми. Крім цього, розміщено панель інструментів з піктограмами, які програмно реалізовані. На формах є можливість додавати, редагувати і видаляти інформацію з бази даних. У програмі використовуються інструменти для фільтрування інформації, яка відображається, а також інші різноманітні вибірки, застосовуючи при цьому складні запити з внутрішнім об'єднанням для збереження повноти даних. Також забезпечено можливість роботи з зображенням на різних формах з різними таблицями.

Окрім цього, реалізовано функцію друку звіту по замовленнях у вигляді таблиці, використовуючи загальну таблицю відображення даних. При цьому використовується кілька графічних функцій зі спеціальної бібліотеки System::Drawing, на основі яких робиться скріншот, якому задаються певні розміри, він центрується на сторінці для друку.

Проаналізовано роботу можливих методів для збереження та виведення зображення для кожного запису в певних таблицях. Після цього зроблено висновок, що передавати шлях і зберігати зображення найкраще, використовуючи властивість локації, якій присвоюється шлях збереження зображення (можливий навіть інтернет-ресурс).

Отже, під час виконання курсового проектування було засвоєно: основи об'єктно-орієнтованого аналізу та проектування систем, створення і реалізація класів, синтаксис, семантичні і функціональні властивості C++ для розроблення об'єктно-орієнтованих програм, засоби об'єктно-орієнтованого програмування Windows – застосунків, засоби створення інтерфейсу користувача для роботи з таблицями бази даних.

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Освітньо-професійна програма «Інженерія програмного забезпечення» від 31 серпня 2020 року. – URL: <https://kep.nung.edu.ua/lecturers#education>.
2. Пітчук Л.В. Курсове та дипломне проектування. Методичні вказівки. – URL: <https://kep.nung.edu.ua/lecturers#education>
3. Юрчишин В.М. Моделювання процесу розробки програмного забезпечення [Текст]: конспект лекцій/В. М. Юрчишин, В. Я. Піх – Івано-Франківськ : ІФНТУНГ, 2015. – 79 с. – (Каф. програмного забезпечення автоматизованих систем).
4. Habr - collaborative blog about IT. [Електронний ресурс] – URL: <https://habr.com/en/>
5. Dut.edu. [Електронний ресурс] – URL: http://www.dut.edu.ua/ua/news-1-626-8002-zastosuvannya-uml-chastina-3-diagrama-klasiv---class-diagram_kafedra-kompyuternih-nauk-ta-informaciyних-tehnologiy
6. Об'єктно-орієнтоване програмування. Частина 1. Основи об'єктно-орієнтованого програмування на мові C++: Навчальний посібник. / Д.В. Настенко, А. Б. Нестерко. – К.: НТУУ «КПІ», 2019. - 76с

					КП.ПІ-18-01.10.03.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		