

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ

Кафедра Комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач випускової кафедри

Аліна САВЧЕНКО.

«_____» _____ 2023р.

КВАЛІФІКАЦІЙНА РОБОТА
(ДИПЛОМНА РОБОТА, ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИЦІ ОСВІТНЬОГО СТУПЕНЯ “МАГІСТР”

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ

ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Тема: “Система планування графіків польотів повітряних суден”

Виконавець: студентка групи УС-211М Маковська Дар'я Юріївна

Керівник: професор Зіатдінов Юрій Кашафович

Нормоконтролер: _____ Ігор РАЙЧЕВ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12 "Інформаційні Технології", 122 "Комп'ютерні науки", "Інформаційні управляючі системи та технології"

ЗАТВЕРДЖУЮ

Завідувач випускової кафедри

_____ Аліна САВЧЕНКО

«_» _____ 2023р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

Маковської Дар'ї Юріївни

(прізвище, ім'я, по батькові)

- 1. Тема роботи:** «Система планування графіків польотів повітряних суден» затверджена наказом ректора №1976 від 29.09.2023 р.
- 2. Термін виконання роботи:** з 02 жовтня 2023 року по 31 грудня 2023 року.
- 3. Вихідні данні до роботи:** загальний аналіз світового транспорту; дослідження історії міжнародних перевезень; аналіз існуючих методів і алгоритмів для реалізації продукту у відповідності до поставленої задачі; проектування системи розробки графіків польотів з застосуванням сучасних методів.
- 4. Зміст пояснювальної записки:** вступ, огляд предметної області, математична модель, розробка програмного додатку для створення графіку польотів повітряних суден, висновок.
- 5. Перелік обов'язкового графічного матеріалу:** діаграма варіантів використання системи, схема основних етапів алгоритму Branch and Price, графік пасажиропотоку в світі, схема процесу формування графіку польоту, схема алгоритму Branch and Price.

6. Календарний план-графік

№ п/п	Завдання	Термін виконання	Підпис керівника
1.	Отримання завдання на дипломну роботу, створення плану дипломної роботи та побудова плану-графіку виконання робіт.	02.10.2023 – 04.10.2023	
2.	Огляд та аналіз наукової літератури по темі дипломної роботи та написання Розділу 1.	05.10.2023 – 23.11.2023	
3.	Написання Розділу 2 дипломної роботи.	24.10.2023 – 25.11.2023	
4.	Написання Розділу 3 і Розділу 4 дипломної роботи. Завершення створення пояснювальної записки дипломної роботи.	26.11.2023 – 10.12.2023	
5.	Оформлення та друк пояснювальної записки.	11.12.2023 – 14.12.2023	
6.	Створення презентації, доповіді та підготовка до захисту дипломної роботи.	15.12.2023 – 20.12.2023	
7.	Підготовка матеріалів дипломної роботи для передачі секретарю ДЕК (папка, конверт, диск із файлом диплому, рецензія, відгук).	21.12.2022 – 22.12.2023	

7. Дата видачі завдання: «02» жовтня 2023 р.

Керівник дипломної роботи _____

(підпис керівника)

Юрій ЗІАТДІНОВ

(П.І.Б.)

Завдання прийняв до виконання _____

(підпис випускника)

Дар'я МАКОВСЬКА

(П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Система планування графіків польотів повітряних суден» складається зі вступу, трьох розділів, висновку, списку бібліографічних посилань та одного додатку і містить 105 сторінок, 27 рисунків, 2 таблиць. Список бібліографічних посилань включає в себе 30 найменувань.

Ключові слова: МАРШРУТИЗАЦІЯ, ПОВІТРЯНІ СУДНА, ПОВІТРЯНІ ТРАСИ, АВІАПЕРЕВЕЗЕННЯ, ОПТИМАЛЬНІ РЕЙСИ МОДУЛЬ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, МЕТОД ГІЛОК І МЕЖ, МЕТОД ГІЛОК І ЦІНИ, JAVA, SPRING FRAMEWORK, СИСТЕМА СТВОРЕННЯ ГРАФІКІВ ПОЛЬОТІВ.

Актуальність. На сьогоднішній день, авіаційний транспорт є незмінним лідером рейтингу серед всіх можливих видів транспорту. Він забезпечує своїх клієнтів відносною швидкістю та регулярністю сполучень, безпекою та комфортом. Розвиток авіаіндустрії та зростання попиту на авіап перевезення загалом призводить до загострення нагальної потреби забезпечення чіткої логічної маршрутизації. Це цілком реально реалізувати завдяки створенню певних циклів рейсів, які складаються з обов'язкових і необов'язкових рейсів. Тому виникла потреба створення продукту для формування графіків польотів. Не зважаючи на закритий повітряний простір через війну в нашій країні це не має зупинити розвиток і новітні розробки в сфері авіації, адже по закінченню війни потрібно буде прикласти багато зусиль у відбудову і оновлення всіх сфер функціонування країни.

Метою дипломної роботи є дослідження методів для розробки графіків польотів, задач направлених на вирішення основного завдання маршрутизації, створення системи для знаходження і побудови повних циклів, що складаються з рейсів які обираються з вибірки вже існуючих. В наслідку, на основі отриманих результатів необхідно реалізувати систему планування графіків польотів повітряних суден.

Для досягнення мети роботи необхідно вирішити такі підзавдання:

- дослідити історію авіап перевезень і авіаційного транспорту загалом;
- детально ознайомитися з поняттям свободи повітряного простору та

вузловий аеропорт;

- визначити основні фази формування графіків польотів;
- сформулювати технічне завдання для розробки;
- провести аналіз принципів і методів вирішення завдання створення

графіків польоту повітряних суден;

- окреслити математичну модель продукту, що проектується;
- визначити основне завдання;
- обрати алгоритм для вирішення задачі і застосувати його;
- ознайомитися з варіантами моделей для розробки програмного

забезпечення;

- обрати мову програмування і допоміжні засоби;
- створити систему на основі описаних методів і алгоритмів.

Об'єктом дослідження є процес аналізу методів маршрутизації авіатранспорту.

Предметом дослідження є методи вирішення задач шляхом розгалужень з урахуванням затрат.

Методи дослідження включають у себе:

- методи побудови графіків польотів повітряних суден;
- метод комбінаторної оптимізації для вирішення цілочисельної задачі

лінійного програмування;

- метод генерації стовпців;
- метод гілок і ціни;
- метод гілок і меж;
- метод водоспадної моделі розробки програмного забезпечення.

Теоретичною основою дипломної роботи стали вітчизняні та зарубіжні дослідження щодо забезпечення якості програмного забезпечення та публікації на

сайтах, присвячені питання маршрутизації, створення графіків польотів і розвитку авіаперевезень загалом.

Теоретична і практична значимість роботи полягає в тому, що на основі отриманих знань:

1) дійсно можна за малий проміжок часу ознайомитися з історією авіаційної галузі та зрозуміти рівень необхідності створення автоматизованої системи розробки графіків польотів повітряних суден;

2) зі підбраного матеріалу є можливість розробити окремий методичний посібник для студентів;

3) розроблено систему для розробки графіків польотів повітряних суден.

На захист виносяться наступні положення:

1) процес аналізу авіаційного транспорту;

2) розробка математичної моделі основного завдання;

3) реалізація системи розробки графіків польотів.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ.....	13
1.1 Загальна характеристика транспорту світу.....	13
1.2 Історія міжнародних повітряних рейсів.....	15
1.2.1 Свободи повітряного простору.....	19
1.2.2 Вузлові аеропорти та міжнародні компанії.....	25
1.3 Стадії формування графіків польотів.....	31
1.4 Аналіз принципів вирішення основного завдання	33
Висновки до Розділу 1	36
РОЗДІЛ 2 МАТЕМАТИЧНА МОДЕЛЬ.....	40
2.1 Постановка задачі.....	40
2.2 Застосування методу Branch and Price	41
2.2.1 Застосування евристичного методу для побудови основного завдання... 44	
2.2.2 Додаткове завдання	45
2.4 Пошук цілочисельного розв'язку основного завдання.....	47
2.3 Адаптація алгоритму методу для створення системи	49
Висновки до Розділу 2	50
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ ДЛЯ СТВОРЕННЯ ГРАФІКУ ПОЛЬОТІВ ПОВІТРЯНИХ СУДЕН	52
3.1 Вибір методології розробки програмного продукту	52
3.2 Проектування програмного продукту	62

3.3 Створення діаграми варіантів використання	64
3.4 Створення інтерфейсу програми	68
Висновки до Розділу 3	73
ВИСНОВКИ	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	83
ДОДАТОК А . ТЕКСТ ПРОГРАМИ.....	87

ВСТУП

На сьогоднішній день важко уявити прогресивну країну без авіасполучення. Натомість в нашій країні наразі відбувається війна, повітряний простір закритий, більшість аеропортів розбомблена і деякі з них ніколи більше не запрацюють. Але це все не має зупиняти розвиток і розробки в різних галузях, адже по закінченню війни потрібно буде прикласти багато зусиль у відбудову і оновлення всіх сфер функціонування країни. Авіагалузь має високий пріоритет на відновлення, адже складно уявити успішну державу без її наявності. Авіація безпосередньо впливає на економіку країни, можливість імпорту і експорту, туризм, який також вносить свій вклад в економіку країни. Тому розвиток авіації має вагоме значення для країн, що розвиваються.

Авіаційний транспорт має неабиякий вплив в світовій економіці. Повітряний транспорт дає змогу швидко доставляти як пасажирів так і товар з точки А в точку Б на дуже масштабні відстані. Це все є основою якісного сполучення. Завдяки авіатранспорту можна доставити людей і вантаж в майже будь-яку точку світу, саме це дає величезний поштовх до глобалізації. Поряд з цим значного розвитку зазнає світова економіка, адже на неї значно впливає розробка новітніх засобів і автоматизація зв'язку і транспорту.

Глобальне економічне зростання сприяє підвищенню попиту на авіаційні перевезення. Потреби зростають прямо пропорційно до міри підвищення самої економіки. Однією з найголовніших причин зростання світової економіки є саме авіаперевезення. Адже збільшення потреб прямо пропорційно впливає на збільшення і розвиток галузі, яка в свою чергу на пряму впливає на світову економіку. Постійне збільшення шляхів зв'язку з глобальною магістраллю авіаперевезень сприяє підвищенню продуктивності і більшому задоволенню потреб, надає полегшений доступ до ринків, покращуючи як внутрішні, так і зовнішні зв'язки між виробництвами. В свою чергу це дає змогу використовувати доступніший шлях до

ресурсів і світових ринків капіталу. Глобальний економічний розвиток можливий тільки при наявності такого необхідного фактора як повітряна галузь. Більше третини всіх товарів і торгів відправляється авіатранспортом, адже це досить швидко, а для товару що псується це ключовий фактор. Тому міжнародний ринковий бізнес нереально уявити без повітряного транспорту.

Повітряне сполучення створює єдину в світі надшвидку транспортну мережу. Вона є необхідною для існування бізнесу. Існування бізнесу з повітряним сполученням сприяє економічному зростанню. Економічне зростання має властивість генерування нових робочих місць. Як наслідок стимулює зовнішню міжнародну торгівлю та туризм.

За останній період, згідно з останніми оцінками Міжгалузевої групи дій з повітряного транспорту (АТАГ), загальний економічний вплив (прямий, непрямий, індукований та пов'язаний з туризмом) світової авіаційної галузі досяг 2,7 трильйона доларів США, що становить близько 3,5 відсотка світового валового внутрішнього продукту (ВВП).

АТАГ була опублікувала доповідь *Aviation: Benefits Beyond Borders*, в якій досліджувалися різні способи, за допомогою яких авіаційна галузь робить свій внесок у світову економіку, а також як вона сприяє сталому розвитку.

Даний звіт широко цитується засобами масової інформації і тим самим спонукає до більш розширеного погляду на те, як авіаційна галузь створює робочі місця та стимулює економічне зростання держави.

Повітряний транспорт дає дуже важливі економічні вигоди:

- Авіаційна галузь забезпечує єдину у світі транспортну мережу, яка є важливою для розвитку бізнесу та туризму. Вона відіграє життєво важливу роль у сприянні економічному зростанню.
- За допомогою авіаційного транспорту перевозиться близько 2 мільярдів пасажирів на рік і 40% міжрегіонального експорту товарів (за значенням).
- 40% міжнародних туристів мають можливість подорожувати повітряним транспортом.
- Авіаційна галузь створює загалом 29 мільйонів робочих місць у всьому світі

- (через прямі, непрямі, індуковані та каталітичні впливи).
- Глобальний економічний вплив авіації (прямий, непрямий, індукований і каталітичний) оцінюється в 2960 мільярдів доларів США, що еквівалентно 8% світового валового внутрішнього продукту (ВВП).
- 900 авіакомпаній світу мають загальний парк з майже 22 000 літаків. Вони обслуговують близько 1670 аеропортів через мережу маршрутів протяжністю кілька мільйонів кілометрів, якими керують близько 160 постачальників аеронавігаційного обслуговування.
- 25% продажів усіх компаній залежать від авіатранспорту. 70% компаній повідомляють, що обслуговування більшого ринку є ключовою перевагою використання авіапослуг. Щодо соціальних переваг, то повітряний транспорт дає значні соціальні вигоди:
- Повітряний транспорт покращує якість життя, розширюючи відпочинок та культурний досвід людей. Він пропонує широкий вибір місць відпочинку по всьому світу. • Повітряний транспорт допомагає підвищити рівень життя та зменшити бідність, наприклад, через туризм.
- Повітряний транспорт може бути єдиним транспортним засобом у віддалених районах, що сприяє соціальному залученню.
- Повітряний транспорт сприяє сталому розвитку. Сприяючи туризму та торгівлі, він створює економічне зростання, забезпечує робочі місця, збільшує надходження від податків та сприяє збереженню заповідних територій.
- Мережа повітряного транспорту сприяє доставці екстреної та гуманітарної допомоги будь-де на землі, а також забезпечує швидку доставку медичних матеріалів та органів для трансплантації.
- Повітряний транспорт сприяє економічному та соціальному прогресу. Очікується, що попит на повітряний транспорт зростатиме в середньому на 3% річних протягом наступних 20 років. Якщо це зростання буде досягнуто, до 2038 року авіатранспортна галузь сприятиме: 13,7 мільйона прямих робочих місць та 1,7 трильйона доларів ВВП у світову економіку 76

- мільйонів робочих місць, включаючи непрямі та індуковані внески, та 4,3 трильйона доларів ВВП [30].

Розглядаючи сферу повітряного транспорту в останні роки варто зауважити наскільки зросло загальне число пасажиропотоку. Це зростання відбулося завдяки створенню нових повітряних маршрутів, а саме повітряних магістралей, які мають протяжність десятки кілометрів. Ще одним важливим чинником зростання став бурхливий та стрімкий розвиток вантажних і пасажирських перевезень в середині країни(внутрішніх) в сфері авіації. Саме кількість перевезень пасажирів і товарів, а також значна динаміка зростання цієї кількості є основою ефективності авіаперевезень.

Добробут населення – відсотковий показник від якого залежить попит на перевезень (як зовнішніх так і внутрішні). Попит є числовою похідною від добробуту населення. Як наслідок, в цьому випадку прямі рейси без пересадок не надають великого збільшення показника попиту.

Якщо розглянути ВВП певної країни і вплив його на темп зростання кількості авіаперевезень то прослідковується лінійна залежність відносно його росту чи спаданню. Якщо ж порівняти прямі і транзитні перевезення, то другі мають мультиефект, бо одним рейсом можна виконати одночасно кілька маршрутів. Це стає реальним при вигідному стратегічному стикуванні і складанні маршрутів, а також при правильному складанні графіків польотів.

Розглядаючи на прикладі Україну та її транспортні вузли важливу роль відіграє точка трансферу Бориспіль в маршрутних сполученнях Заходу і Сходу країни. Пункт трансферу являє собою проміжне місце де пасажиру необхідно зробити пересадку на інший рейс для подальшого слідування маршрутом. Наявність таких пунктів надала значне зрушення в сфері новацій авіагалузі в Україні.

- Початок роботи з видом трансферних перевезень піднімає питання розробки новітнього підходу до планування графіків польотів повітряних суден (ПС), при цьому обов'язковою вимогою до нових проектів є саме автоматизація процесів.

РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальна характеристика транспорту світу

Однією з найважливіших видів діяльності суспільства в сучасному світі є транспортна система. Потреби суспільства в будь-яких видах перевезень, стабільне функціонування світового господарства, можливість сполучень різних точок на карті світу – це все забезпечує транспорт, який також є основою міжнародного географічного поділу праці. Варто зазначити, що світова транспортна система утворюється сукупністю всіх транспортних шляхів світу. Певні умови впливають на вибір використання різних видів транспорту в залежності від конкретних потреб.

Відповідно до рівня розвитку транспортної системи країни поділяються на такі:

- *високорозвинені країни, ті, що мають всі види транспорту* (наприклад, держави Європи, США, Канада);
- *країни, що мають високий рівень розвитку більшості видів транспорту* (наприклад, Японія, бо в країні не представлено трубопровідних та річкових перевезень; Австралія, Нова Зеландія, Південна Африка, бо перелічені країни майже не мають річкового транспорту);
- *країни, в яких переважає один певний вид транспорту* (в Індії, Аргентині, Марокко переважає саме залізничний вид перевезень, а ось автомобільний — в Афганістані, Саудівській Аравії, Ефіопії, а також у Судані, Конго, Парагваї – домінуючим видом являється річковий транспорт) .

Кафедра КІТ(47)				НАУ 23 24 09 000 ПЗ			
Виконала	Маковська Д.Ю.			Огляд предметної області	Літера	Аркуш	Аркушів
Керівник	Зіатдінов Ю.К.					13	27
Консульт.					УС-211М 122		
Н-контр.	Райчев І.Е.						

В країнах Європи, Північної Америки з найбільш розвинутими транспортними системами сконцентрована більша частина залізничних колій, повітряних шляхів, автомобільних шляхів, автопарку та авіапарку. У вище перелічених країнах вже досягнутий високий технічний рівень розвитку транспорту, а саме:

- відносна безпека;
- велика швидкість перевезень;
- регулярне оновлення авіа-, автопарку;
- відносна дешевизна квитків;
- регулярність;
- масовість перевезень вантажів та пасажирів, що дає змогу знизити собівартість перевезень.

Варто відзначити такі країни з високим рівнем розвитку транспорту як Тайвань, Республіка Корея, країни Перської затоки(в яких добувається нафта), Японія. Також дуже швидким темпом оновлюється та осучаснюється транспортна система країни Китай.

Більшій частині країн, що розвиваються, притаманний саме досить низький темп і рівень розвитку транспортної системи. Їм властивий застарілий малопотужний рухомий склад. Прикладом слугують країни Азії, Африки та Латинської Америки, в яких розповсюджений гужовий транспорт(який пересувається силою м'язів різних тварин, таких як собак, оленів, слонів, буйволів, лам, віслюків, верблюдів, волів, коней): в'ючний(тварина несе вантаж безпосередньо на собі) та тягловий (тварина упряжена в сани, віз, бричку, примітивний екіпаж).

Рівень розвитку різноманітних видів транспорту виражається безліччю чинників:

- довжина системи шляхів сполучення;
- кількість актуального складу на ходу;
- кількість задіяного економічно активного населення в транспортній системі;
- обсяг перевезень, вантажо- та пасажирообігом.

Наймолодшим серед усіх видів транспорту є авіаційний транспорт, але не дивлячись на це саме за темпами розвитку він значно випереджає всі інші види транспорту. Останнім часом роль повітряного транспорту в пасажиро-перевезеннях значно зростає. Особливістю даного транспорту є те, що через відносно високу собівартість перевезень, самий цей вид транспорту використовуються для перевезення вантажів, що швидко псуються: тропічних фруктів, екзотичних квітів, а також пошти й коштовностей, тобто тих продуктів і засобів для яких швидкість доставки з умовної точки А до точки Б є найголовнішим чинником. За даним показником авіаційний транспорт поступається всього лише автомобільному та залізничному, але при цьому в міжконтинентальних видах перевезень є безсумнівним лідером. Найпопулярніші країни за авіаперевезеннями пасажирів: США, Канада, Велика Британія, Австралія, Франція, Німеччина, Японія. П'ятдесят п'ять відсотків усіх перевезень припадають на міжнародні лінії(перевезення пасажирів і вантажів транспортом з перетином одного або декількох державних кордонів), а інші сорок п'ять відсотків – на внутрішні(в межах однієї країни).

Основну частину світового повітряного транспорту становить мережа аеропортів. Загалом їх налічується біля 25 тисяч. Найбільшими з них є ті, в яких пасажирообіг на рік становить понад 10 млн осіб. Найбільша їх чисельність базується в Сполучених Штатах Америки. Більшість пасажирів приймає аеропорт «Хартсфілд-Джексон» (Атланта) у США (92,5 млн щорічно), «Пекін-Столичний» у Китаї (77,4 млн), «Хітроу» (Лондон) у Великій Британії (69,4 млн), «О'Хара» (Чикаго) в США (66,6 млн), «Міжнародний аеропорт Токіо» в Японії (62,3 млн). Серед аеропортів Європи (окрім британського «Хітроу») вирізняються за пасажиропотоками «Шарль де Голь» (Париж) у Франції (61 млн), «Франкфурт» (Франкфурт-на-Майні) у Німеччині (56,4 млн), «Схіпхол» (Амстердам) у Нідерландах (49,8 млн), «Мадрид Барахас» (Мадрид) в Іспанії (49,6 млн)[27].

1.2 Історія міжнародних повітряних рейсів

Багато тисяч років до винаходів літальних апаратів людство намагалось підняти

в повітря різноманітні предмети, тварин та звичайно ж навіть себе. Хтось намагався стрибати величезної висоти намагаючись спіймати хвилю повітряного потоку і балансувати, хтось досліджував тканини і робив експерименти з тканинами щоб визначити найбільш підходящу наприклад для повітряних зміїв. Більшість таких спроб була невдалими але їх результати давали підґрунтя для удосконалення експериментів, які пізніше стали основою формувань висновків і міцним фундаментом для нових розробок льотних апаратів.

В наукових матеріалах досить часто можна побачити термін «повітряні перевезення», які ще називають «повітряні рейси», ці два терміни можна пояснити як перевезення пасажирів і будь-яких вантажів з точки А в точку Б. Ці рейси/перевезення виконуються різними транспортними засобами, які можуть підтримувати керований політ.

Перший політ пасажирів датується 1914 роком. З того часу відбулися радикальні зміни в галузі повітряних перевезень. Значним рушієм у вдосконаленні авіаційного транспорту в світі стали Перша та Друга світові війни. Як відомо, значення переваги в повітрі надає глобальні привілеї противнику, тому саме ці події значною мірою дали масштабний поштовх до розвитку повітряного транспорту.

Перша світова війна дала підґрунтя для великого розвитку в сфері авіації, що стало фундаментом для реалізації нових експериментів і виготовлення повітряних суден вже в часи Другої світової війни.

Якщо ж розглядати сьогодні, то варто зазначити, що повномасштабна війна в Україні в 2022 також слугувала глобальним поштовхом у розвитку української авіації, а також її оновленням. Наразі, українські інженери і науковці працюють над новими проектами і розробками установок Проти повітряної оборони, військової авіації та повітряними дронами, щоб в подальшому забезпечувати вищий рівень захисту і мати перевагу в повіті, що значно полегшить хід війни.

Такі події призводять до популяризації авіаперевезень, адже являються невід'ємними й досить ефективними елементами військових стратегій, які полегшують реалізацію масштабних воєнних тактик. Це сприяє відкриттю великої кількості нових аеропортів, а також збільшенню кількості кваліфікованих авіапілотів

і персоналу з обслуговування авіації, що сприяє збільшенню кількості робочих місць.

Значний крок відбувається і в розвитку реактивної авіації. Це дає змогу повітряним перевезенням зайняти лідерське місце в пасажирських і вантажних перевезеннях.

Деякі країни ризикуючи почали залучати авіацію в сфери функціонування своєї країни, до прикладу Сполучені Штати Америки одними з перших розробили проект доставки пошти авіарейсами і запустили його в дію. В цей час країни Європи працювали над створенням нових авіаліній і розширенням меж авіасполучень.

Значним проривом в авіації стало розробка і поширення використання реактивних двигунів. Це значною мірою вплинуло швидкість польоту повітряного транспорту, що майже вдвічі скоротило час польоту повітряних суден. На рисунках 1.1 та 1.2 схематично зображено різницю у часі сполучення з Нью-Йорку до будь-якої точки на землі, де на першому малюнку зображений час польоту на повітряному судні з поршневым двигуном, а на другому – з реактивним.

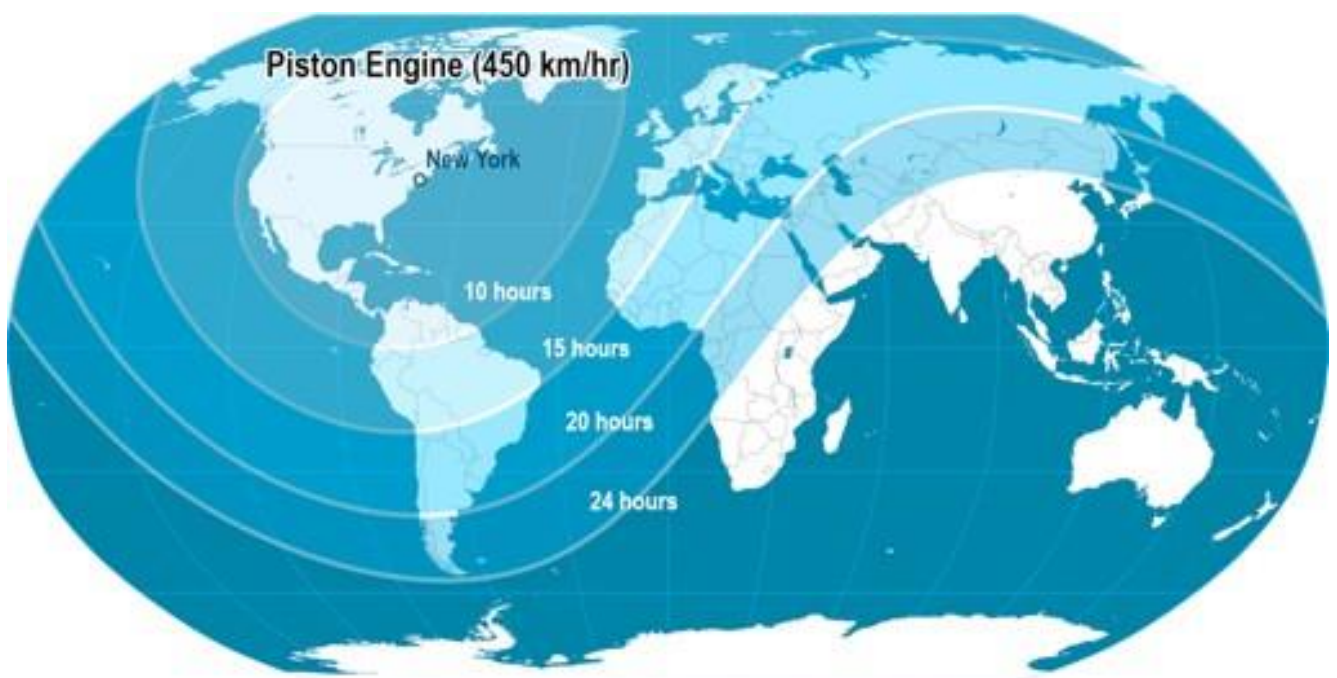


Рис.1.1 Час польоту на повітряному судні з поршневым двигуном

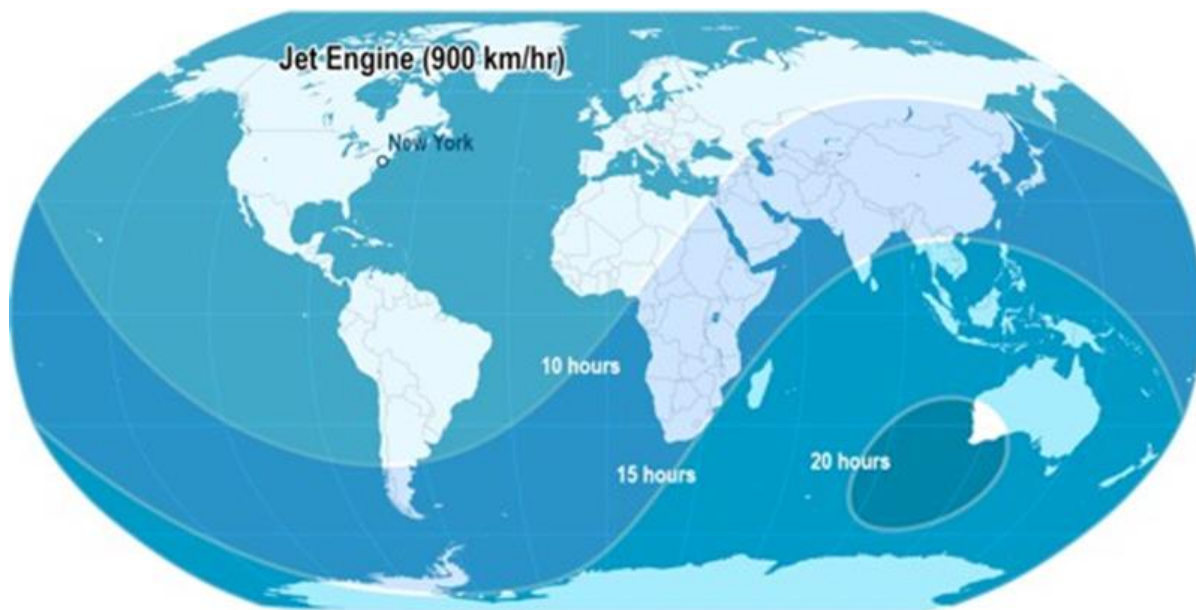


Рис.1.2 Час польоту на повітряному судні з реактивним двигуном

На сьогодні, авіаперевезення є найбільш домінуючим способом міжконтинентальних перевезень, це пояснюється швидкістю сполучення, яка має значну перевагу в порівнянні з будь-яким іншим видом транспорту.

Ще в 1960-х роках повітряні перевезення не мали такої популярності, адже даний тип транспорту був досить небезпечним, за статистикою в середньому за день ставалось три авіапригоди. Завдяки цьому пасажирів часто боялися подорожувати авіатранспортом, а власники бізнесу не довіряли авіації перевозити свою продукцію щоб не зазнати значних збитків.

Але прогрес розвитку авіагалузі призвів до шаленого зменшення числа ризиків та авіапригод під час рейсів. Вже в 2018 році за рік зафіксовано тільки дев'ять інцидентів за рік [6], що наразі дає змогу зробити висновок, що даний вид транспорту є досить безпечний.

Важливим для світової торгівлі є перевезення літаками вантажів. За оцінками експертів частка таких транспортувань наразі в міжнародній економіці становить менше одного відсотка, та за вартістю такі товари займають 35% від загальної ціни перевезених вантажів[6].

Важливою перевагою повітряних перевезень є їх швидкість, тому сама

авіарейси є найбільш поширеними для перевезень вантажів, які необхідно якнайшвидше транспортувати на масштабні відстані за найкоротший проміжок часу, або в тих випадках, коли швидкість доставки має велике значення, хоч такий транспорт є досить затратним.

Історія цивільної авіації досить довга та яскрава, як і список досягнень в її сфері. Важко переоцінити вклад і вплив авіації на сучасний світ включаючи найрізноманітніші сфери життя суспільства, як з позитивної сторони (проявляється в розквіті міжнародного туризму, розповсюдження культурного різноманіття, розвиток міжнародного експорту та імпорту), так і з негативної (масштабне природне забруднення, стрімке розповсюдження інфекційних захворювань непритаманних іншим континентам).

1.2.1 Свободи повітряного простору

В цивільній і комерційній авіації існує сталий набір правил, що надають право певній авіакомпанії на вхід в повітряний простір і посадку в іншій країні. Щодо вибору маршруту авіатранспорт в порівнянні з будь-яким іншим видом транспорту має значно глобальнішу свободу, адже авіалінії мають здатність охоплювати озера, моря, канали, океани, гори, пустелі, важкодоступні місця і майже будь-які перепони.

Але з розвитком авіаційної промисловості, з еволюцією процесів пасажирських перевезень і міжнародних вантажних рейсів постала необхідність створення певних правил користуванням повітряним простором, щоб не допускати авіакатастроф і критичних ситуацій. Ці процеси розвитку і еволюції були пов'язані напяму з підготовкою, а потім і ходом Другої світової війни, яка в свою чергу загострювала проблему пасажирських міжконтинентальних рейсів. Вирішенням цих проблем стала прийнята і підписана в 1944 році було Чиказька конвенція.

Вже до 1940-их років глобальна індустріальна промисловість і

популяризація міжнародного авіаційного транспорту в розвинених державах призводить до виникнення масштабної проблеми, адже подальший успішний розвиток авіагалузі був би можливий тільки за єдиної умови – об'єднання і узгодження певних чітких правил користування повітряним простором інших держав, які також були зацікавлені в розвитку міжнародних сполучень. Виникнула масштабна мережа міжнародних пасажирських і вантажних повітряних сполучень, що дало підґрунтя для низки проблем різного характеру включаючи політичні, комерційні, технічні, індустріальні, промислові проблеми. Ці всі проблеми мало вирішити вкладення певного списку/акту правил, якого б дотримувалась більшість держав в інтересах розвитку торгівлі, пасажирських перевезеннях, розвитку туризму та глобальної індустріалізації. Ключовим аспектом даної конвенції мало стати заключення миру між державами, які приймали її.

Проблеми, які піднімалися під час прийняття конвенції були пов'язані зі стурбованістю сторін можливого зародження юридичних і економічних суперечок. Такі конфлікти найчастіше виникали при порушенні суверенності повітряними судами державних кордонів іншої країни без дозволу. Ще одним важливим чинником було недотримання прав і невиконання обов'язків у зв'язку з польотами однієї країни в межах або через територію інших країн. Загострилось приватне питання міжнародних повітряних сполучень.

Стало зрозуміло що вирішення цих виникнувши проблем можливо лише шляхом укладання певних угод серед міжнародних авіаперевізниками, на базі окремих міждержавних або багатосторонніх договорів.

З часом все більше загострювалась потреба системної організації міждержавних взаємозв'язків під час повітряних рейсів авіатранспорту. Поставала проблема щодо розподілу і зафіксування відповідальної сторони серед учасників авіадіяльності як і на рівні міжнародних і міжконтинентальних перевезень, так і на рівні внутрішніх польотів в середині держави.

Прийняття і встановлення всеосяжних міжнародних певних норм і правил для врегулювання цих проблем і потреб слугувало б забезпеченням

сприятливих умов для подальшого розвитку і домінування над іншими видами транспорту. Таким чином було прийняте рішення формування певної низки принципів і правил, які б в свою чергу забезпечували такі аспекти діяльності авіації:

- умови, які зазначалися в угоді мали виконуватись і дотримуватись чітко по стандартам всіма країнами учасниками цього договору;
- надання безпеки, високої результативності і регулярності авіасполучень;
- високий ступінь схожості врегулювання взаємозв'язків, які утворюються при повітряних міжнародних перевезеннях між різними державами;
- умов для правильного і ефективного функціонування, що в подальшому має призвести до стрімкого розвитку і вдосконалення міжнародної авіаційної діяльності транспорту.

Такими принципами і правилами стали дев'ять свобод повітряного простору (Freedoms of the air, див. детальніше Таблицю 1.1 та Рисунок 1.3).

Свободи повітряного простору

№	Опис	Приклад
I	Проліт над іноземною країною (без посадки).	З Мексики до Канади мексиканською авіакомпанією над США.
II	Дозаправка (або технічне обслуговування) без висадки пасажирів.	Дозаправка в Ірландії при польоті з Великої Британії в США британською авіакомпанією.
III	З "домашньої" країни в іншу.	Політ австралійською авіакомпанією з дому в Китай.
IV	Політ з "третьої" країни додому.	Політ з Чилі бразильців додому.
V	Політ між двома іноземними країнами в рейсі, що починається чи завершується власною (-ій).	Рейс з ОАЕ в Нову Зеландію, здійснений авіакомпанією Emirates (ОАЕ) із зупинкою в Австралії.
VI	Право на політ з однієї іноземної країни до іншої та зупинкою у власній з нетехнічних причин.	Політ з Австралії до США французькою авіакомпанією із зупинкою у Французькій Полінезії.
VII	Переліт між двома іноземними країнами без "заходу" до власної.	З Португалії до Німеччини ірландською авіакомпанією.

VIII	Політ в іноземній країні з продовженням у власній.	Рейс англійської авіакомпанії між Сан-Франциско та Лондоном, з повною зупинкою в Нью-Йорку.
IX	Право літати в іноземній країні без продовження рейсу у власній.	Переліт з Лондона в Стамбул італійською авіакомпанією.

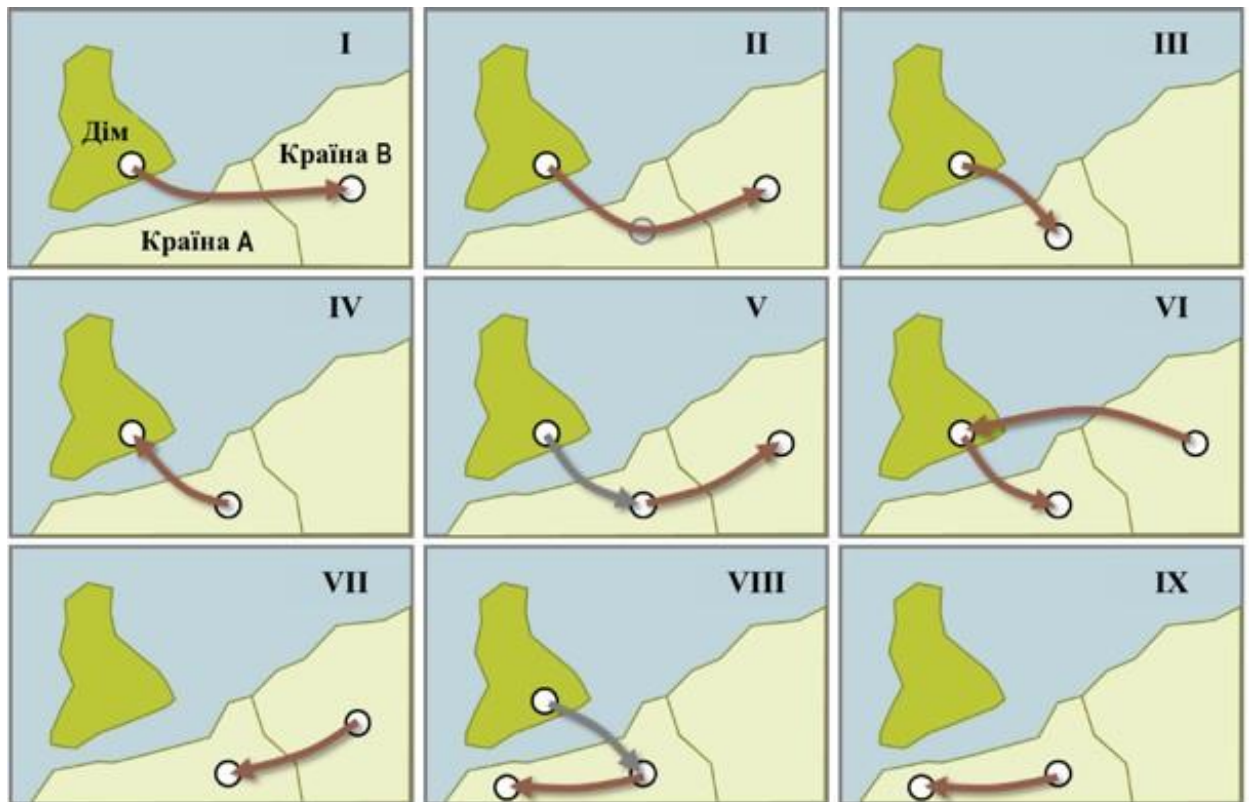


Рисунок 1.3 Ілюстрація конвенції Свобод повітряного простору

З 1944 року діє положення Чиказької конвенції, яка регулює принципи і правила діяльності авіаційного транспорту і правила поведіння транспорту в повітряному просторі як своєї країни, так і в інших. Складається ця угода з дев'яносто шести статей, які регулюють відносини держав між собою, при цьому всі учасники конвенції зобов'язані виконувати ці правила без виключень.

Для зручнішого використання, SARPs подається у вигляді Додатків до Чиказької конвенції (Annexes to the Convention on International Civil Aviation). На

сьогоднішній день виділяють вісімнадцять Додатків, які охоплюють найголовніші напрямки діяльності міжнародної авіагалузі. Деякі з них (наприклад, Додаток 16) прийняті багатьма державами як національних стандартів.

Склад Додатків до Чиказької конвенції про міжнародну цивільну авіацію:

- 1. Видача свідоцтва особовому складу (Personnel Licensing);
- 2. Правила польотів (Rules of the Air);
- 3. Метеорологічне забезпечення міжнародної аеронавігації (Meteorological Service for International Air Navigation);
- 4. Аеронавігаційні карти (Aeronautical Charts);
- 5. Одиниці виміру, що підлягають використанню в повітряних і наземних операціях (Units of Measurement to be Used in Air and Ground Operations);
- 6. Експлуатація повітряних суден (Operation of Aircraft);
- 7. Національні та реєстраційні знаки повітряних суден (Aircraft Nationality and Registration Marks);
- 8. Льотна придатність повітряних суден (Airworthiness of Aircraft);
- 9. Спрощення формальностей (Facilitation);
- 10. Авіаційна телекомунікаційна зв'язок (Aeronautical Telecommunications);
- 11. Обслуговування повітряного руху (Air Traffic Services);
- 12. Пошук і порятунок (Search and Rescue);
- 13. Розслідування авіаційних подій (Aircraft Accident and Incident Investigation);
- 14. Аеродроми (Aerodromes);
- 15. Служби аеронавігаційної інформації (Aeronautical Information Services);
- 16. Охорона навколишнього середовища (Environmental Protection);
- 17. Безпека: захист міжнародної цивільної авіації від актів незаконного втручання (Security: Safeguarding International Civil Aviation Against Acts of Unlawful Interference);
- 18. Безпечне перевезення небезпечних вантажів по повітряю (The Safe

Значним поштовхом у розвитку авіасфери стало створення окремих служб повітряного транспорту з перевезення вантажу. Найчастіше такі служби функціонують використовуючи інші повітряні траси, не такі як в пасажирських мережах. Пояснюється це так: саме пасажирські служби фокусуються на туристів, коли в свою чергу служби з перевезення вантажу на вантажовідправників. Це значно вплинуло на появу появи дій з полегшеними повітряними суднами, які вирізняються своєю великою вантажомісткістю. Не варто забувати й те що в сфері перевезень є комбіновані перевізники (наприклад, Lufthansa), які є досить поширеними в світі. Такі перевізники поєднують і виконують задачі як пасажирських так і вантажних перевезень, що забезпечує їх змішаний флот літаків адаптованих під різні задачі.

Сезонність – це головна особливість повітряних перевезень мереж авіації. Піковими сезонами є напередодні Різдва, і деякі конкретні свята, до яких є затребуваним доставка якої-небудь тематичної продукції, як от наприклад квіти до Дня матері, святого Валентина, міжнародного жіночого Дня. За статистикою липень та серпень є піковими для пасажирських авіаперевезень. Це зумовлено максимальною популярністю туризму в цей період, адже саме ці місяці є так званими «лідерами» туристичного сезону завдяки сприятливій погоді.

А ось у Китаї апогей пасажирських повітряних перевезень фіксується в дні року, найближчі до Весняного фестивалю, або так званого Місячного Нового року що припадає на січень або лютий.

1.2.2 Вузлові аеропорти та міжнародні компанії

Майбутнє розвитку міжнародного ринку повітряних перевезень та рівень його розвитку на сьогодні напряму пов'язаний зі створенням спільних альянсів авіакомпаній, що формуються навколо аеропортів. На їх базі закладається мережа повітряних перевезень, що значною мірою збільшує число повітряних перевезень авіакомпаніям, які є членам альянсу, на базі цих аеропортів. Зрозуміло що існування

конкуренції на міжнародному ринку повітряних перевезень це класичний прояв боротьби альянсів авіаперевізників і масштабних аеропортів-хабів, які охоплюють глобальну частину ринку споживачів.

Вузловим аеропортом(хабом) можна вважати розподільчий аеропорт, який являє собою фундаментальний комунікаційний вузол радіальної системи повітряних сполучень. Він являє собою місце, де відбувається пересадка пасажирів з одного на інший авіарейс. Такі хаби розповсюджені переважно в великих або середніх за чисельністю населення містах.

Аналізуючи авіаційний ринок, відслідковуються два види розвитку авіаперевезень. Перший – на основі системи point-to-point, полягає в виконанні прямих польотів із пункту вильоту і в пункт призначення (часто зустрічається в США). Другий – який полягає в тому, що пасажир дістається до вузлового аеропорту(хабу) і в ньому має пересадку на інший рейс. Така схема притаманна країнам Європи. Головною умовою таких рейсів є те, що їх графіки були такими щоб пасажир швидко міг перейти на рейс стикування, при йому часу має бути достатньо щоб пасажир зміг не поспішаючи це зробити. Отже, основною задачею аеропортів-хабів є збір усіх пасажиропотоків з великого числа рейсів і перерозподіл цього потоку на рейси стикування. Важливою умовою для створення аеропорту-хабу налагоджена до ідеальності швидка техніка обслуговування пасажирів та обробки багажу. Це дає змогу налагодити достатній ступінь швидкості обслуговування клієнтів що є найголовнішою вимогою для нормального функціонування аеропортів з системою хабів.

Такі новітні потреби в авіасполученнях значною мірою коригують основи формування політики головних авіабудівних корпорацій. До прикладу компанія «Boeing»(США) вважає що майбутнє за швидкісними, високоекономічними лайнерами середньої пасажиромісткості, які зможуть влаштувати безпосадкове повітряне сполучення між безліччю міст світ, при цьому минаючи великі аеропорти, адже такі аеропорти найчастіше максимально навантажені. Західноєвропейський авіабудівний альянс «Airbus» приділяють свою увагу суперлайнеру А-380, що має здатність перевозити пасажирів з найбільш комфортними умовами при цьому

вміщуючи понад півтисячі пасажирів за один рейс. Але важливим нюансом є те, що такі повітряні судна матимуть змогу приймати тільки найбільші аеропорти в світі. Це призводить до того, що пасажирам прийдеться діставатися до свого пункту призначення на літаках але вже меншої ємності.

Враховуючи всі ці особливості провідні авіабудівні компанії розробляють різні види повітряних суден (ПС). Зважаючи на потенціал аеропортів і наявність глобального пасажиропотоку, не варто чекати що літаки з середньою пасажиромісткістю (за американською моделлю) будуть долітати до кожного аеропорту. Такі літаки найімовірніше будуть доставляти пасажирів до регіональних аеропортів, звідки їм доведеться добиратись до кінцевого пункту призначення місцевими авіалініями або взагалі іншим видом транспорту.

Отже, з цього слідує що в цих двох напрямках є нагальна проблема в аеропортах-хабах, їх масштаб і певні особливості будуть визначатися декількома пунктами: від пасажиропотоку, типу наявних в флоті повітряних суден, рівню стратегічної необхідності.

В Сполучених Штатах Америки на сьогоднішній день активно діють найбільші в усьому світі вузлові аеропорти-хаби, такі як: Атланта, Чикаго, Лос-Анджелес, Нью-Йорк, Х'юстон та інші. В таких вузлових аеропортах приблизний пасажиропотік фіксується в межах кілька десятків мільйонів осіб за рік.

Гострий занепад використання авіасполучення стався в 2019 році, саме такого впливу завдала пандемія коронавірусу на сферу авіації. Значне падіння обсягу пасажирських повітряних перевезень. Цей показник який впав аж на шістдесят відсотків і навіть фактично повернувся до рівня обсягу пасажирських перевезень 2003 року. На рисунку 1.4 можна наглядно побачити різкий спад обсягу пасажиропотоку під час пандемії.

World passenger traffic evolution 1945 – 2020

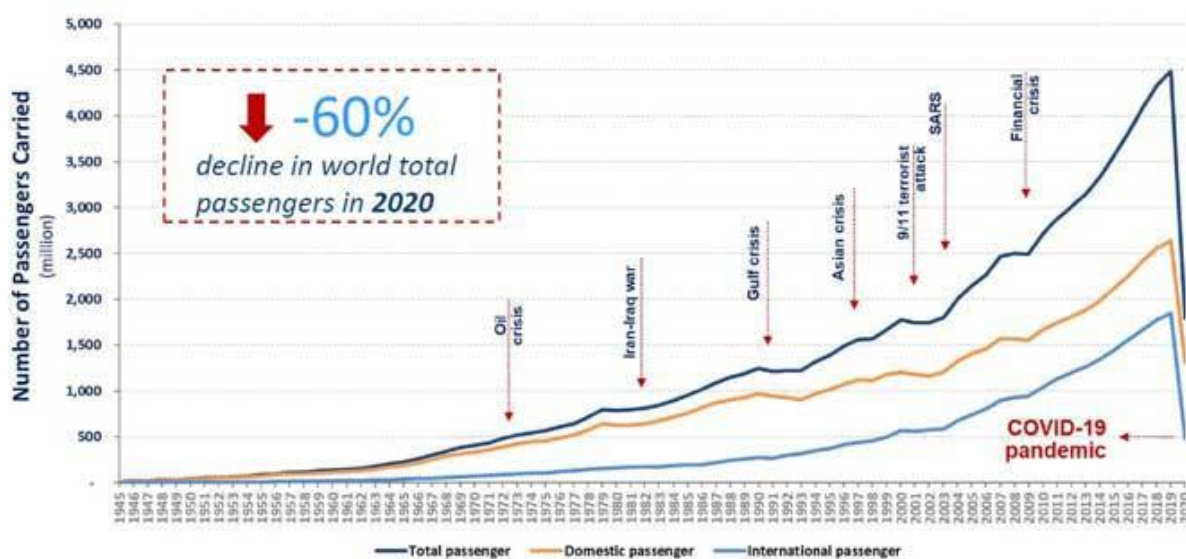


Рисунок 1.4 Графік пасажиропотоку в світі

Більшість авіаліній мають централізовані мережі, а ось локальні аеропорти найбільших компаній ринку – одні з найнавантажениших у світі.

Найголовніша суть мережі з вузловими аеропортами-хабами це застосування проміжного аеропорту-хабу в побудові транспортних маршрутів. На цій фазі можуть виникнути складнощі в збереженні цілісності графіків, бо переважна кількість пасажирів використовують транзитні рейси.

Аеропорти-хаби виконують найважливішу роль – регуляцію авіашляхів для повітряних суден. Даний процес дає змогу розглянути хаотичний шлях авіарейсів і перебудувати його в більш логічний, чіткий і систематичний, що неабияк полегшує створення чітких графіків польотів повітряних суден. На рис 1.5 візуально показано різницю хаотичних польотів до і після регуляції за допомогою вузлового аеропорту-хабу.

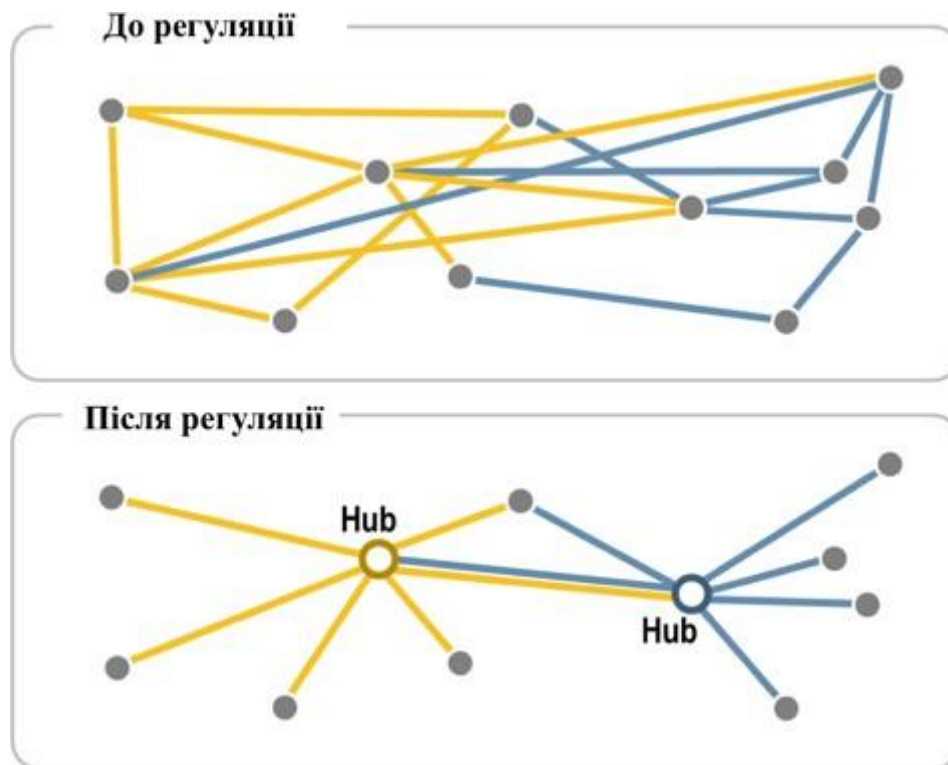


Рис 1.5 Повітряні траси до та після регуляції за допомогою хабу.

Популярність процесу регуляції і створення нових вузлових аеропортів-хабів довгі роки не мало великого темпу поширення. Це було зумовлено тим, що в різних країнах були різні норми регуляції, принципи і правила що перешкоджали розвитку і обмежували стрімке поширення і створення регуляторних хабів. На вирішення цієї проблеми прийшло утворення міжнародних альянсів. Вони мали на увазі складання непримусових договорів для об'єднання авіакомпаній зі спільною метою – посилення конкурентних позицій країн-партнерів.

При цьому авіакомпанії лишались незалежними в комерційному плані. При цьому це давала значну перевагу цим компаніям, яка досягалась шляхом зниження трансакційних розходів, а також рівномірного поділу небезпек серед компаній-партнерів. На сьогоднішній день існують три домінуючих повітряних альянсів:

- Star Alliance – найстаріший авіаційний альянс, утворений навесні 1997 року. Засновниками виступили п'ять авіакомпаній: Lufthansa, Air Canada, SAS, Thai Airways та United Airlines. Подібне об'єднання та угода про широке

співробітництво стало першим в історії цивільної авіації.

- SkyTeam. Другий за величиною і наймолодший авіаальянс, заснований влітку 2000 року. Ініціаторами створення виступили чотири авіакомпанії: Air France, Aeroméxico, Delta Air Lines та Korean Air. Загальний повітряний флот SkyTeam складається з 3600 літаків, що виконують понад 15 400 вильотів на день. Головний офіс альянсу знаходиться в амстердамському аеропорту Схіпхол (Amsterdam Airport Schiphol).
- Oneworld. Третій серед найвпливовіших альянсів. Був створений авіакомпаніями American Airlines, British Airways, Qantas Airways, Cathay Pacific та Canadian Airlines у лютому 1999 року. Штаб-квартирою спілки призначили Нью-Йорк. За 2019 рік рейсами альянсу скористалися 557 мільйонів пасажирів, що становило 18 відсотків усього ринку авіаперевезень.[29]

Аналізуючи всі аспекти створення і подальшої діяльності альянсів авіакомпаній можна виділити декілька основних переваг їх утворення:

- кодшерінг – це так зване групове використання літаків однієї компанії кількома авіаперевізниками на деякому маршруті. Авіакомпанії складають договір про групову торговельне використання певного авіарейсу. За даною угодою у пасажирів є можливість використати більш зручні стикові маршрути через вузловий хаб-аеропорт;
- партнери альянсу мають право збувати місця на рейсах один одного;
- оптимізація з'єднань – спільні термінали, що прискорюють зв'язки, наприклад, термінал №3 Міжнародного аеропорту Пекіна, де всі авіакомпанії Star Alliance, які обслуговують місто, розташовані спільно;
- зменшення затрат шляхом контролювання конкурентоспроможності через використання єдиних авіаційних тарифів;
- специфікація і унікальність обслуговування;
- колективний реліз квитків та рекламування;
- оптимізація графіків польотів завдяки розклад груповій розробці кількості

польотів.

Важливі договори про комерційну співпрацю надають змогу вирішити питання спільного признання перевізних угод тка паперів, оформлення перевезень, групові та пульні(від пул - вид монополії, суть якої об'єднання ринку до загального фонду) типи використання авіаліній.

В минулому столітті всі застали розквіт і глобальне зростання популяризації перевезень пасажирів і вантажів, але навіть на сьогоднішній день існує безліч вагомих невикористаних проблем і недоліків цієї галузі щоб стати безсумнівним лідером перевезень. Найголовнішою проблемою авіатранспорту на сьогодні досі лишається тероризм. Саме цей чинник займає чи не найбільшу частку авіапригод за статистикою останніх років. Тому головною задачею сфери авіаперевезень наразі є забезпечення безпеки та вдосконалення технологій та систем контролю її.

Ще однією глобальною проблемою галузі є високий рівень забруднення від пального. Тому критично необхідно започаткувати використання екологічно чистого пального. Третім головним питанням лишається оптимізація графіків польотів повітряних суден, адже наразі фіксується значне збільшення навантаження на лінії та авіатраси.

1.3 Стадії формування графіків польотів

Кожної секунди в повітряному просторі перебуває приблизно одинадцять тисяч повітряних суден[7]. Необхідною умовою продуктивної оптимізації є вдало розроблений графік польотів повітряних суден, який забезпечує безпеку польотів і регулює форсування авіації в залежності від потреб пасажирів або попиту тих чи інших товарів (рис. 1.6).

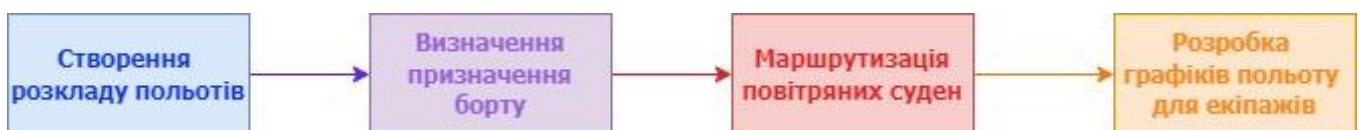


Рисунок 1.6 Схема процесу формування графіку польоту

Головною метою кожного етапу послідовності є пошук рішення для застосування на наступному кроці процесу [8].

Першочерговим етапом розробки графіка польотів повітряних суден є загальний розклад для авіації.

На цій стадії основними є декілька чинників, які мають значний вплив на розрахунок ймовірних характеристик своєї фінансової дальності. Серед таких чинників варто виділити:

- розробка прогнозування попиту пасажирів/постачальників продукції на певний продукт, в нашому випадку – рейси;
- співставлення інформації про наявність затребуваних ресурсів

Наслідком стабільної правильної діяльності під час першої фази матимемо можливість розробити низку можливих рейсів, які авіакомпанія в змозі зробити відповідно до своїх потенційних затребувань та відносно своєї реальної на те здатності.

Другою стадією розробки графіків польотів повітряних суден служить співставлення типу призначення повітряного судна на рейси, які були складені і заплановані на першому етапі планування графіків.

Основною задачею планування є коректне виконання рейсу. На це значний вплив має технічні параметри судна: структура і будова фюзеляжу, чисельність пасажирів, місткість, вантажопідйомність та ін. Метою цієї фази є визначення чисельності авіалайнерів та їх моделі конкуренти. Ці моделі в подальшому будуть застосовані авіакомпаніями назаплановані рейси, які планувались на першій стадії етапів формування графіків польотів.

Виконане складання переліку польотів з зазначенням певного номеру борту і маршруту за яким він слідує сигналізує про можливість переходу на наступний етап. На цьому етапі створюється графік польотів повітряних суден та їх чітка маршрутизація. Ця фаза є найскладнішою і трудозатратною, адже це пов'язано з великою кількістю регламентацій роботи членів бортового судна (переважна більшість з них керуються трудовим законодавством країни, в якій розташована авіакомпанія, та міжнародними регуляторними актами) та використанням тієї чи іншої моделі літака [9].

В даній магістерській дипломній роботі наведений саме останній крок формування графіків польотів повітряних суден. Планування графіку робіт моделюється як детермінована задача цілочисельного лінійного програмування, метою якого є мінімізація витрат на оплату робочого часу персоналу на борту літака [10], [11]. Авіакомпанії вигідно скорочувати частку оплачуваного часу, що не використовується безпосередньо для виконання польотів [12], [13].

1.4 Аналіз принципів вирішення основного завдання

Вже протягом багатьох років завдання створення графіків польотів повітряних суден є досить популярною темою для безлічі наукових праць. Транспортна маршрутизація завжди мала глобальний вплив на це завдання.

В 1968 році було створене один чи не найперший опис методу для вирішення задачі створення розкладу для бортової команди. Відсутність автоматизованої обчислювальної техніки (комп'ютерів) з необхідною обчислювальною потужністю пояснює те, що в цьому дослідженні були описані тільки евристичні методи, адже не можливо було отримати точні числові результати [14].

Згодом, через кілька років після того була написана інша наукова праця. Дана робота описувала методи, що надавали змогу автоматизувати процес планування графіків польотів [15]. При цьому значною проблемою лишалась мала кількість і дуже низька якість обчислювальних машин, а також доступних ресурсів. Тому науковці не мали змоги повною мірою відмовитись від евристичних методів.

В подальші роки після цього (період четвертого і п'ятого покоління електронно-обчислювальних машин ЕОМ) був зафіксований значний скачок в розвитку і відкриттях у галузі електронно-обчислювальної техніки. Головним досягненням цього періоду була розробка безлічі точних алгоритмів для знаходження найбільш оптимального вирішення задачі побудови графіків польоту повітряних суден.

Згодом, вже в 2004 році групою науковців було створено систему прийняття рішень для планування завдань для команд літака. Основою цієї системи полягала в

методі Branch and Price [16].

У прикладній математиці, Branch and Price – це метод комбінаторного оптимізація для вирішення завдань цілісного лінійного програмування (ILP) і змішаного цілого лінійного програмування (MILP) з багатьма змінними. Метод являє собою гібрид методів переходу та прив'язки та створення стовпців.

Branch and Price – це метод гілок та зв'язків, у якому до кожного вузла дерева пошуку можна додавати стовпці до релаксації лінійного програмування (релаксації LP). Суттю методу є те, що на початку алгоритму групи стовпчиків виключаються з релаксації LP, щоб знизити вимоги для обчислень і обсяг пам'яті, а потім стовпці знову додаються до релаксації LP, якщо це необхідно.

Даний підхід ґрунтується на спостереженні, адже для великих задач більшість стовпців будуть небазисними, а їх відповідна змінна дорівнює нулю в будь-якому оптимальному розв'язку. Отже, переважна більшість стовпців не мають ніякого значення для вирішення проблеми[24].

Схему розв'язку задач за допомогою методу Branch and Price можна побачити на рисю 1.7.

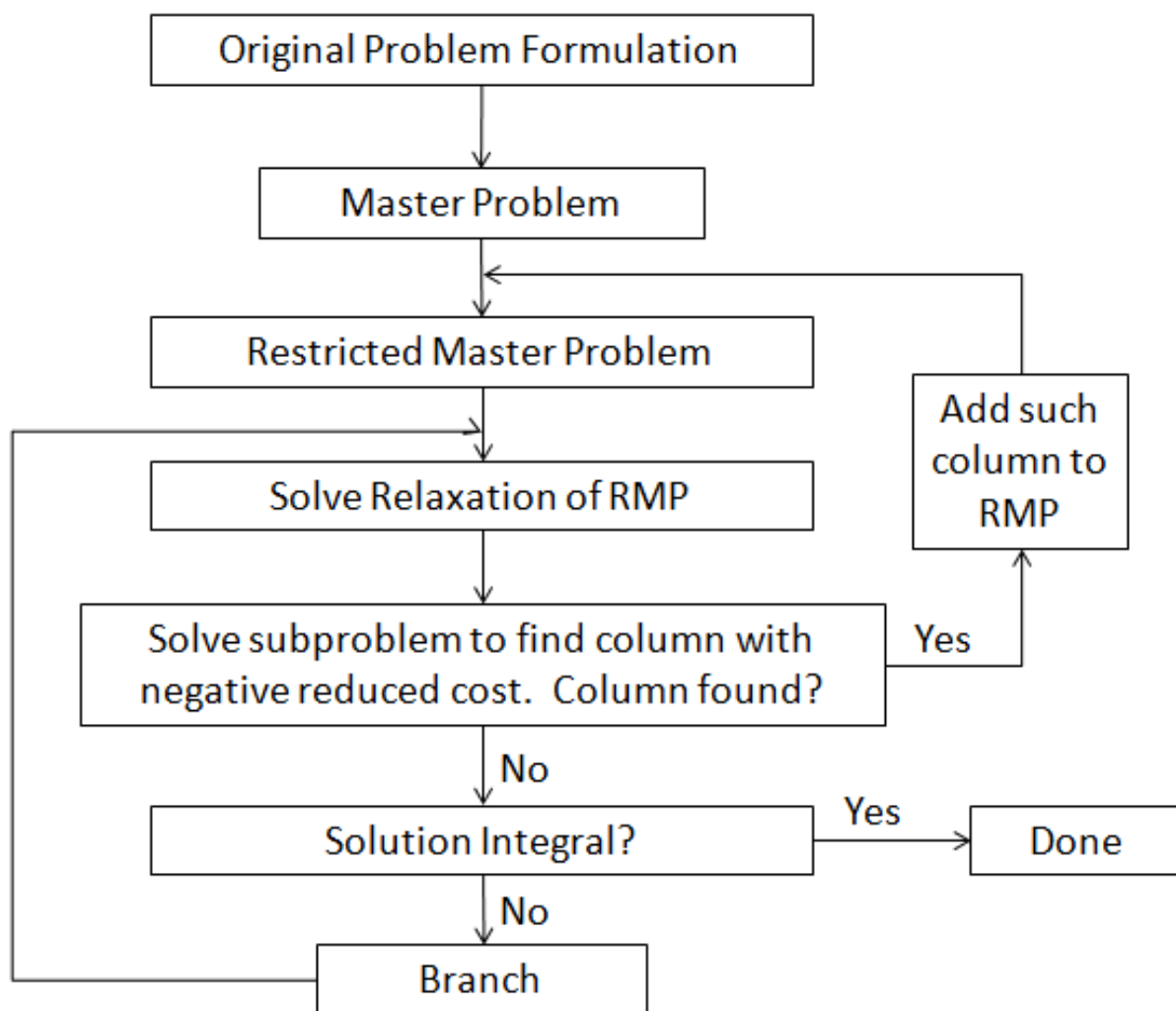


Рис. 1.7 Схема Branch and Price

Сам метод являє собою метод комбінаторної оптимізації для вирішення цілочисельної задачі лінійного програмування з великою кількістю змінних [17]. Відома дослідниця у сфері транспортної логістики Синтія Барнхарт разом з командою опублікували роботу про використання методу Branch and Price в даній галузі [18].

Алгоритм Branch and Price суміщає в собі метод гілок і меж та метод генерації стовпців для матричної задачі [19]. Метод гілок і меж вперше був запропонований в 1960 році та включає в себе три етапи: розбиття на гілки, вибір гілки і обчислення меж [20]. Даний алгоритм відзначається своєю доцільністю. На кожному етапі відбувається відсікання гілок, головною умовою якого є умова що обчислена межа наразі гірше за нинішнє ліпше рішення.

Метод генерації стовпців застосовується для вирішення завдань, в яких фігурують безліч змінних. В такому випадку доцільно алгоритм поділити на дві окремі складові. Першою складовою буде основна(глобальна) задача, в свою чергу підзадача(другорядна) буде другою складовою.

Ціль вирішення першої складової – відшукати межі для підмножини змінних для поставленого завдання. Головною умовою для початку розв’язання підзадачі є проведення релаксації стовпців і отримання вирішення двоїстої задачі.

Метою другої складової (підзадачі) є знаходження інших змінних, яких не було при розв’язанні першої складової. Завдання знаходження границь в двоїстій задачі еквівалентне меті знаходження іншої змінної для першої складової. В алгоритмі Branch and Price друга складова виконує пошук цього обмеження для забезпечення нормального здійснення більшого переліку границь в двоїстій задачі. Внесення ще одної нової змінної в першу складову задачі викликає до оновлення двоїстих змінних. Це призводить до змін цільової функції другої складової алгоритму Branch and Price.

Регуляція стовпчиків здійснюється до того моменту, доки усі поставлені границі двоїстої задачі не будуть виконуватися. В свою чергу це дає змогу зрозуміти, що найкраще значення релаксованої задачі виявлено [21]. За умови що відшукане рішення ціле, то в такому разі буде вважатись що початкове завдання вирішене, в протилежній ситуації необхідною умовою буде подальше розгалуження вже саме по дробовим змінним [22].

Висновки до Розділу 1

Починаючи з ХХ століття галузь авіаційного транспорту зазнала значних змін. Глобальний розвиток цієї індустрії припадає саме на другу половину ХХ століття, адже якраз повітряні перевезення стають свіжим ковтком повітря в галузі існуючих видів транспорту.

Масштабним поштовхом до розвитку авіації слугували Перша та Друга світові

війни. Саме вони грали ключову роль у зростанні технічного прогресу, адже літаки стали необхідним атрибутом для вдалих військових стратегій і тактик.

Розвиток галузі був необхідним ще і для створення нових робочих місць, що дало змогу залучити великий відсоток безробітне населення до вдосконалення сфери, це посприяло трохи м'якшому виходу населення з занепаду після бойових дій.

Також стають можливими міжконтинентальні сполучення, що дає змогу поширенню нового типу торгівлі, імпорту та експорту між країнами. Стрімко розвивається туризм, який навіть стає майже одразу для деяких країн ключовою складовою економіки країни.

Залучення науковців до розвитку сфери авіаційного транспорту і повітряних перевезень призвело до новітніх розробок і вдосконалення методів які застосовувались в цій сфері. Справжнім глобальним проривом слугувало створення реактивного двигуна, що майже вдвічі збільшував максимальну швидкість польоту повітряних суден, завдяки цьому досягненню стала можлива міжнародна торгівля продуктами, що швидко псуються.

Ще одним важливим кроком в авіаіндустрії став початок застосування радіохвиль. В цей час розвивались методи й алгоритми навігаційного координування. Це все стало вершиною післявоєнних технологій.

Саме повітряні перевезення вантажу і пасажирів за безліччю аспектів є поза конкуренцією всіх інших видів перевезень. Високий рівень безпеки наразі максимально намагаються надати авіаперевізники.

Наразі основною проблемою авіаційних перевезень лишається питання безпеки. Наразі індустрія розвивається і спрямовує багато сил на запобігання тероризму. В порівнянні минулим століттям індустрія авіації зазнала глобального позитивного впливу на формування безпеки пасажирів і вантажів. Ще одним напрямку розвитку авіагалузі є захист навколишнього середовища і досягнення мінімального негативного впливу на середовище, адже основною задачею наразі є саме мінімізація негативного впливу, бо наразі науковці ще не мають вдалих досягнень щодо повного припинення шкідливого впливу авіації на екологію.

Глобальний розвиток і популяризація авіаційного транспорту

пояснюється мінімальною кількістю перешкод, адже майже кожна країна наразі всіма силами намагається ці перешкоди усунути.

Найголовніші визначальні правила, що застосовуються до повітряного транспорту між собою, а також протягом проведення авіапольоту над проміжними країнами, були складені ще в першій половині ХХ століття, а найважливіші з них внесені до Чиказької конвенції в 1944 році і мають загальну назву Свободи повітряного транспорту.

Значним прогресом перевезень в авіаційній сфері стало створення незалежних грузових авіакомпаній, які найчастіше форсують у інших мережах, аніж пасажирські авіакомпанії. Але існують і такі перевізники які можуть виконувати і пасажирські і вантажні перевезення, це пов'язано з наявністю в їх флоті повітряних суден для різних видів повітряних перевезень.

В плануванні графіків польотів повітряних суден важливу роль займає сезонність. Адже сезонність впливає на чисельність авіаперевезень в певний момент року. В Європі та Північній Америці до прикладу, піковими для авіаперевезень місяцями вважають липень та серпень, а для вантажних перевезень часова структура зміщується на грудень та січень.

Значним проривом стало в авіагалузі стало створення поняття вузлових аеропортів-хабів, що виконують роль домашніх аеропортів глобальних авіакомпаній, що функціонують як транзитні вузли. Завданням науковців з транспортної маршрутизації полягає в створенні оптимального графіку польотів повітряних суден між хабами-аеропортами.

Створення повітряних альянсів закривають вимоги своїх клієнтів на всіх типах ринків, навіть за умови що в певному місці компанія слабо представлена або не існує зовсім. Повітряні альянси наразі налічують понад п'ятдесят найбільших авіакомпаній.

Найголовнішими перевагами створення таких союзів є те, що компанії мають змогу проводити обмін кодами, оптимізувати і налагоджувати стикування рейсів, максимально оптимізувати графіки польотів транзитних рейсів, географічна спеціалізація та колективний маркетинг.

Оптимально розроблений графік польотів відіграє провідну роль при правильній взаємодії одинадцяти тисяч пасажирських повітряних суден, які щомиті перебувають в небі.

Створення графіку польотів повітряних суден можна розподілити на чотири основні фази цього процесу:

- 1-ий: створення розкладу польотів повітряних суден;
- 2-ий: визначення призначень бортів;
- 3-тій: маршрутизація бортів;
- 4-ий: розробка графіку для польоту відповідного бортового екіпажу.

Важливо, що жодна з цих фаз не може бути пропущена чи змінена, при цьому певні алгоритми радять паралельно обробляти два кінцевих етапи для оптимізування і прискорення роботи над графіком. В даній магістерській дипломній роботі наведений саме останній крок формування графіків польотів повітряних суден – моделювання як детермінована задача лінійного програмування. Основною ціллю якого є мінімізація затрат авіакомпанії на заробітні плати членів екіпажу літака.

Перші розробки та запропоновані методи були в більшості своїй обмежені потужностями тодішніх електронно-обчислювальних машин, але згодом цю перешкоду здолали і вже у 2004 році був розроблений метод Branch and Price. Метою роботи стало його вдосконалення і розвиток відповідно до масштабних потужностей нинішньої.

РОЗДІЛ 2 МАТЕМАТИЧНА МОДЕЛЬ

2.1 Постановка задачі

Завдання створення розкладу для бортового екіпажу моделюємо як задачу про розбиття. Маршрут позначаємо як певну циклічну сукупність суміжних рейсів, які команда екіпажу встигає пролетіти за один рейс.

Тоді задача може бути сформульована наступним чином [10]:

$$\sum_{j \in R} c_j b_j \rightarrow \min$$
$$\sum_{j \in R} a_{ij} b_j = 1, \forall i \in D^* \quad (1)$$
$$b_j \in [0,1],$$

де $r \in R$, а R – множина всіх можливих маршрутів для даного розкладу рейсів з визначеним певним екіпажем. D – множина всіх можливих рейсів, а $D^* \subset D$ – підмножина тільки обов'язкові рейси.

Кафедра КІТ(47)				НАУ 23 24 09 000 ПЗ			
Виконала	Маковська Д.Ю.			Математична модель	Літера	Аркуш	Аркушів
Керівник	Зіатдінов Ю.К.					40	12
Консулт.					УС-211М 122		
Н-контр.	Райчев І.Е.						

$$a_{ij} = \begin{cases} 1, \text{ якщо } i \in R; \\ 0, \text{ якщо } i \notin R. \end{cases}$$

$$b_j = \begin{cases} 1, \text{ якщо } r \text{ входить в роз'язок;} \\ 0, \text{ якщо } r \text{ не входить в роз'язок.} \end{cases} \quad (2)$$

$$c_j = \sum_{j \in R} c_i - \text{вартість маршруту } j$$

В даній дипломній роботі визначення рейсів також може бути більш розширеним. За єдиної умови: якщо екіпажам надати дозвіл на здійснення обов'язкових рейсів декілька разів.

Під обов'язковим рейсом тут мається на увазі такі, що приносять максимальну економічну вигоду для авіаційної компанії, тоді необов'язковими рейсами вважаються ті, здійснення яких не приносять суттєвого прибуткового доходу.

2.2 Застосування методу Branch and Price

Branch and Price – це модифікований метод гілок і меж. Для пошуку розв'язку задачі лінійного програмування з великою кількістю змінних застосовується метод генерації стовпців [23]. На Рисунку 2.1 можна побачити ключові кроки даного алгоритму.

Алгоритм Branch and Price суміщає в собі метод гілок і меж та метод генерації стовпців для матричної задачі [19]. Метод гілок і меж включає в себе три етапи: розбиття на гілки, вибір гілки і обчислення меж [20]. Даний алгоритм відзначається своєю доцільністю. На кожному етапі відбувається відсікання гілок, головною умовою якого є умова що обчислена межа наразі гірше за нинішнє ліпше рішення.

Метод генерації стовпців застосовується для вирішення завдань, в яких фігурують безліч змінних. В такому випадку доцільно алгоритм поділити на дві окремі складові. Першою складовою буде основна(глобальна) задача, в свою чергу підзадача(другорядна) буде другою складовою.

Завдання знаходження границь в двоїстій задачі еквівалентне меті знаходження іншої змінної для першої складової. В алгоритмі Branch and Price друга складова виконує пошук цього обмеження для забезпечення нормального здійснення більшого переліку границь в двоїстій задачі. Внесення ще одної нової змінної в першу складову задачі викликає до оновлення двоїстих змінних. Це призводить до змін цільової функції другої складової алгоритму Branch and Price.

Регуляція стовпчиків здійснюється до того моменту, доки усі поставлені границі двоїстої задачі не будуть виконуватися. В свою чергу це дає змогу зрозуміти, що найкраще значення релаксованої задачі виявлено [21]. За умови що відшукане рішення ціле, то в такому разі буде вважатись що початкове завдання вирішене, в протилежній ситуації необхідною умовою буде подальше розгалуження вже саме по дробовим змінним [22].

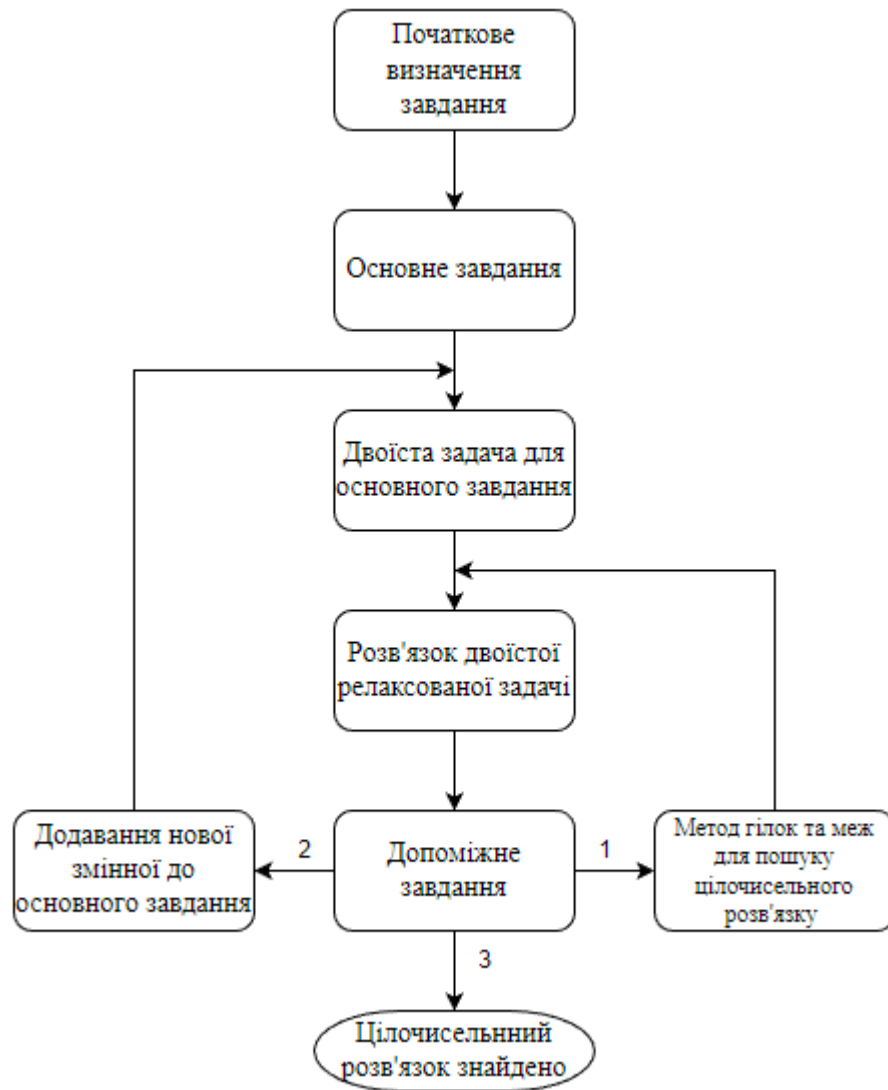


Рисунок 2.1 Схема основних етапів алгоритму Branch and Price

На етапі пошуку рішення допоміжної задачі алгоритм має три гілки продовження:

1. за умови, що всі обмеження двоїстої задачі виконані, але розв'язок основного завдання не є цілим числом;
2. за умови, що обмеження двоїстої задачі невиконане зовсім;
3. за умови, що обмеження двоїстої задачі виконані і розв'язок основного завдання являє собою ціле число. У цьому випадку початкове завдання вважається виконаним з отриманим розв'язком.

2.2.1 Застосування евристичного методу для побудови основного завдання

Евристичним називається алгоритм або емпіричне правило, яке ймовірно, але не обов'язково призведе до позитивного результату. Варто наголосити на тому, що даний алгоритм забезпечує тільки ймовірність позитивного результату, а не сто відсоткове точне вирішення задач.

Наприклад, в даному прикладі звичайною евристикою є „посилення переваги”. Це означає що є можливість надати перевагу якомусь певних груп варіантів за яким небудь аспектом. Скорочення числа варіантів призводить до того, що дерево рішень стає значно коротшим і ймовірно може призвести до підвищення умовної переваги. Цей алгоритм не гарантує позитивного вирішення завдання, але збільшує його вірогідність.

В даному підрозділі представлено описання евристичного алгоритму, метою якого є пошук певної підмножини змінних, яка в подальшому має ввійти до основного завдання. Реалізація даного евристичного методу потрібна для шукання комплекту можливих маршрутів для авіаційного екіпажу, який би включав всі необхідні рейси з мінімальною вартістю.

Основна дія алгоритму повернення підмножини змінних (набору допустимих маршрутів), для якого надалі слідує розв'язання основного завдання.

Одержимо вирішення за допомогою поетапного додавання курсу(маршруту) до загальної множини маршрутів. При цьому, в свою чергу кожен маршрут отримується послідовним додаванням рейсів.

Головною метою слугує забезпечення виконання всіх необхідних рейсів циклічними маршрутами (*цикл* – набір сусідніх рейсів, у яких перший та останній аеропорти співпадають). Наслідком цього є те, що кожен маршрут має обов'язково включати в себе якнайбільше польотів, адже в їх кількості зацікавлена компанія перевізник.

Перелік обов'язкових рейсів включає в себе авіарейси з кардинально різною вартістю. Головна суть цього евристичного алгоритму – пошук найдорожчої складової з певної підмножини таких складових і перекрити витрати на нього

першочергово.

На наступному етапі відбувається пошук циклічних маршрутів, щоб перший і останній аеропорти в одній зміні екіпажу мають бути ідентичними. Першочерговою задачею є забезпечення побудови рейсів замкнутого типу.

Ключовою властивістю алгоритму вважається першочергове перекриття всіх обов'язкових рейсів та спроби виконання алгоритму без залишку непокритих рейсів.

Одержана підмножина R дозволяє визначити основне завдання. Значення цільової функції релаксованого завдання та двоїстої до нього ідентичні в оптимальних точках. Релаксація задачі надає змогу спрощення пошуку рішення (обмеження $b_j \in [0,1]$ на $0 \leq b_j \leq 1$). Двоїсте завдання в свою чергу оформлюється так:

$$\begin{aligned} \sum_{i \in I} e_i &\rightarrow \max \\ \sum_{i \in I} a_{ij} e_i &\leq b_j, \forall j \in R \\ e_i &\geq 0 \end{aligned} \quad (3)$$

У двоїстої задачі вже експоненційна кількість границь. Для її розв'язку першочергово розбирається тільки підмножина границь. Допоміжне завдання алгоритму Branch and Price дозволяє відшукати границі для двоїстої задачі, що найчастіше порушується, і доповнити основне завдання новою змінною, яка відповідає цьому обмеженню.

2.2.2 Додаткове завдання

Основною метою даного завдання є пошук порушеного обмеження для двоїстої задачі. В подальшому в цільовій функції додаткової задачі будуть застосовуватись

одержані раніше значення двоїстих перемінних. Варто зазначити, що дана цільова функція утворюється шляхом визначення певних обмежень з пов'язаною з нею задачею.

$$\sum_{n \in j} (c_n - e_n^*) \quad (4)$$

Першим кроком присвоюємо параметр e^* розв'язком двоїстої задачі. При цьому c_n – буде вартістю n -го рейсу. За умови $n \notin D^*$, значення $e^* = 0$.

У випадку коли цільова функція додаткового завдання приймає які-небудь від'ємні значення, шукані обмеження рахується виявленими. Саме метод гілок і обмежень (Branch and Bound) застосовується для такого виду пошуку.

За допомогою застосування евристичного алгоритму відбувається пошук вихідного рішення, як результат відбувається значна оптимізація методу. Далі отримана межа вводиться як змінна в основне завдання, яке було сформоване першим. Наступний крок – розв'язок релаксованого завдання ще раз.

Важливо: в даній дипломній роботі залучаються показники рейсів тільки за три дні. Мається на увазі, що будь-який спланований рейс має тривати не більше сімдесяти двох годин. При цьому за розглянутий часовий проміжок кількість маршрутів між одними і тими ж аеропортами необмежена.

При моделюванні графіку польотів на прикладі орієнтованого графа, то аеропорти будуть вершинами, а маршрути будуть ребрами. В результаті одержимо мультиграф (обов'язкова ознака – ребра кратні)

Надалі граф-графік перетворюється в реберний граф для полегшення виконання методу гілок і меж. Одержимо орієнтований граф зі зваженими вершинами, але важливо що він вже не буде мультиграфом ні в якому випадку. Для визначення ваги ребер одержаного графа для можливості переходу назад до графу зі зваженими ребрами застосовують формулу:

$$\frac{w(a) + w(b)}{2}, \quad (5)$$

де a, b – вершини реберного графа, $w(a)$ – вага вершини a , а $w(b)$ – вага вершини b .

2.4 Пошук цілочисельного розв'язку основного завдання

Одержане необхідне цілочисельне рішення є розв'язком основного завдання. У тому випадку, коли обмеження виконано, але одержані перемінні в початковому завданні не є цілими числами, тоді у використаному алгоритмі Branch and Price необхідно застосувати метод Branch and Bound (гілок і меж) для знаходження цілочисельного оптимального рішення.

За допомогою основного завдання, а також підзавдання шукаються оптимальні нецілочисельні значення змінних для вихідної задачі. Поміж отриманих значень можуть існувати і цілочисельні, і навіть дробові значення. Розгалуження здійснюється тільки для дрібних змінних з фіксованим значенням цілих. Для кожної змінної реалізується розгалуження по двох гілках: $b_j = 0$ та $b_j = 1$. На рисунку 2.2. зображено схему процесу розгалуження цілочисельного розв'язку.

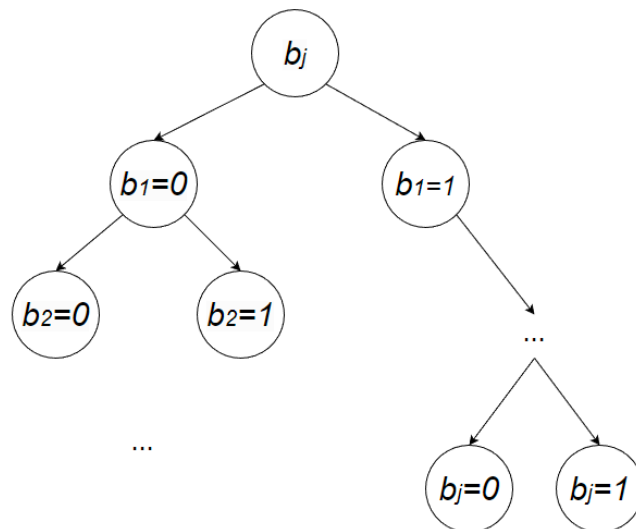


Рисунок 2.2. Процес розгалуження при пошуку цілочисельного розв'язку

Для пошуку цілочисельного розв'язку, а також знаходження найбільш порушеного обмеження існує нижня межа. Раніше на передуючому кроці методу значення цієї межі вже було визначено. Отже, отримане значення цільової функції початкового завдання з нецілочисельними змінними є саме тією нижньою межею, під час порівняння з якою відсікаються підмножини рішень, які очевидно не є оптимальними.

При реалізації методу гілкування можливе по двом різним гілкам, за умови коли $b_j = 0$ та за умови $b_j = 1$. Для вдалого застосування методу Гілок та меж потрібно обов'язково повторно пройти усі головні фази метода з певними зафіксованими змінними.

- $b_j = 0$:

за умови $b_j = 0$ для головного завдання вводиться обмеження $b_j \leq 0$. Обов'язковою умовою є те, що в допоміжному завданні ця гілка не використовується під час першого пошуку первинного вирішення завдання. Варто зауважити, що загальна вартість забороненого маршруту також не має бути на рівні поточної нижньої границі під час процесу подальшого розгалуження для найбільш порушеного обмеження.

- $b_j = 1$:

за умови $b_j = 1$ до головного завдання маємо обов'язково вести обмеження $b_j \geq 1$. В даному відгалуженні не ігнорується цей маршрут, а отже, як результат усі фази виконуються ідентично до того, як під час шуканні нецілочисельного розв'язку поставленої задачі.

Розгалуження відбувається до того моменту поки абсолютно всі змінні основного завдання не набудуть значення 1 або 0. Розв'язком основного завдання отримаємо набір маршрутів, який вміщує в себе обов'язкові рейси і здійснюється командами авіаційних ліній.

2.3 Адаптація алгоритму методу для створення системи

Попереджуючи виникнення проблем, які можуть створитися під час вирішення поставленого завдання обов'язково треба автоматизувати алгоритм. Це є обов'язковим та необхідним першим етапом у створенні комп'ютерної моделі алгоритму Brunch and Price.

Сортування всієї множини, яка надходить до входу, що першочергово унеможливорює створення єдиної математичної моделі – це найпоширеніша проблема, яка може утворитися при створенні завдання формування графіків польотів. Це значною мірою впливає на основні ресурси, які є обов'язковими при розробці. Під такими ресурсами мається на увазі загальна пам'ять приладу та час роботи.

Також методи такого типу вирізняються однією негативною властивістю – їхньою неспроможністю вирішення завдань третьої та четвертої фази створення графіків одночасно. Цими етапами є маршрутизація та складання розкладу. Як результат, це критично збільшує проміжок часу пошуку вирішення завдання і це все відбувається з розвитком багатопоточності в надсучасних технологіях що є недопустимим в даній задачі.

Рішення цієї проблеми можливо завдяки адаптації даного алгоритму з застосуванням сучасних мов програмування з урахуванням заощадження ресурсів.

На початковому етапі на вхід системи буде надходити набір усіх можливих існуючих польотів. В цей набір входять як головні рейси, які необхідно виконати в обов'язковому порядку, та другорядні Другорядні рейси це такі, які не несуть вагомого економічного прибутку вигоди для авіакомпанії.

На другій фазі відбувається запуск циклу на пошук рейсу, що здійснюється найпершим. Важливо: не критично якщо знайдений рейс буде необов'язковим. Саме ця умова є особливою ознакою шуканого адаптованого алгоритму. Здійснення цього етапу дає змогу не оминати рейси, що можливо б бути завершені, але завдяки певному часу вильоту були б початково відсіяні ще до першої фази.

Характерною особливістю розробки та адаптації алгоритму буде залучення рекурсії до його методу функціонування. Вірність виконання буде відбуватися за

основним критичним ключем. Таким ключем в даному алгоритмі будуть – час відправлення попереднього екіпажу, а також час прильоту наступного літака.

Основою умовою залучення обраного рейсу до отриманого циклу буде сума витрат на їх здійснення, тобто обчислення необхідних та обов'язкових витрат на кожній фазі.

Вагомою необхідною особливістю алгоритму є обов'язкове виключення повторень здійснених рейсів. Такі рейси мають бути обов'язково видалені якщо вони вже додані до будь-якого з циклів.

Вдалим результатом виконання програми має бути набір циклів, в кожному з яких обов'язково має міститись хоча б один обов'язковий рейс, а також окремий набір обов'язкових рейсів, які не вдалося залучити до жодного з отриманих циклів.

Висновки до Розділу 2

Актуальною проблемою сьогодення лишається питання маршрутизації повітряних суден. Маршрутизація має забезпечувати чітку систему перевезень пасажирів і матеріальних речей з будь-якої точки на карті світу в іншу найкращим шляхом, оптимізованим по витратам на час, а також економічно вигідним. В даному випадку, під шляхом мається на увазі замкнутий набір рейсів, які має змогу здійснити екіпаж за визначений період часу.

Варто було дослідити наявні алгоритми з аспекту економічної доцільності, яку вони пропонують. Адже це значною мірою впливає на їх можливість задовольнити в повній мірі потреби сучасних авіаційних компаній.

Аналіз всіх переваг і недоліків існуючих підходів в результаті дав такий наслідок: сучасні наявні методи не забезпечують оптимізовану маршрутизацію в повному обсязі. Тому виникла нагальна потреба осучаснення, перебудови, розвитку і оновлення технічного аспекту. Як наслідок, створення комп'ютерних моделей на базі алгоритмів дослідженого задання транспортної маршрутизації.

Ключову роль в дослідженні грає алгоритм Branch and Price. Це пояснюється

тим, що даний алгоритм є відносно новою розробкою в сфері авіаперевезень, як пасажирських, так і матеріальних.

В результаті досліджень, як наслідок отримали такий висновок: на початковій фазі головного завдання обов'язково вивчати питання мінімізування витрат, а не виключно питання збільшення зисків. Такий додатковий новий критерій сприяв створенню певного обмеження, що впливає на рівень оптимізації, що в свою чергу в майбутньому повпливає на удосконалення вивчення утвореного питання.

Створена удосконалена модель вирішення завдання планування пасажирських повітряних перевезень має вагому перевагу в адаптації до певних існуючих технічних ресурсів. Серед них вирізняються такі: унеможливлення перевантаження стеку, яке забезпечується оптимізованим повторювальним (рекурсивним) підходом, а також можливості застосування багатопоточності до тих фаз процесу, які являються функціонально-незалежними.

Даний алгоритм можна вважати універсальним, адже він оскільки спроможний бути застосованим в абсолютно протилежних галузях діяльності. До прикладу, має змогу бути застосованим на практиці суміжних видів перевезень, але окрім тих, маршрут яких не є обмеженим наявністю і протяжністю полотна.

На сьогоднішній день, технічній глобалізації притаманні вкрай швидкий ритм розвитку, тому можна зробити висновок що автоматизований і адаптований алгоритм буде в подальшому розвиватися і покращуватись відповідно до сучасного попиту пасажирів і перевізників.

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ ДЛЯ СТВОРЕННЯ ГРАФІКУ ПОЛЬОТІВ ПОВІТРЯНИХ СУДЕН

3.1 Вибір методології розробки програмного продукту

Перед початком розробки програмного продукту потрібно виділити час на безпосереднє проектування системи відповідно до останніх сучасних розробок та прогресивних методах життєвого циклу програмних продуктів. Даний етап є обов'язковим для зменшення загального витраченого часу на реалізацію методів та недопускання класичних помилок на початковому етапі.

Процес вибору певного підходу проектування програмного продукту із безлічі наявних залежить від різних чинників, серед яких такі: видом поставлених задач, відведених на розробку грошей, деяких індивідуальних вимог та ін.

В даному розділі досліджуємо чотири головних види методології для розробки програмних продуктів, оцінюємо їх переваги, а також недоліки.

Існують такі види методів розробки:

- водоспадна(каскадна) модель(Waterfall Model);
- V-модель;
- інкрементна модель (Incremental Model);
- Гнучка модель (Agile Model).

Водоспадна модель (Waterfall Model)

Першою класично розглядають каскадну, або водоспадну, модель циклу розробки програмного забезпечення. Процес створення зображений на Рисунку 3.1.

Кафедра КІТ(47)				НАУ 23 24 09 000 ПЗ			
Виконала	Маковська Д.Ю.			Огляд предметної області	Літера	Аркуш	Аркушів
Керівник	Зіатдінов Ю.К.					52	26
Консульт.					УС-211М 122		
Н-контр.	Райчев І.Е.						

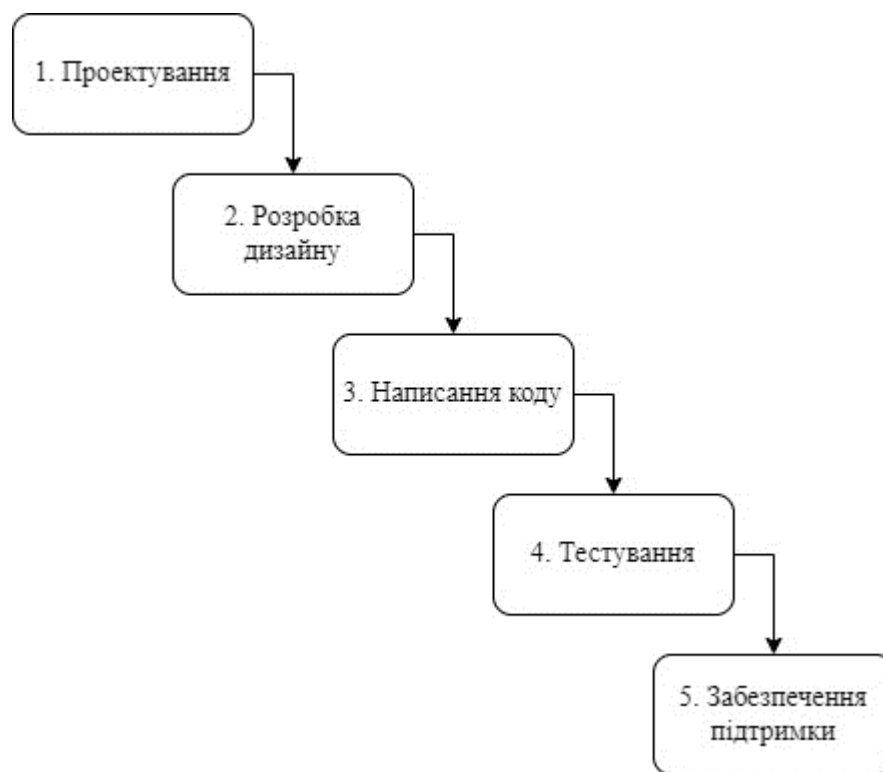


Рисунок 3.1 Водоспадна модель розробки програмного забезпечення

Такий тип моделі розробки програмного забезпечення є найстарішим. У даній моделі кожна попередня фаза розробки яка відповідає певному етапу життєвого циклу програмного забезпечення, продовжує передуючу. Це означає, що перехід на нову фазу розробки можливий тільки в тому випадку, коли попередній є повністю завершеним.

Каскадна модель розробки програмного забезпечення наразі вже не така практична, але при цьому вона лишається досить нескладною і доступною. Наразі, каскадна модель розробки програмного забезпечення залучається тільки глобальними ІТ-компаніями до масштабних і багатозадачних розробок, адже в більшості випадків дані проекти допускають і забезпечують абсолютне контролювання всіх існуючих небезпек.

Переваги та недоліки використання каскадної моделі:

- абсолютне документування всіх фаз розробки;
- зрозуміле нерозмите планування строків на розробку, а також затрат;
- чіткість процесів;
- обов'язкове складання ще на початковій фазі розробки всього загального

- обсягу вимог до системи;
- за потреби розширення списку вимог на інших фазах розробки буде відбуватись зворотній перехід на початковий етап, а отже, як наслідок виконання роботи заново починаючи з першої фази.

Даний вид моделі значно втратив свої лідерські позиції, але при цьому ще досі використовується. На сьогоднішній день програмістами використовуються значно сучасніші і більш гнучкі методи та моделі розробки ПЗ.

Доцільно застосовувати каскадну модель в таких проектах:

- з конкретними і чітко окресленими вимогами, при чому для них не очікується переміна даних сформульованих вимог вже в подальшому ході розробки;
- які мають намір переходу з одного виду платформи на інакшу. Як наслідок, вимоги лишаються такі самі ж, але змінюється тільки наприклад системне оточення та/або мова програмування;
- за умови, що розробник не буде проводити тестування, адже його проведенням буде займатись замовник або наприклад інша компанія.

Отже, як висновок, можна сказати що найчастіше каскадна модель застосовується коли технічне завдання проекту відоме і чітке, проект не має суперечливих задач, у порівняно немасштабних проектах, при умові доступності кваліфікованого персоналу.

V-модель

V-модель являє собою кращу версію звичайної каскадної моделі розробки програмного забезпечення. На всіх фазах розробки здійснюється ревізія поточного ходу розробки. Це потрібно задля того, щоб впевнитись в реальності переходу на подальший етап.

Особливістю даної моделі є те, що в ній етап тестування розпочинається аж на

початку – під час формулювання вимог. До того ж, для всіх подальших фаз розробки створений власний ступінь тестового покриття.

Для певного ступеня тестування створюється так званий певний тест-графік. Це означає що паралельно в той же час тестування поточної стадії, розробник створює план для тестування подальшого етапу.

В сучасній V-моделі всім фазам проектування і розробки системи відповідає певна стадія тестування. По ліву сторону так званої літери V відбувається певні етапи розробки, які являють собою низхідну послідовність, при цьому по праву сторону знаходяться етапи її тестування. Горизонтальними стрілками позначено відповідність стадіям розробки і тестування.

Переваги V-моделі розробки ПЗ:

- суворя поетапність;
- на початкових стадіях формуються графіки тестування, а також умови верифікації продукту;
- на фоні каскадної моделі значно поліпшений тайм-менеджмент;
- Міжрівневе тестування.

Недоліки V-моделі розробки програмного забезпечення:

- низький рівень дослідження можливих ризиків;
- розробка платформи відбувається на стадії створення коду(в середині процесу розробки);
- слабка гнучкість моделі;
- не застосовується практика паралельних подій, а отже, як результат не реалізовано метод динамічного внесення змін.

Доцільно застосовувати V-модель програмного забезпечення в таких проектах:

- яким необхідно значно більше тестове покриття, аніж в каскадній моделі

- розробки програмного забезпечення;
- з обмеженнями на затрати, а також на час.

Отже, ключовою відмінністю від каскадної моделі розробки програмного забезпечення від V-моделі є спрощене тестування, а також головна особливість – реальність введення змін до проекту вже на останніх стадіях розробки.

Схема розробки програмного забезпечення за V-моделлю зображена на Рисунку 3.2.

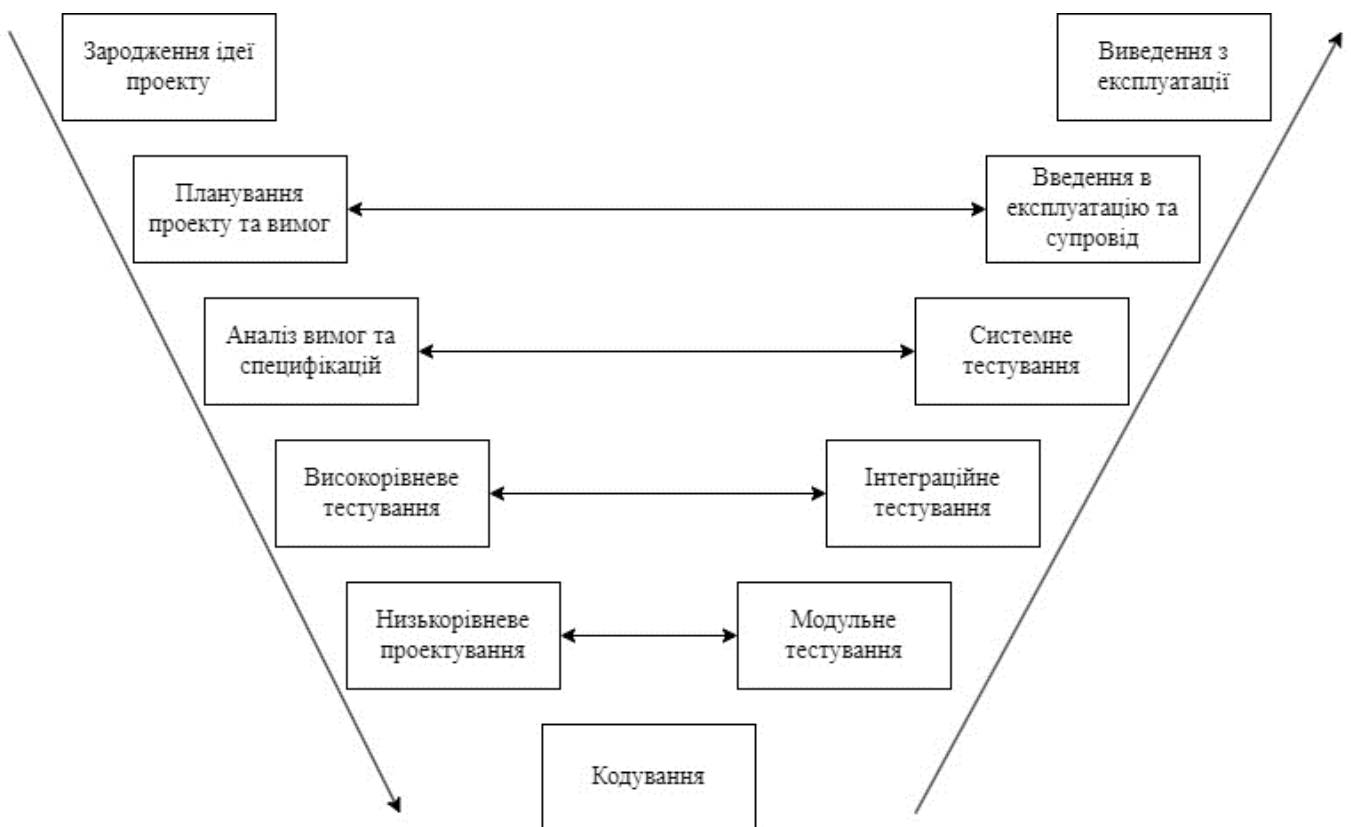


Рисунок 3.2 – V-модель розробки програмного забезпечення

Інкрементна модель (Incremental Model)

В основі цієї моделі розробки програмного забезпечення наявна лінійна послідовність фаз, при цьому в декілька версій, які і носять назву інкрменти. Завдяки

цьому відбувається удосконалення продукту, даний процес відбувається систематично протягом усього життєвого циклу розробки програмного забезпечення допоки він не закінчиться.

На першій стадії розробки формулюються вимоги до кінцевого програмного продукту, а вже впродовж наступних подальших фаз розробка відбувається так званими «послідовними версіями». Ці послідовні версії являють собою завершені продукти, які обов'язково мають правильно працювати.

Переваги інкрементної моделі розробки програмного забезпечення:

- наявна можливість оцінки абсолютно всіх версій кінцевих розроблених продуктів;
- наявна можливість оцінки ризиків, які можуть виникнути при розробці;
- адаптоване спокійне звикання до залучення новітніх технологій.

Недоліки інкрементної моделі розробки програмного забезпечення:

- обов'язково має бути означений кінцевий програмний продукт на першій фазі розробки(самий початок життєвого циклу), даний процес відбувається з метою розбиття на окремі ітерації;
- цілісність системи має ризик бути порушеною завдяки постійним або навіть періодичним перемінам в будові системи;
- строки виконання основного завдання системи мають властивість змінюватись, це може бути викликано незабезпеченістю ресурсів здачі системи можуть бути збільшені через обмеженість необхідних.

Процес роботи інкрементної моделі розробки програмного забезпечення зображений на рисунку 3.3.



Рисунок 3.3 Інкрементна модель розробки програмного забезпечення

Спіральна модель (Spiral model)

В даного виду моделі розробки програмного забезпечення вирізняється життєвий цикл продукту. Його можна візуалізувати як певну спіраль, при цьому її початком буде центр, який візуалізує фазу планування. З початкової точки в центрі спіраль далі прямує з поступовим проходженням кожної фази розробки. При проходженні повного витка від початку до кінця, одержуємо в кінці одну з кінцевих версій. Кожен такий пройдений цикл гарантуватиме отримання прототипу, який обов'язково буде протестований і входить в перелік усіх можливих версій та його доповнює. За умови що версія розробленої системи задовольняє всі початкові вимоги, вона може бути реалізована.

Переваги спіральної моделі розробки програмного забезпечення:

- в даній моделі чітко дотримується керування і контроль сформульованими ризиками розробки;
- наявна можливість створення адвентивних функції навіть на заключних фазах розробки;
- забезпеченість гнучкого проектування.

Недоліки спіральної моделі розробки програмного забезпечення:

- затратний аналіз ризиків на всіх стадіях проектування системи;
- постійна оцінка різних версій призводить до безлічі нових ітерацій, результатом чого стає значне збільшення проміжку часу до затвердження кінцевого варіанту розробленої системи;
- є менш доцільним в застосуванні до малих проектів.

Процес роботи спіральної моделі розробки програмного забезпечення зображений на рисунку 3.4.



Рисунок 3.4 Спіральна модель розробки програмного забезпечення

Гнучка модель (Agile Model)

Дана модель розробки програмного забезпечення базується на ітеративному методі розробки. Являє собою низку методів проектування, призначенням яких являється швидка реалізація якісного програмного забезпечення.

Переваги гнучкої методології розробки програмного забезпечення:

- пряма взаємодія з замовником і командою розробників – метод дає змогу взаємодії замовника та проекту на будь-якій стадії розробки, що в результаті сприяє високому рівню комунікації команди розробників і замовника, це в свою чергу допомагає розробникам чіткіше зрозуміти технічне завдання, а також зрозуміло окреслити бажаний кінцевий програмний продукт;
- прозорість – мається на увазі, що замовник має змогу залучатися до етапів проекту, впливати на пріоритетність поставлених завдань і вимог, складати списки черги ітерацій, а також додавання нових удосконалених версій програмного забезпечення;
- швидке очевидне отримання кінцевого продукту – забезпечується застосуванням сталого графіку циклів, залучення додаткових функції відбувається досить спритно і майже завжди має чіткий очевидний майбутній результат, це призводить до реальності здійснення бета-тестування програмного забезпечення на раніших фазах розробки;
- очікувані затрати і передбачуваний розклад – стабільність і фіксація інтервалу циклу сприяє формуванню очікуваних затрат і чітких меж поставлених завдань перед командою розробників, які мають бути обов'язково вирішені за спланованим графіком;
- придатність до редагування – в кожній новій ітерації методу є можливість поліпшувати список вимог, змінювати і розширювати основні завдання продукту що розробляється;

Недоліки гнучкої методології розробки програмного забезпечення:

- нечіткість вимог;
- складність оцінки завершеності проекту, адже на кожному кроці

- розробки програмного забезпечення можливе коригування основних завдань та зміна або доповнення вимог;
- зростання часу та об'єму роботи при внесенні змін на будь-якій стадії розробки.

Даний метод розробки програмного забезпечення доцільно використовувати за таких умов:

- розробка проектів з постійною динамічною зміною вимог до кінцевої системи;
- коли є певний бюджет, який виділено на проект і певну кількість ітерацій, але в майбутньому є перспектива покращення кінцевого продукту і збільшення кількості витрат на його поліпшення.

Основний принцип роботи гнучкого методу розробки програмного забезпечення зображений на Рис. 3.5.

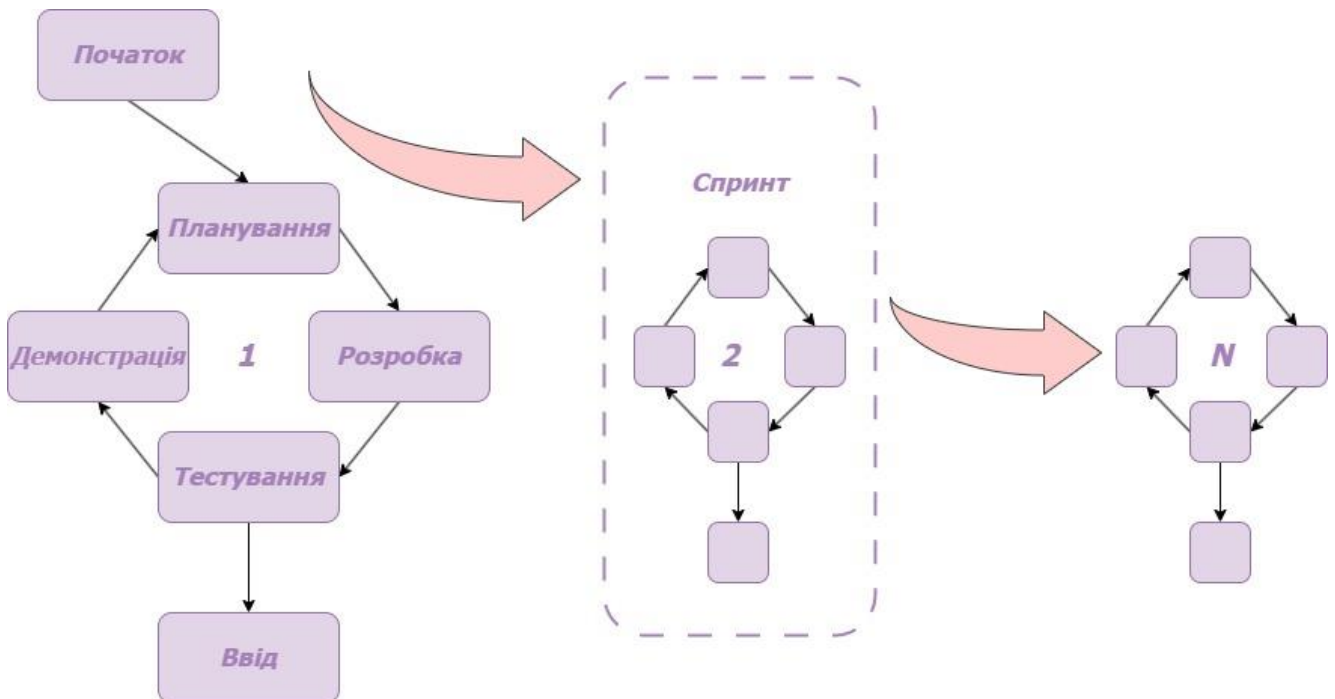


Рисунок 3.5 Гнучка модель розробки програмного забезпечення

Аналізуючи поставлене основне завдання, доступні до виконання проекту

ресурси, а також особливо наявність чітко зазначених вимог до кінцевого продукту, окреслена чіткість і не знаходження яких-небудь протиріч в початковому ТЗ продукту, що розробляється, а також передбачувано немасштабного програмного продукту було вирішено застосовувати водоспадну модель життєвого циклу розробки програмного забезпечення. Даний метод забезпечить цілковито повне документування всіх фаз розробки на початковій фазі проектування, надасть можливість чіткого планування строків на майбутню розробку, а також витрав на її реалізацію, а також забезпечить чіткість процесів усіх етапів проектування.

3.2 Проектування програмного продукту

На цій фазі розробки кінцевого програмного продукту відбувається безпосереднє вирішення завдань які виникають при початковій фазі проектування.

Першочергове завдання яке постає – підбір методу розробки програмного забезпечення та вибір алгоритмів для вирішення отриманої задачі. Дані алгоритми та методи були описані в попередніх розділах, чітко окреслено їх переваги та недоліки, принципи за якими вони працюють, а також обґрунтовано чому вибір пав саме на них.

Обраний підхід базується на алгоритмі Branch and Price, що являє собою метод розробки, пристосований до новітніх програмних розробок та технічних винаходів сьогодення.

Ключовою особливістю даного алгоритму слугує метод відбору першого рейсу. Він відбувається шляхом сортування всіх існуючих рейсів і знаходженням такого, що є найпершим в черзі на здійснення. В інших же методом початковим береться обов'язковий рейс.

Дане рішення було запропоноване для створення розширених циклів для залучення більшої можливої кількості рейсів. При цьому ці цикли можуть включати в себе не тільки обов'язкові рейси визначені авіаційною компанією.

Ще одним ключовим аспектом розробки системи є зниження витрат шляхом оптимізації використання, що в подальшому має призвести до зростання попиту на

перевезення такого виду.

Обов'язковим на цій фазі розробки є формулювання чітко окресленої збірки технологій, які будуть застосовані в даній розробці.

Першочерговим кроком в цьому проекті є визначення виду представлення даних. Після аналізу існуючих вимог і особливостей конкретної розробки системи, слідує висновок, що найдоцільніше буде використання цілочисельного типу представлення даних `int`. Це значно полегшує розробку, адже такий тип даних являє собою найпростіший.

Але, в подальшому для нових апгрейдів та можливості застосування системи в реальності, а не на прикладі, обов'язковим буде застосування особливого типу для часу та масштабних чисел.

Для реалізації обраного і наведеного вище алгоритму буде використана мова програмування JavaScript.

Саме ця мова програмування була обрана не випадково. Варто зазначити перелік переваг, які безпосередньо полегшують завдання створення продукту, серед яких такі:

- кросплатформність даної мови програмування – особливістю цієї мови є наявність певної системи, що має змогу здійснювати фізично і запускати код. Дана система являє собою віртуальна машину Java(Java Virtual Machine).
- поширеність даної мови – надає змогу легко вирішувати проблеми які виникають під час програмування, адже даною мовою володіють більшість розробників, які можуть з легкістю покрити великий об'єм проблем та питань які можуть з'явитись під час розробки;
- надійність – завдяки строгому типізуванню зменшується обсяг можливих проблем, адже абсолютно всі змінні обов'язково типізуються ще до початку компіляції. Особливістю компілятора є те, що він може самостійно визначити і надати розробнику інформацію про місце де була попущена неточність;
- об'єктно-орієнтованість;
- відносно проста мова програмування;
- достатня гнучкість – можливість розробки проекту різної складності від

- інтернет-магазину до штучного інтелекту.

Призначенням розробленого продукту має бути забезпечення потреб індустріального напрямлення. На сьогодні наявні два різних методи стек Spring framework та Java EE. Java EE в якійсь мірі виконує Spring. При цьому, варто зауважити, що саме Java EE найчастіше використовується в глобальних масштабних проектах з можливістю технічної підтримки в майбутньому цього проекту.

Spring Framework, або просто Spring – один із найпопулярніших фреймворків для створення веб-додатків на Java. [25]

Spring дозволяє створювати програми зі звичайних об'єктів Java (plain old Java objects, POJO) і застосовувати до них корпоративні сервіси.[26]

Аналізуючи ці два методи, зупинилися саме на методі було Spring framework, адже застосування цього методу дозволяє використовувати так звану Spring mvc (model view controller). Завданням якої є генерація сторінок в веб-сервісі на сервері, а також використання Spring dependency injection. Дана технологія дає можливість розробки застосувань, при якій незалежні класи поєднуються в єдине ціле для їх використання. При цьому їх робота поодиноці абсолютно визначена, на відміну роботи всією групою.

Головною метою розробки системи є виконання алгоритму. Тому не є доцільним використовувати глобальні бази даних, адже достатньо буде полегшеного окремого сховища, який в подальшому буде доступний до редагування, або зберігання інформації про польоти та існуючі вже виконані цикли.

3.3 Створення діаграми варіантів використання

Діаграма варіантів використання(так звана Use Case) надає можливість представлення ролей та їх взаємодії як з системою, так і між собою. Але при цьому

не дає уявлення про послідовність етапів виконання. Дана діаграма слугує для схематичного зображення функціональних вимог. Тобто зображення того, що дана система спроможна робити зі сторони користувача. Основні елементи даної діаграми наведені на Рисунку 3.6.

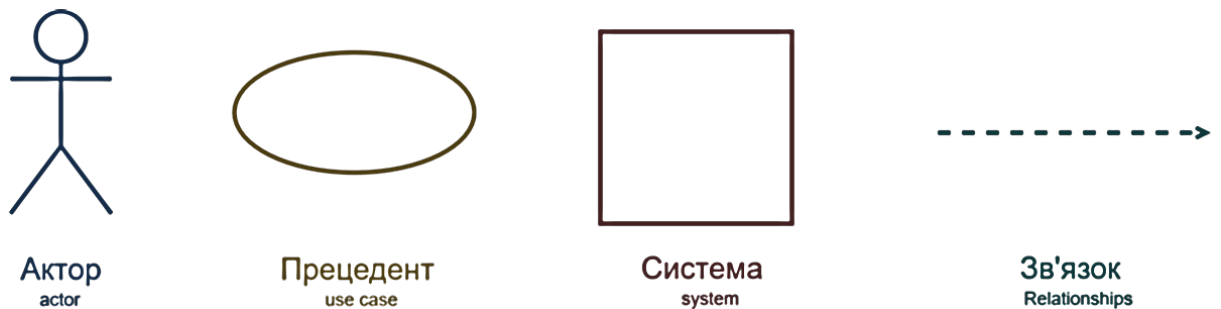


Рис.3.6 Елементи діаграми використання

Цей тип діаграм формується на початкових етапах розробки програмного продукту, призначенням якої є спрощене представлення системи, чітке формулювання технічної задачі, визначення функціональних вимог.

Визначення актора системи

Актором системи вважається певний об'єкт, який безпосередньо взаємодіє з системою, яка розробляється, але при цьому перебуває не всередині неї, а знаходиться за її межами. На діаграмі позначається схематичним чоловічком.

Актором в нашій системі буде розробник графіків польотів, який безпосередньо буде взаємодіяти з системою розробки розкладу.

Визначення випадків використання (прецедентів)

Випадків використання (прецеденти) позначають поведінку системи, тобто що саме має можливість виконувати дана система. Описують перелік можливої діяльності програми, певних призначень або задач.

На схемі позначаються у формі еліпса з підписом дії(дієсловом).

В нашій системі існують такі прецеденти:

- Завантажити CSV файлу з існуючими рейсами. Можливість вставлення

- файлу з даними існуючих рейсів, з яких нам треба створити певний цикли з обов'язковими і необов'язковими рейсами.
- Ввести номер рейсу. Можливість зазначення номеру рейсу з переліку всіх існуючих рейсів.
- Ввести місце вильоту. Можливість зазначення місця початку рейсу, тобто місця вильоту літака.
- Ввести місце призначення. Можливість зазначення місця, куди має прилетіти літак, тобто місце призначення рейсу.
- Ввести значення рейсу. Можливість ввести значення true або false що відповідає за обов'язковість рейсу, де значення true відповідає обов'язковим рейсам, а значення false – необов'язковим рейсам.
- Додати рейс. Можливість створення ще одного рейсу.
- Змінити рейс. Можливість редагування елементів рейсу(місця вильоту, місця призначення, обов'язковості або номеру рейсу).
- Видалити рейс. Можливість видалення обраного рейсу із загального списку існуючих.

Визначення системи

Системою виступає, те що проектується. Розробкою можуть виступати різноманітні сайти, програмні застосунки, модулі ПЗ.

Позначається на діаграмі прямокутником, всередині якого знаходяться прецеденти. Всередині вгорі підписується назва даної системи.

В даному проектуванні такою системою є система розробки графіків польотів.

Визначивши всі найголовніші елементи діаграми варіантів використання та сформулювавши і означивши їх на прикладі системи, що проектується в даній роботі складемо діаграму варіантів використання Системи розробки графіків польотів. Кінцевий варіант діаграми варіантів використання наведений на рис. 3.7.

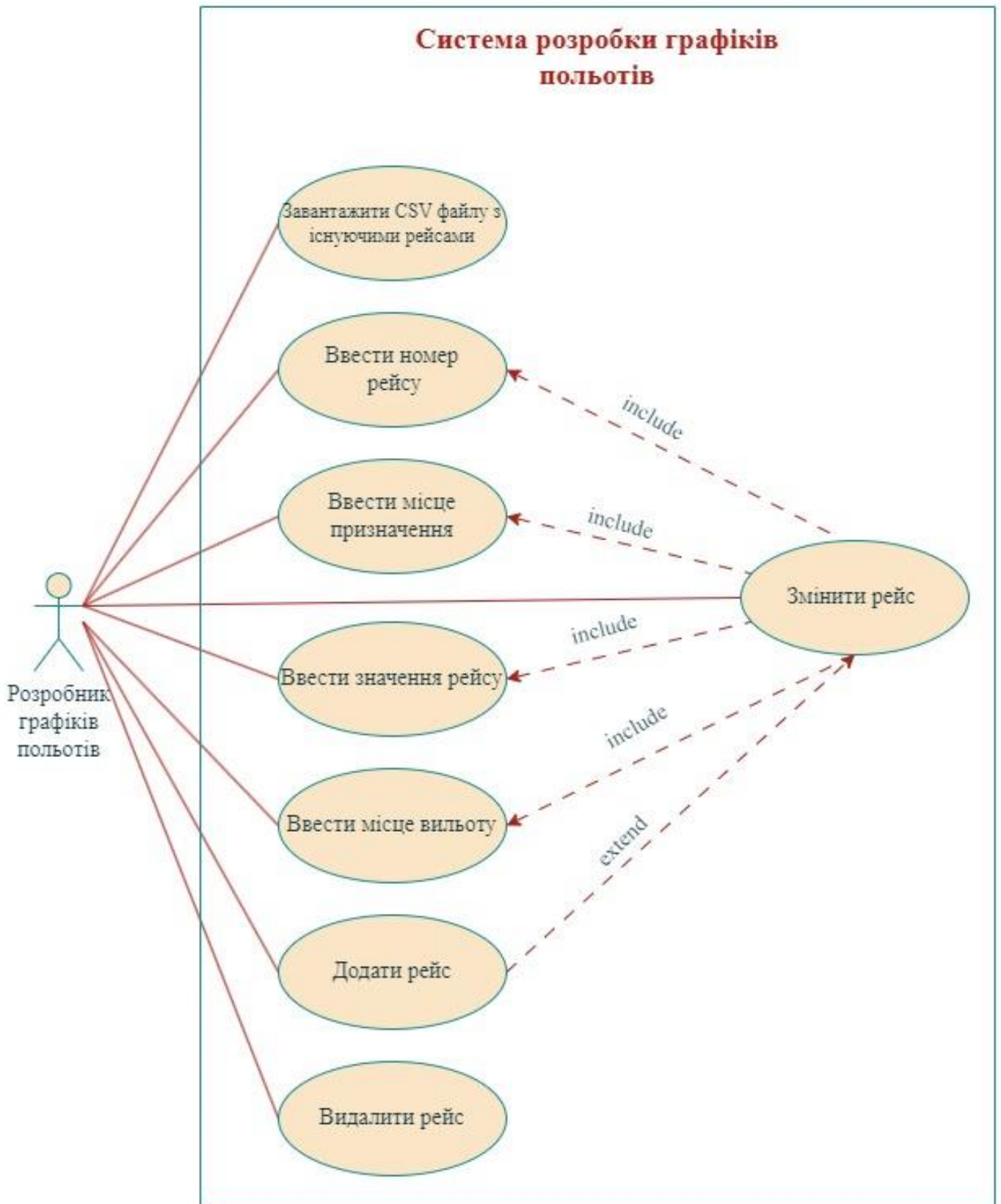


Рисунок 3.7 Діаграма варіантів використання Системи розробки графіків польотів повітряних суден

3.4 Створення інтерфейсу програми

За допомогою будь-якого браузера відкриваємо додаток (Рис.3.8).



Рисунок 3.8 Вихідне вікно розробленої системи

Розшифровка кожної складової продукту наведена далі в Таблиці 3.1

Розшифрування основних елементів

<i>Назва елемента</i>	<i>Призначення та властивості елемента</i>
<i>Calculate</i>	Запуск алгоритму та отримання відповіді
<i>Choose file</i>	Вибір файлу (файл .csv з розділювачем "кома")
<i>Upload CSV file</i>	Завантаження обраного файлу
<i>Number</i>	Номер рейсу (не може повторюватися)
<i>From</i>	Місце вильоту (не може співпадати з пунктом призначення)
<i>To</i>	Пункт призначення
<i>Income</i>	Прибуток авіакомпанії (максимізується в задачі)
<i>Expanses</i>	Витрати компанії (мінімізуються)
<i>Departure time</i>	Час відправки (не може співпадати з часом прибуття)
<i>Arrival time</i>	Час прибуття
<i>Mandatory</i>	Значення рейсу (обов'язковий чи ні)
<i>Add or update flight</i>	Додати чи оновити рейс
<i>Remove</i>	Видалити рейс (з'являється після додавання хоча б одного рейсу)

Абсолютно усім показникам, не враховуючи елемент Mandatory(обов'язковість), маємо задати тип даних `int` для полегшення реалізації розробки. Елемент Mandatory в заданому варіанті має задаватись шляхом обирання значень `true` (так) або `false` (ні) з створеного випадючого списку.

В системі реалізована валідація зазначених даних. На Рис. 3.9 можна побачити процес валідації.



Рисунок 3.9 Процес валідації

Створено тест різних місць відправлення і прильоту для формулювання вимог на час відправки для певного рейсу (він обов'язково має бути меншим за час прильоту).

Реалізовано можливість оновлення даних щодо рейсу, шляхом залучення його номеру.

Поетапно цей процес матиме такий вигляд:

- *перший етап: введення певного рейсу* (Рисунок 3.10);



Рисунок 3.10 Перший етап.

- *другий етап: за потреби можливість внесення змін в рейс;*

Процес введення відбувається таким чином: записуємо змінені дані, при цьому залишаючи номер рейсу, який підлягає зміні. Даний процес можна побачити на Рисунку 3.11.



Рисунок 3.11 Другий етап

- третій етап: одержання відредагованого рейсу, після внесення змін.

Даний процес зображений на Рис. 3.12.

Source	Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
0	1	0	0	100	20	100	200	false
1	2	4						true

Рисунок 3.12 Третій етап

В поле Choose file додаємо обраний файл і за допомогою елемента Upload CSV file завантажуюємо його. За допомогою цього процесу наглядно перевіримо зразок роботи з даними, які завантажуються з іншого джерела(в даному випадку CSV-файлу). На Рисунку 3.13 зображений цей етап.

Source	Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
1	1	2	120	24	123	321	true	
2	3	4	120	24	134	431	false	
3	5	2	120	24	145	541	true	
4	1	4	120	24	156	651	false	
5	3	2	110	22	178	871	true	
6	5	4	110	22	190	981	false	
12	5	4	100	20	222	477	false	
13	3	2	100	20	267	435	true	
16	5	3	125	25	287	500	true	
17	4	3	140	28	268	500	true	
19	4	1	140	28	444	900	true	
24	2	1	135	27	575	700	false	
14	3	4	130	26	708	1030	false	
20	2	3	140	28	769	1600	false	
33	5	3	135	27	839	1000	true	
32	1	2	125	25	857	1011	false	
18	2	5	140	28	888	1000	false	
21	4	5	115	23	906	1400	true	
13	1	2	130	26	908	1300	true	
25	3	5	135	27	930	1022	true	
29	5	3	125	25	1000	1111	true	
30	1	2	125	25	1001	1200	false	
15	5	2	130	26	1100	1300	true	
23	4	1	105	21	1111	1300	true	
7	1	2	110	22	1234	3210	true	
8	3	4	110	22	1245	5421	false	
22	2	1	115	23	1300	1000	false	
9	5	2	100	20	1345	5632	true	
23	4	3	115	23	1355	1600	true	
26	2	3	105	21	2199	2500	false	
10	1	4	100	20	2344	2445	false	
24	2	5	115	23	2399	2600	false	
28	1	2	105	21	2414	2700	false	
16	3	1	130	26	3200	3900	false	
27	4	8	105	21	3456	4800	true	

Рисунок 3.13 Приклад третього етапу

Результатом даного третього етапу будуть певні набори циклів, головною умовою яких є наявність хоча б одного з обов'язкових рейсів, з розрахунком часом.

Результат даного етапу зображений на Рисунку 3.14.

Detected cycles

from: 4, departure time: 268

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
17	4	3	140	28	268	500	true
14	3	4	130	26	708	1020	false

from: 4, departure time: 444

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
19	4	1	140	28	444	900	true
13	1	2	130	26	908	1300	true
22	2	1	115	23	1300	1600	false
10	1	4	100	20	2344	2445	false

from: 5, departure time: 267

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
31	5	3	125	25	267	500	true
35	3	5	135	27	930	1022	true

from: 2, departure time: 888

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
18	2	5	140	28	888	1000	false
15	5	2	130	26	1100	1300	true

from: 1, departure time: 123

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
1	1	2	120	24	123	321	true
34	2	1	135	27	575	700	false

from: 5, departure time: 145

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
3	5	2	120	24	145	541	true
24	2	5	115	23	2399	2600	false

from: 5, departure time: 222

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
12	5	4	100	20	222	477	false
21	4	5	115	23	906	1400	true

from: 3, departure time: 134

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
2	3	4	120	24	134	431	false
23	4	3	115	23	1355	1600	true

from: 3, departure time: 178

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
5	3	2	110	22	178	871	true
26	2	3	105	21	2199	2500	false

from: 1, departure time: 156

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
4	1	4	120	24	156	651	false
25	4	1	105	21	1111	1300	true

from: 5, departure time: 190

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
6	5	4	110	22	190	981	false
27	4	5	105	21	3456	4500	true

from: 3, departure time: 267

Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
11	3	2	100	20	267	435	true
20	2	3	140	28	769	1600	false

Рисунок 3.14 Отримані цикли після проведення третього етапу

Неодмінно внизу отримаємо ще додатковий список обов'язкових рейсів, які, на жаль, не були задіяні в жодному з вище зазначених циклах.

Приклад такого списку представлений на Рисунку 3.15.

Detected mandatory flights without cycles							
Number	From	To	Income	Expenses	Departure time	Arrival time	Mandatory
33	5	3	135	27	839	1000	true
13	1	2	130	26	908	1300	true
31	5	3	125	25	267	500	true
29	5	3	125	25	1000	1111	true
7	1	2	110	22	1234	3210	true
9	5	2	100	20	1345	5632	true

Рисунок 3.15 Список незадіяних в циклах обов'язкових рейсів

Висновки до Розділу 3

Безпосереднє проектування вважається вагомим та необхідною фазою створення програмного забезпечення, при чому основну роль грають рівень актуальності методів і алгоритмів розробки та масштаби їх застосування в світових межах розробки програмних продуктів.

Дуже важливим є процес вибору підходу, адже треба порівняти всі переваги і недоліки, вимоги до кінцевого продукту, а також затрати на його розробку і використання та забезпечення технічної підтримки в майбутньому.

Проаналізовано основні моделі розробки програмного забезпечення, зокрема водоспадну, V-модель, інкрементну, спіральну та гнучку.

Першою розглянутою моделлю була водоспадна. Серед її особливостей варто виділити такі: в даній моделі відбувається абсолютне документування, чіткість строків і витрат на розробку, конкретний опис процесів, формування всіх вимог тільки на початковому етапі, в іншому разі відбувається перехід на початкову фазу і початок планування наново.

Другою розглянутою моделлю була V-модель розробки ПЗ. Вона вирізняється такими властивостями: чітка поетапність, вже на початку проектування за допомогою такої моделі на створюється розклад тестування, покращений тайм-менеджмент, можливість реалізації міжрівневого тестування, при цьому посередня обробка можливих ризиків, не має можливості застосування динамічного внесення правок, головна особливість – реальність введення змін до проекту вже на останніх стадіях розробки та спрощене тестування.

Третьою розглянутою моделлю розробки програмного забезпечення була інкрементна. Їй притаманні такі особливості: наявність реалізації оцінювання всіх можливих версій кінцевих продуктів та оцінки ризиків при проектуванні, адаптоване залучення нових технологій, а також обов'язково має бути означений кінцевий програмний продукт на першій стадії проектування, при цьому не забезпечується цілісність системи завдяки постійним змінам в системі, немає чіткого уявлення терміну виконання. Отже, даний тип застосовується до задач з строго визначеними запитами, які в майбутньому можуть передбачати зміни в основному поставленому завданні.

Четвертою розглянутою моделлю була спіральна. Відзначимо такі її властивості: строге виконання керування і контролю можливих існуючих ризиків, розробка адвентивних функції на кінцевих стадіях проектування, гнучкість проектування, при цьому дороговартісний огляд ризиків, постійна велика кількість ітерацій при внесенні змін.

Останньою розглянутою була гнучка модель, яка вирізняється такими особливостями: можливість взаємодіяти з замовником і командою розробників напряму, прозорість проектування, отримання кінцевого продукту за чітким термін виконання, очевидні конкретні затрати і конкретність кінцевого продукту, при цьому розмите формулювання вимог, розширення часу та об'єму роботи при внесенні змін на будь-якій стадії розробки.

Провівши чіткий аналіз найпопулярніших моделей, а також зібравши певний конкретний список вимог до кінцевого продукту, зважаючи на чіткість описаних вимог було обрано використання водоспадної моделі розробки програмного

забезпечення. Даному вибору також посприяло ще декілька факторів, а саме:

- не знаходження протиріччя в основній задачі проектування;
- невеликий масштаб реалізуємого проекту.

Серед безлічі сучасних існуючих алгоритмів вибір пав саме на алгоритм Branch and Price. Це пояснюється значною адаптовністю його до новітніх технічних можливостей. Застосовано специфічний метод – вибір найпершого за часом можливого рейсу. А не так, як відбувалося в інших методах, коли першим рейсом обирався найперших з переліку обов’язкових. В результаті, такий підхід дав змогу значно розширити цикли.

Саме мова програмування Java була застосована в створенні програмного продукту. Реалізація проекту відбулася за допомогою певного фреймворку застосунків Spring framework. Сховищем даних слугує файл CSV. Це значно спрощує реалізацію проекту. Таке спрощення пояснюється пріоритетністю створення саме алгоритму даної розробки.

Особливістю кінцевого розробленого продукту є його висока універсальність. Значною перевагою даного продукту є можливість в майбутньому імплементувати адвентивні можливості, необхідність яких може виникнути в майбутньому під час активного використання.

Зосередженню саме на авіаційних перевезеннях сприяв значний попит на такий тип перевезень в зв’язку з розвитком міжнародного туризму та глобальною міжнародною торгівлею.

ВИСНОВКИ

Аналізуючи всі галузі нашого життя, варто зазначити що особливе місце займає саме авіаційна галузь. Реалії сьогодення роблять авіацію невід'ємною частиною нашого життя. Без авіаційної галузі зараз складно уявити масштабні складові нашого життя, однією з яких є світова торгівля. Авіакомпанії наразі забезпечують сполучення майже будь яких двох точок на карті світу, при цьому забезпечуючи відносно швидкий, безпечний та комфортний спосіб перевезень.

Значний розквіт галузі забезпечений технічними відкриттями, що мають значний внесок в глобальний розвиток нинішньої індустрії. З дня першого польоту дана галузь розширила своє використання аж в сімдесят дев'ять разів. Важливим є те, що ні єдиний вид транспорту не може похвалитись таким успіхом. Це призвело до того, що повітряний транспорт наразі має неабиякий вплив на світову економіку загалом та її розвиток зараз і в майбутньому.

Авіаційний транспорт має найбільшу перевагу серед будь яких інших видів транспорту – можливість досить шпитного переміщення як пасажирів, так і товарів з цінником в десятки мільярдів доларів до точок по всій карті світу.

Ще одним невід'ємним аспектом даної галузі є вплив на роботу та відпочинок людства. Важлива роль в таких невід'ємних складових життя людей допомагає в постійному оновленні, розвитку галузі та технічному прогресі загалом.

Надання нових робочих місць та можливість працевлаштування в даній сфері забезпечує пониження рівня бідності, за рахунок зростання економіки.

Важливим направленням в даній галузі стало інвестування коштів в розвиток світової торгівлі, що сприяє глобалізації світової економіки, а також інвестування в новітні розробки, що кожної секунди забезпечують зростання безпеки і відносної швидкості доставки.

Наразі авіаційна галузь ти чи іншим чином має вагомий вплив на більшість сфер діяльності людства. Сучасний світ, на сьогодні, нереально уявити без існування галузі повітряних перевезень. Той рівень залучення авіації в повсякденне життя людства наразі незамінний жодним іншим видом існуючого транспорту.

На сьогодні, сфера авіації вже має масштабну сторінку в загальній історії розвитку всіх видів транспорту. Від розробки і першого застосування реактивних двигунів, які на той час стали глобальним проривом в галузі повітряного транспорту, нині минуло вже більше сотні років.

Вагомий вплив на розвиток технічного прогресу і навчання висококваліфікованого персоналу мали Перша та Друга світові війни, адже авіація дала змогу здійснювати найскладніші бойові стратегії і тактики, а перевага в повітрі майже завжди означала перемогу сторони конфлікту в якій вона була.

Ще одну важливу роль відіграло забезпечення здійснення міжконтинентальних сполучень, що вплинуло на створення міжнародної торгівлі та світового туризму.

Варто зауважити, що в порівнянні з іншими видами транспорту саме авіасполучення є безумовними лідерами за безліччю критеріїв оптимальності, зручності, безпеки і швидкості перевезень. Це і пояснює чому саме авіаперевезення найчастіше обирають пасажери.

Дана сфера постійно продовжує свій розвиток і ніколи не стоїть на місці. З кожним днем рівень забезпечуваної безпеки зростає задля попередження терористичним діям, а також боротьбі з небезпечними загрозами. Галузь невпинно намагається забезпечити максимальну безпеку своїх пасажирів.

Ще одним аспектом, в який авіаіндустрія розвиває і інвестує багато своїх ресурсів ще екологічність свого діяння. Значно поширюється важливість зменшення рівня забруднення від викидів діяльності авіаційного транспорту.

Успіх застосування на практиці повітряних перевезень пояснюється насамперед дозволом на пересікання повітряного простору інших держав. Найголовніші вимоги які визначають нормальну діяльність повітряного транспорту, а також нетоксичну взаємодію між транзитними країнами(Свободи повітряного транспорту)сформовані в Чиказькій конвенції ще в 1944 році і функціонують по сьогоднішній день.

Цікавим кроком в розвитку авіаційної сфери стало створення вантажних компаній, що забезпечують доставку вантажів саме авіаційним транспортом. Але в подальшому стали з'являтися компанії які б поєднували б і пасажирські і вантажні

перевезення, адже на сьогодні, мультизадачність грає вагому роль в будь якій галузі діяльності де б вона не була застосована. Тому такі авіакомпанії значною мірою займають панівні перші сходинки в рейтингу існуючих компаній.

Спостерігається чітка сезонність серед повітряних перевезень. Даний аспект має вагоме значення в створенні графіків польотів повітряних суден. Найактивнішим періодом в Європі та Північній Америці є липень та серпень, при цьому для Японії наприклад найбільш туристичним сезоном є період цвітіння сакури, а ось пік вантажних перевезень фіксується в зимовий період(саме перші два місяці), більшою мірою це пов'язано з новорічними святами і необхідністю забезпечення великої кількості товарів, які в подальшому стають подарунками .

Забезпечення логічності і значної оптимізації маршрутів досягається за допомогою локальних аеропортів-хабів. Реалізація вдалої транспортної маршрутизації є основною задачею в створенні певного розкладу рейсів, які пов'язують ці хаби.

Створення повітряних альянсів дало змогу розширити межі покриття і задоволення абсолютно всіх потреб що можуть виникнути навіть в тих місцях, де вони не домінують.

Щохвилини в повітряному просторі перебувають одинадцять тисяч пасажирських літаків, при цьому глобальне значення грає чітко , коректно і безпомилково планування графік польотів повітряних суден. Розробка такого роду розкладу відбувається в чотири основні фази:

- розробка графіку польотів;
- визначення призначення борту;
- забезпечення вдалої маршрутизації;
- розробка графіку певної бортової команди.

В нашій розробці проектування відбувається саме кінцевої фази розробки розкладу. Основне завдання якої подається у вигляді чітко визначеного завдання лінійного програмування. Основною задачею цього завдання є мінізування затрат на виконання рейсів мережею авіакомпаній. А результатом належного виконання даної

задачі має стати найдовший цикл, що складається як з обов'язкових, так і необов'язкових рейсів, що бере свій початок з найпершого за часом відправки рейсу, який може бути навіть необов'язковим.

Вагомою дилемою в розвитку авіації стало питання маршрутизації повітряного транспорту. Загалом під терміном маршрутизація мається на увазі низка із певних окремих підборів властивостей визначеного курсу, по якому відбувається курсування повітряних суден, головною умовою яких є вдале застосування логістичних норм і вимог до перевезень пасажирів, а також вантажів.

Мета предмету процесу маршрутизації досягає свого піку актуалізації за наявності безлічі варіантів розподілення повітряних суден по існуючим повітряним потокам. Вивчення даного предмету та його безпосередній аналіз дає змогу забезпечити створення таких циклів рейсу, які б в свою чергу змогли б перекрити всі вимоги, забезпечити максимальний комфорт та максимально мінімізувати затрачений на рейс час.

Вирішенням питання маршрутизації слугувало створення певної окремої системи з чітко окресленими кордонами, які носять назву торгівельні шляхи або транспортні повітряні шляхи.

Тільки за останні пів століття було підняте питання маршрутизації та початку вирішення вже існуючих проблем її застосуванням на практиці. Найперші спроби розробки алгоритмів та нових методів найчастіше були в значній мірі недостатніми, адже це пояснювалось низьким рівнем індустріалізації в світі, а також слабким технічним оснащенням існуючих на той момент електронно-обчислювальних машин.

З плином часу ці проблеми вирішувались з поширенням глобального технічного розвитку. І вже на початку 2000-них років відбувся реліз нового методу під назвою Branch and Price.

Розробка даного методу сприяла подальшому активному пошуку нових рішень в сфері існуючих невирішених задач і проблем. З часом відбувалося покращання методу, що пояснюється розвитком новітніх технологій і техніки загалом.

На сьогоднішній день, актуальність питання забезпечення вдалої маршрутизації повітряних потоків не втрачає своїх позицій. В результаті чого

продовжується активний синтез розроблених методів і алгоритмів з метою їх вдосконалення, яке може забезпечуватись шляхом оптимізації економічного впливу і мінімізації витрат на їх запровадження.

Ще одним вагомим аспектом є здатність алгоритму покрити всі нинішні вимоги авіаційних компаній, які наразі є необхідними для здійснення рейсів. Створення такого типу досліджень розкрило потребу технічного завдання методу, в подальшому це призвело до розробки певних комп'ютерних макетів з використанням методів вже вивченого завдання маршрутизації повітряних суден.

Алгоритм Branch and Price має значний вплив на маршрутизацію повітряних потоків, адже досі являє собою майже найновішу розробку в сфері новітніх проектувань в галузі авіації.

В наслідок проведених досліджень отримано такий результат: визначено що під час фази формулювання основного завдання варто враховувати не тільки аспект збільшення доходів, а ще й пошук умов, за яких досягається максимальна мінімізація затрат на реалізацію рейсів. Залучення даного критерію до списку впливових чинників пояснюється спробою введенням ще одного об'єкту для дослідження, що в свою чергу матиме вплив на подальше вдосконалення цього методу.

Довершена модель вирішення основного завдання планування авіаційних перевезень характеризується прерогативою в галузі широкої адаптивності до існуючих технічних ресурсів, які забезпечуються на самому початку проектування.

Описаний вище алгоритм Branch and Price можна вважати універсальним, адже він може використовуватися в абсолютно інших існуючих на сьогодні галузях життя.

Розвиток технічної галузі відбувається з нечуваною швидкістю. Це призводить до висновку, що даний алгоритм матиме в майбутньому шанс на удосконаленні і залучення його в нові процеси життєдіяльності та забезпеченні перекриття всіх нових потреб сучасних користувачів.

Під час створення програмного продукту вагому роль грає саме фаза проектування системи. Адже вибір підходу впливатиме на актуальність розробки та загалом її доцільності. Варто звертати увагу на рівень належності останнім

розробкам, тобто доцільність використання застосованих методів і алгоритмів.

Процес обрання конкретного методу спирається на певний перелік чинників, а саме серед них такі: специфічність основного завдання проектування, бюджету на розробку, особливих специфічних вимог замовників та безліч інших.

Розповсюдженими серед моделей розробки програмного забезпечення є такі види моделей: каскадна, V-модель, інкрементна, спіральна та гнучка.

Головною перевагою каскадної моделі розробки програмного забезпечення є її строгість документування.

А ось V-модель розробки програмного забезпечення відзначається своєю чіткою поетапністю та можливістю реалізації міжрівневого тестування.

При цьому інкрементна модель розробки програмного забезпечення застосовується до поставлених основних з чітко визначеними вимогами, які в майбутньому можуть передбачати зміни в основному поставленому завданні.

В спіральній моделі розробки програмного забезпечення варто відзначити строге забезпечення керування і контролю можливих існуючих ризиків в системі.

Гнучка модель розробки програмного забезпечення вирізняється можливістю взаємодії замовника і командою розробників.

Базуючись на присутності в нашому поставленому основному завданні строго сформованих вимог до системи, а також не знаходженні протиріч в основній задачі, було вирішено надати перевагу водоспадній моделі розробки програмного забезпечення.

За осново взято метод Branch and Price. Реалізований метод першим рейсом обирає найперший по часу відправлення рейс не звертаючи увагу на його обов'язковість. Це дає змогу значно розширити можливі цикли шуканих рейсів.

Проектування і розробка системи відбувається за допомогою об'єктно-орієнтованої строго типізованої мови програмування Java та заліченням до проекту Spring framework.

Створена система відзначається легкістю у використанні, універсальністю застосування(може бути адаптована під інші види транспорту), а також нескладним

процесом залучень адвентних доповнень.

Реалізація авіаційного міжконтинентального сполучення дала змогу розвитку світової економіки, шляхом створення світової торгівлі. Наразі, забезпечення людства усім необхідним можна на пряму спів вставити з забезпеченням авіасполучень.

СПИСОК ВИКОРИСТАНИХ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. Number of flights performed by the global airline industry from 2004 to 2020. [Електронний ресурс]. — Режим доступу: <https://www.statista.com/statistics/564769/airline-industry-number-of-flights/> — Дата доступу: Жовтень 2023. — Назва з екрана.
2. Хто користувався українською небом в 2019 році: ТОП-10 авіакомпаній. [Електронний ресурс]. — Режим доступу: <https://biz.liga.net/all/transport/novosti/kto-chasche-vseh-letal-nad-ukrainoy-v-2019-godu-infografika> — Дата доступу: Жовтень 2023. — Назва з екрана.
3. Employment. [Електронний ресурс]. — Режим доступу: <https://aviationbenefits.org/economic-growth/supporting-employment/> — Дата доступу: Жовтень 2023. — Назва з екрана.
4. Which major expenses affect airline companies? [Електронний ресурс]. — Режим доступу: <https://www.investopedia.com/ask/answers/040715/what-are-major-expenses-affect-companies-airline-industry.asp> — Дата доступу: Жовтень 2023. — Назва з екрана.
5. Air Transport. [Електронний ресурс]. — Режим доступу: https://transportgeography.org/?page_id=1765 — Дата доступу: Жовтень 2023. — Назва з екрана.
6. Кількість літаків в небі побилло рекорд. [Електронний ресурс]. — Режим доступу: <https://lenta.ru/news/2018/07/02/recordflightnumber/> — Дата доступу: Жовтень 2023. — Назва з екрана.
7. Barnhart, C., Cohn, A.M., Johnson, E.L, Klabjan, D., Nemhauser, G.L., Vance, P.H. *Handbook of Transportation Science*, Springer Link, 2003. [Електронний ресурс]. — Режим доступу: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.5219&rep=rep1&type=pdf> — Дата доступу: Жовтень 2023. — Назва з екрана.
8. Andersson, E., Housos, E., Kohl, N., Wedelin D. *Operations Research in the Airline Industry*, pp.228-258, 1998. [Електронний ресурс]. — Режим доступу: <https://studfile.net/preview/7091353/page:16/> — Дата доступу: Жовтень 2023. — Назва з екрана.

з экрана.

9. Andres, V., Valdes V. Integrating Crew Scheduling and Rostering Problems. 2010. [Электронный ресурс]. — Режим доступа: http://amsdottorato.unibo.it/2705/1/vera_valdes_victor_andres_tesi.pdf — Дата доступа: Листопад 2023. — Назва з экрана.
10. Smallen.D. 2015 U.S.-Based Airline Traffic Data. *Washington: Bureau of Transportation Statistics*, 2016. - [Электронный ресурс]. — Режим доступа: http://www.rita.dot.gov/bts/press_releases/bts018_16Archetti, — Дата доступа: Листопад 2023. — Назва з экрана.
11. С., Speranza M.G. A survey on matheuristics for routing problems. 2013. - [Электронный ресурс]. — Режим доступа: <https://www.unibs.it/sites/default/files/ricerca/allegati/quaderno%20wpdem%2011.pdf> — Дата доступа: Листопад 2023. — Назва з экрана.
12. Özdemir U. Methodology for crew-pairing problem in airline crewscheduling. 2004. - [Электронный ресурс]. — Режим доступа: <https://www.cmpe.boun.edu.tr/~gungort/theses/Methodology%20for%20CrewPairing%20Problem%20in%20Airline%20Crew%20Scheduling.pdf>— Дата доступа: Листопад 2023. — Назва з экрана.
13. Wren A. A review of computer scheduling of buses and crews. *Proceedings of Public Transport Analysis Seminar*, pages 42–47, 1968
14. Nicoletti B. Automatic crew rostering. *Transportation Science*, 9(1):33–42, 1975. [Электронный ресурс]. — Режим доступа: http://www.academia.edu/4119857/Automatic_Crew_Rostering — Дата доступа: Листопад 2023. — Назва з экрана.
15. Freling, R., Lentink, R., Wagelmans A. A decision support system for crew planning in passenger transportation using a flexible branch-and-price algorithm. *Annals of Operations Research: Staff Scheduling and Rostering: Theory and Applications*, Part I, 127(1–4):203–222, 2004. [Электронный ресурс]. — Режим доступа: https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&mlr_RyWg&sig2=kvC353p4ZvXHIBZONVBqIg — Дата доступа: Листопад 2023. — Назва з экрана.
16. Branch and price. [Электронный ресурс]. — Режим доступа: https://en.wikipedia.org/wiki/Branch_and_price — Дата доступа: Листопад 2023. — Назва з экрана.

17. Barnhart, C., Marla, L., Jiang, H. Optimization approaches to airline industry challenges: Airline Schedule Planning and Recovery. *DROPS*, 2009. [Електронний ресурс]. — Режим доступу: <http://drops.dagstuhl.de/opus/volltexte/2009/2188/pdf/09261.BarnhartCynthia.ExtAbstract.2188.pdf> — Дата доступу: Листопад 2023. — Назва з екрана.
18. Vance, P. H., Atamturk, A. Technical Report TLI/LEC-97-06, Georgia Institute of Technology, Atlanta, GA, 1997. [Електронний ресурс]. — Режим доступу: <http://www.ieor.berkeley.edu/~atamturk/pubs/united.pdf> — Дата доступу: Листопад 2023. — Назва з екрана.
19. Doig, A. G., Land, A. H. An automatic method of solving discrete programming problems. *Econometrica*. 28 (3). pp. 497–520, 1960. [Електронний ресурс]. — Режим доступу: <http://www.jstor.org/stable/1910129?origin=crossref> — Дата доступу: Листопад 2023. — Назва з екрана.
20. Ehrgott, M., Tind, J. Column Generation in Integer Programming with Applications in Multicriteria. *Optimization. Technical Report of the Faculty of Engineering*, 2007. [Електронний ресурс]. — Режим доступу: <http://www.math.ku.dk/~tind/integercolumnWEB> — Дата доступу: Листопад 2023. — Назва з екрана.
21. Savelsbergh, M. A Branch-and-Price Algorithm for the Generalized Assignment Problem. *Georgia Tech*, 1997. [Електронний ресурс]. — Режим доступу: <http://www2.isye.gatech.edu/~ms79/publications/or45.pdf> — Дата доступу: Листопад 2023. — Назва з екрана.
22. Vanderbeck, F., Wolsey, L. An exact algorithm for IP column generation. *Operations Research Letters*, 19, 151 – 159, 1995. [Електронний ресурс]. — Режим доступу: <https://www.math.u-bordeaux.fr/~fvanderb/papers/ipcgPap.pdf> — Дата доступу: Листопад 2023. — Назва з екрана.
23. Рейтинг мов програмування 2020. [Електронний ресурс]. — Режим доступу: <https://dou.ua/lenta/articles/language-rating-jan-2020/> — Дата доступу: Листопад 2023. — Назва з екрана.
24. Branch and price. [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Branch_and_price#cite_note-lectureSlides-1 — Дата

доступу: Листопад 2023. — Назва з екрана.

25. Spring для лінивих. Основи, базові концепції та приклади з кодом. Частина 1. [Електронний ресурс]. — Режим доступу: <https://javarush.com/groups/posts/spring-framework-java-1>— Дата доступу: Листопад 2023. — Назва з екрана.

26. Фреймворк Spring та його особливості. [Електронний ресурс]. — Режим доступу: <https://highload.today/uk/frejmwork-spring-ta-jogo-osoblivosti/>— Дата доступу: Листопад 2023. — Назва з екрана.

27. Розділ IV. ТРЕТИННИЙ СЕКТОР ГОСПОДАРСТВА. ТЕМА 1. ТРАНСПОРТ. ТРАНСПОРТ СВІТУ. [Електронний ресурс]. — Режим доступу: https://subject.com.ua/textbook/geography/9klas_7/41.html — Дата доступу: Листопад 2023. — Назва з екрана.

28. Конвенція про міжнародну цивільну авіацію/ [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B2%D0%B5%D0%BD%D1%86%D1%96%D1%8F_%D0%BF%D1%80%D0%BE_%D0%BC%D1%96%D0%B6%D0%BD%D0%B0%D1%80%D0%BE%D0%B4%D0%BD%D1%83_%D1%86%D0%B8%D0%B2%D1%96%D0%BB%D1%8C%D0%BD%D1%83_%D0%B0%D0%B2%D1%96%D0%B0%D1%86%D1%96%D1%8E — Дата доступу: Листопад 2023. — Назва з екрана.

29. Авіаційні альянси. [Електронний ресурс]. — Режим доступу: <https://trips.com.ua/ohlyady/aviatsiyeni-alyansy/> — Дата доступу: Листопад 2023. — Назва з екрана.

30. 8-10 Вікторова Є.О..pdf [Електронний ресурс]. — Режим доступу: <https://dspace.nau.edu.ua/bitstream/NAU/55129/1/8-10%20%D0%92%D1%96%D0%BA%D1%82%D0%BE%D1%80%D0%BE%D0%B2%D0%B0%20%D0%84.%D0%9E..pdf>— Дата доступу: Грудень 2023. — Назва з екрана.

ДОДАТОК А

ТЕКСТ ПРОГРАМИ

```
package airways.calculation;
import airways.data.Cycle;
import airways.data.Flight; import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
import java.util.NoSuchElementException;
import java.util.Optional;
import java.util.stream.Collectors;
import java.util.stream.Stream;
public class Calculator {
    private final List<Flight> allFlights = new ArrayList<>(); // всі рейси
    private final List<Cycle> cycles = new ArrayList<>(); // обчислені цикли
    private final List<Flight> mandatoryFlights = new ArrayList<>(); // обов'язкові
    рейси
    private final List<Flight> mandatoryFlightsWithoutCycles = new
    ArrayList<>(); // обчислені обов'язкові рейси поза циклами
    public Calculator(List<Flight> allFlights) {
        if (allFlights == null) {
            throw new IllegalArgumentException("all flights list is null");
        }
        // Робляться копії списків для можливості видалення елементів в
    процесі роботи алгоритму
        this.allFlights.addAll(allFlights);
    this.allFlights.stream().filter(Flight::isMandatory).forEach(mandatoryFlights::add);
    }
```

```

public List<Cycle> getCycles() {
    return cycles;
}

public List<Flight> getMandatoryFlightsWithoutCycles() {return
    mandatoryFlightsWithoutCycles;
}

public void perform() {
    List<Cycle> allCycles = new ArrayList<>();
    // Для всіх рейсів
    while (!allFlights.isEmpty()) {
        // Порядок перебору - від рейсів, що починають найраніше
        final Flight origin = getEarliestFlight();
        // Знайти цикли, що містять обов'язкові рейси
        getCyclesWithMandatoryFlights(origin).forEach(allCycles::add);
        // Видалити рейс, для якого ми шукали цикли - він вже не потрібний
        allFlights.remove(origin);
    }
    // Для всіх обов'язкових рейсів
    while (!mandatoryFlights.isEmpty()) {
        // Порядок перебору - від рейсів із найбільшим прибутком
        final Flight mostValuable = getMandatoryFlightWithMaxIncome();
        // Знайти для такого рейсу найдешевший за витратами цикл
        final Optional<Cycle> cheapestOpt = getCheapestCycle(allCycles,
mostValuable);
        // Видалити рейс, для якого ми шукали цикли - він вже не потрібний
        mandatoryFlights.remove(mostValuable); if
(cheapestOpt.isPresent()) {
            final Cycle cheapest = cheapestOpt.get();
            // Додати знайдений найдешевший цикл до результату
            cycles.add(cheapest);

```


// Видалити всі цикли, які мають рейси, спільні зі знайденим - рейс відбувається лише раз

```
    allCycles = removeIntersectedCycles(allCycles, cheapest);
  } else {
    // Жодного циклу із таким рейсом не знайдено
    mandatoryFlightsWithoutCycles.add(mostValuable);
  }
}
}
private Optional<Cycle> getCheapestCycle(List<Cycle> allCycles, Flight
flight) {
  return allCycles.stream()
    .filter(cycle -> cycle.contains(flight))
    .min((c1, c2) -> c1.getExpenses(Calculator::getWaitExpenses) -
c2.getExpenses(Calculator::getWaitExpenses));
}
private List<Cycle> removeIntersectedCycles(List<Cycle> list, Cycle
baseCycle) {
  return list.stream()
    .filter(cycle -> cycle.getFlights().stream().noneMatch(flight ->
baseCycle.contains(flight)))
    .collect(Collectors.toList());
}
private Stream<Cycle> getCyclesWithMandatoryFlights(Flight origin) {
  return detectCycles(new
Cycle(origin)).stream().filter(Cycle::containsMandatory);
}

private Flight getEarliestFlight() {
  return allFlights.stream().min((f1, f2) -> f1.getDepartureTime() -
```

```

f2.getDepartureTime())
    .orElseThrow(() -> new NoSuchElementException());
}
private Flight getMandatoryFlightWithMaxIncome() {
    return mandatoryFlights.stream().max((f1, f2) -> f1.getIncome() -
f2.getIncome())
    .orElseThrow(() -> new NoSuchElementException());
}

private List<Cycle> detectCycles(Cycle base) {
    final Flight last = base.getLast();
    // Перевірити лише цикли, що мають хоча два рейси, отже, можуть
бути замкненими
    if (base.containsBeforeLast()) {
        // Час відправлення останнього рейсу має бути після часу прибуття
передостаннього:
        final Flight beforeLast = base.getBeforeLast();
        if (last.getDepartureTime() < beforeLast.getArrivalTime()) {
            // Далі ми рухатися по цьому ланцюжку не можемо - відсікаємо
Гілку return Collections.emptyList();
        }
        if (last.getTo() == base.getReturnPoint()) { #!/перевіряємо чи може бути
циклом
            // Останній рейс у ланцюжку повертається у початкову точку -цикл
знайдено:
            return Arrays.asList(base);
        }
    }
    // Продовжити рекурсивно для всіх сусідніх рейсів:
    return getNeighbours(last)

```

```

        .map(flight -> detectCycles(new Cycle(base, flight))) //!це рекурсія
        .flatMap(List::stream)
        .collect(Collectors.toList());
    }

    private Stream<Flight> getNeighbours(Flight flight) {
        return allFlights.stream().filter(next -> next.getFrom() == flight.getTo());
    }

    private static int getWaitExpenses(int time) {
        final double waitPrice = 0.3; // TODO треба поекспериментувати із
різними коефіцієнтами
        return (int) Math.round(time * waitPrice);
    }
}

package airways;

import java.util.List;
import java.util.ArrayList;
import airways.calculation.Calculator;

import airways.data.Cycle;
import airways.data.Flight;
import airways.data.FlightValidator;

import airways.data.FlightsStorage;

import airways.upload.csv.CsvParser;

import java.io.IOException;
import java.util.HashMap;

```

```

import java.util.Map;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping; import
org.springframework.web.bind.annotation.ModelAttribute; import
org.springframework.web.bind.annotation.PostMapping; import
org.springframework.web.bind.annotation.RequestParam; import
org.springframework.web.multipart.MultipartFile;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;

```

```
@Controller
```

```
@Scope("session")
```

```
public class MainController {
```

```
    private final FlightsStorage sourceFlightStorage = new FlightsStorage();
```

```
    private final List<Cycle> cycles = new ArrayList<>();
```

```
    private final List<Flight> mandatoryFlightsWithoutCycles = new
ArrayList<>();
```

```
    private final Map<String, String> validationErrors = new HashMap<>();
```

```
    private final List<String> importErrors = new ArrayList<>();
```

```
// Відкрити головну сторінку
```

```
@GetMapping("/")
```

```
public String home(Model model) {
```

```
    clearErrors();
```

```
    return goHome(model);
```

```
}
```

```
@PostMapping("/")
```

```

public String addFlight(@ModelAttribute Flight, Model model) {
    clearErrors();
    final Map<String, String> valErrors = new FlightValidator().apply(flight);
    if (valErrors.isEmpty()) {
        sourceFlightStorage.store(flight);
    } else {
        validationErrors.putAll(valErrors);
    }
    return goHome(model);
}

@GetMapping("/remove")
public String removeFlight(@RequestParam(name = "number", required = true)
int number, Model model) {
    sourceFlightStorage.remove(number);
    clearErrors();
    return goHome(model);
}

@PostMapping("/uploadCsv")
public String uploadCsv(@RequestParam("file") MultipartFile file,
    RedirectAttributes, Model model) throws IOException {
    cycles.clear();
    sourceFlightStorage.clear();
    mandatoryFlightsWithoutCycles.clear();
    clearErrors();
    final CsvParser parser = new CsvParser(file.getBytes());
    parser.perform();
    parser.getFlights().forEach(flight -> sourceFlightStorage.store(flight));
    importErrors.addAll(parser.getErrors());
    return goHome(model);
}

```

```

@GetMapping("/calculate")
public String calculate(Model model) {
    cycles.clear();
    mandatoryFlightsWithoutCycles.clear();
    clearErrors();
    final Calculator = new Calculator(sourceFlightStorage.list());calculator.perform();
    cycles.addAll(calculator.getCycles());

mandatoryFlightsWithoutCycles.addAll(calculator.getMandatoryFlightsWithoutCycles());
    return goHome(model);
}

private String goHome(Model model) {
    model.addAttribute("flight", new Flight());
    model.addAttribute("source", sourceFlightStorage.list());
    model.addAttribute("cycles", cycles);
    model.addAttribute("mandatoryFlightsWithoutCycles",
mandatoryFlightsWithoutCycles);
    model.addAttribute("validationErrors", validationErrors);
    model.addAttribute("importErrors", importErrors);
    return "home";
}

private void clearErrors() {
    importErrors.clear();
    validationErrors.clear();
}
}

package airways.data; public class Flight {

```

```

private int number; // номер. Унікальний ключ
private int from; // аеропорт, звідки рейс починається
private int to; // аеропорт, де рейс закінчується
private int income; // прибуток від рейсу або пріоритет рейсу
private int expenses; // витрати на рейс
private int departureTime; // час відльоту
private int arrivalTime; // час прильоту
private boolean mandatory; // рейс є обов'язковим?
public int getNumber() {
    return number;
}
public void setNumber(int number) {
    this.number = number;
}
public int getFrom() {
    return from;
}
public void setFrom(int from) {
    this.from = from;
}
public int getTo() {
    return to;
}
public void setTo(int to) {
    this.to = to;
}
public int getDepartureTime() {
    return departureTime;
}
}

```

```
public void setDepartureTime(int departureTime) {
    this.departureTime = departureTime;
}
public int getArrivalTime() {
    return arrivalTime;
}
public void setArrivalTime(int arrivalTime) {
    this.arrivalTime = arrivalTime;
}
public boolean isMandatory() {
    return mandatory;
}
public void setMandatory(boolean mandatory) {
    this.mandatory = mandatory;
}
public int getIncome() {
    return income;
}
public void setIncome(int income) {
    this.income = income;
}
public int getExpenses() {
    return expenses;
}
public void setExpenses(int expenses) {
    this.expenses = expenses;
}
}
```

```
package airways.data; import java.util.List;
```



```

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class FlightsStorage {

    // Внутрішній контейнер рейсів. Забезпечує унікальність рейсу за номером
    private final Map<Integer, Flight> data = new HashMap<>();

    public void store(Flight flight) {
        if (flight != null) {
            data.put(flight.getNumber(), flight);
        }
    }

    public void remove(int number) {
        if (data.containsKey(number)) {
            data.remove(number);
        }
    }

    public List<Flight> list() {
        final List<Flight> result = new ArrayList<>(data.values());
        result.sort((f1, f2) -> f1.getDepartureTime() - f2.getDepartureTime());
        return result;
    }

    public void clear() {
        data.clear();
    }
}

```

```

}
package airways.data;
import java.util.HashMap;
import java.util.Map;
import java.util.function.Function;

public class FlightValidator implements Function<Flight, Map<String, String>> {

    @Override
    public Map<String, String> apply(Flight flight) {
        final Map<String, String> result = new HashMap<>();
        // Місце відправлення та місце призначення мають бути різними
        if (flight.getFrom() == flight.getTo()) {
            result.put("to", "Start and end points of flight are the same");
        }

        // Час відправлення має бути меншим, ніж час прибуття
        if (flight.getDepartureTime() > flight.getArrivalTime()) {
            result.put("arrivalTime", "Arrival time is less than departure time");
        }
        return result;
    }

}

<!DOCTYPE html>

<html xmlns:th="https://www.thymeleaf.org"> <!-- мова розмітки, входить в
спрінгБут -->

```

```
<head>
```

```
<title>AIRWAYS application</title>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<style type="text/css">
```

```
.error {
```

```
  color: red;
```

```
}
```

```
td {
```

```
  vertical-align: top;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>AIRWAYS application:
```

```
<a href="#" th:href="@{/calculate}">Calculate</a></h1>
```

```
<form method="POST" enctype="multipart/form-data" th:action="@{/uploadCsv}"
```

```
>
```

```
<input type="file" name="file" /> <input type="submit" value="UploadCSV
```

```
file" />
```

```
</form>
```

```
<ul th:if="${!importErrors.isEmpty()}">
```

```
  Import errors:
```

```
<li th:each="error : ${importErrors}" th:text="${error}"/>
```

```
</ul>
```

```
<hr/>
```

```
<h2>Source</h2>
```

```
<form action="#" th:action="@{/}" th:object="${flight}" method="POST">
```

```
  <table>
```

```
    <tr>
```

```
      <th>Number</th>
```

```
      <th>From</th>
```

```
      <th>To</th>
```

```
      <th>Income</th>
```

```
      <th>Expenses</th>
```

```
      <th>Departure time</th>
```

```
      <th>Arrival time</th>
```

```
      <th>Mandatory</th>
```

```
      <th></th>
```

```
    </tr>
```

```
  <tr>
```

```
    <td>
```

```

        <input type="number" th:field="* {number}"/>
    </td>
    <td>
        <input type="number" th:field="* {from}"/>
    </td>
    <td>
        <input type="number" th:field="* {to}"/>
        <div class="error" th:if="{validationErrors.containsKey('to')}"
            th:text="{validationErrors.get('to')}" />
    </td>
    <td>
        <input type="number" th:field="* {income}"/>
    </td>
    <td>
        <input type="number" th:field="* {expenses}"/>
    </td>
    <td>
        <input type="number" th:field="* {departureTime}"/>
    </td>
    <td>
        <input type="number" th:field="* {arrivalTime}"/>
        <div class="error"
            th:if="{validationErrors.containsKey('arrivalTime')}"
            th:text="{validationErrors.get('arrivalTime')}" />

```

```
</td>
```

```
<td>
```

```
<select th:field="* {mandatory}">
```

```
<option value="true" th:value="true">true</option>
```

```
<option value="false" th:value="false">false</option>
```

```
</select>
```

```
</td>
```

```
<td>
```

```
<input type="submit" value="Add or update flight"/>
```

```
</td>
```

```
</tr>
```

```
<tr th:each="flight : ${source}">
```

```
<td th:text="${flight.number}" />
```

```
<td th:text="${flight.from}" />
```

```
<td th:text="${flight.to}" />
```

```
<td th:text="${flight.income}" />
```

```
<td th:text="${flight.expenses}" />
```

```
<td th:text="${flight.departureTime}" />
```

```
<td th:text="${flight.arrivalTime}" />
```

```
<td th:text="${flight.mandatory}" />
```

```
<td>
```

```
<a href="#"
```

```
th:href="@{/remove(number=${flight.number})}">Remove</a>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</form>
```

```
<div th:if="{!cycles.isEmpty()}">
```

```
<hr/>
```

```
<h2>Detected cycles</h2>
```

```
<table th:each="cycle : {cycles}">
```

```
<tr>
```

```
<td colspan="7" th:text="{cycle.getTag()}" />
```

```
</tr>
```

```
<tr>
```

```
<th>Number</th>
```

```
<th>From</th>
```

```
<th>To</th>
```

```
<th>Income</th>
```

```
<th>Expenses</th>
```

```
<th>Departure time</th>
```

```
<th>Arrival time</th>
```

```
<th>Mandatory</th>
```

```
</tr>
```

```
<tr th:each="flight : {cycle.getFlights()}">
```

```
<td th:text="{flight.number}" />
```

```
<td th:text="{flight.from}" />
```

```
<td th:text="{flight.to}" />
```

```
<td th:text="{flight.income}" />
```

```
<td th:text="{flight.expenses}" />
```

```
<td th:text="{flight.departureTime}" />
```

```
<td th:text="{flight.arrivalTime}" />
```

```
<td th:text="{flight.mandatory}" />
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
<div th:if="{!mandatoryFlightsWithoutCycles.isEmpty()}">
```

```
<hr/>
```

```
<h2>Detected mandatory flights without cycles</h2>
```

```
<table>
```

```
<tr>
```

```
<th>Number</th>
```

```
<th>From</th>
```

```
<th>To</th>
```

```
<th>Income</th>
```

```
<th>Expenses</th>
```

```
<th>Departure time</th>
```

```
<th>Arrival time</th>
```

```
<th>Mandatory</th>
```

```
</tr>
```

```
<tr th:each="flight : {mandatoryFlightsWithoutCycles}">
```

```
<td th:text="{flight.number}" />
```

```
<td th:text="{flight.from}" />
```

```
<td th:text="{flight.to}" />
```

```
<td th:text="{flight.income}" />
```



```
<td th:text="{flight.expenses}" />
```

```
<td th:text="{flight.departureTime}" />
```

```
<td th:text="{flight.arrivalTime}" />
```

```
<td th:text="{flight.mandatory}" />
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
</body>
```

```
</html>
```