

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:
**«Модель інформаційно-аналітичної системи
пасажиропотоків»**

Студента 2 курсу, 2м групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Муквіча Андрія
Леонідовича

Науковий керівник кандидат
педагогічних наук, доцент
кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Жирова Тетяна
Олександрівна

Гарант освітньої програми
доктор економічних наук,
професор кафедри інженерії
програмного забезпечення
та кібербезпеки

підпис гаранта

Токар Володимир
Володимирович

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«10» листопада 2020 р.

Завдання

на випускний кваліфікаційний проєкт студентів

Муквічу Андрію Леонідовичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Модель інформаційно-аналітичної системи пасажиропотоків»

Затверджена наказом ректора від «30» листопада 2020 р. № 3224

2. Строк здачі студентом закінченого проєкту 25 листопада 2021 р.

3. Цільова установка та вихідні дані до проєкту

Мета проєкту: комплексне дослідження програмного забезпечення моделювання та прогнозування, принципів його роботи та можливостей

Об'єкт дослідження: власне програмна модель та супутні компоненти, що потрібні для її роботи

Предмет дослідження є розробка програмного модулю, що здатен забезпечити роботу імітаційної моделі інформаційно-аналітичної системи на високому рівні

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТРАНСПОРТНА ІНФРАСТРУКТУРА ТА ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ: ВЗАЄМОЗВ'ЯЗОК ТА СТАТИСТИЧНИЙ АНАЛІЗ ПРОБЛЕМИ

1.1. Глобальне та статистичне обґрунтування проблеми

1.2. Аналіз програмних засобів, використовуваних для вирішення задачі

1.3. Висновок до розділу 1

РОЗДІЛ 2. ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ТА ВИВЧЕННЯ ІНСТРУМЕНТІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Сутність імітаційної моделі: принципи роботи та ключові поняття

2.2. Типи та принципи побудови імітаційних моделей

2.3. Вивчення інструментів для розробки пз та план дій щодо практичної реалізації моделі

2.4. Висновок до розділу 2

РОЗДІЛ 3. АНАЛІЗ ТЕХНІЧНОГО РІШЕННЯ ПРОГРАМНОГО ПРОДУКТУ «FORECASTER»

3.1. Початок роботи програми

3.2. Задання технічних характеристик терміналів

3.3. Меню імітації

3.4. Аналіз методів та процедур, що забезпечують роботу програми

3.5. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ТЕХНІЧНЕ ЗАВДАННЯ

ДОДАТКИ

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2020	21.09.2020
2.	<i>Розробка та затвердження завдання на проєкт магістра (стац/заоч)</i>	06.11.2020/ 22.12.2020	22.12.2020
3.	<i>Вступ та перелік літературних джерел</i>	27.02.2021	27.02.2021
4.	<i>Розробка технічного завдання</i>	20.03.2021	20.03.2021
5.	<i>Розділ 1. Транспортна інфраструктура та імітаційне моделювання: взаємозв'язок та статистичний аналіз проблеми</i>	16.04.2021	16.04.2021
6.	<i>Розділ 2. Основні теоретичні відомості про імітаційне моделювання та вивчення інструментів розробки програмного забезпечення</i>	24.05.2021	24.05.2021
7.	<i>Розділ 3. Аналіз технічного рішення програмного продукту «Forecaster»</i>	21.06.2021	21.06.2021
8.	<i>Розділ 4. (Об'єднано з Розділом 3)</i>	20.09.2021	20.09.2021
9.	<i>Розробка програми та методики тестування</i>	18.10.2021	18.10.2021
10.	<i>Написання наукової статті</i>	22.05.2021	22.05.2021
11.	<i>Керівництво користувача</i>	21.10.2021	21.10.2021
12.	<i>Висновки та пропозиції</i>	01.11.2021	01.11.2021
13.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	03.11.2021	03.11.2021
14.	<i>Підготовка автореферату та презентації доповіді</i>	03.11.2021	03.11.2021
15.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	22.11.2021 – 25.11.2021	22.11.2021
16.	<i>Здача зброшурованої випускного кваліфікаційного проєкту</i>	25.11.2021	25.11.2021
17.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	26.11.2021	26.11.2021
18.	<i>Підготовка до публічного захисту випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «10» листопада 2020 р.

8. Науковий керівник випускного кваліфікаційного проєкту Жирова Т.О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Токар В.В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Муквіч А.Л.

(прізвище, ініціали, підпис)

АНОТАЦІЯ

Відповідно до мети дослідження, дана робота присвячена комплексному дослідженню програмного забезпечення моделювання та прогнозування, принципів його роботи та можливостей для його впровадження на інфраструктурних об'єктах типу «аеропорт».

Розробку програмного модулю було здійснено у середовищі Microsoft Visual Studio мовою C#.

Робота з дисципліни «Об'єктно-орієнтоване програмування» на тему «Модель інформаційно-аналітичної системи пасажиропотоків» містить:

1. Загальний обсяг роботи: 47 сторінок
2. Кількість рисунків: 27
3. Кількість таблиць: 2

Ключові слова: C#, модель, аналітика, система, імітація

ABSTRACT

According to the purpose of research, this work is devoted to a comprehensive study of modeling and forecasting software, the principles of its operation and opportunities for its implementation in infrastructure facilities such as "airport".

The software module was developed in Microsoft Visual Studio in C #.

The work on the subject "Object-Oriented Programming" on the topic "Model of information and analytical system of passenger flows" consists of:

1. Total volume of work: 47 pages
2. Number of drawings: 27
3. Number of tables: 2

Keywords: C#, Visual Studio, model, analytics, systems, imitation

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ІАС – інформаційно-аналітична система

IATA (англ. International Air Transport Association) – Міжнародна Асоціація повітряного транспорту

					<i>КНТЕУ 121– 02-14.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. кафедри		Криворучко О.В.		29.09.20	<i>Модель інформаційно-аналітичної системи пасажиропотоків</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Жирова Т.О.		29.09.20		<i>ПС</i>	<i>2</i>	<i>45</i>
Гарант		Токар В.В.		29.09.20		<i>Факультет інформаційних технологій, 2 курс, 2м група</i>		
Розроб.		Муквіч А.Л.		29.09.20				
					<i>Перелік умовних скорочень</i>			

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1 ТРАНСПОРТНА ІНФРАСТРУКТУРА ТА ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ: ВЗАЄМОЗВ'ЯЗОК ТА СТАТИСТИЧНИЙ АНАЛІЗ ПРОБЛЕМИ	6
1.1. Глобальне та статистичне обґрунтування проблеми	6
1.2. Аналіз програмних засобів, використовуваних для вирішення задачі.....	11
1.3. Висновок до розділу 1	12
РОЗДІЛ 2 ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ТА ВИВЧЕННЯ ІНСТРУМЕНТІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	14
2.1. Сутність імітаційної моделі: принципи роботи та ключові поняття	14
2.2. Типи та принципи побудови імітаційних моделей	17
2.3. Вивчення інструментів для розробки ПЗ та план дій щодо практичної реалізації моделі.....	22
2.4. Висновок до розділу 2	25
РОЗДІЛ 3 АНАЛІЗ ТЕХНІЧНОГО РІШЕННЯ ПРОГРАМНОГО ПРОДУКТУ «FORECASTER»	26
3.1. Початок роботи програми.....	26
3.2. Задання технічних характеристик терміналів.....	27
3.3. Меню імітації	30
3.4. Аналіз методів та процедур, що забезпечують роботу програми.....	39
3.5. Висновки до розділу 3	42
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	45
ТЕХНІЧНЕ ЗАВДАННЯ	46
ДОДАТКИ	48
<i>Додаток А</i>	48
<i>Додаток Б</i>	49
<i>Додаток В</i>	50
<i>Додаток Г</i>	51

<i>КНТЕУ 121– 02-14.МР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>
Зав. кафедри		Криворучко О.В.		22.12.20
Керівник		Жирова Т.О.		22.12.20
Гарант		Токар В.В.		22.12.20
Розроб.		Муквіч А.Л.		22.12.20
<i>Модель інформаційно-аналітичної системи пасажиропотоків</i>				
<i>Зміст</i>				
		<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
		3	3	45
<i>Факультет інформаційних технологій, 2 курс, 2м група</i>				

ВСТУП

В процесі росту світової популяції та науково-технологічного прогресу, по мірі зростання кількості та потужності транспортних засобів та їх ролі у глобалізації, світовий пасажиропотік стрімко зростає. Так, наприклад, ряд найбільш завантажених аеропортів світу, таких як Міжнародний аеропорт Лос-Анджелеса, Міжнародний аеропорт Пекін-Шоуду або Міжнародний аеропорт Дубай, згідно даних Airport Council International, здатні обслуговувати більше 80 мільйонів пасажирів у рік, а деякі, такі як Міжнародний аеропорт імені Гартсфілда, вже перевершили позначку у 100 мільйонів. Однак для ефективного, безпечного та прогнозованого функціонування складних інфраструктурних споруджень, таких як аеропорт, вкрай необхідно володіти відповідними засобами, здатними прогнозувати результати діяльності подібних об'єктів на кілька кроків вперед у візуально доступному та інформативному вигляді.

Рішенням такого завдання може бути створення *імітаційної моделі інформаційно-аналітичної системи* (далі – ІАС). Основними перевагами підходу моделювання є можливість для розробника, конструктора, виробника або персоналу загалом заздалегідь передбачити можливий перебіг функціонування об'єкту ще до того, як сценарій його роботи буде безпосередньо реалізовано у реальному житті. Це дозволяє значно зменшити вірогідність непередбачуваних подій, про які без наявності моделі можна дізнатися лише по факту їх виникнення, при цьому не ризикуючи жодними втратами, адже весь «збиток» прийме на себе не сама система, а лише її

Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121– 02-14.МР</i>			
Зав. кафедри	Криворучко О.В.			27.02.21	<i>Модель інформаційно-аналітичної системи пасажиропотоків</i>	Стадія	Аркуш	Аркушів
Керівник	Жирова Т.О.			27.02.21		В	4	45
Гарант	Токар В.В.			27.02.21		Факультет інформаційних технологій, 2 курс, 2м група		
Розроб.	Муквіч А.Л.			27.02.21				
					<i>Вступ</i>			

віртуальна модель. Такий тип програм широко використовується для проектування таким чином, щоб кінцевий програмний продукт був максимально наближений до характеристик матеріального об'єкта, роботу якого він моделює. Моделі здатні змінювати свої початкові прогнози в реальному часі у разі, якщо користувач вносить зміни до початкових параметрів, або бажає оцінити ефект цих змін за допомогою моделі, перед тим як втілювати їх на об'єкті.

Головною *метою* дослідження є комплексне вивчення програмного забезпечення моделювання та прогнозування, принципів його роботи та можливостей для його впровадження на інфраструктурних об'єктах типу «аеропорт».

Об'єктом дослідження є власне програмна модель та супутні компоненти, що потрібні для її роботи.

Предметом дослідження є розробка програмного модулю, що здатен забезпечити роботу імітаційної моделі інформаційно-аналітичної системи на високому рівні, зменшити вірогідність перебоїв у її роботі та налагодити постійний контакт між усіма ключовими для роботи підприємства пристроями.

Задачами дослідження є вивчення предметної області інформаційно-аналітичних систем моделювання, а також практична реалізація проекту та його подальше впровадження у доступному для кінцевого користувача вигляді.

Головним *методом* дослідження є моделювання, що є наслідком необхідності побудови моделі у результаті роботи.

Практичне значення дослідження полягає у задоволенні потреб підприємств транспортної галузі в сучасних інформаційно-аналітичних системах, які б дали їм можливість оперативно відслідковувати усі зміни показників пасажиропотоку та іншої діяльності.

					<i>КНТЕУ 121– 02-14.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		5

РОЗДІЛ 1

ТРАНСПОРТНА ІНФРАСТРУКТУРА ТА ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ: ВЗАЄМОЗВ'ЯЗОК ТА СТАТИСТИЧНИЙ АНАЛІЗ ПРОБЛЕМИ

1.1. Глобальне та статистичне обґрунтування проблеми

Не дивлячись на тимчасову дисфункцію, викликану пандемією COVID-19, господарча цінність аеропортів та авіаційних перельотів як найбільш ефективного та часто єдиного засобу пересування не викликає сумнівів. Так, по даним Міжнародної асоціації повітряного транспорту (далі – IATA), світовий авіаційний пасажиропотік зріс з 2.5 млрд. чоловік у 2009 році до 4.5 мільярдів у 2018. Великою є і заповненість світового ринку новими авіасуднами – так, в середньому в небі постійно знаходиться від 11 до 13 тисяч різноманітних авіаційних суден, що забезпечує стабільний пасажиропотік усім провідним авіакомпаніям світу.

Аеропорти, як і інші транспортні об'єкти на зразок морських портів чи сухопутних вокзалів, є куди більш важливим економічним фактором, ніж компанії-перевізники, що їх експлуатують. Такі інфраструктурні установи являють собою об'єднувальний центр діяльності – їх роль багатогранна, оскільки внесок у забезпечення їх роботи вноситься з боку великої кількості учасників – клієнтів, компаній-посередників, обслуговуючих організацій і так далі (рис. 1.1).

					<i>КНТЕУ 121– 02-14.МР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. кафедри		Криворучко О.В.		16.04.21	<i>Модель інформаційно-аналітичної системи пасажиропотоків</i>	Стадія	Аркуш	Аркушів
Керівник		Жирова Т.О.		16.04.21		P1	6	45
Гарант		Токар В.В.		16.04.21		Факультет інформаційних технологій, 2 курс, 2м група		
Розроб.		Муквіч А.Л.		16.04.21				
					<i>Транспортна інфраструктура та імітаційне моделювання: взаємозв'язок та статистичний аналіз проблеми</i>			



Рис. 1.1. Приклад універсальної ролі аеропорту

На значну роль транспортної галузі вказує і її фінансовий результат, який вона генерує. Так, фінансові показники галузі перевезень сягають позначки у 3.8 трильйонів \$, здійснивши приріст більш ніж вдвічі в порівнянні з минулим століття, із них близько 700 мільярдів \$ припадає на галузь авіаційних перевезень (близько 20%).

Однак управління настільки габаритними, у вищій мірі багатолюдними та багатокомпонентними об'єктами, їх правильна експлуатація є завданням підвищеної складності, яке дуже важко, а іноді й повністю неможливо вирішити без підтримки обчислювальних засобів та реалізації приблизної схеми роботи інфраструктурного об'єкту в цифровому вигляді за допомогою засобів програмного забезпечення. Таке рішення, реалізоване в тому чи іншому вигляді, здатне, як мінімум, забезпечити наступні переваги:

1. **Підвищена безпека.** Актуальність цієї точки зору пояснюється розумінням того, що транспортна інфраструктура є місцем особливо великого скупчення людей – як пасажирів так й обслуговуючого персоналу – а значить, можливість попереднього передбачення її роботи з боку програмних засобів є серйозним підґрунтям на шляху до убезпеченості від нештатних або потенційно небезпечних ситуацій.
2. **Пришвидшення прийняття рішень.** Якщо керуючий персонал, експлуатуючий той чи інший транспортний об'єкт, отримає якісний

прогноз щодо перебігу його діяльності на заданий період наперед, це суттєво полегшить локальний менеджмент та швидкість прийняття рішень.

3. Підвищення комфорту. Автоматизація процесу прогнозування звільнить людину від ряду функцій, обчислення яких раніше довелося б робити самостійно, давши персоналу час та можливість зосередитись на більш важливих завданнях (рис 1.2).



Рис. 1.2. Порівняння швидкості отримання підсумкового прогнозу за допомогою моделі так без неї (згідно даних групи AnyLogic)

Найбільш зручним та ефективним способом реалізації такої схеми, здатної відтворити головні подробиці роботи досліджуваного об'єкту у цифровому вигляді, а потім спрогнозувати результати їх діяльності на

майбутній період, є *імітаційна модель інформаційно-аналітичної системи*. Як вже зазначалося, у випадку інфраструктури застосування подібного ПЗ є особливо актуальним, враховуючи що ефективна робота інфраструктури напряму впливає на комфорт та навіть безпеку людини, яка користується її можливостями. Імітаційна модель стає більш необхідною на особливо складних її зразках, таких як аеропорт, морський порт або вокзал. В умовах збільшення трафіку та площі подібних об'єктів стає очевидною нездатність людини самотійно врахувати усі технологічні аспекти їх функціонування, спрогнозувати майбутні показники пасажиропотоку та бути в готовності швидко вносити зміни до базових прогнозів у відповідності до зовнішніх умов, які змінюються та можуть вплинути на такий об'єкт (наприклад, зміна погодних умов, аварія чи затримка поїзда/літака і т.д.). З ціллю уникнення подібних ситуацій використання моделі моделювання пасажиропотоку, яка б працювала в реальному часі, є найбільш вигідним виходом з уразливого положення *невизначеності та низької прогнозованості* майбутнього (рис 1.3).



Рис. 1.3. Графічна ілюстрація ефективності моделювання

					КНТЕУ 121–02-14.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

У якості *відносного недоліку* імітаційного моделювання можна відзначити, що кожне рішення, отримане таким способом, має обмежений характер, оскільки воно відповідає фіксованим елементам структури, алгоритмам, значенням параметрів і т.д. Незважаючи на важливі відмінності, межа між цифровими моделями багато в чому умовна, тому що всі вони використовують математичні моделі та обчислювальні процедури. Інакше кажучи, математичні моделі є однією з найважливіших основ імітації.

Яскравий доказ доцільності моделювання - успішна реалізація імітаційної моделі на складному та важливому інфраструктурній об'єкті – реалізація такої моделі в рамках супроводження роботи аеропорту Франкфурта (рис. 1.4).



Рис. 1.4. Ілюстрація головного терміналу Франкфуртського аеропорту

У певний момент компанія Fraport AG, експлуататор летовища, зіткнулася з обмеженнями щодо подальшого будівництва своєї інфраструктури, тому було вирішено розробити систему керування пасажиропотоком для збільшення ефективності використання існуючих площ та підвищення якості обслуговування пасажирів. Наприклад, це

					<i>КНТЕУ 121– 02-14.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		10

завдання могло бути вирішено за допомогою скорочення затримок пасажиропотоків. Загалом, робота системи управління терміналом, розроблена в результаті, заснована на вимірі поточного пасажиропотоку та прогнозуванні його змін.

1.2. Аналіз програмних засобів, використовуваних для вирішення задачі

1.2.1. Загальні відомості

Найменування розроблюваного проекту – програма «The Forecaster», або «Прогнозист».

Головна сутність технічного завдання полягає у розробці програмного забезпечення, здатного організувати ефективний в реальному часі прогноз пасажиропотоку аеропорту на основі його властивостей та технічних можливостей, введених користувачем раніше у якості базових параметрів. ПЗ повинно надавати користувачу можливість самостійно вибирати, скільки терміналів міститиме летовище та які характеристики будуть притаманними кожному із них; результатом роботи має стати перелік матеріалів у вигляді графіків та діаграм, в яких має бути засвідчено результати симуляції у вигляді прогнозу пасажиропотоку як летовища загалом, так і окремих його терміналів зокрема.

1.2.2. Мета та призначення ПЗ

Мета програмного забезпечення – створити ефективний програмний модуль, сумісний з реальними алгоритмами роботи аеропортів, здатний прогнозувати результати діяльності аеропортів у реальному часі з метою полегшення управління ними, незалежно від їх основних параметрів та характеристик проектування.

					<i>КНТЕУ 121–02-14.МР</i>	Аркуш
						11
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

1.2.3. Вимоги до системи

Вимоги до системи в цілому виокремлюють загальні концепції її роботи. До них належать наступні вимоги:

- I. Програмний продукт має бути портативним та не вимагати додаткового встановлення на комп'ютери підприємства. Користувачі повинні отримати здатність приступити до роботи, щойно виконуваний файл програмного продукту потрапить на їх робочі комп'ютери.
- II. Програма повинна містити можливість вибору користувачем головних характеристик аеропорту, діяльність якого піддаватиметься моделювання; до них належать такі параметри, як кількість терміналів, кількість реєстраційних стійок в кожному з них, стан транспортного сполучення з аеропортом і т.д. (див. Розділ 3); вікно вибору режиму повинно бути доступним й інтуїтивно зрозумілим.
- III. Програма повинна бути здатною змінювати свої першопочаткові прогнози у разі, якщо користувач вносить зміни вже після того, як симуляція та моделювання розпочалися.
- IV. Результат роботи має бути розгорнутим і містити повні свідчення щодо прогнозованого пасажиропотоку аеропорту на добу вперед. Результат має бути представленим у вигляді графіків й діаграм.
- V. Швидкість моделювання повинна знаходитися у допустимих кордонах для імітаційних моделей – від 5 до 10 хвилин або менше.

1.3. Висновок до розділу 1

Таким чином, можна зробити висновок, що обґрунтування для створення імітаційної моделі такого типу є більш ніж доцільним і базується в

					<i>КНТЕУ 121–02-14.МР</i>	Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата		

першу чергу на статистичних показниках та світових тенденціях, які переконливо доводять необхідність розвитку та удосконалення імітаційних моделей як одного з небагатьох засобів управління, що здатен забезпечити управління тим об'ємом складних функцій, які виконує об'єкт інфраструктури протягом технологічного циклу своєї діяльності.

Подальші дослідження в предметній області потенційно здатні призвести до створення ефективної, потужної та безпечної для пасажирів моделі керування аеропортом.

					<i>КНТЕУ 121–02-14.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

РОЗДІЛ 2.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ТА ВИВЧЕННЯ ІНСТРУМЕНТІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Сутність імітаційної моделі: принципи роботи та ключові поняття

Імітаційне моделювання являє собою метод дослідження, сутність якого полягає у заміні системи, яку потрібно дослідити, її моделлю, здатною в достатній точності описати у собі основні принципи роботи цієї системи та врешті-решт отримати стійку статистику шляхом проведення експериментів.

Проведення експериментів з моделлю називається **імітацією**. До імітаційного моделювання вдаються у наступних випадках:

- 1) Проведення дослідів на реальному об'єкті надто дорого або така можливість не передбачена технічно;
- 2) Вимагається дослідження системи у реальному часі – побудова моделі менш складного типу неможлива;
- 3) У системі, яку потрібно дослідити, присутні взаємозв'язки, наслідки, нелінійності, а також випадкові величини.

Ціль імітаційного моделювання – відтворити поведінку досліджуваної системи і обґрунтувати основні закономірності її роботи, що досягається шляхом аналізу найбільш істотних взаємозв'язків між елементами моделі.

Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 – 02-14.МР</i>			
Зав. кафедри	Криворучко О.В.			24.05.21	<i>Модель інформаційно-аналітичної системи пасажиропотоків</i>	Стадія	Аркуш	Аркушів
Керівник	Жирова Т.О.			24.05.21		P2	14	45
Гарант	Токар В.В.			24.05.21				
Розроб.	Муквіч А.Л.			24.05.21				
					<i>Основні теоретичні відомості про імітаційне моделювання та вивчення інструментів розробки програмного забезпечення</i>	Факультет інформаційних технологій, 2 курс, 2м група		

Великою перевагою імітаційного моделювання є його властивість здійснювати процес моделювання у режимі реального часу. У разі, якщо така необхідність виникає, час у моделі можна піддати керуванню – пришвидшувати його плин у разі роботи з довгостроковими процесами та уповільнювати у разі, якщо процеси короткострокові. Як зазначалося вище, користувачу дозволено імітувати поведінку тих об'єктів, реальні експерименти з якими дорогі, неможливі чи небезпечні. Зі збільшенням ролі персонального комп'ютера, як і з розвитком комп'ютерної техніки загалом, керівництво великою кількістю технологічних процесів, як правило, супроводжується комп'ютерним імітаційним моделюванням, реалізація якого може бути здійснена у 3D, 2D форматах, у вигляді живих графіків і т.д.

Загальний перелік застосування імітаційного моделювання широкий, однак зазвичай сфера його застосування вкладається у наступні галузі:

- Бізнес процеси, бізнес-симуляція
- Бойові дії
- Динаміка населення
- Дорожній рух, залізничні перевезення, авіаперевезення
- ІТ-інфраструктура
- Математичне моделювання історичних процесів
- Логістика, транспортування
- Пішохідна динаміка
- Виробництво, ринок та конкуренція
- Сервісні центри
- Ланцюжки поставок
- Вуличний рух
- Управління проектами
- Економіка охорони здоров'я, екосистема
- Інформаційна безпека, релейний захист

					<i>КНТЕУ 121–02-14.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

Введення змінних	на виході системи	на вході системи
Спосіб знаходження рішення	алгоритм обчислень	експериментальний аналіз досліджуваного об'єкту за допомогою комп'ютера
Кількість випробувань для виводу рішення	одне випробування	допускається серія випробувань
Характер рішення	точні рішення	рішення на основі вірогідності
Знаходження оптимального рішення у разі побудови моделі	гарантовано	не гарантовано
Залучення спеціалізованого програмного забезпечення	небагато	багато (допускається варіативність програмних рішень)
Рішення складних завдань	важко (за рахунок обмеженості функціоналу моделі)	доступно
Застосовуваність на практиці	обмежено	необмежено
Близькість моделі до об'єкту, що вивчається	достатньо ускладнена	максимально наближена
Клас об'єктів, що вивчаються	звужений	розширений

2.2. Типи та принципи побудови імітаційних моделей

Наразі виділяються три головних підходи імітаційного моделювання.

Агентне моделювання — достатньо молодий по віку тип моделювання, що знайшов найбільше застосування у роботі з децентралізованими системами. Сутність підходу – дослідження роботи не всієї системи відразу, а лише обраного ряду її активних членів. Мета агентних моделей — отримати уявлення про правила, згідно яким функціонує система, шляхом визначення засад поведінки груп окремих її

						Аркуш
						17
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121–02-14.МР	

компонентів індивідуально, та дослідження взаємодії цих компонентів між собою (рис. 2.1).

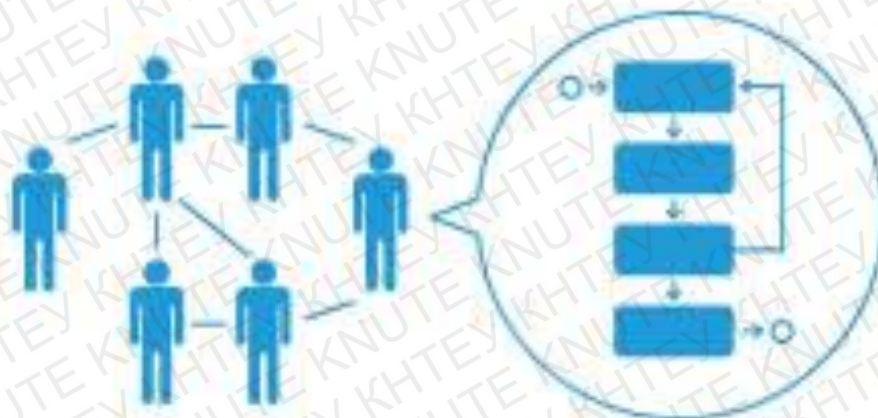


Рис. 2.1. Агентне моделювання – ставка на індивідуальні складові системи (зображено у вигляді фігурок людей)

На початку роботи агентного моделювання спершу відбувається процес ідентифікації активних компонентів системи, які й мають назву **агентів**. У якості агентів може виступати будь-яка сутність, від якої залежить успішна робота досліджуваної системи: домогосподарства, люди, транспорт, обладнання, товари та компанії. Щойно агенти ідентифіковані, відбувається встановлення зв'язків між ними і формалізується загальний вигляд середовища, в рамках якого вони функціонують. Індивідуальна активність кожного з агентів визначає поведінку системи у глобальному сенсі, чого виявляється достатньо, щоб зрозуміти закономірності та сформулювати модель.

Агентне моделювання, як правило, не оперує надто складними сутностями і дає можливість зосередитися тільки на окремих об'єктах, правилах їх поведінки та взаємодії. Агентна модель, що виходить, представляє собою низку взаємодіючих між собою об'єктів, що відображають зв'язки у форматі, максимально наближеному до реалій справжнього світу. За

рахунок цього агентне моделювання може виявитися вкрай корисним для розуміння та контролю складних соціальних й бізнес-процесів.

Дискретно-подійне моделювання — підхід до моделювання, що дає можливість відійти від безперервного обліку всіх можливих подій і аналізувати лише найбільш значущі події системи, що моделюється, такі як: «очікування», «обробка», «завантаження», «розвантаження» і т.д.

Причина впровадження даного підходу полягає в тому, що значну частку бізнес-процесів буває зручно описувати саме як послідовність окремих одна від одної дискретних подій. Наприклад, транспортний засіб прибуває на базу, прямує до пункту розвантаження, розвантажується та покидає місцевість. Саме задля моделювання подібних процесів іноді застосовується дискретно-подійний підхід моделювання.

При такому підході, для демонстрації, прямування залізничного складу з точки 1 до точки 2 представиться моделлю як дві окремі події: відправлення та прибуття. Сам рух потягу є лише затримкою події (або між подіями), але не подією як такою. Ці події та рух між ними можна представити у графічному вигляді.

Моделювання дискретно-подійним методом означає, що досліджуваний об'єкт чи система моделюються на неглибокому рівні абстракції. Глибинні деталі, такі як технічні характеристики транспортного засобу, зазвичай моделлю не враховуються. Дискретно-подійне моделювання популярне у виробничій галузі, при вирішенні питань логістики в та охороні здоров'я (рис 2.2).



Рис. 2.2. Дискретно-подійне моделювання – ставка на відтворення ключових подій (представлено у вигляді блоків та ліній)

					КНТЕУ 121–02-14.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		19

Системна динаміка - парадигма моделювання, сутність якої полягає в побудові графіків й діаграм для дослідження глобальних впливів одних параметрів системи на інші у режимі реального часу. Такий вид моделювання є найбільш вдалим рішенням, якщо користувач бажає наочно дослідити причинно-наслідковий зв'язок між роботою системи та впливом на неї зовнішніх (або внутрішніх) явищ та подій.

Властивість підходу полягає в тому, що системна динаміка дозволяє моделювати складні системи на достатньо абстрактному рівні, не беручи до уваги маленькі деталі, такі як індивідуальні властивості окремих товарів, предметів або людей. Моделі такого типу дають можливість отримати масштабне уявлення про систему та можуть використовуватися для стратегічного планування. Наприклад, коли аеропорт розробляє модель свого пасажиропотоку, йому зовсім необов'язково моделювати поведінку кожного пасажера окремо. Однак йому необхідно врахувати інші показники більш високого рівня, такі як, наприклад, стан терміналів, їх кількість, ступінь розвитку інфраструктури використовуваної для підвозу пасажирів до летовища і т.д.

Головний спосіб представлення результатів діяльності моделі – графіки та діаграми (рис 2.3).



Рис. 2.3. Системна динаміка – ставка на графіки та діаграми відслідковування процесів в живому часі

						КНТЕУ 121– 02-14.МР	Аркуш
							20
Зм.	Аркуш	№ докум	Підпис	Дата			

Найбільш поширений та інтуїтивно зрозумілий спосіб відтворення засад роботи імітаційної моделі – **модульний**. Нижче приведено схематичний принцип роботи моделі з його допомогою (рис. 2.4).



Рис. 2.4. Схема взаємодії блоків імітаційної моделі

Алгоритми моделі реалізуються у блоці імітації процесів функціонування об'єкта, яка знаходиться у блоці 2. В даному секторі відміряється відлік модельного часу, опрацьовуються логічні та динамічні засади взаємодії між елементами моделі, здійснюється процес експериментування та збираються дані, необхідні для прогнозу та збору оцінок функціонування об'єкта.

Блок імітації випадкових впливів (блок 1) існує для створення значень випадкових величин і процесів, що особливо важливо для моделі типу «системна динаміка», що працює з абстрактними величинами. До його складу входять засоби реалізації алгоритмів моделювання випадкових впливів із необхідними властивостями. У блоці обробки результатів імітації (блок 3) відбувається розрахунок поточних, отриманих моделлю на початку її роботи, та підсумкових (прогнозованих) значень характеристик системи, що є результатами експериментів з моделлю.

2.3. Вивчення інструментів для розробки ПЗ та план дій щодо практичної реалізації моделі

Для розробки ПЗ «Airport Forecaster», здатного втілити у власному програмному модулі імітаційну модель моделювання пасажиропотоку аеропорту, було обрано середовище програмування Microsoft Visual Studio 2019, платформа – ASP.NET. Тип моделі – **системна динаміка**. Вибір на користь саме цього середовища було зроблено з кількох причин:

1. Універсальність мови C#, як і середовища Visual Studio, та їх повна придатність до розгортання й програмування графіків і діаграм, що особливо важливо для зазначеного типу імітаційного моделювання;
2. Придатність та зручність середовища Visual Studio C# для здійснення відладки програми, а також транспортування її на інші комп'ютери, навіть якщо на них відсутнє середовище програмування;
3. Інтуїтивна зрозумілість середовища програмування та засобів, що воно містить.

В цілому, Visual Studio C# надає весь необхідний перелік засобів для створення багатьох різноманітних прикладів ПЗ, і дана імітаційна модель не є винятком.

Для реалізації відображення та руху графіків, діаграм та інших елементів, що складають ядро імітаційної моделі, використовуватиметься такий елемент, як **Chart**, прямий переклад з англійського – «діаграма». Елемент є вбудованим членом панелі елементів Microsoft Visual Studio 2019, та призначений для побудови графіків, діаграм, схем, кривих та усіх інших компонентів з цього ряду, здатних продемонструвати зміну досліджуваного процесу в динаміці. Даний компонент є основоположним у розроблюваному програмному забезпеченні та складає ядро програми (рис. 2.5).

						Аркуш
					<i>КНТЕУ 121–02-14.МР</i>	22
Зм.	Аркуш	№ докум	Підпис	Дата		

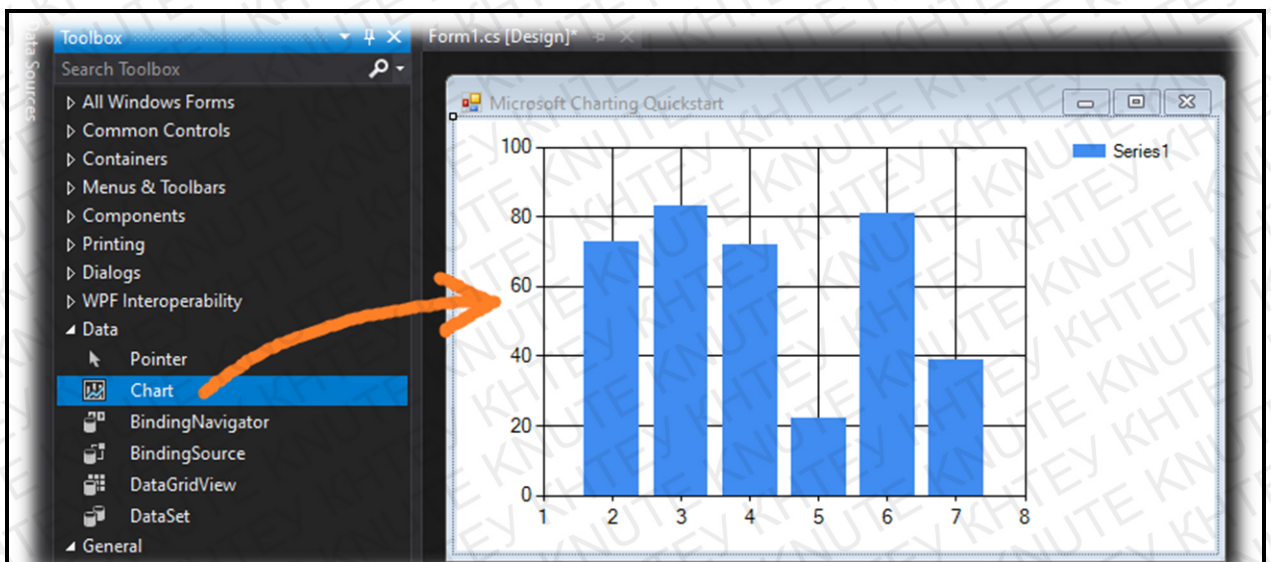


Рис. 2.5. Базовий вигляд елемента Chart

Сутність роботи програмного забезпечення – класична для роботи моделей типу «системна динаміка». Користувач повинен ввести базові для моделі статичні величини, що описують аеропорт загалом – передусім, кількість його терміналів та їх технічні характеристики. Після цього дана інформація повинна бути передана у елемент Chart та інтерпретована на екрані у графічному вигляді. Результат роботи розроблюваної імітаційної моделі повинен складати мережу графічних діаграм, які б містили прогноз пасажиропотоку летовища на 24 години вперед. Фінальні результати, отримувані в кінці циклу моделювання, повинні давати загальне уявлення того, яких меж може сягнути пасажиропотік аеропорту з урахуванням технологічних характеристик терміналу, інфраструктури аеропорту, а також з урахуванням різного роду випадковостей та нештатних ситуацій на зразок відміни рейсів чи погодних змін.

Послідовність необхідних дій в процесі інтерпретації даних у графічний вигляд та створення графіку розміщено у Таблиці 2.2.

2.4. Висновок до розділу 2

У даному розділі було проведено більш глибокий аналіз явища імітаційних моделей, зокрема, їх типологію, способи організації, а також розглянуто перелік технічних дій, необхідних для досягнення побудови графіку моделі як її ключового компоненту

					<i>КНТЕУ 121–02-14.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		25

РОЗДІЛ 3

АНАЛІЗ ТЕХНІЧНОГО РІШЕННЯ ПРОГРАМНОГО ПРОДУКТУ «FORECASTER»

Даний розділ задля зручності та кращої цілісності даних об'єднає у собі, окрім аналізу алгоритмів побудови ПЗ, також і поетапний опис роботи програмного продукту у вигляді тексту та скріншотів.

3.1. Початок роботи програми

Програмне забезпечення зустрічає користувача з меню вибору транспортного об'єкту, пасажиропотік якого буде моделюватися з допомогою програми. Як і вказано у Розділі 2, цільовим об'єктом моделювання є підприємство типу «аеропорт», в той час як, тим не менше, інші транспортні структури на зразок вокзалу чи морського порту є об'єктами подальших досліджень широкої та об'ємної на ресурси предметної області імітаційного моделювання.

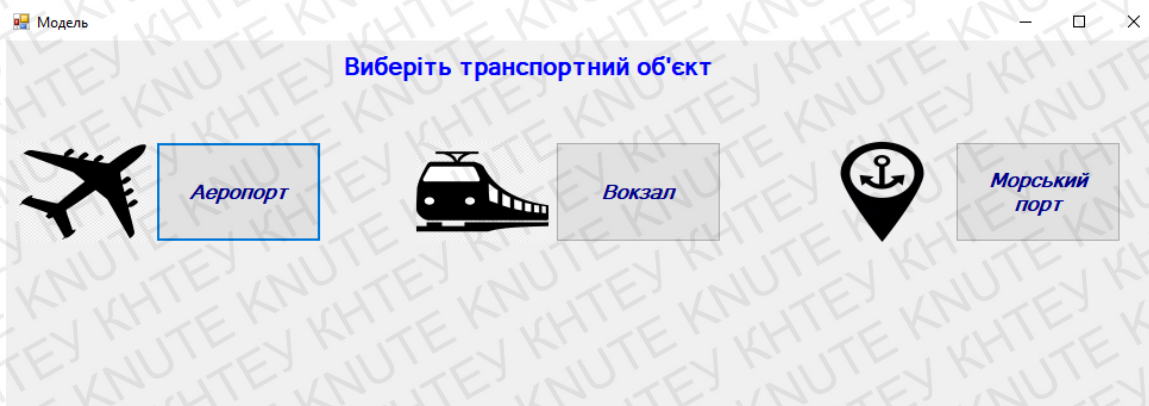


Рис. 3.1. Меню вибору типу транспортного об'єкту

Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121– 02-14.МР</i>			
Зав. кафедри	Криворучко О.В.			21.06.21	<i>Модель інформаційно-аналітичної системи пасажиропотоків</i>	Стадія	Аркуш	Аркушів
Керівник	Жирова Т.О.			21.06.21		РЗ	26	45
Гарант	Токар В.В.			21.06.21				
Розроб.	Муквіч А.Л.			21.06.21				
					<i>Аналіз технічного рішення програмного продукту «forecaster»</i>			
						Факультет інформаційних технологій, 2 курс, 2м група		

Щойно тип об'єкту, робота котрого планується для моделювання, обрано, користувачу пропонується обрати його базові характеристики. Це - необхідний мінімум, від яких можна було б відштовхнутися, щоб продовжувати роботу програми.

Оскільки аналізованим об'єктом є аеропорт, робота якого широко залежна від роботи кожного окремо взятого терміналу, користувачу для початку пропонують обрати їх кількість, щоб в подальшому перейти до більш детального аналізу кожного з них окремо (рис. 3.2).

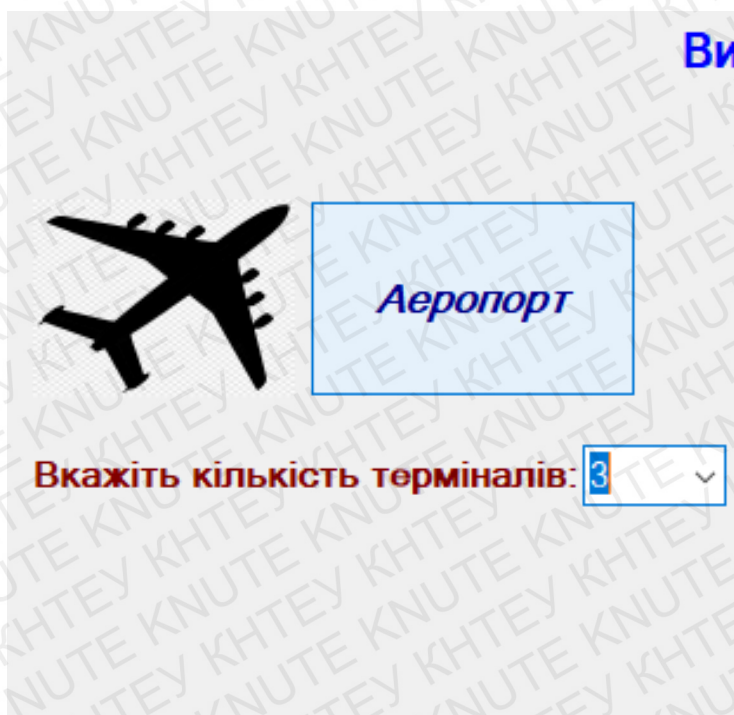


Рис. 3.2. Вибір користувачем кількості терміналів аеропорту

Слід зазначити, що кількість терміналів напряму впливає на технологічні потужності аеропорту – як правило, чим їх більше, тим більше пасажирів потенційно зможе обслуговувати летовище.

3.2. Задання технічних характеристик терміналів

Задавши, скільки терміналів містить аеропорт, користувач отримує можливість задати основні технічні параметри для кожного із них. Як видно

на рис. 3.2., у даному варіанті імітації термінали були представлені в кількості трьох одиниць. Це означає, що наступне меню, яке розкривається перед користувачем, міститиме інформаційні картки у кількості 3 шт., де кожна картка відповідає одному терміналу (рис. 3.3):



Рис. 3.3. Картки терміналів

Користувач отримує можливість передати програмі усі найважливіші якості, якими може володіти термінал, що здатні найбільш значущо вплинути на його спроможність обслуговування пасажирів аеропорту. Найбільш критичними, без яких рух пасажирів далі неможливий взагалі, є наявність у терміналі стійок реєстрації, в яких пасажир реєструється на рейс, пунктів догляду вантажу, інакше кажучи пунктів безпеки, та пунктів паспортного контролю, де пасажир підтверджує свою особистість.

Тим не менше, існує безліч інших параметрів та складових, здатних вплинути на швидкість роботи аеропорту найбільш серйозним чином —

наприклад, допомогти уникнути заторів або натовпів. Сюди входять наступні категорії:

1. Рухомі елементи
 - 1.1. Пасажирські ліфти
 - 1.2. Вантажні ліфти
 - 1.3. Ескалатори
2. Системи докування
 - 2.1. Самохідні трапи
 - 2.2. Телескопічні трапи
3. Транспортне сполучення
 - 3.1. Дорожнє (вибрано за замовчуванням)
 - 3.2. У вигляді рельсового рухомого складу

Дані фінального розрахунку годинного пасажиропотоку для кожного з терміналів можуть серйозно корелюватися в залежності від того, наскільки добре він оснащений і яка інфраструктура об'єднує його з зовнішнім світом.

Оскільки модель досліджує глобальну динаміку замість того, щоб концентруватися на кожному пасажирі індивідуально, система розрахунків намагається виходити з середнього значення продуктивності роботи того чи іншого елемента, наявного в терміналі. Наприклад, відомо, що середньостатистичний темп обслуговування одного пасажирів під час паспортного контролю складає ~2 хвилин. Такий показник й береться за основу для розрахунків, однак ефект від нього може погіршитися, якщо, наприклад, стан інфраструктури аеропорту незадовільний й існує вірогідність виникнення людських або транспортних заторів.

Такий підхід актуальний для всієї системи взаємозв'язків у рамках єдиної «екосистеми» аеропорту – погіршення у одному секторі здатне негативно вплинути на результативність іншого, навіть якщо його робота сама по собі не викликає жодних нарікань.

					<i>КНТЕУ 121–02-14.МР</i>	Аркуш
						29
Зм.	Аркуш	№ докум	Підпис	Дата		

По мірі заповнення карток терміналів та натиснення на команду «ОК», картки блокуються (рис 3.4). Це свідчить про готовність переходу до наступного меню – меню імітації.



Рис. 3.4. Картки заповнені

3.3. Меню імітації

Меню імітації представляє собою ядро програми та кульмінацію її роботи. Користувач отримує можливість побудувати «живі» графіки роботи аеропорту, що можуть змінюються у режимі реального часу, а сам користувач має можливість вносити ці зміни самостійно – це актуально для таких подій, як відміна рейсу, зміна погодних умов, серйозні затори чи аварії в інфраструктурі і т.д.

Частина меню імітації, яка містить панель управління, відображена на рис. 3.5.

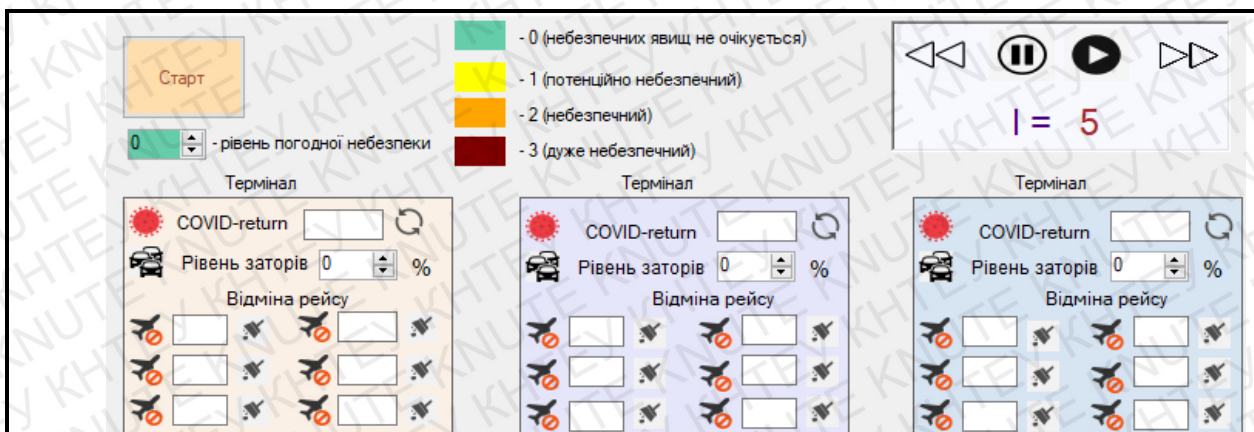


Рис. 3.5. Панель управління меню імітації

Дане меню дозволяє керувати процесом імітації з боку користувача. В ньому існує можливість як починати, призупиняти, пришвидшувати (інтервал у секундах відображено показником I), сповільнювати, чи продовжувати процес, так і вносити інформацію про події в процесі діяльності аеропорту, які надходять і ефект яких має бути оцінено програмою. Серед них – зміна рівня погодної небезпеки, щодо чого існує окрема інструкція, відміна окремих рейсів, зміна ступеню заторів на шляхах до аеропорту, а також повернення пасажирів у зв’язку з коронавірусом.

Трохи нижче розміщене одне з меню Chart з погодинним графіком доби та умовними позначеннями терміналів, на даний момент - пусте у зв’язку з тим що процес імітації ще не було розпочато (рис. 3.6).



Рис. 3.6. Меню побудови графіків

Загальний вигляд «активного» меню програми (рис. 3.7)

						Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121–02-14.МР	

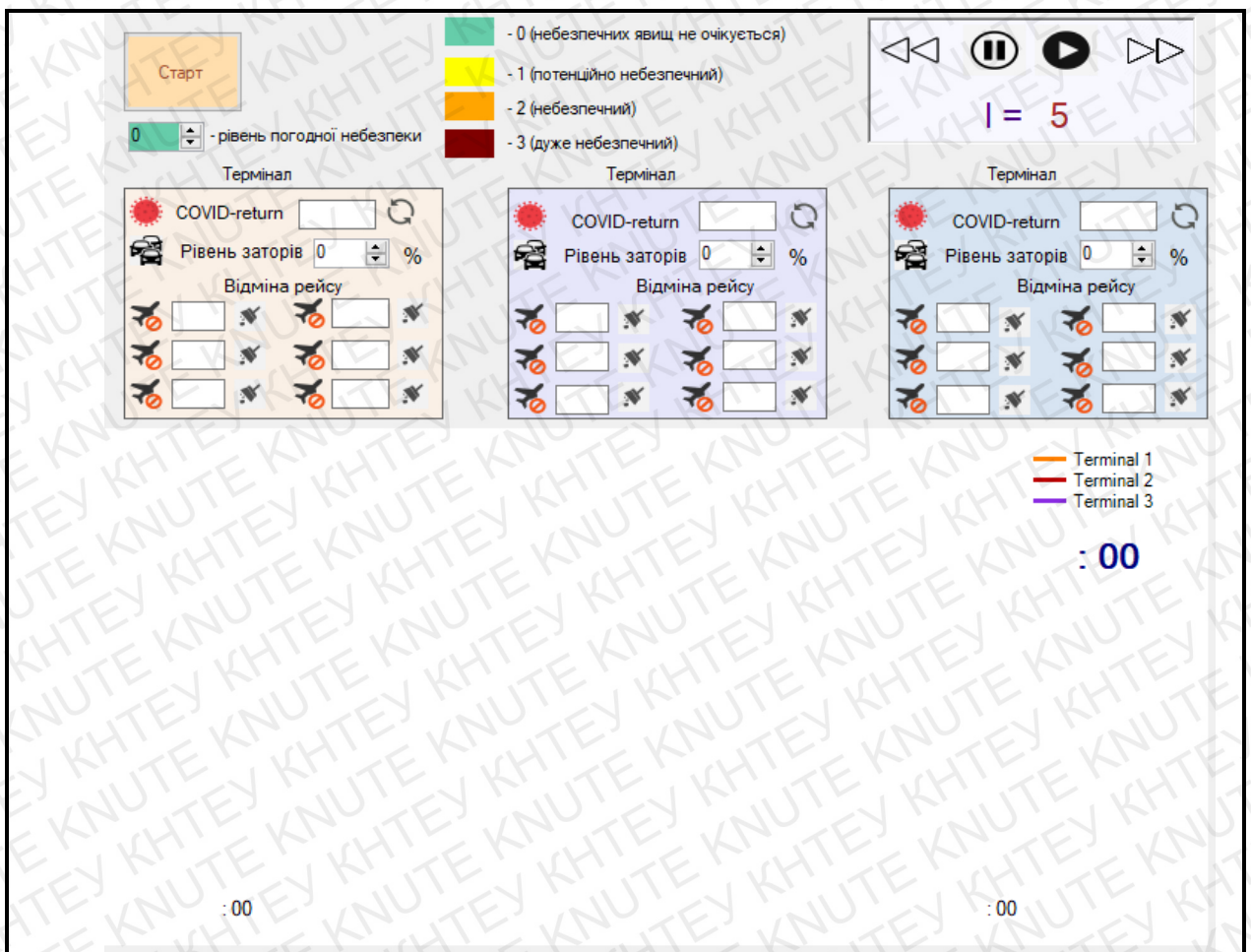


Рис. 3.7. Загальний вигляд

Запуск імітації відбувається за допомогою натиснення клавіші «Старт». Щойно її натиснуто, компонент Chart починає процес креслення кривих пасажиропотоку для кожного з терміналів, інформацію яких користувач попередньо заповнював за допомогою карток терміналів. Колір кривої відповідає легенді кожного терміналу, розміщеній у верхньому правому кутку графіка. Варто зазначити, що у разі вибору меншої кількості терміналів у складі аеропорту, кількість кривих також буде менше і деякі з них не будуть креслитись взагалі. Для кращого орієнтиру зверху графіку зображено години доби. Початок процесу креслення зображений на рис. 3.8.

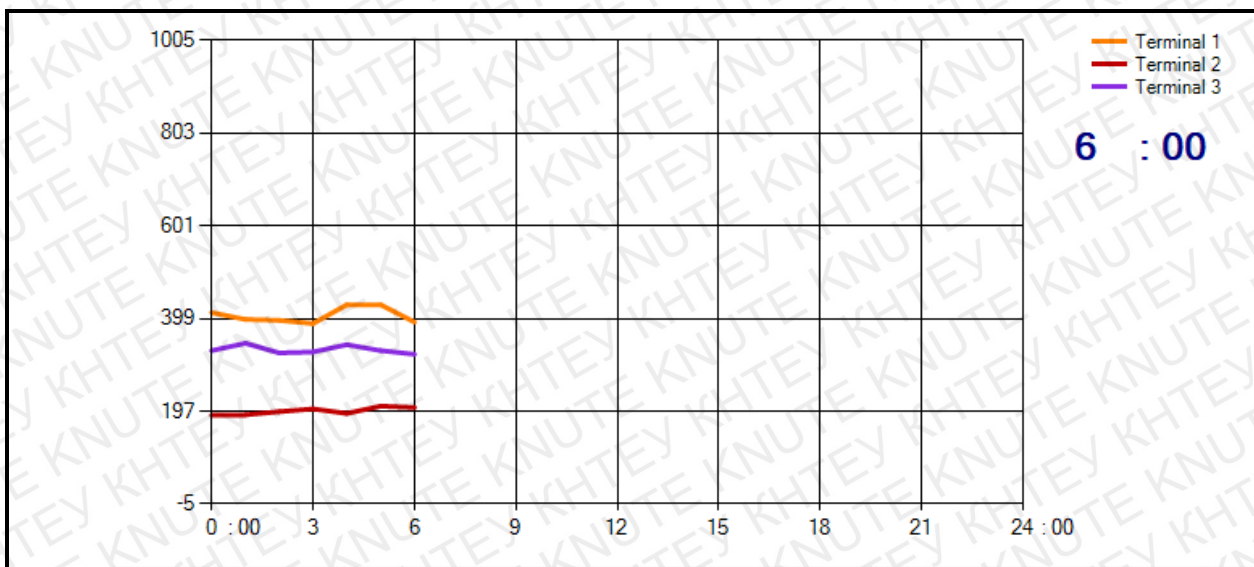


Рис. 3.8. Початок креслення

Однак для того, щоб імітаційна модель мала сенс, вона повинна бути здатною реагувати на зміни, що здатні корегувати початковий прогноз пасажиропотоку, яким би він не був. Відповідно, ці зміни повинні бути відображені на кривій графіку і засвідчені у вигляді його стрибків і коливань.

Для досягнення мети, як видно на рис. 3.5., у програму й було додано панель керування. Нижче, на рис. 3.9. та інших відтворена демонстрація роботи цього підходу. Як видно з ілюстрації, щойно колір у панелі погодної небезпеки змінився на малиновий і досяг рівня «3», пасажиропотік відчув вкрай суттєве погіршення, яке зберігатиметься, допоки рівень погодної небезпеки не буде знижено користувачем до більш м'якого або допоки погода не нормалізується в повній мірі. Подібний підхід до імітаційного моделювання, який повністю відповідає концепції системної графіки, буде актуальним й у інших ситуаціях, його також можна буде помітити на ряді наступних рисунків.

Різноманіття способів впливу на пасажиропотік та експериментів з ним – на рис. 3.9., 3.10. та 3.11.

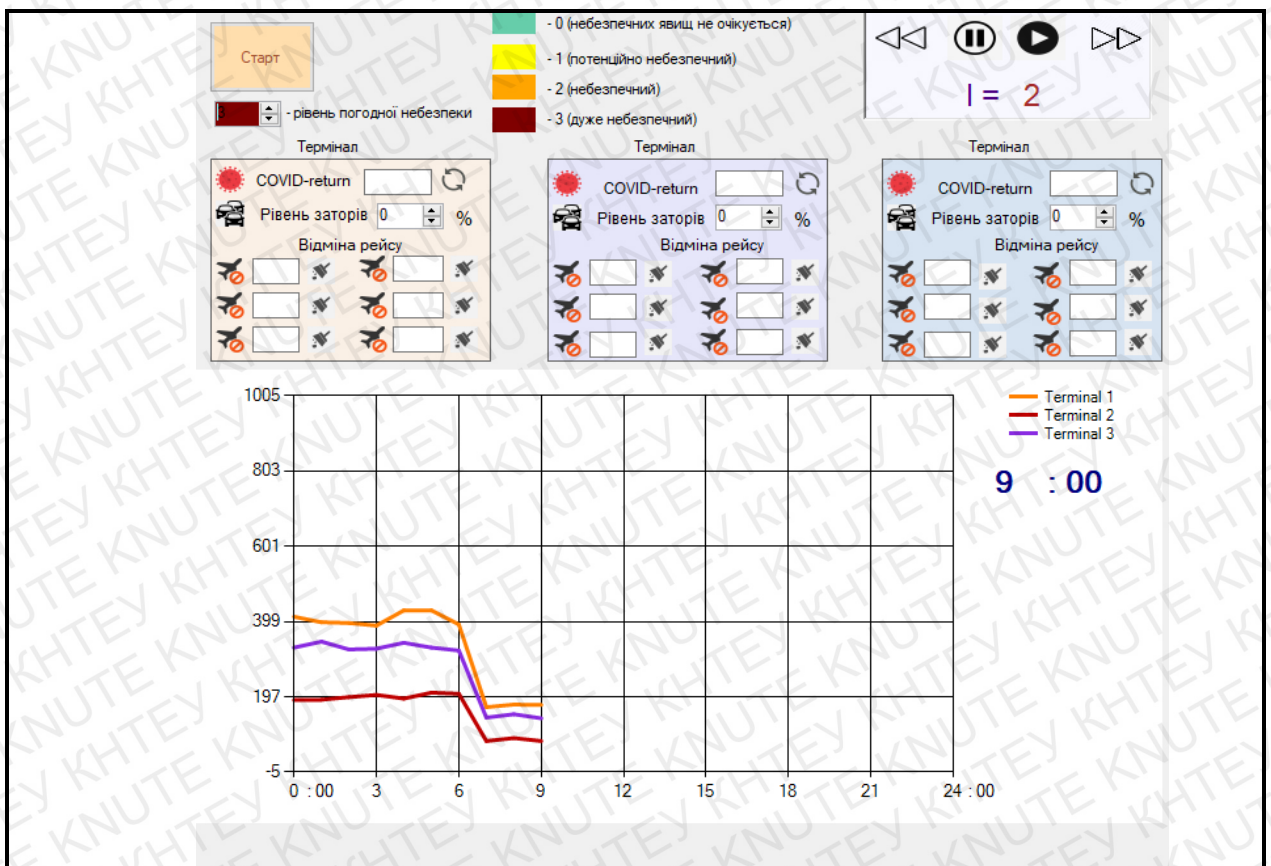


Рис. 3.9. Користувач суттєво погіршує рівень погоднії небезпеки – ефект

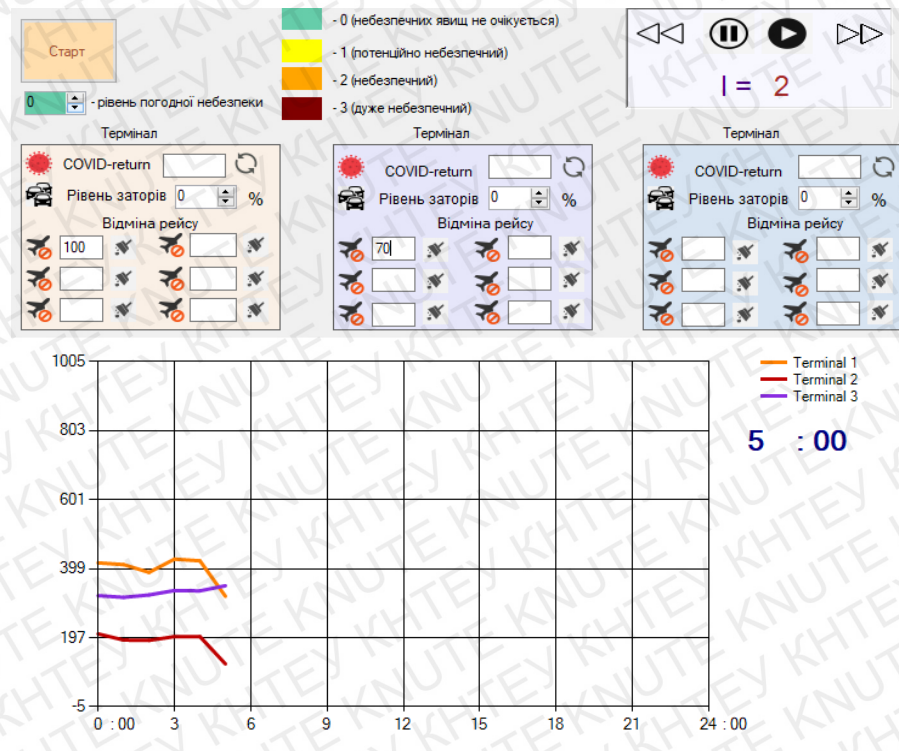


Рис. 3.10. Користувач повідомив про відміну ряду рейсів – ефект

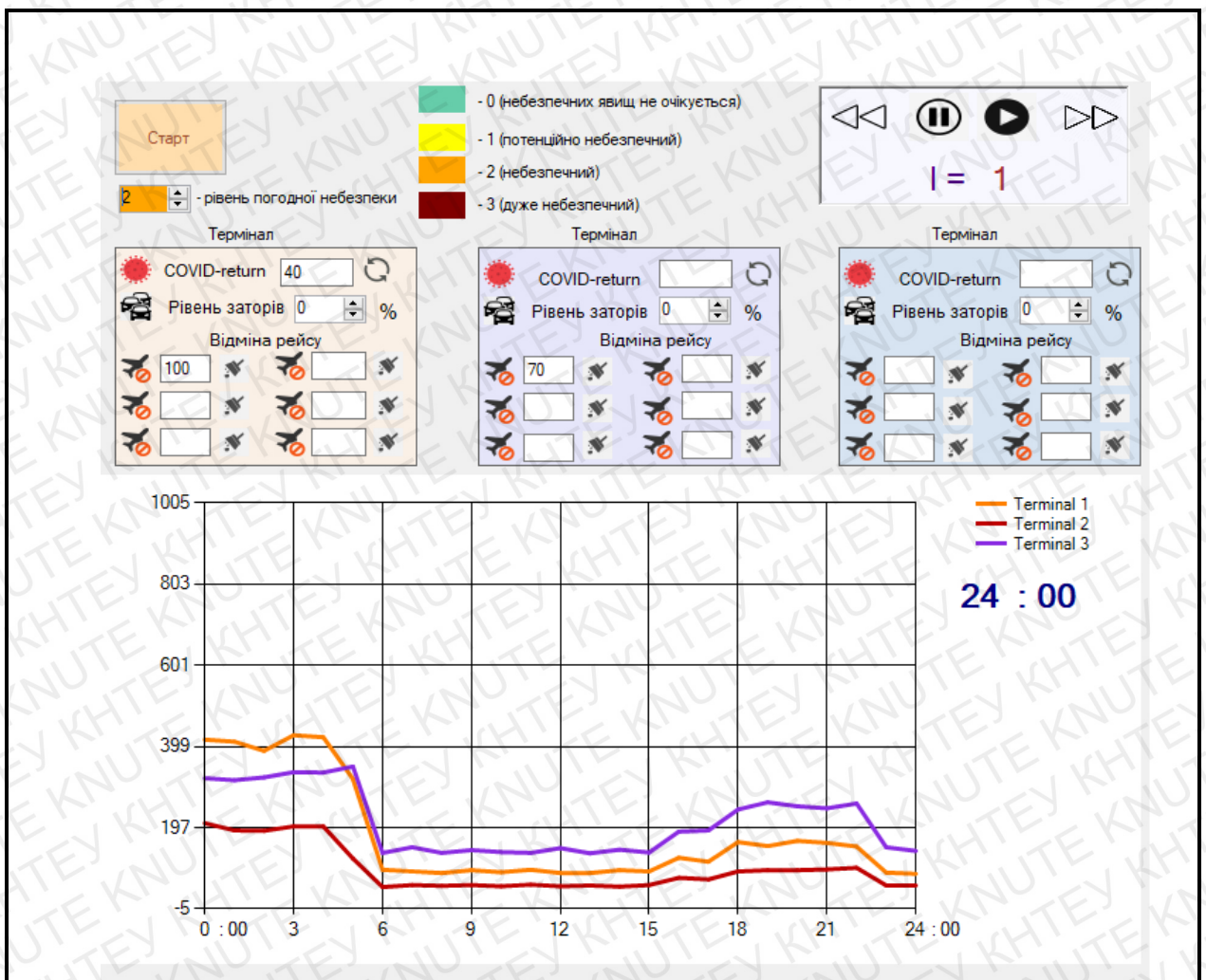


Рис. 3.11. Впродовж доби сталася серія нештатних подій відразу - ефект

Щойно експеримент здійснено і модель проаналізувала пасажиропотік на весь період доби – настає необхідність підведення підсумків експерименту, а саме звітності, у якій вказувалась б підсумкова проекція (по версії моделі) пасажиропотоку аеропорту як на всю добу наперед, так і на кожну годину зокрема. Для здійснення поставленого завдання було створено ще кілька елементів Chart, однак на відміну від кривих, вигляд відображуваної там графіки приймає характер діаграм у вигляді колонок, а генеруються вони у правому вікні програми. Їх зображено на рис. 3.12.

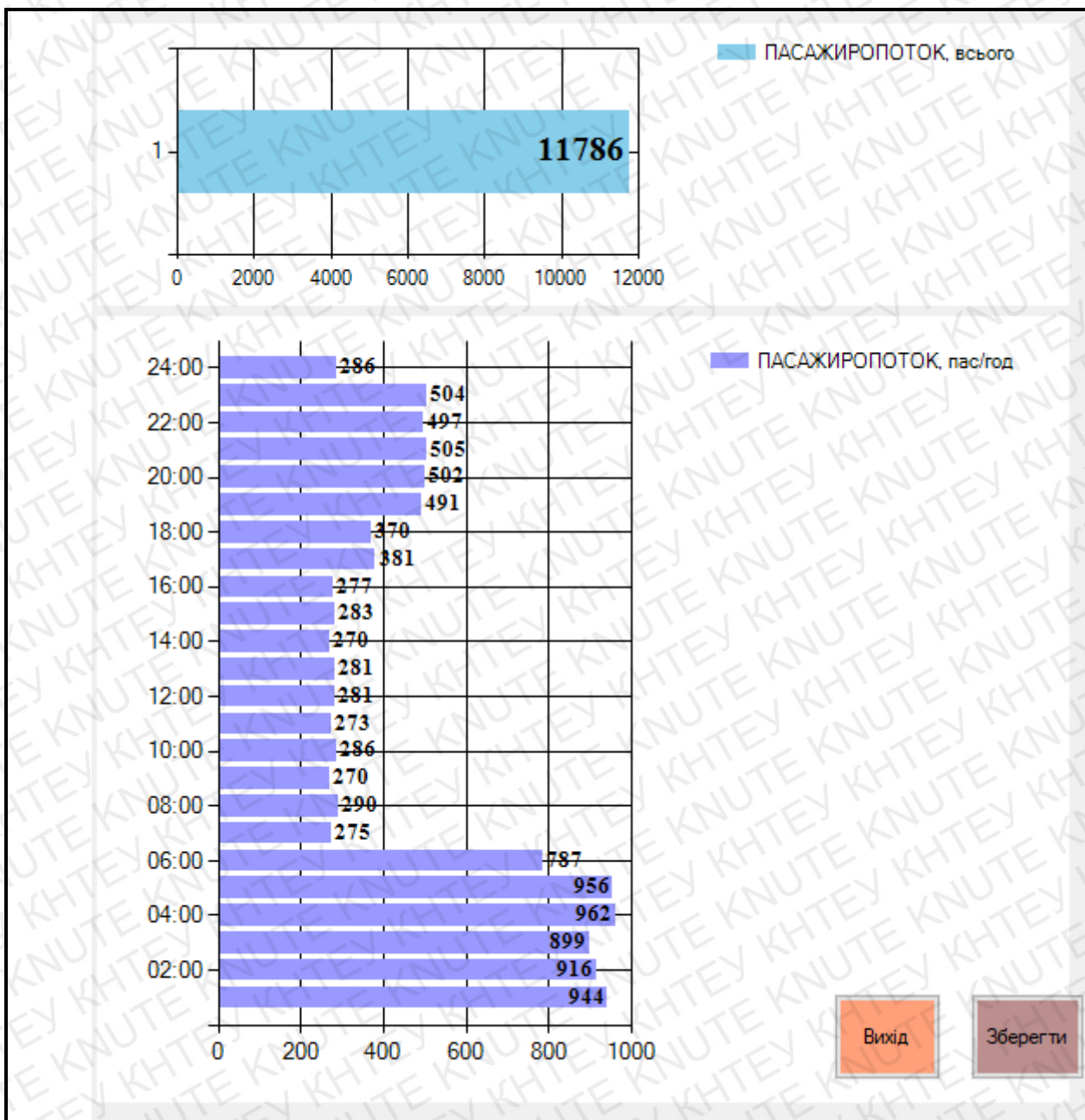


Рис. 3.12. Підсумкова проєкція пасажиропотоку – по годинно та за добу

Як видно на рис. 3.11., у програму вбудовано таймер. Його значення просувається вперед, індикуючи користувачу, яку саме частину доби зараз опрацьовує імітаційна модель. Добігаючи позначки 24:00, раунд проєкції завершується і перед користувачем з'являються до цього недоступні діаграми. Загальний ж вигляд програми зображено на рис. 3.13.

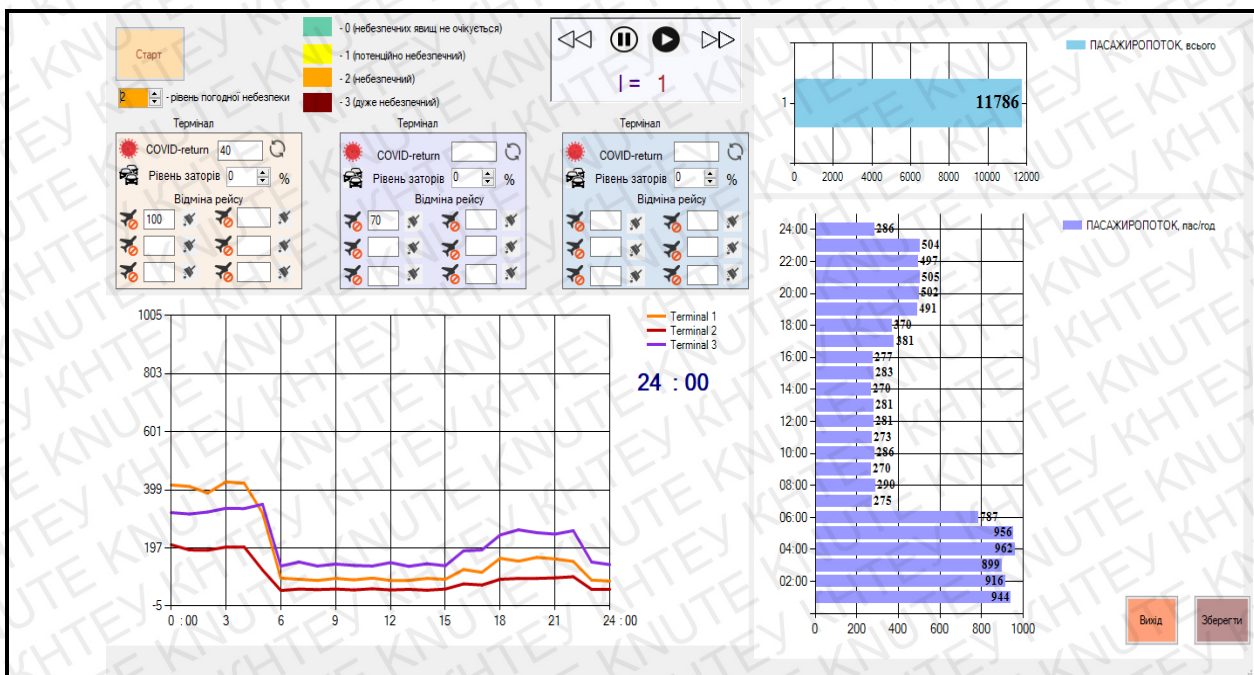


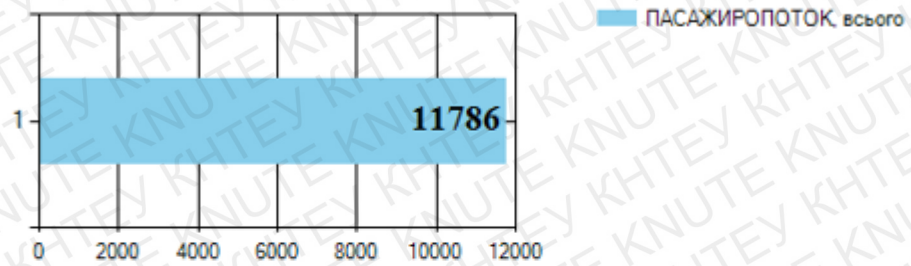
Рис. 3.13. Загальний вигляд програми

Як видно на рисунку, користувач може вийти з програми або сформувати звіт, щоб зберегти результати роботи. Звіт формується командою «Зберегти» і з'являється у форматі Microsoft Word. Особливості звіту наступні – в назву документу фіксується дата та час його створення, щоб користувач завжди розумів, коли саме було створено документ. Сам звіт, окрім вищезазначених діаграм, містить інформацію про кількість відмінених рейсів, якщо вони є, середній пасажиропотік аеропорту за годину, а також проєкцію того, скільки пасажирів аеропорт не обслужив через різноманітні події.

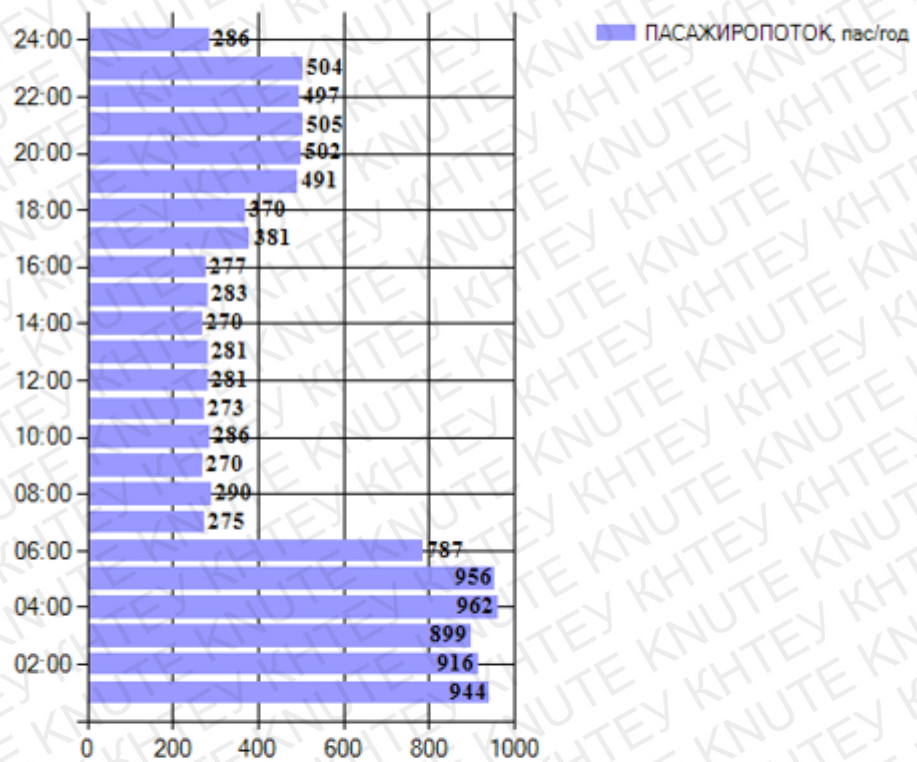
Звіт зберігається у корінному каталозі програми, що поруч з виконуваним файлом імітаційної моделі. Він доступний для відкриття, редагування та переправлення.

Якщо користувач не бажає створювати звіт, він може вийти з програми, натиснувши команду «Вихід».

ПРОГНОЗ МОДЕЛЮВАННЯ ДОБОВОГО ПАСАЖИРОПОТОКУ



ПРОГНОЗ ПАСАЖИРОПОТОКУ ПО ГОДИНАМ



ВСЬОГО ВІДМІНЕНО РЕЙСІВ: 2

СЕРЕДНІЙ ПАСАЖИРОПОТІК/год: 491

ПОТЕНЦІЙНО МОЖЛИВИЙ ПАСАЖИРОПОТОК: +12334

Рис. 3.14. Звіт у форматі .doc, підсумовуючий результати роботи моделі

3.4. Аналіз методів та процедур, що забезпечують роботу програми

Не дивлячись на достатньо немалі об'єми програми, яка містить 3 форми, безпосередньо модель формують кілька основних функцій і процедур. Перша з них – побудова графіку (рис. 3.15).

```
int terminal1 = Int32.Parse(textBox3.Text);
int terminal2 = Int32.Parse(textBox4.Text);
int terminal3 = Int32.Parse(textBox5.Text);

timer1.Enabled = true;

numericUpDown1.Maximum = terminal1;
numericUpDown2.Maximum = terminal2;
numericUpDown7.Maximum = terminal3;
numericUpDown1.Minimum = 0;
numericUpDown2.Minimum = 0;
numericUpDown7.Minimum = 0;

numericUpDown1.Value = terminal1;
numericUpDown2.Value = terminal2;
numericUpDown7.Value = terminal3;

chart1.ChartAreas[0].AxisY.Maximum = terminal1 + terminal2 + terminal3;
chart1.ChartAreas[0].AxisY.Minimum = -5;

chart1.ChartAreas[0].AxisX.Minimum = 0;
chart1.ChartAreas[0].AxisX.Maximum = 24;

chart1.ChartAreas[0].AxisX.Interval = 3;
```

Рис. 3.15. Процедура побудови кривої пасажиропотоку

Для початку, відбувається під'єднання згенерованих раніше (рис. 3.4.) значень пасажиропотоків кожного з терміналів до графічного поля Chart, де пасажиропотік кожного з терміналів позначається як terminal1, terminal2 та terminal3 відповідно. Самі ж змінні для зручності перенесені у поля прокрутки відомі як “numericupdown”, де «up» та «down» означають, що графік відреагує зменшенням або збільшенням своєї кривої у разі прокрутки елемента вгору чи вниз.

Після цього, за допомогою виразів «chart1.ChartAreas[0].AxisY.Maximum =>» відбувається встановлення мінімуму та максимуму графіку по кривій x та y відповідно. Йде визначення формату осей та інтервалу їх роботи – в даному випадку за допомогою виразу «chart1.ChartAreas[0].AxisX.Maximum = 24;» максимальним значенням осі X обрано показник 24, що відповідає 24 годинам, які й повинна обробити модель, в той час як вираз «chart1.ChartAreas[0].AxisX.Interval = 3;» вказує, що умовне позначення годин на осі X здійснюється у форматі раз у 3 години: 00:00, 03:00 і т.д.

Для того, щоб мати змогу будувати графік у реальному часі, відповідна функція була додана у ще одну важливу подію – Timer1Tick (рис. 3.16). Сам код побудови вказаний як «chart1.Series[0].Points.AddXY(timenow, value);», де value – значення пасажиропотоку терміналу.

```

ссылка: 1
private void timer1_Tick(object sender, EventArgs e)
{
    DateTime timenow = DateTime.Now;
    int value = Convert.ToInt32(numericUpDown1.Value);
    int value2 = Convert.ToInt32(numericUpDown2.Value);
    int value3 = Convert.ToInt32(numericUpDown7.Value);

    if (textBox4.Text == "0")
    {
        chart1.Series[0].Points.AddXY(timenow, value);
    }
    else if (textBox4.Text != "0")
    {
        chart1.Series[0].Points.AddXY(timenow, value);
        chart1.Series[1].Points.AddXY(timenow, value2);
    }
    if (textBox5.Text != "0")
    {
        chart1.Series[0].Points.AddXY(timenow, value);
        chart1.Series[1].Points.AddXY(timenow, value2);
        chart1.Series[2].Points.AddXY(timenow, value3);
    }
}

```

Рис. 3.16. Забезпечення безперервності побудови кривої за рахунок синхронізації процесу з таймером

						Аркуш
						40
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121–02-14.МР	

Ще однією задачею, обов'язковою в рамках завдання, є додавання можливості інтерактивної зміни графіку у сторону як зростання, так і спадання, щойно користувач вносить зміни у цифри пасажиропотоку. Дана ціль досягається з допомогою процедури «numericUpDown_ValueChanged», де, як зазначалося вище, елемент numericUpDown зберігає у собі значення пасажиропотоку терміналу на даний момент. Для того, щоб змінити його, разом з тим змінити й графік, було створено відповідну процедуру, що з допомогою операторів if та else опрацьовує події збільшення або зменшення значення numericUpDown з відповідними зрушеннями й для графіку.

```

Decimal OldValue2;
ссылка:1
private void numericUpDown5_ValueChanged(object sender, EventArgs e)
{
    if (numericUpDown5.Value > OldValue2)
    {
        numericUpDown2.Value = numericUpDown2.Value - numericUpDown2.Value * 1 / 10;
    }
    else if (numericUpDown5.Value < OldValue2)
    {
        numericUpDown2.Value = numericUpDown2.Value + numericUpDown2.Value * 1 / 10;
    }
    else
    {
        return;
    }
    OldValue2 = numericUpDown5.Value;
}

Decimal OldValue3;

```

Рис. 3.17. Процедура зміни динаміки графіку

Нарешті, заповнення діаграм, відображаючих погодинні зміни пасажиропотоку, відбувається вельми схожим принципом відносно способу побудови всіх інших графіків, де v1-v21 являють собою попередньо зчитані значення потоку пасажирів у підсумку кожної окремої години в процесі імітації. Лівише відображено точну годину – це зроблено, щоб графік був інтуїтивно зрозумілим (рис. 3.18).

```

chart2.Series[0].Points.AddXY("01:00", v1);
chart2.Series[0].Points.AddXY("02:00", v2);
chart2.Series[0].Points.AddXY("03:00", v3);
chart2.Series[0].Points.AddXY("04:00", v4);
chart2.Series[0].Points.AddXY("05:00", v5);
chart2.Series[0].Points.AddXY("06:00", v6);
chart2.Series[0].Points.AddXY("07:00", v7);
chart2.Series[0].Points.AddXY("08:00", v8);
chart2.Series[0].Points.AddXY("09:00", v9);
chart2.Series[0].Points.AddXY("10:00", v10);
chart2.Series[0].Points.AddXY("11:00", v11);
chart2.Series[0].Points.AddXY("12:00", v12);
chart2.Series[0].Points.AddXY("13:00", v13);
chart2.Series[0].Points.AddXY("14:00", v14);
chart2.Series[0].Points.AddXY("15:00", v15);
chart2.Series[0].Points.AddXY("16:00", v16);
chart2.Series[0].Points.AddXY("17:00", v17);
chart2.Series[0].Points.AddXY("18:00", v18);
chart2.Series[0].Points.AddXY("19:00", v19);
chart2.Series[0].Points.AddXY("20:00", v20);
chart2.Series[0].Points.AddXY("21:00", v21);
chart2.Series[0].Points.AddXY("22:00", v22);
chart2.Series[0].Points.AddXY("23:00", v23);
chart2.Series[0].Points.AddXY("24:00", v24);

```

Рис. 3.18. Відображення погодинного графіку пасажиропотоку

3.5. Висновки до розділу 3

У Розділі 3 було комплексно розглянуто роботу ПЗ «Forecaster» на практиці. Це включило в себе вивчення правил та закономірностей побудови моделей через програму, вивчення найважливіших методів та процедур, що забезпечують роботу програми, а також наочна демонстрація роботи імітаційної моделі, яку утворює програма.

					<i>КНТЕУ 121–02-14.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		42

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Середовище програмування Microsoft Visual Studio C# надає широкий спектр інструментів для розробки програмного забезпечення будь-якого типу. В цю категорію входять імітаційні моделі інформаційно-аналітичних систем, здатні забезпечувати оновлюваний потік інформації на одиницю реального часу, що є особливо важливим у відслідковуванні та регулюванні результатів діяльності різноманітних підприємств, що особливо актуально для транспортних об'єктів, діяльність яких базується на забезпеченні пасажирського потоку.

Створена в результаті наукової діяльності модель ІАС аеропорту надає користувачу розгорнутий та відкритий потік інформації по годинно, аналізуючи пасажирський потік аеропорту впродовж цілої доби його діяльності. Окрім здатності надавати інформацію, модель здатна серйозно її корегувати та видозмінювати, враховуючи різноманітні події, що можуть вплинути на результат роботи летовища, такі як втрати пасажиропотоку через вплив коронавірусу, відміни рейсів, погіршення погодних умов, ускладнене прибуття до аеропорту через завантаженість зовнішньої інфраструктури і т.д.

Результатом моделювання є ряд інформативних діаграм, які містять інформацію про те, скільки пасажирів обслуговує аеропорт при обраному користувачем сценарії подій. Сюди входить як по годинна діаграма, так і діаграма з аналітикою за всю добу цілком.

					<i>КНТЕУ 121– 02-14.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. кафедри	Криворучко О.В.			01.11.21	<i>Модель інформаційно-аналітичної системи пасажиропотоків</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Жирова Т.О.			01.11.21		<i>ВП</i>	<i>43</i>	<i>45</i>
Гарант	Токар В.В.			01.11.21				
Розроб.	Муквіч А.Л.			01.11.21				
					<i>Висновки та пропозиції</i>	Факультет інформаційних технологій, 2 курс, 2м група		

Щойно діаграми сформовані, користувачу доступно транспортувати їх у зручний та універсальний формат Microsoft Office Word. Окрім власне діаграм, модель додає в звіт кілька додаткових характеристик: кількість відмінених рейсів, якщо вони мали місце, середній пасажиропотік у годину, а також висновок щодо того, яку кількість пасажирів аеропорт міг би обслужити потенційно.

Проведена науково-дослідна робота дає широку базу для подальших досліджень. Розробка ІАС аналогічного принципу для інших транспортних об'єктів, таких як вокзал чи морський порт, є перспективним напрямом подальшого прогресу в рамках досліджень імітаційних моделей ІАС, що, однак, вимагатиме серйозного переопрацювання підходу до розрахунку та побудови таких моделей.

					<i>КНТЕУ 121–02-14.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		44

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Троельсен Ендрю, Япиксе Філіп, Pro C# 7.0 на платформах .NET та .NET Core / Ендрю Троельсен, Філіп Япиксе. – Apress, 8 видання, 20 січня 2018. – 1372 ст.
2. Альбахарі Бен, C# 7.0 - Миттева довідка для програмістів C# 7.0 / Бен Альбахарі - O'Reilly Media, 1 видання, 3 липня 2017. – 240 ст.
3. Альбахарі Джозеф, C # 7.0 в двох словах / Джозеф Альбахарі, - O'Reilly Media, 7 видання, 7 листопада 2017. - 1056 ст.
4. Марк Дж. Прайс, C # 7.1 і .NET Core 2.0 - сучасні крос-платформні розробки / Марк Дж. Прайс - Packt Publishing, 3-тє виправлене видання, 30 листопада 2017. - 800 ст.
5. Роб Майлз, Зелена книга C# / Майлз Роб. - Kindle Edition, 6 січня 2014. - 322 ст.
6. Джеймі Чан, C# за один день / Чан Джеймі. - Learn Coding Fast, 1 видання, 20 жовтня 2015. – 161 ст.
7. Міжнародна асоціація повітряного транспорту IATA – Керівництво польотів [Електронний ресурс]. - Режим доступу: <https://www.iata.org/>

					<i>КНТЕУ 121 – 02-14.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. кафедри		Криворучко О.В.		27.02.21	<i>Модель інформаційно-аналітичної системи пасажиропотоків</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Жирова Т.О.		27.02.21		<i>СВД</i>	45	45
Гарант		Токар В.В.		27.02.21		Факультет інформаційних технологій, 2 курс, 2м група		
Розроб.		Муквіч А.Л.		27.02.21				
<i>Список використаних джерел</i>								

ТЕХНІЧНЕ ЗАВДАННЯ

Технічне завдання – важливий етап на шляху розробки додатку, оскільки процес власне розробки ПЗ неможливий, якщо немає чіткого формулювання того, що саме потрібно зробити і як.

Технологічні вимоги до ПЗ:

- 1) Мова програмування – С#
- 2) Платформа – ASP.NET 7.0
- 3) Середовище розробки – Microsoft Visual Studio 2019
- 4) Кількість форм: 3 форми
 - 4.1. Форма 1 – вибір типу модельованого об'єкту
 - 4.2. Форма 2 – заповнення основних характеристик об'єкту та його терміналів
 - 4.3. Форма 3 – імітація та звіт по результатам роботи ІАС
- 5) Процедури: 3 (креслення графіку, регулювання швидкості та звіту)

Функціональні вимоги:

- 1) Можливість зупинки процесу роботи ІАС в будь-який момент
- 2) Можливість прискорення або уповільнення роботи моделі по бажанню користувача
- 3) Програмний модуль повинен моделювати роботу об'єкту на період 24 годин (однієї доби) з моменту запуску програми
- 4) Можливість формування звіту в універсальному та зручному для транспортування форматі (такому як Microsoft Office Word)

					<i>КНТЕУ 121 – 02-14.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. кафедри		Криворучко О.В.		20.03.21	<i>Модель інформаційно-аналітичної системи пасажиропотоків</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Жирова Т.О.		20.03.21		<i>ТЗ</i>	<i>46</i>	<i>47</i>
Гарант		Токар В.В.		20.03.21				
Розроб.		Муквіч А.Л.		20.03.21				
					<i>Технічне завдання</i>	Факультет інформаційних технологій, 2 курс, 2м група		

5) Здатність графічного відтворення даних – вимагається застосування графіків та діаграм

6) Утворення діаграм повинно відбуватися автоматично у момент досягнення ПЗ 24-годинної відмітки моделювання

1.1. Діаграма 1 – результат пасажиропотоку по годинно

1.2. Діаграма 2 – результат пасажиропотоку за добу

Вимоги до сумісності:

1) Можливість запуску на операційних системах сімейства Windows версій від XP до Windows 11

2) Вимога портативності. Програма має працювати у повній мірі без необхідності додаткового встановлення її компонентів на комп'ютері. Виконуваний файл ПЗ повинен мати .exe формат.

3) Звіти, створені по результатам роботи ПЗ, мають бути сумісними і доступними для збереження на різних пристроях та операційних системах, для чого вимагається їх збереження у спрощеному форматі .doc.

					<i>КНТЕУ 121–02-14.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		47

ДОДАТКИ

Додаток А

Модуль «Лістинг функції початку процесу імітації»

```
countSeconds = 0;
suspended = 0;
label32.Text = "0";
label33.Text = "0";
label34.Text = "0";
foreach (var series in chart2.Series)
{
    series.Points.Clear();
}
foreach (var series in chart1.Series)
{
    series.Points.Clear();
}
label30.Text = "";
numericUpDown3.Value = numericUpDown3.Minimum;
numericUpDown4.Value = numericUpDown4.Minimum;
numericUpDown5.Value = numericUpDown5.Minimum;
numericUpDown6.Value = numericUpDown6.Minimum;
timer2.Enabled = false;
time2 = 0;
progress = 0;
int terminal1 = Int32.Parse(textBox3.Text);
int terminal2 = Int32.Parse(textBox4.Text);
int terminal3 = Int32.Parse(textBox5.Text);
timer1.Enabled = true;
numericUpDown1.Maximum = terminal1;
numericUpDown2.Maximum = terminal2;
numericUpDown7.Maximum = terminal3;
numericUpDown1.Minimum = 0;
numericUpDown2.Minimum = 0;
numericUpDown7.Minimum = 0;
numericUpDown1.Value = terminal1;
numericUpDown2.Value = terminal2;
numericUpDown7.Value = terminal3;
chart1.ChartAreas[0].AxisY.Maximum = terminal1 + terminal2 + terminal3;
chart1.ChartAreas[0].AxisY.Minimum = -5;
chart1.ChartAreas[0].AxisX.Minimum = 0;
chart1.ChartAreas[0].AxisX.Maximum = 24;
chart1.ChartAreas[0].AxisX.Interval = 3;
```

Модуль «Лістинг події кривої креслення графіку»

```
private void timer1_Tick(object sender, EventArgs e)
{
    int value = Convert.ToInt32(numericUpDown1.Value);
    int value2 = Convert.ToInt32(numericUpDown2.Value);
    int value3 = Convert.ToInt32(numericUpDown7.Value);
    resultRandoma = rnd.Next(value - value * 1 / 8, value);
    resultRandoma2 = rnd.Next(value2 - value2 * 1 / 8, value2);
    resultRandoma3 = rnd.Next(value3 - value3 * 1 / 8, value3);

    if (textBox4.Text == "0")
    {
        chart1.Series[0].Points.AddXY(countSeconds, resultRandoma);
    }
    else if (textBox4.Text != "0")
    {
        chart1.Series[0].Points.AddXY(countSeconds, resultRandoma);
        chart1.Series[1].Points.AddXY(countSeconds, resultRandoma2);
    }
    if (textBox5.Text != "0")
    {
        chart1.Series[0].Points.AddXY(countSeconds, resultRandoma);
        chart1.Series[1].Points.AddXY(countSeconds, resultRandoma2);
        chart1.Series[2].Points.AddXY(countSeconds, resultRandoma3);
    }
    countSeconds++;
    timer2.Enabled = true;
}
```

Модуль «Лістинг функції креслення діаграм»

```
int allairport = v1 + v2 + v3 + v4 + v5 + v6 + v7 + v8 + v9 + v10 + v11 + v12 + v13 + v14 + v15 + v16 + v17 + v18 + v19 +  
v20 + v21 + v22 + v23 + v24;
```

```
chart2.Series[0].IsValueShownAsLabel = true;
```

```
chart3.Series[0].IsValueShownAsLabel = true;
```

```
chart2.ChartAreas[0].AxisX.Interval = 2;
```

```
chart2.Series[0].Font = new Font("Times", 10, FontStyle.Bold);
```

```
chart3.Series[0].Font = new Font("Times", 15, FontStyle.Bold);
```

```
chart2.Series[0].Points.AddXY("01:00", v1);
```

```
chart2.Series[0].Points.AddXY("02:00", v2);
```

```
chart2.Series[0].Points.AddXY("03:00", v3);
```

```
chart2.Series[0].Points.AddXY("04:00", v4);
```

```
chart2.Series[0].Points.AddXY("05:00", v5);
```

```
chart2.Series[0].Points.AddXY("06:00", v6);
```

```
chart2.Series[0].Points.AddXY("07:00", v7);
```

```
chart2.Series[0].Points.AddXY("08:00", v8);
```

```
chart2.Series[0].Points.AddXY("09:00", v9);
```

```
chart2.Series[0].Points.AddXY("10:00", v10);
```

```
chart2.Series[0].Points.AddXY("11:00", v11);
```

```
chart2.Series[0].Points.AddXY("12:00", v12);
```

```
chart2.Series[0].Points.AddXY("13:00", v13);
```

```
chart2.Series[0].Points.AddXY("14:00", v14);
```

```
chart2.Series[0].Points.AddXY("15:00", v15);
```

```
chart2.Series[0].Points.AddXY("16:00", v16);
```

```
chart2.Series[0].Points.AddXY("17:00", v17);
```

```
chart2.Series[0].Points.AddXY("18:00", v18);
```

```
chart2.Series[0].Points.AddXY("19:00", v19);
```

```
chart2.Series[0].Points.AddXY("20:00", v20);
```

```
chart2.Series[0].Points.AddXY("21:00", v21);
```

```
chart2.Series[0].Points.AddXY("22:00", v22);
```

```
chart2.Series[0].Points.AddXY("23:00", v23);
```

```
chart2.Series[0].Points.AddXY("24:00", v24);
```

```
chart3.Series[0].Points.AddXY("1", allairport);
```

```
int f1 = Convert.ToInt32(textBox3.Text);
```

```
int f2 = Convert.ToInt32(textBox4.Text);
```

```
int f3 = Convert.ToInt32(textBox5.Text);
```

```
int allbasic = (f1 + f2 + f3) * 24;
```

```
int difference = allbasic - allairport;
```

```
label33.Text = difference.ToString();
```

```
int average = allairport / 24;
```

```
label34.Text = average.ToString();
```

Модуль «Лістинг функції створення звіту»

```

private void button4_Click(object sender, EventArgs e)
{
    this.chart3.SaveImage(".\\Chart1.png", ChartImageFormat.Png);
    this.chart2.SaveImage(".\\Chart2.png", ChartImageFormat.Png);
    myimage = Image.FromFile(".\\Chart1.png");
    myimage2 = Image.FromFile(".\\Chart2.png");
    Microsoft.Office.Interop.Word._Application oWord = new Microsoft.Office.Interop.Word.Application();
    Microsoft.Office.Interop.Word._Document oDoc = oWord.Documents.Add();
    oWord.Visible = true;
    Object oMissing = System.Reflection.Missing.Value;
    var oPara1 = oDoc.Content.Paragraphs.Add(ref oMissing);
    oPara1.Range.Text = "ПРОГНОЗ МОДЕЛЮВАННЯ ДОБОВОГО ПАСАЖИРОПОТОКУ";
    oPara1.Range.InsertParagraphAfter();
    System.Drawing.Image Picture = myimage;
    var oPara2 = oDoc.Content.Paragraphs.Add(ref oMissing);
    Thread thread = new Thread(() => Clipboard.SetImage(Picture));
    thread.SetApartmentState(ApartmentState.STA);
    thread.Start();
    thread.Join();
    oPara2.Range.Paste();
    oPara2.Range.InsertParagraphAfter();
    Object oMissing2 = System.Reflection.Missing.Value;
    var oPara3 = oDoc.Content.Paragraphs.Add(ref oMissing);
    oPara3.Range.Text = "ПРОГНОЗ ПАСАЖИРОПОТОКУ ПО ГОДИНАМ";
    oPara3.Range.InsertParagraphAfter();
    System.Drawing.Image Picture2 = myimage2;
    var oPara4 = oDoc.Content.Paragraphs.Add(ref oMissing);
    Thread thread2 = new Thread(() => Clipboard.SetImage(Picture2));
    oPara3.Range.InsertParagraphAfter();
    oPara3.Range.Text = "ВСЬОГО ВІДМІНЕНО РЕЙСІВ: " + label32.Text;
    oPara3.Range.InsertParagraphAfter();
    oPara3.Range.Text = "СЕРЕДНІЙ ПАСАЖИРОПОТІК/год: " + label34.Text;
    oPara3.Range.InsertParagraphAfter();
    oPara3.Range.Text = "ПОТЕНЦІЙНО МОЖЛИВИЙ ПАСАЖИРОПОТОК:" + "+" + label33.Text;
    oPara3.Range.InsertParagraphAfter();
    thread2.SetApartmentState(ApartmentState.STA); //Set the thread to STA
    thread2.Start();
    thread2.Join();
    oPara4.Range.Paste();
    oPara4.Range.InsertParagraphAfter();
    oDoc.SaveAs(FileName: Environment.CurrentDirectory + "\\Звіт_" + DateTime.Now.ToString("yyyyMMdd_hhmmss") + ".doc");
    oDoc.Close();
    oWord.Quit();
    myimage.Dispose();
    myimage2.Dispose();
}

```