

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут аерокосмічних технологій

Кафедра авіа- та ракетобудування

«На правах рукопису»
УДК 629.7.023.46

До захисту допущено:
В. о. завідувача кафедри
_____ Петро ЛУК'ЯНОВ
«__» _____ 2024 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Літаки і вертольоти»

зі спеціальності 134 «Авіаційна та ракетно-космічна техніка»

на тему: «Створення типових шпангоутів в системі Siemens NX»

Виконав:

студент II курсу другого (магістерського) рівня, групи АЛ-21мп
Литвиненко Іван Олександрович _____

Керівник:

Доцент кафедри АРБ, к.т.н.,
Бондар Юрій Іванович _____

Рецензент:

Доцент кафедри СКЛА, к.т.н.,
Бобков Ю.В. _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут аерокосмічних технологій
Кафедра авіа- та ракетобудування

Рівень вищої освіти – другий (магістерський)

Спеціальність – **134 «Авіаційна та ракетно-космічна техніка»**

Освітньо-професійна програма «Літаки і вертольоти»

ЗАТВЕРДЖУЮ
В. о. завідувача кафедри
_____ Петро ЛУК'ЯНОВ
«__» _____ 2023 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Литвиненку Івану Олександровичу

1. Тема дисертації «Створення типових шпангоутів у системі *Siemens NX*», науковий керівник дисертації доцент кафедри Бондар Юрій Іванович, к.т.н., затверджена наказом по університету від «03» __11__ 2023 р. № 5134-с
2. Термін подання студентом дисертації __26 грудня 2023 р.
3. Об'єкт дослідження – Технологія створення шпангоутів у *CAD* системі
4. Вихідні дані:
 - Типові шпангоути
 - Типові вирізи під стрингери
 - Майстер-геометрія обшивки та стрингерів
 - Площина шпангоуту
5. Перелік завдань, які потрібно розробити
 - 5.1. Аналітичний огляд конструктивно-силового набору планера літака
 - 5.2. Огляд типових шпангоутів, розробка класифікатора.
 - 5.3. Методика побудови шпангоутів у *CAD* системі *Siemens NX*.
 - 5.4. Розробка інструменту для створення шпангоутів.
 - 5.5. Розробка допоміжних інструментів для користувача.
 - 5.6. Тестування інструменту.
 - 5.7. Розробка стартап-проекту.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу
- 6.1. Аналітичний огляд конструктивно-силового набору планера літака
- 6.2. Розробка класифікатора профілів.
- 6.3. Типові профілі шпангоутів літака із розробленого класифікатора.
- 6.4. Інтерфейс інструменту.
- 6.5. Результат використання інструменту.
- 6.6. Розробка стартап-проекту.
7. Орієнтовний перелік публікацій
- 7.1. Стаття у фаховому виданні.
- 7.2. Доповідь на науково-технічній конференції з публікацією тез.
8. Дата видачі завдання 08.11.2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Аналітичний огляд конструктивно-силового набору планера літака	до 15.02.2022 р.	
2.	Огляд типових шпангоутів, розробка класифікатора.	до 01.02.2023 р.	
3.	Методика побудови шпангоутів у CAD системі <i>Siemens NX</i> .	до 21.03.2023 р.	
4.	Розробка інструменту для створення шпангоутів.	до 03.06.2023 р.	
5.	Розробка допоміжних інструментів для користувача.	до 20.09.2023 р.	
6.	Тестування інструменту.	до 27.10.2023 р.	
7.	Розробка стартап-проекту.	до 05.12.2023 р.	
8.	Оформлення пояснювальної записки та ілюстративного матеріалу. Попередній захист	до 26.12.2023 р.	
9.	Доопрацювання матеріалів дисертації. Перевірка на плагіат	до 10.01.2024 р.	

Студент

Іван ЛИТВИНЕНКО

Науковий керівник

Юрій БОНДАР

Реферат

Магістерська дисертація: "Створення типових шпангоутів в системі Siemens NX", 92 сторінок, 65 рисунків, 18 таблиць, 12 посилань.

Актуальність: інженерно-технічна сфера сучасного індустріального середовища постійно розвивається, вимагаючи від фахівців постійного удосконалення та впровадження нових технологій для підвищення ефективності проектування та виробництва. Однією з ключових складових цього процесу є розробка програмного забезпечення, спрямованого на полегшення та оптимізацію простих та повторюваних завдань. В контексті системи Siemens NX, яка є однією з передових інтегрованих платформ для механічного проектування і виробництва. Застосування стандартних шаблонів значно полегшує та прискорює створення нових проектів.

Мета роботи: розробка інструменту для створення типових шпангоутів в CAD-системі Siemens NX, допоміжних програм для користувача та класифікатора профілів шпангоутів та вирізів під стрингер в ньому.

Об'єкт дослідження: Технологія створення шпангоутів в CAD-системі.

Предмет дослідження: Автоматизація створення шпангоутів в CAD-системі Siemens NX.

Методи дослідження: аналіз конструктивно-силової схеми літака, типових шпангоутів, їх перерізів та вирізів під стрингер; дослідження методології побудови шпангоута в CAD системі; визначення основних необхідних функцій Siemens NX для побудови шпангоута; аналіз бібліотеки NXOpen та можливостей створення інтерфейсу інструменту за допомогою стандартних функцій Siemens NX.

Наукова новизна одержаних результатів: розроблено інструмент для створення типових шпангоутів в системі Siemens NX, який дає можливість

побудувати шпангоут двома способами, також розроблено класифікатор профілів шпангоутів та вирізів в них під стрингер, в яку користувач зможе додавати свої профілі та використовувати в подальшому в інструменті.

Практичне значення одержаних результатів: автоматизація проектування шпангоутів за рахунок спеціального інструменту для побудови шпангоутів, який використовує класифікатор профілів шпангоутів та вирізів під стрингер, що зменшує час на створення шпангоута та спрощує проектувальний процес.

Апробація результатів дисертації: V науково-практична конференція студентів та молодих вчених «Авіа-ракетобудування: Перспективи та напрямки розвитку».

Публікації: XV International Conference of Students and Young Scientists «INTELLIGENCE.INTEGRATION.RELIABILITY» 2023-го року.

Ключові слова: літак, конструкція, фюзеляж, силовий елемент, шпангоут, стрингер, виріз, CAD-система, Siemens NX, автоматизація, проектування, створення, інструмент, класифікатор.

Abstract

Master's thesis: "Creating standard frames in the Siemens NX system", 92 pages, 65 figures, 18 tables, 12 references.

Relevance: The engineering and technical sphere of the modern industrial environment is constantly evolving, requiring specialists to constantly improve and implement new technologies to increase the efficiency of design and production. One of the key components of this process is the development of software aimed at facilitating and optimizing simple and repetitive tasks. In the context of the Siemens NX system, which is one of the leading integrated platforms for mechanical design and manufacturing. The use of standard templates greatly facilitates and accelerates the creation of new projects.

Purpose of the study: Development of a tool for creating standard frames in the Siemens NX CAD system, auxiliary programs for the user and a classifier of frames profiles and cutouts for stringers in it.

The object of study: Technology for creating frames in a CAD system.

Subject of research: Automation of the creation of frames in the Siemens NX CAD system.

Methods of research: analysis of the structural and power scheme of the aircraft, typical spars, their cross-sections and cutouts for the stringer; study of the methodology for building a spar in a CAD system; identification of the main necessary Siemens NX functions for building a spar; analysis of the NXOpen library and the possibilities of creating a tool interface using standard Siemens NX functions.

The results and their novelty: developed a tool for creating standard frames in the Siemens NX system, which makes it possible to build a frame in two ways, and developed a classifier of frames profiles and cutouts for stringers, to which the user can add their own profiles and use them in the tool in the future.

Recommendations for the use of work: automation of the design of frames due to a special tool for building frames, which uses a classifier of frames and stringer cutouts, which reduces the time for creating a frame and simplifies the design process.

Approbation of the results of the dissertation: V the scientific and practical conference of students and young scientists "Aircraft and rocket construction and design: Perspectives and directions of development"

Publications: XV International Conference of Students and Young Scientists "INTELLIGENCE.INTEGRATION.RELIABILITY" 2023th year.

Keywords: aircraft, construction, fuselage, primary element of structure, frame, stringer, cutout, CAD system, Siemens NX, automation, design, creation, tool, classifier.

Зміст

Перелік умовних позначень	11
Вступ. Актуальність теми та її практичне значення	12
1. Аналітичний огляд конструктивно-силового набору літака.	14
1.1 Фюзеляж.....	14
1.2. Крило.....	20
1.3. Оперення.....	29
Висновок	32
2. Огляд типових шпангоутів, розробка класифікатора.	33
2.1. Виготовлення шпангоутів.	34
2.2. Розробка класифікатора.	36
Висновок	38
3. Методика побудови кожного класу шпангоутів у CAD системі Siemens NX.	39
Висновок	46
4. Розробка програми для створення шпангоутів.	47
4.1. NXOpen	47
4.2. Інтерфейс інструменту	48
4.3. Розробка інструменту	51
4.4. Основні функції використовуються в інструменті.....	52
Висновок	57
5. Розробка допоміжних програм для користувача.	58
5.1. Інтерфейс допоміжного інструменту для додавання профілів шпангоута.....	58
5.2. Інтерфейс допоміжного інструменту для додавання профілів вирізів під стрингери в шпангоуті	59
Висновок	60
6. Тестування програми.....	61
6.1. Загальне тестування інструменту.....	61
6.2. Тестування окремих випадків.....	68
Висновок	83
Висновки	84

АЛ-2109мп.70.01.00.00 ПЗ				
Зм.	Арк.	№ документа	Підпис	Дата
		Литвиненко І.О.		
		Бондар Ю.І.		
		Поваров С.А.		
		Лук'янов П. В.		
Створення типових шпангоутів в системі Siemens NX			Літ.	Аркуш
			9	92
<i>КПІ ім. Ігоря Сікорського Каф. АРБ гр. АЛ-21мп</i>				

Перелік використаних джерел	86
Додаток 1	88
Додаток 2	89
Додаток 3	90
Додаток 4	91
Додаток 5	92
Додаток 6	93

Перелік умовних позначень

CAD – система автоматизованого проектування (Computer-aided Design).

PLM – керування життєвим циклом виробу (Product Lifecycle Management).

БКС – базова конструктивна схема.

ЛА – літальний апарат.

КСС – конструктивно-силова схема.

ВО – вертикальне оперення.

ГО – горизонтальне оперення.

СПГО – суцільно-поворотне горизонтальне оперення.

Вступ. Актуальність теми та її практичне значення

Сучасні технології в області машинобудування та аерокосмічної промисловості вимагають високої точності та швидкості в процесі розробки та виробництва. Одним із ключових етапів цього процесу є створення шпангоутів - структурних елементів, які забезпечують міцність та стійкість конструкцій. В даному рефераті розглянуто програму для створення шпангоутів в системі Siemens NX.

Siemens NX є інтегрованою системою проектування та виробництва, яка забезпечує повний цикл розробки виробів. Ця система володіє широким набором інструментів для створення та аналізу складних тривимірних моделей. Використання Siemens NX значно полегшує роботу інженерів та дозволяє швидше впроваджувати нові розробки.

Шпангоути є важливим силовим елементом літального апарату. Створення їх в системі Siemens NX може вимагати значних зусиль та часу вручну. Розробка інструменту для автоматизації цього процесу дозволить інженерам ефективно та швидко створювати шпангоути, зменшуючи час проектування.

Актуальність роботи: інженерно-технічна сфера сучасного індустриального середовища постійно розвивається, вимагаючи від фахівців постійного удосконалення та впровадження нових технологій для підвищення ефективності проектування та виробництва. Однією з ключових складових цього процесу є розробка програмного забезпечення, спрямованого на полегшення та оптимізацію простих та повторюваних завдань.

Мета роботи: розробка інструменту для створення типових шпангоутів в CAD-системі Siemens NX, допоміжних програм для користувача та класифікатора профілів шпангоутів та вирізів під стрингер в ньому.

Задачі дослідження:

1. Аналіз конструктивно-силового набору літака та основних силових елементів.

2. Дослідити методологію моделювання шпангоутів в CAD-системі Siemens NX.

3. Розробити інструмент для побудови шпангоутів в CAD-системі Siemens NX та класифікатор профілів шпангоутів та вирізів під стрингер, а також допоміжні інструменти для зручного використання класифікатора.

4. Оцінити параметри економічної ефективності наукового дослідження та розробити на його основі стартап-проект.

Об'єкт дослідження: Технологія створення шпангоутів в CAD-системі.

Предмет дослідження: Автоматизація створення шпангоутів в CAD-системі Siemens NX.

Наукова новизна: розроблено інструмент для створення типових шпангоутів в системі Siemens NX, який дає можливість побудувати шпангоут двома способами, також розроблено класифікатор профілів шпангоутів та вирізів в них під стрингер, в яку користувач зможе додавати свої профілі та використовувати в подальшому в інструменті.

Практичне значення отриманих результатів: автоматизація проектування шпангоутів за рахунок спеціального інструменту для побудови шпангоутів, який використовує класифікатор профілів шпангоутів та вирізів під стрингер, що зменшує час на створення шпангоута та спрощує проектувальний процес.

1. Аналітичний огляд конструктивно-силового набору літака.

1.1 Фюзеляж

Фюзеляж літака призначений для розміщення екіпажу, обладнання та цільового навантаження. У фюзеляжі може розташовуватись паливо, шасі, двигуни. Представляючи собою будівною основою конструкції літака, він об'єднує в силовому сенсі в одне ціле всі його частини [1].

В загальній конструктивно-силовій схемі (КСС) літака фюзеляж займає особливе місце, представляючи собою силову ланку, на якій повинні бути врівноважені всі навантаження: масові, аеродинамічні, а також виникаючі в місцях кріплення до нього інших агрегатів літака та розташованих всередині нього вантажів, механізмів та обладнання. Ці обставини викликають великі труднощі в процесі проектування, так як вузли кріплення агрегатів та установок механізмів, стикові з'єднання та гермовідсіки, вирізи та експлуатаційні роз'єми, розраховані у відповідності до прийнятими для них коефіцієнтами безпеки та заданими перевантаженнями, складають надзвичайне різноманіття розподілених навантажень [1].

Сили, які діють на фюзеляж під час польоту та при посадці, можна розділити на наступні групи [2]:

1. Масові сили конструкції фюзеляжу.
2. Масові сили агрегатів, вантажів та обладнання розташованих в фюзеляжі.
3. Сили, що приходять до фюзеляжу від прикріплених до нього частин літака.
4. Аеродинамічні сили, які розподілені по поверхні фюзеляжу.
5. Для герметичних фюзеляжів – сили внутрішнього тиску.

Сили прикладені до фюзеляжу, розглядаються в двох площинах: паралельній площині симетрії літака та перпендикулярній площині симетрії літака. Під дією навантажень конструкція літака зазнає деформацій. Від сил, які діють паралельно площині симетрії, фюзеляж згинається у вертикальній

площині. Під дією сил прикладених до фюзеляжу в горизонтальній площині фюзеляж згинається в горизонтальній площині, а також при навантаженні на вертикальне оперення фюзеляж скручується. Таким чином, конструкція фюзеляжу в загальному випадку працює на вертикальний згин та здвиг, горизонтальний згин та здвиг та кручення [2].

За конструктивно-силовою схемою фюзеляжі діляться на фермові, балкові та змішані (фермово-балкові) [3].

Фермові фюзеляжі (рис. 1.1) – це просторова ферма, яка складається з чотирьох плоских форм (двох вертикальних та двох горизонтальних), зв'язаних між собою поперечним набором. Кожна плоска ферма складається з поясів, спільних для двох суміжних ферм, стійок та розкосів. Поперечний набір складається з діагональних просторових стержнів [3].

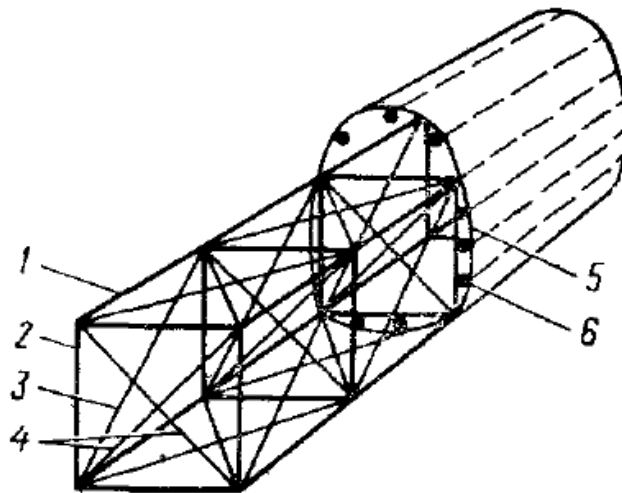


Рис. 1.1. Фермовий фюзеляж.

1 – пояс; 2 – стійка; 3 – розкос; 4 – розчалки;

5 – профільований шпангоут; 6 – стрингер.

Така просторова ферма сприймає всі діючі навантаження на фюзеляж. Для надання фюзеляжу обтічної форми на ферми ставляться профільовані шпангоути, до яких кріпляться стрингери. Каркас фюзеляжу закривається обшивкою [3].

Фермові фюзеляжі доцільно робити для не швидкісних легких літаків, на яких може бути застосована полотняна обшивка. В такому випадку фермовий фюзеляж може мати меншу масу, ніж балковий. Встановлення профільованих шпангоутів, стрингерів та металевої обшивки, які забезпечують гарні аеродинамічні форми фюзеляжу та якість поверхні, але не беруть участь в силовій роботі, призводить до значного збільшення маси конструкції [3].

Балкові фюзеляжі (рис. 1.2). Прагнення надати фюзеляжу грани аеродинамічні форми, отримати високу якість поверхні та створити відносно легку, міцну та жорстку конструкцію, яка б дозволяла повністю використовувати внутрішній об'єм, призвело до широкого використання балкових фюзеляжів [3].

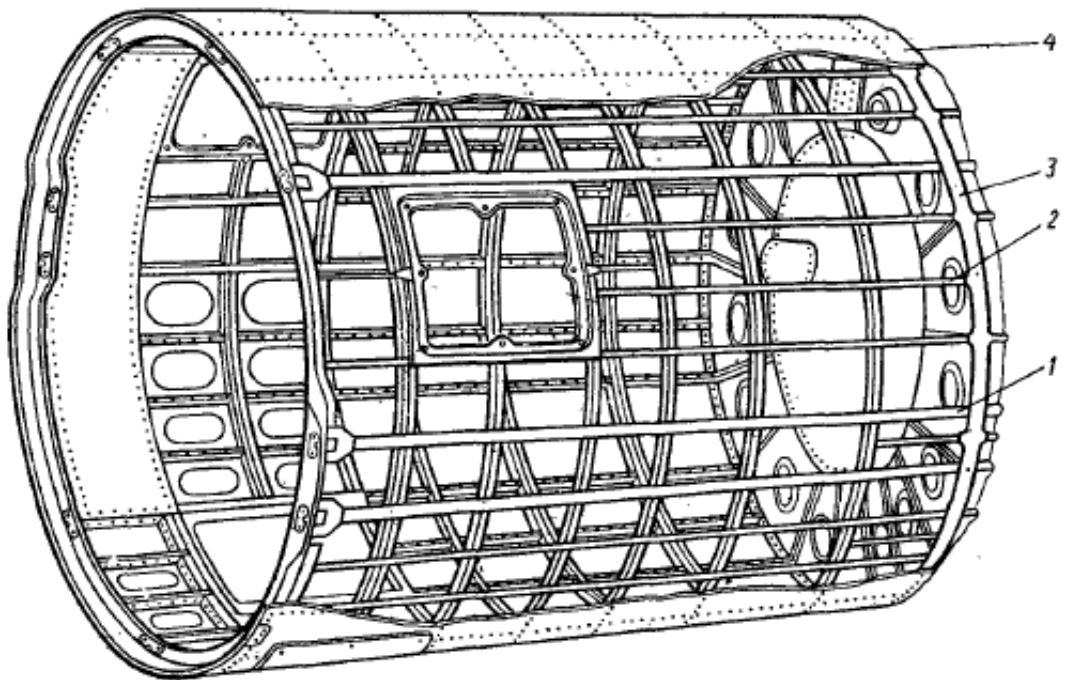


Рис. 1.2. Балковий фюзеляж

1 – лонжерон; 2 – стрингер; 3 – шпангоут; 4 – обшивка.

Конструкція балкового фюзеляжу складається з жорсткої обшивки, повздовжнього набору (лонжерони та стрингери) та поперечного набору – шпангоутів [3].

Лонжерони – це потужні елементи повздовжнього набору, які сприймають майже весь згинальний момент фюзеляжу [3].

Стрингери слугують для підкріплення обшивки та сприймають разом з нею згинальний момент [3].

Шпангоути забезпечують збереження заданої форми поперечних перерізів фюзеляжу, підкріплюючи обшивку та набір стрингерів, та сприймають місцеве аеродинамічне навантаження та навантаження від підкріплених до них агрегатів [3].

Обшивка фюзеляжу утворює його поверхню, передає аеродинамічне навантаження на каркас і бере участь в роботі фюзеляжу на згин та кручення [3].

Всі балкові фюзеляжі в залежності від ступеня участі в силовій роботі окремих елементів можна розділити на лонжеронні, стрингерний (фюзеляж типу напівмонокок) та безстрингерний (фюзеляж типу монокок) [3].

Лонжеронний фюзеляж (рис. 1.3) має повздовжній набір, який складається з чотирьох лонжеронів та відносно невеликого числа стрингерів, які мають дуже мало площу поперечного перерізу. До поперечного набору належать шпангоути. Каркас покривається порівняно тонкою жорсткою обшивкою. Основними силовими елементами такого фюзеляжу являються чотири лонжерони, які сприймають майже весь згинаючий момент. Слабкий набір стрингерів служить для підкріплення обшивки. Відносно тонка обшивка сприймає крутний момент [3].

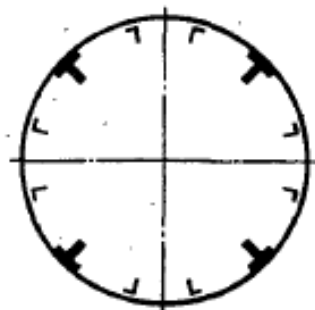


Рис. 1.3. Поперечний переріз лонжеронного фюзеляжу.

Стрингерний фюзеляж (рис. 1.4) має повздовжній набір, який складається з великої кількості стрингерів, поперечний – з шпангоутів. Каркас покривається жорсткою обшивкою. Набір стрингерів разом з обшивкою сприймає згинаючий момент фюзеляжу. Частка згинаючого моменту, який сприймається обшивкою, залежить від товщини обшивки та частоти стрингерного набору. Чим товстіше обшивка і чим більша відстань між стрингерами, тим більшу участь бере обшивка в роботі на згин. Крутий момент сприймається обшивкою. Таким чином, в обшивці стрингерного фюзеляжу виникають не тільки дотичні, але і нормальні напруження. Шпангоути в стрингерному фюзеляжу встановлюються частіше, ніж в лонжеронному. Це покращує роботу стрингерів та обшивки при стисканні [3].

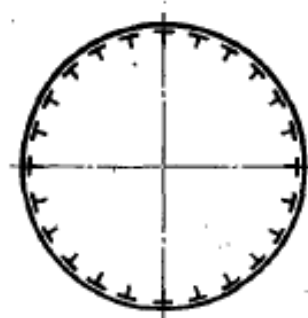


Рис. 1.4. Поперечний переріз стрингерного фюзеляжу

Безстрингерний фюзеляж (рис. 5) представляє собою конструкцію, яка складається з порівняно товстої обшивки або шаруватої обшивки з заповнювачем, яка підкріплена тільки шпангоутами. В такому фюзеляжі всі навантаження сприймає обшивка, і в ній потім виникають і нормальні, і дотичні напруження. В місцях вирізу обшивки та в місцях прикладення концентрованих навантажень в таких фюзеляжах встановлюються стрингери та інші посилюючі елементи [3].

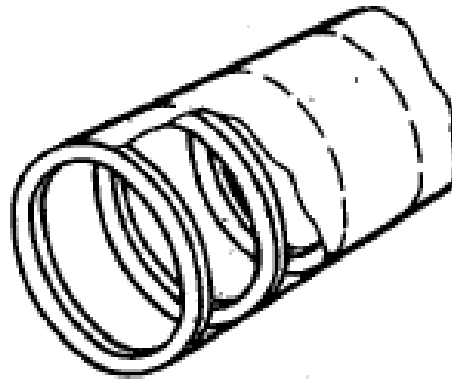


Рис. 1.5. Відсік безстрингерного фюзеляжу

Порівняльна характеристика різних типів балкових фюзеляжів

Найбільш вигідним в масовому сенсі міг би бути безстрингерний фюзеляж з обшивкою змінної товщини. Його конструкція має високу живучість, дозволяє раціонально використовувати внутрішній об'єм. Однак велика кількість вирізів в обшивці, необхідність передачі на неї концентрованих зусиль від інших агрегатів літака потребують місцевих зусиль у вигляді накладок та окантовок профілями. Маса цих зусиль на безстрингерному фюзеляжі при великій кількості різних вирізів стає настільки великою, що ця схема стає менш вигідною, ніж стрингерна чи лонжеронна [3].

Якщо фюзеляж має дуже великі вирізи, то найбільш вигідною може стати виявитися лонжеронна схема. В такому фюзеляжі тонка обшивка і тому простіше виконувати підкріплення великих вирізів, при цьому масові затрати будуть меншими [3].

При наявності в конструкції великої кількості не дуже великих вирізів найбільш вигідним є стрингерний фюзеляж. Наявність частого стрингерного набору дозволяє облегшити роботу обшивки, простіше здійснювати підкріплення вирізів. Тому ця схема зараз широко застосовується, особливо для важких літаків [3].

Дуже часто застосовуються балкові фюзеляжі комбінованої схеми [3].

1.2. Крило.

Конструктивно-силову схему крила зазвичай утворюють повздовжні та поперечні елементи та обшивка. До повздовжнього набору належать лонжерони, повздовжні стінки та стрингери [4].

Лонжероном – це потужна балка з верхнім та нижнім поясом, що сприймає значну частину згинального моменту крила та поперечної сили. Один з поясів в цей час працює на розтяг, інший – на стиск. Стінка лонжерона сприймає поперечну силу і частину крутного моменту, працюючи при цьому на зсув [4].

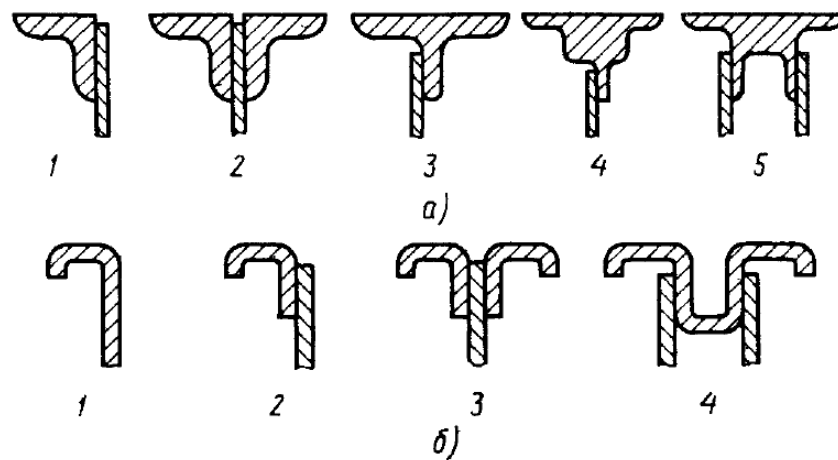


Рис. 1.6. Типові перерізи балкових лонжеронів

а – пояси з пресованих та катаних профілів; б – пояси з гнутих профілів

Повздовжня стінка – це елемент конструкції, менш потужний, ніж лонжерон, і розташований по розмаху на всій довжині крила чи на певній його ділянці. Повздовжній стінки сприймають поперечну силу і частково крутний момент крила [4].

За конструкцією і розташуванню повздовжні стінки аналогічні лонжеронам крила, але з досить слабкими поясами, тому їх часто називають додатковими лонжеронами чи лонжеронами-стінками [4].

Стрингер – повздовжній елемент, який разом з поясами лонжеронів та обшивкою сприймає згинальний момент. Крім того стрингери передають місцеве аеродинамічне навантаження з обшивкою на нервюри та підкріплюють

обшивку збільшуючи її критичне навантаження та зменшуючи деформацію від місцевого аеродинамічного навантаження. Стрингер зазвичай представляє собою профіль у вигляді кутика, швелера, тавра та інших форм перерізів (рис. 1.7) [4].

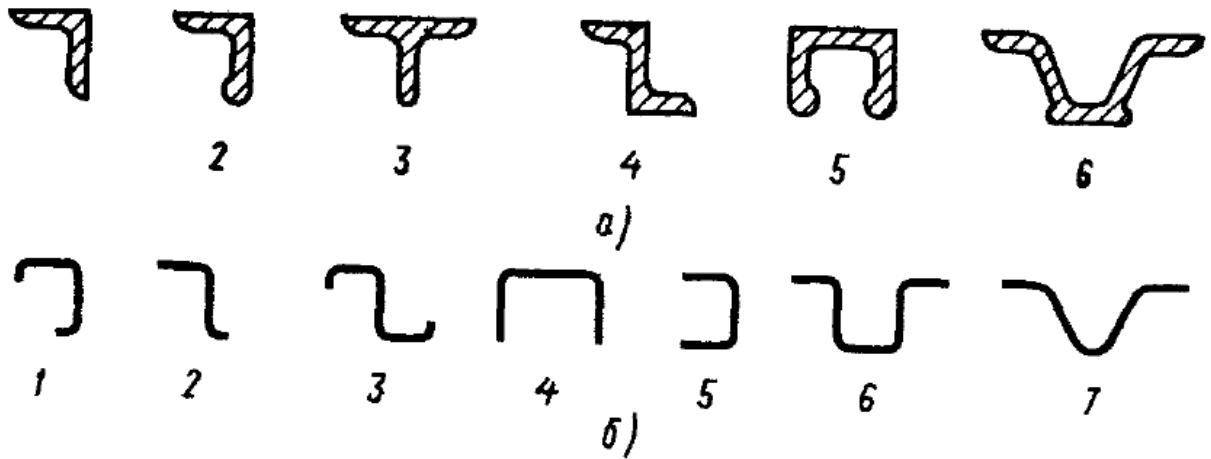


Рис. 1.7. Типові перерізи стрингерів
а – пресовані профілі; б – гнуті профілі

Поперечний набір крила зазвичай складається з нервюр, які представляють собою тонкостінну [4].

Нервюри (рис. 1.8) створюють форму крила заданого профілю та передають місцеві аеродинамічні навантаження на лонжерони та повздовжні стінки, посилюють стрингери та обшивку, збільшуючи їх критичне навантаження [4].

Відстань між нервюрами визначається з даних розрахунків на міцність крила та залежить від швидкості літака, величини питомого навантаження на крило, товщини обшивки та конструктивної схеми крила [4].

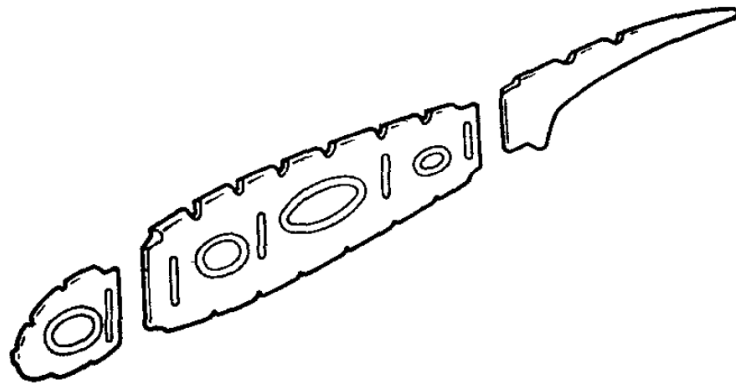


Рис. 1.8. Нормальна балкова нервюра

Обшивка герметизує конструкцію крила та утворює задану для нього форму. Нерівності на зовнішній поверхні крила повинні бути зведені до мінімуму. Обшивка сприймає місцеве аеродинамічне навантаження. Вона може працювати в залежності від конструкції крила на здвиг і на стиск чи розтяг від загального згинання та кручення [4].

Існують різноманітні конструктивні схеми крила. За основну ознаку, яка характеризує тип конструкції крила, можна взяти характер роботи та ступінь використання обшивки чи повздовжнього набору при згині та крученні крила. За цими ознаками розрізняють лонжеронні, кесонні та моноблочні конструктивні схеми крил (рис. 1.9) [4].

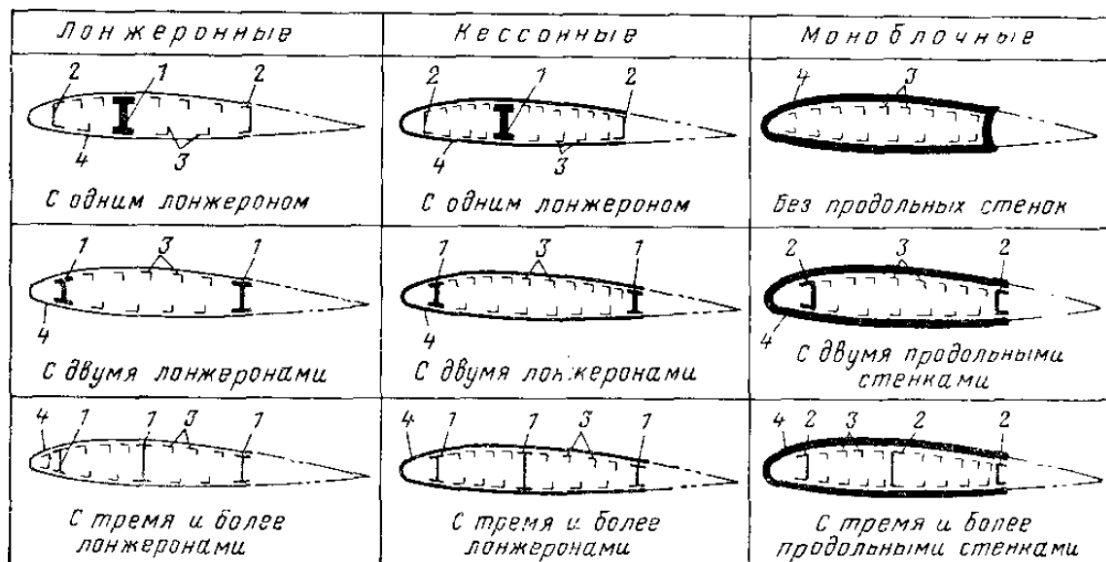


Рис. 1.9. Конструктивні схеми крила

1 – лонжерони; 2 – повздовжні стінки; 3 – стрингери; 4 - обшивка

Якщо згинальний момент в основному сприймається лонжеронами, які мають потужні пояси, то таке крило називають лонжеронним. Залежно від кількості лонжеронів розрізняють однолонжеронні, дволонжеронні та багатолонжеронні крила [4].

Кесонним крилом – це крило, де обшивка разом з повздовжніми елементами, які її підкріплюють, сприймає всі види навантажень, які діють на крило. В такі конструкції може бути один, два, чи більше лонжеронів з ослабленими поясами. Кесонні крила представляють собою поєднання лонжеронної та моноблочної конструктивно-силових схем [4].

Для моноблочного крила, як конструктивно-силової схеми, притаманно використання обшивки в ролі основного силового елемента, який працює при згині та крученні. Таким чином, моноблочним крилом називають крило, в конструкції якого повздовжні при згині сприймаються обшивкою і стрингерами по всьому поперечному контуру крила. Лонжерони в моноблочних крилах відсутні, а замість них встановлюються повздовжні стінки. Зустрічаються випадки, коли конструкція крила виконана за змішаними конструктивно-силовими схемами. Наприклад, в кореневій частині застосована дволонжеронна схема, яка потім переходить в моноблочну. Такі крила називають крилами змішаної силової схеми [4].

Лонжеронні крила.

Такі крила мають лонжерони та порівняно тонку обшивку, яка підкріплена стрингерами та нервюрами. Товщини обшивки та набору, який її підкріплює, визначається з умов роботи на здвиг від кручення. При згині крила тонка обшивка в зжатій зоні має низьке значення критичного напруження. В результаті участь обшивки в сприйнятті згинаючого моменту обмежується роботою її в розтягнутій зоні, де вона лише розвантажує нижні пояси лонжеронів [4].

Кількість лонжеронів в крилі визначається розмірами та формою крила. Найчастіше використовуються одно- або дволонжеронні крил, однак при великій хорді крила може бути доцільна багатолонжеронна конструкція [4].

В одно лонжеронному крилі (рис. 1.10) лонжерон зазвичай розташований в місці максимальної висоти профілю. Для отримання контуру, який зможе сприймати кручення, та бази для кріплення елеронів та засобів механізації на однолонжеронному крилі на 65-70% його хорди встановлюється повздовжня стінка. Інколи також ставиться повздовжня передня стінка [4].

Конструкції однолонжеронних крил мають прості стикові вузли для кріплення до центроплану і дозволяють гарно використовувати внутрішні об'єми [4].

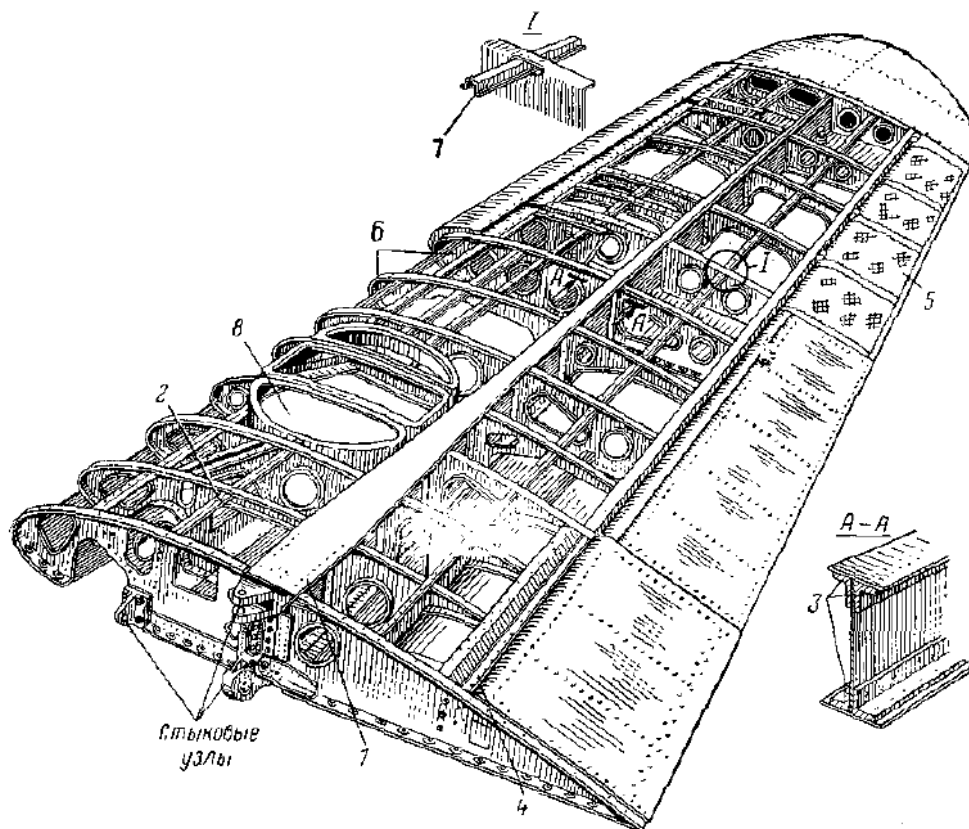


Рис. 1.10. Однолонжеронне крило

1 – лонжерон; 2 – передня повздовжня стінка; 3 – пояс лонжерона; 4 – задня повздовжня стінка; 5 – елерон; 6 -

В дволонжеронному крилі (рис. 1.11) передній лонжерон зазвичай розташовується на 15-25% хорди, а задній – на 60-70%. Максимальна висота профілю у такого крила не використовується для раціонального розподілення матеріалу в конструкції, тому маса дволонжеронного крила при рівних параметрах завжди більша, ніж в однолонжеронного. На кручення працює контур, утворений обшивкою та стінками переднього та заднього лонжеронів [4].

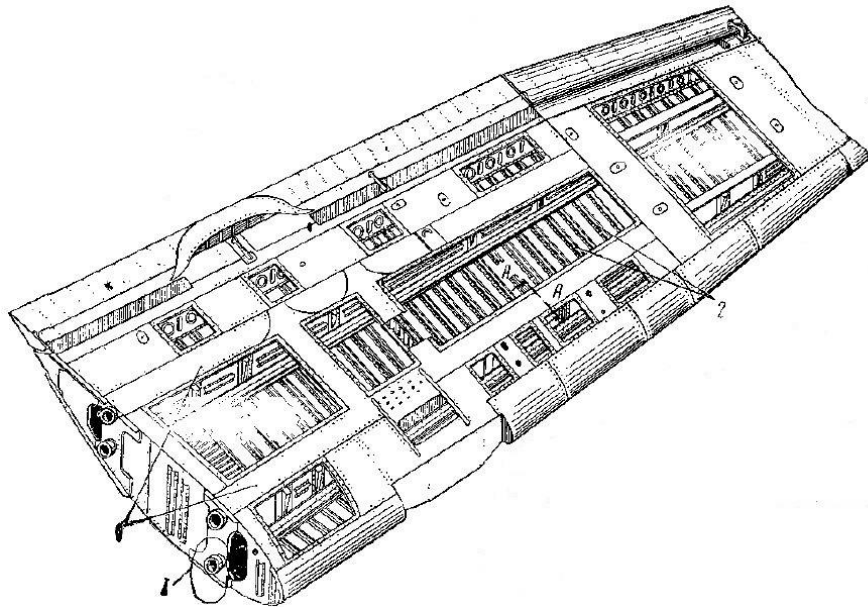


Рис. 1.11. Дволонжеронне безстрингерне крило з тонкою обшивкою
1 – лонжерони; 2 – нервюри.

Типовою конструкцією багато лонжеронного крила може слугувати трьохлонжеронне крило. Лонжерони представляють собою тонкостінні балки двотаврового перерізу змінних розмірів. Пояси лонжеронів мають складний профіль, у якого з зовнішньої сторони зроблені пази для кріплення обшивки. Стінки лонжеронів в місцях кріплення до них нервюр підкріплені стійками. В кореневій частині крила на лонжеронах є стикові вузли. До заднього лонжерона кріпиться хвостова частина крила, вузли підвіски елеронів та закрилків. В нервюр, виконаних з листів, верхня та нижня кромки відігнуті для кріплення обшивки. Нервюри складаються з чотирьох частин. Обшивка кріпиться заклепками до поясів лонжеронів, нервюр, та стрингерам [4].

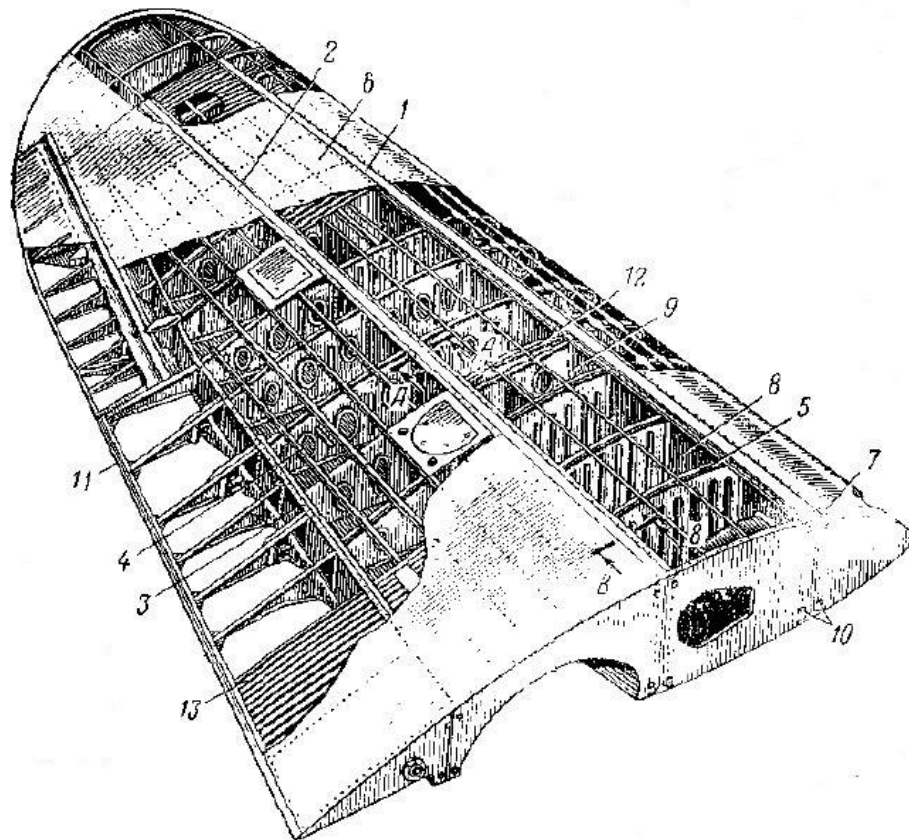


Рис. 1.12. Трьохлонжеронне крило з тонкою обшивкою

1, 2, 3 – лонжерони; 4 – стрингер; 5 – нервюра; 6 – обшивка; 7 – профіль пояса лонжерона; 8 – стінка лонжерона; 9 – стійка стінки лонжерона; 10 – стикові вузли; 11 – задній (хвостовий) стрингер; 12 – посилена нервюра; 13 – гофра.

Кесонні крила.

Збільшення швидкостей літаків підвищило вимоги і до конструкції їх несучих поверхонь. Тонка гладка обшивка не забезпечувала достатньої жорсткості та міцності на всіх режимах польотів, в результаті чого з'явилась необхідність збільшити товщину обшивки і посилити її стрингерами. Водночас, щоб не збільшувати масу конструкції виникло питання про більш ефективне застосування обшивки в роботі крила при згині [4].

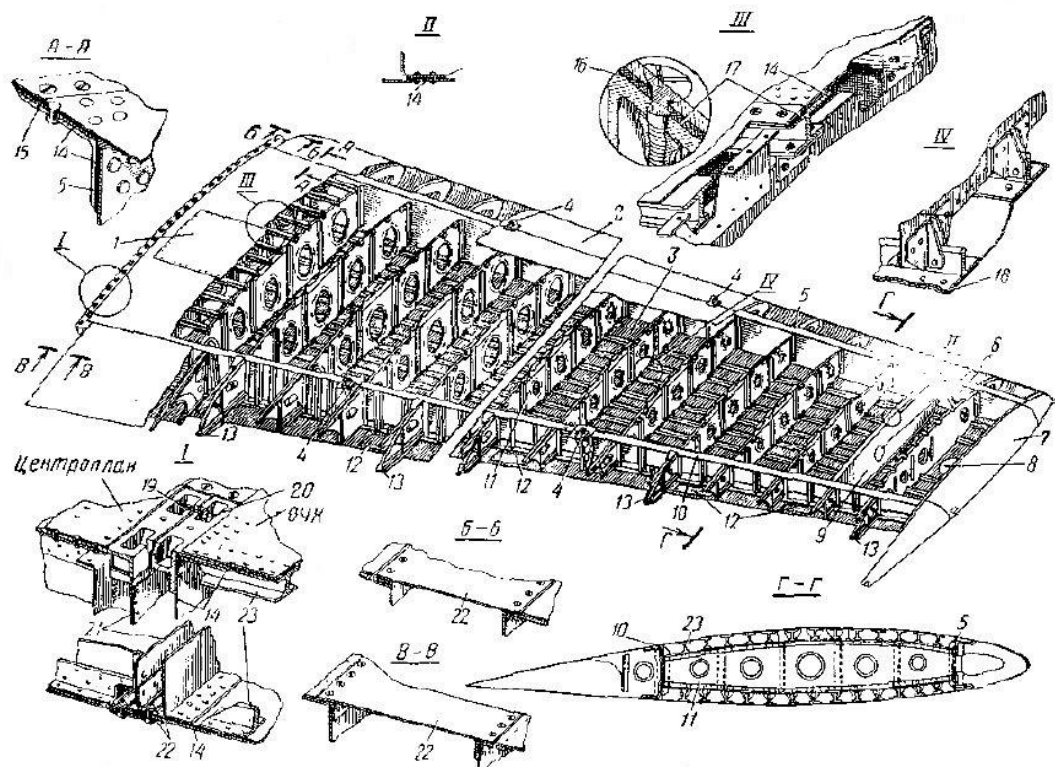


Рис. 1.13. Знімна частина крила кесонної конструкції

1 – знімна панель; 2 – носок; 3 – бак (паливний відсік); 4 – транспортувальні вузли; 5 – передній лонжерон; 6 – люки; 7 – кінцевий обтічник; 8 – люк; 9 – торцева нервюра паливного відсіку; 10 – задній лонжерон; 11 – нервюра; 12 – люки; 13 – вузли підвіски елеронів; 14 – ущільнювальна плівка; 15 – протиобліднювач; 16 – кільце; 17 – прокладка; 18 – ущільнювач; 19 – самоконтрольна гайка; 20 – болт; 21 – рознімна нервюра; 22 – з’єднувальна стрічка; 23 – стрингер.

В однолонжеронному крилі з задньою повздовжньою стінкою чи в дволонжеронних конструкціях обшивку посилювали стрингерами стрингерним набором чи гофром, щоб при зменшенні перерізів поясів лонжеронів забезпечити сприймання згинаючого моменту. Така конструкція крила отримала назву кесонної [4].

Стрингери та обшивка мають високі критичні напруження та ефективно беруть участь в роботі крила на згин, сприймаючи більшу частину згинаючого моменту [4].

Верхня панель крила в найбільш важких випадках працює на стиск, тому її роблять з більш товстою обшивкою, ніж нижню. Стрингерний набір верхньої панелі іноді заміняють гофром [4].

Моноблочні крила

В моноблочному крилі (рис. 1.14) відсутні потужні повздовжні елементи крила – лонжерони, тому згин та кручення сприймаються обшивкою та стрингерами [4].

Замість порівняно товстої обшивки та частого повздовжнього та поперечного наборів в конструкції крила можна використовувати жорсткі панелі, які ефективно працюють при згині та крученні крила [4].

Поперечна сила сприймається двома, трьома чи більше повздовжніми стінками, та частково обшивкою [4].

Крутний момент сприймається обшивкою та повздовжніми стінками, які утворюють разом з обшивкою замкнуті контури, які в свою чергу забезпечують крилу жорсткість на кручення. В тонкого моноблочного крила може бути доцільним застосування потужних панелей без стрингерів та нервюр [4].

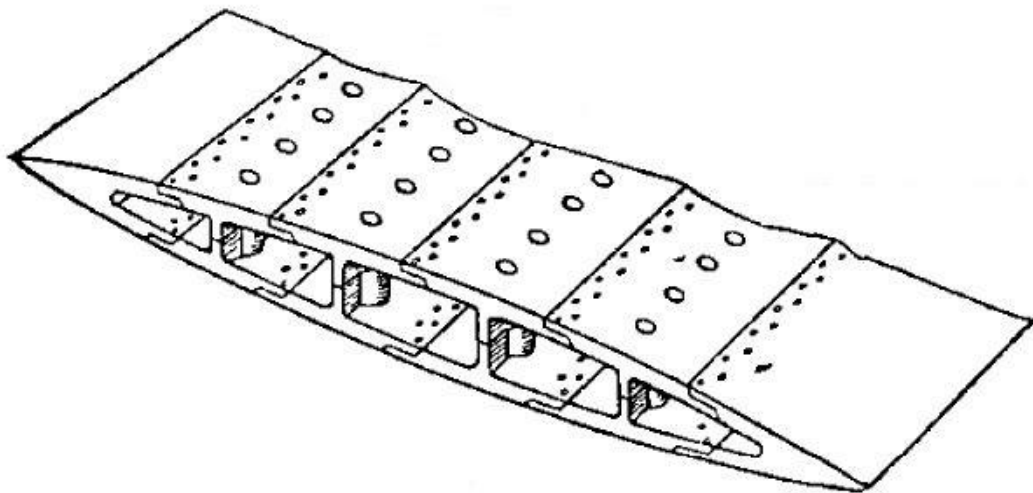


Рис. 1.14. Крило моноблочної конструкції

Напруження в перерізах моноблочного крила розраховується так само, як у кесонного крила.

1.3. Оперення.

Горизонтальне оперення (ГО) призначено для забезпечення повздовжньої стійкості та керованості літака, а вертикальне оперення (ВО) – шляхову [1].

Розташування частин оперення суттєво впливає на ефективність та масу оперення. В зоні супутнього струменя, особливо за крилом, мають місце великі скоси потоку. Це зменшує ефективність оперення при його попаданні в таку зону. Дуже небезпечна наявність вихрів в такій зоні, що можуть викликати вібрацію частин оперення. На рисунку 1.15 показані всі схеми розташування частин оперення [1].

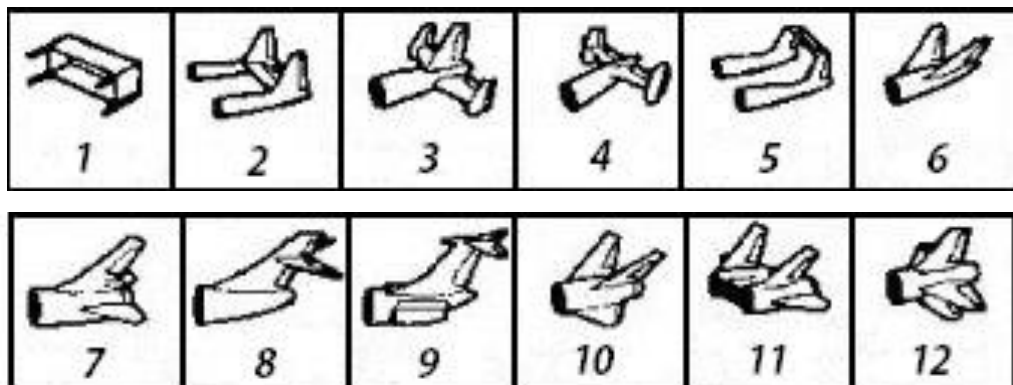


Рис. 1.15. Схеми розташування частин оперення.

1 – квадратне; 2 – Н-подібне; 3 – багатокільове; 4 – рознесене двокільове; 5 – Л-подібне; 6 – V-подібне; 7 – нормальне; 8 – з середньорозміщеним ГО; 9 – Т-подібне; 10 – Y-подібне; 11 – двокільове з СПГО; 12 – хрестоподібне.

Рациональним розташуванням ГО може бути винесення його вниз (нижню частину фюзеляжу) чи вгору від супутнього струменя (Т-подібне оперення), або змінивши аеродинамічну схему, що дозволить винести ГО перед крилом (качка) чи взагалі відмовитись від нього (літаюче крило, безхвостка) [1].

При Т-подібній схемі (рис. 1.16) оперенні ГО винесене вгору від супутнього струменя. В такій схемі ГО являється кінцевою шайбою для ВО, збільшуючи його ефективне подовження. Це дозволяє зменшити площу кіля та його масу. Однак конструкція оперення ускладнюється, а передача навантажень від ГО через киль на фюзеляж потребує суттєвого посилення кіля та збільшення

його маси. Ця схема отримала широке застосування на пасажирських літаках [1].

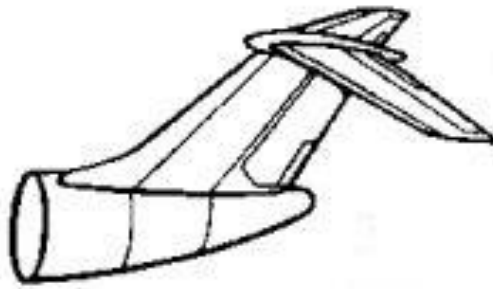


Рис. 1.16. Т-подібне оперення.

В схемі качка (рис. 1.17) можливо отримати перевагу за рахунок зменшення площі крила та його маси, так як підйомна сила ГО при балансуванні літака додається до підйомної сили крила (в класичній схемі сила на ГО зменшує підйомну силу літака). В такій схемі можна також отримати перевагу в аеродинамічній якості. Однак ГО, яке розташоване спереду, затіняє крило, що вимагає більших значень підйомної сили на злітно-посадкових режимах, а також великі втрати на балансування зменшують переваги цієї схеми [1].

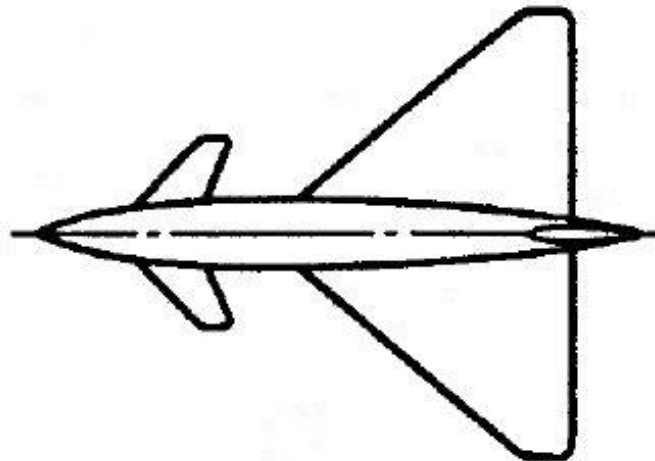


Рис. 1.17. Аеродинамічна схема «качка».

Стабілізатор може бути роз'ємним та нероз'ємним. Роз'ємний стабілізатор складається з двох половин – лівої та правої, які прикріплюються до

фюзеляжу чи кіля. Кіль та роз'ємний стабілізатор представляють собою консольні балки, а не роз'ємний стабілізатор – двохопорну палку з двома консолями. Конструктивно-силові схеми стабілізатора та кіля мало чим відрізняються від конструктивно-силових схем крила. Для КСС стабілізатора та кіля може бути застосована та сама класифікація, що і для схем крила. Характер навантаження та робота окремих силових елементів аналогічні навантаженню та роботі цих елементів в крилі. Конструкція силових елементів – лонжеронів, стрингерів, нервюр та обшивки нічим не відрізняється від конструкції цих елементів в крилі [3].

Висновок

В цьому розділі було проведено аналіз конструктивно-силового набору літака. Спершу був проведений аналіз фюзеляжу та розглянуто основні сили, що діють на фюзеляж, конструктивно-силові схеми фюзеляжів та їхні основні силові елементи. За конструктивно-силовою схемою фюзеляжі ділять на фермові та балкові, а балкові в свою чергу на лонжеронні, стрингерні та безстрингерні в залежності від участі в силовій роботі окремих елементів. Проведено порівняльну характеристику різних типів балкових фюзеляжів.

Також була розглянута конструктивно-силова схема крила, основні силові елементи та типи конструктивно-силових схем. Основною ознакою, за якою характеризують тип конструкції крила є характер роботи та ступінь використання обшивки чи повздовжнього набору при згині та крученні крила. За цією ознакою виділяють лонжеронні, кесоні та моноблочні конструктивно-силові схеми крил.

Останньою була розглянута конструктивно-силова схема оперення літака, розташування його частин, що дуже впливає на ефективність та масу оперення. Конструктивно-силові схеми стабілізатора та кіля мало чим відрізняються від конструктивно-силових схем крила. Для КСС стабілізатора та кіля може бути застосована та сама класифікація, що і для схем крила.

2. Огляд типових шпангоутів, розробка класифікатора.

Поперечний набір каркаса фюзеляжу складають шпангоути. За своєю конструкцією та призначенням в забезпеченні міцності вони можуть бути нормальними, тобто надають форму фюзеляжу та підкріплюють обшивку, та посиленими, які виконують ті ж функції, але додатково сприймають сконцентровані сили прикладені до фюзеляжу [3].

Нормальні шпангоути складають основний поперечний набір фюзеляжу, їх кількість та розташування визначаються досягаючи найбільш ефективного використання обшивки для забезпечення міцності фюзеляжу. Нормальні шпангоути виготовляються з пресованих, або гнутих по контуру перерізу фюзеляжу профілів і складаються, як правило з декількох частин [4].

В загальному випадку зовнішні сили, які навантажують шпангоути врівноважуються на обшивці потоком розподілених дотичних зусиль. В перерізах самих шпангоутів при цьому виникають внутрішні згинальний момент, поперечна та осьова сили [5].

Міцність шпангоутів та їх параметри, як правило, визначають з умови навантаження їх згинальним моментом. Разом зі шпангоутами завжди працює обшивка, жорсткість якої на згин порівняно зі шпангоутами близька до нуля [5].

Нормальні шпангоути.

Основне призначення нормальних шпангоутів складається з сприйняття місцевого аеродинамічного навантаження та забезпечення відповідної опори для стрингері та обшивки. Під дією аеродинамічних навантажень шпангоути є врівноваженими і в основному працюють на розрив. Тому в більшості випадків для не швидкісних, невеликих літаків нормальні шпангоути детально не розраховуються, а вибираються з конструктивних поглядів. Але для великих літаків та для фюзеляжів великого подовження параметри шпангоута потрібно

розрахувати. Розраховують також шпангоути для фюзеляжів складної форми [5].

Характерні перерізи нормальних шпангоутів показані на рис. 2.1.



Рис. 2.1. Перерізи нормальних шпангоутів.

Стандартами передбачено різноманітні форми вирізів, які відповідають розмірам і типу стрингера. Для підвищення витривалості та попередження раннього утворення тріщин у вирізах всіх типів треба робити якнайбільші радіуси вершин вирізу чи виконувати їх у вигляді півкола з відбортунням. Отвори полегшення, які роблять в стінках шпангоута чи в складених стінках, бажано розміщувати таким чином, щоб не орієнтувати ймовірне направлення тріщини на гострі кути вирізів [5].

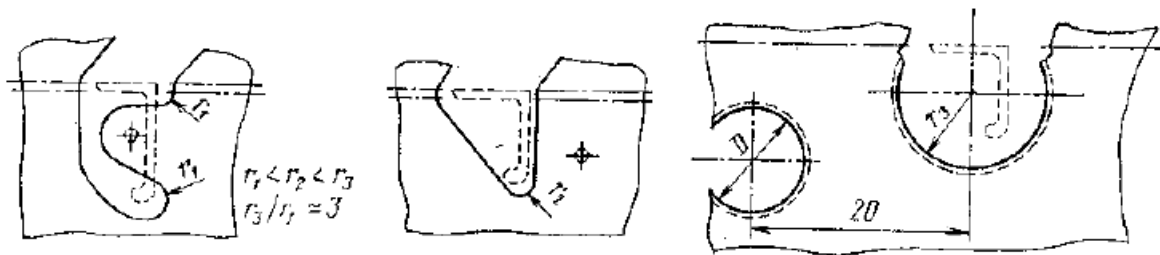


Рис. 2.2. Типові вирізи в стінках шпангоута

2.1. Виготовлення шпангоутів.

Обробка тиском. У виробництві літальних апаратів обробкою тиском отримують деталі з жароміцних і магнітних сплавів штамповкою на молотах і пресах; отримання обшивок ЛА одинарної і подвійної кривизни згинанням, прокаткою, обтяжкою або обтяжкою з розтягом; листових деталей складних форм витяжкою; безшовні тонкостінні оболонки постійної і перемінної товщини з оребренням розкаткою та пресуванням [6].

Планери літаків мають до 70% деталей з листів, профілів та труб; їх виготовляють з використанням операції **листового штампування**. Обсяг таких робіт становить нині 10...12 % усієї трудомісткості виготовлення літака. Основними групами деталей є зовнішні обшивки різної складності форм, одинарної та подвійної кривизни з розмірами до 10×2 м і більше, товщиною до 6...8 мм; монолітні панелі, стрингери, шпангоути, нервюри. Матеріал: алюмінієві сплави Д16, В95, АМг6, сталі Х18Н10Т, титанові сплави. Виготовлення деталей методом штампування полягає у пластичному деформуванні зовнішніми навантаженнями плоских листових та прямолінійних профільних та трубчастих заготовок, у результаті якого забезпечується їх формозміна [6].

Обробка різанням. Механічна обробка або обробка різанням широко застосовується при виготовленні авіаційних деталей, а також при доробці вузлів і агрегатів у місцях їх з'єднання. Основними групами деталей, що виготовляються механічною обробкою, є (рис. 34): монолітні панелі, нервюри, шпангоути, лонжерони, деталі та вузли шасі та гідравлічних підйомників, окремі елементи силового набору: стрингери, полиці лонжеронів та нервюр, фітинги, стикувальні вузли та гребінки, підмоторні рами тощо. Деталі, що підлягають механічній обробці, виконані з алюмінієвих, магнієвих і титанових сплавів, високоміцних, головним чином, хромонікелевих сталей. Заготовки для деталей поставляються у вигляді поковок та штамповок, пресованих профілів та панелей, у вигляді виливків та сортового прокату. Найбільший обсяг механічної обробки посідає фрезерування (близько 75%). Стрингери, лонжерони, монолітні панелі та інші елементи силового набору зазнають фрезерування; точіння застосовується для деталей шасі, підйомників; свердління - для утворення отворів під заклепки та болти. Для обробки деталей поздовжнього та поперечного силового набору в даний час застосовують поздовжньо-фрезерні верстати, на яких можна обробляти деталі довжиною до 20-30 м [6].

Кожному виду профілю відповідатиме ескіз створений в файлі типу .prt з відповідною назвою. Вибравши конкретний профіль використовуючи інструмент для побудови шпангоуту, відповідні розміри ескізу зміняться на розміри вибраного профілю зазначені в базі даних.

Аналогічний класифікатор буде для профілів для вирізів під стрингер в шпангоуті. Додавати профілі можна буде за допомогою додаткового інструменту в системі Siemens NX. Проте, щоб додати профіль в бібліотеку не потрібно задавати розміри, треба буде створити профіль та обрати його в меню інструмента.

Висновок

В цьому розділі було оглянуто типи шпангоутів. Їх розрізняють за конструкцією та призначенням та ділять на нормальні та посилені. Нормальні надають фюзеляжу форму та підкріплюють обшивку. Також розглянуто основні перерізи нормальних шпангоутів. До них належать двотавровий, тавровий, кутиковий, швелерний, зетподібний перерізи, а також швелерний переріз з компенсатором.

Далі розглянуто форми вирізів під стрингер. В залежності від розмірів і типу стрингера вони бувають різноманітних форм.

Останньою частиною розділу була розробка класифікатора для майбутнього інструмента. Він буде складатись з двох файлів MS Excel. Перший буде представляти собою бібліотеку перерізів шпангоутів, поділену по їхнім видам. Другий буде містити ескізи вирізів під стрингер. Буде створено два додаткових інструменти для зручного додавання нових елементів класифікатора для подальшого використання в інструменті для побудови шпангоута.

3. Методика побудови кожного класу шпангоутів у CAD системі Siemens NX.

Побудова шпангоутів будь-якого класу здійснюється на основі базової конструктивної схеми (БКС) (рис. 3.1), до якої належать майстер-геометрія обшивки, площина шпангоуту, а також майстер-геометрія стрингерів у випадках коли необхідно виконати вирізи під них в шпангоуті. Часто допоміжними елементами для побудови шпангоуту є інші конструктивні елементи літака до яких кріпиться шпангоут.

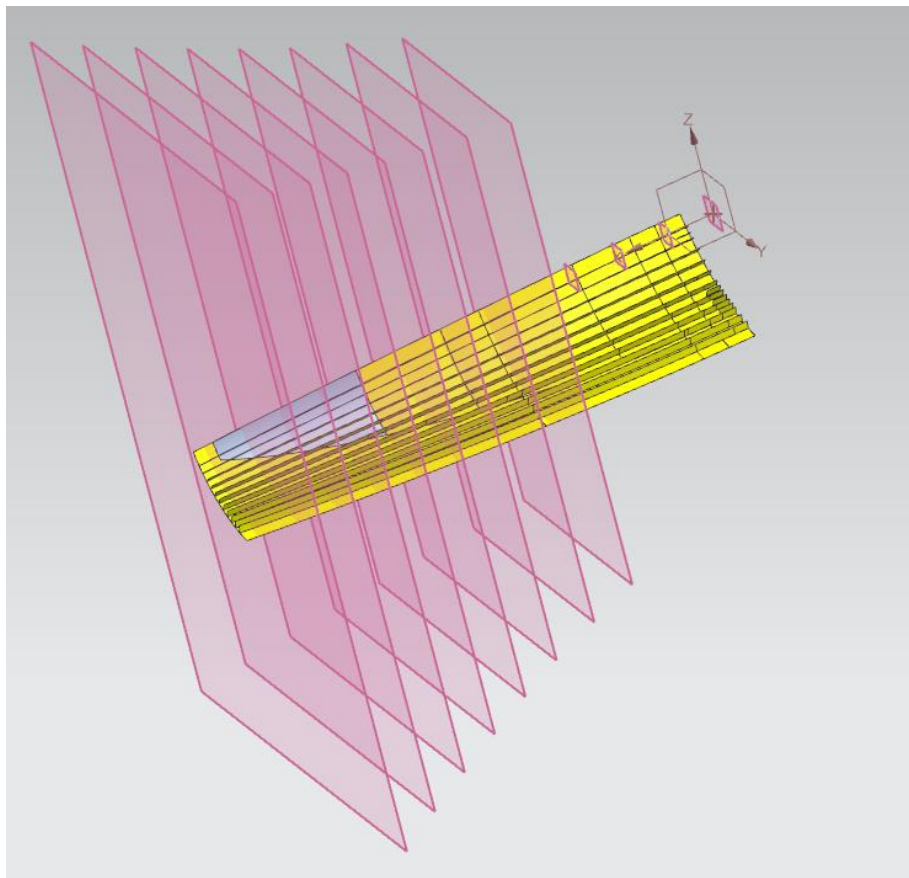


Рис. 3.1. Приклад БКС літака.

Існує безліч шляхів побудови шпангоуту в CAD системі Siemens NX. Як правило, для цього застосовуються прості функції, до яких належать створення точки, лінії, координатної площини, обмеженої поверхні, кривої перетину, ескізу, фаски та інших простих елементів, а також функції «Витягування», «Товщина», «Лінійчата поверхня», «Заокруглення ребра», «Обрізка кривої», «Обрізка поверхні», «Обрізка тіла», «Зшивка» та багато інших (рис. 3.2). Також

застосовують складніші функції такі як «Складний фланець», «Авіаційне відбортування» та інше (рис. 3.3).

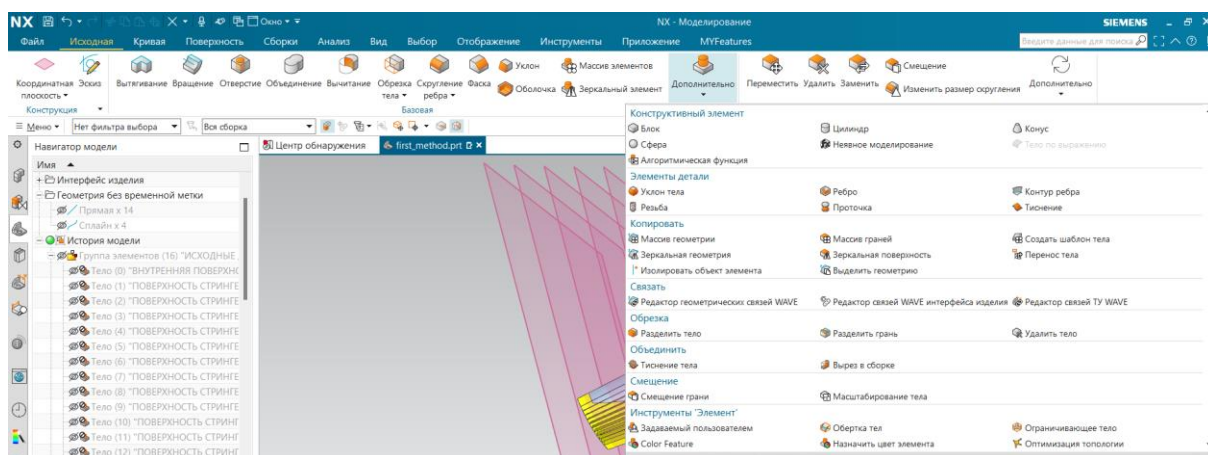


Рис. 3.2. Основні інструменти для 3D-моделювання твердих тіл

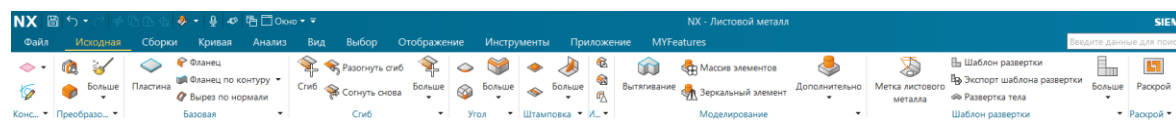


Рис. 3.3. Основні інструменти для листових тіл

Розглянемо найпоширеніші та найдоцільніші варіанти побудови шпангоуту в CAD системі Siemens NX на прикладі зет подібного профілю.

Перший спосіб. Маючи поверхню обшивки, площину шпангоуту знаходимо лінію по якій буде йти шпангоут за допомогою функції «Крива перетину» (рис. 3.4).

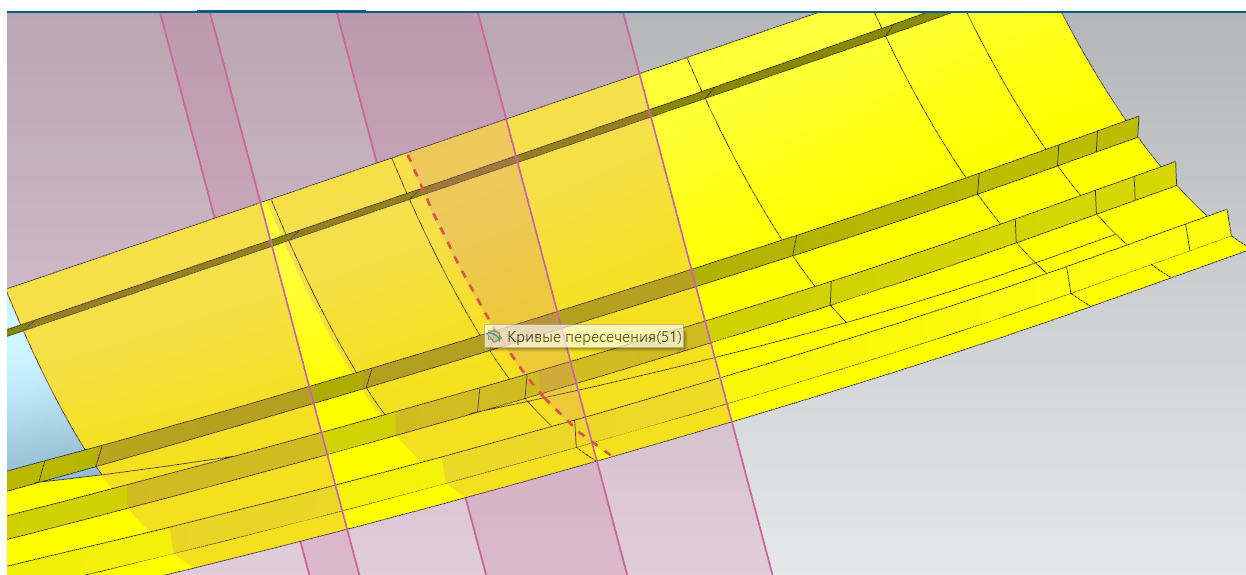


Рис. 3.4. Крива перетину.

Після цього знаючи висоту шпангоуту та його повздовжні межі(часто між двома стрингерами) можемо отримати поверхню обмежену чотирма лініями (рис. 3.5).

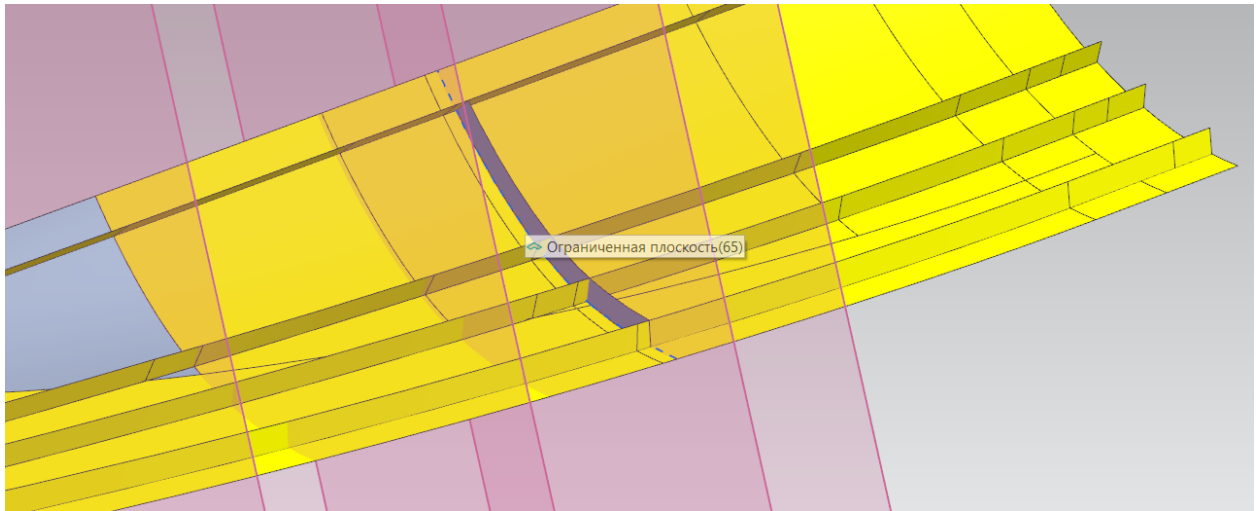


Рис. 3.5. Обмежена поверхня.

Тепер за допомогою функції «Пластина» задаємо товщину стінки шпангоута (рис. 3.6).

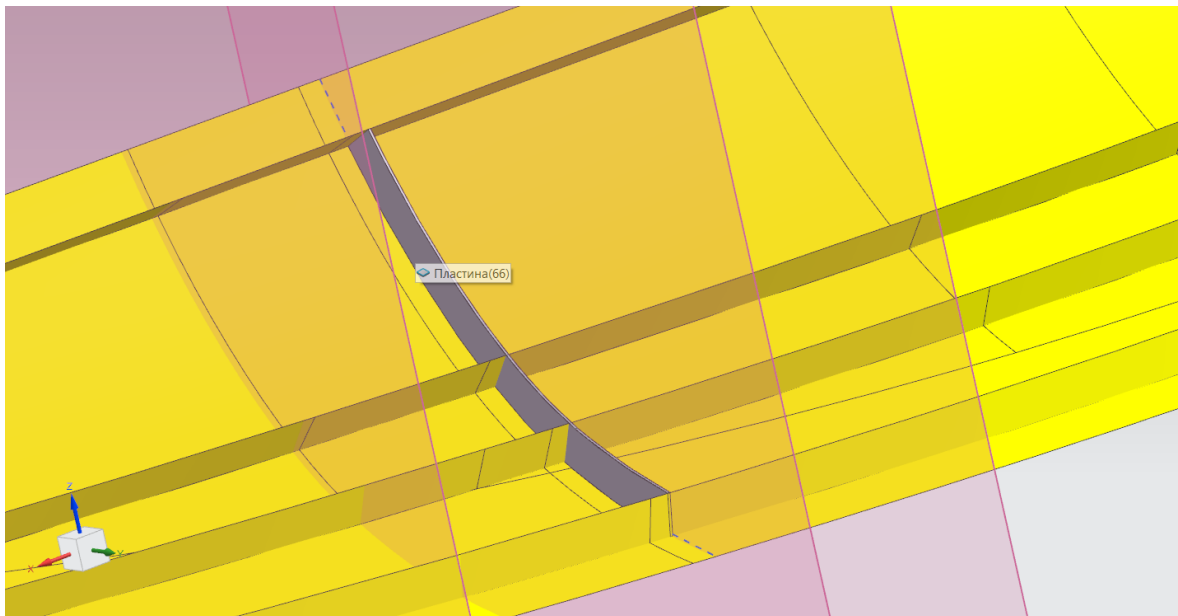


Рис. 3.6. Пластина.

Після цього створюємо ескіз з профілями вирізів під стрингери та за допомогою функції «Витягування» в режимі «Віднімання» утворюємо самі вирізи в пластині (рис. 3.7).

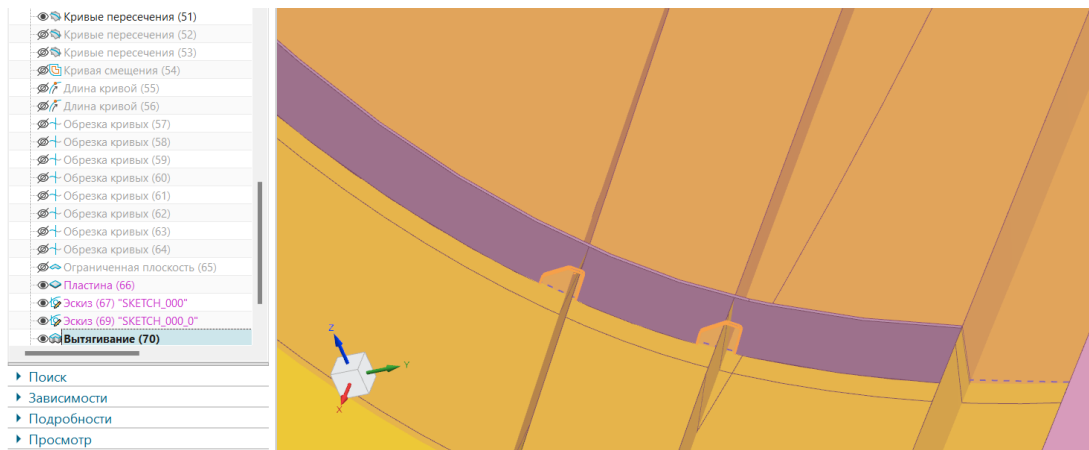


Рис. 3.7. Утворення вирізів під стрингер.

Наступним кроком буде утворення полиць шпангоута. Це можна зробити за допомогою функції «Складний фланець» (рис. 3.8). Цій функції необхідно задати ребро бази, поверхню обшивки, параметри заокруглення, довжину, а також за необхідності інші параметри.

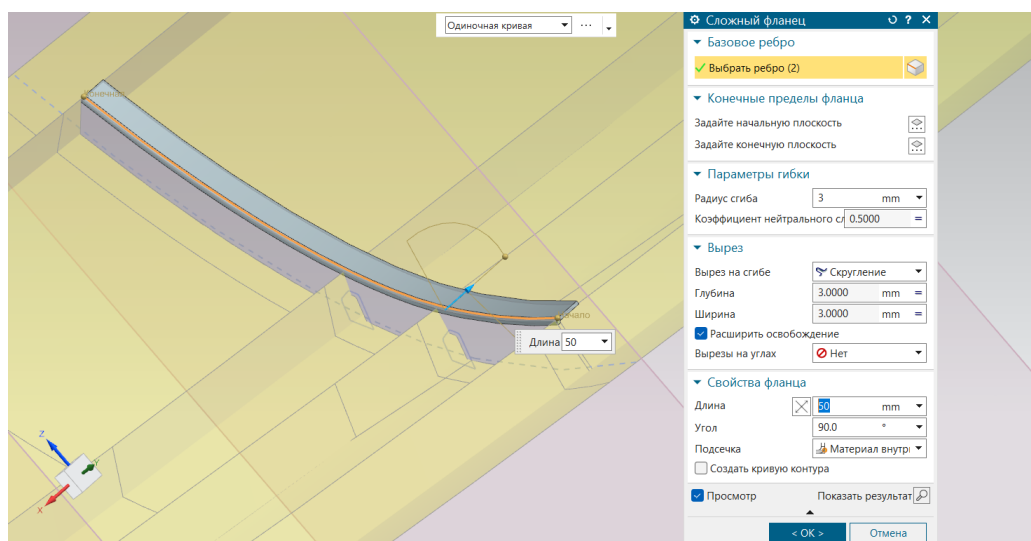


Рис. 3.8. Створення полички шпангоута

Фінальним кроком буде створення заокруглень, фасок та отворів в шпангоуті, а також додаткових побудов залежно від інших особливостей шпангоута.

Другий спосіб доволі схожий на перший, але виглядає більш простим. Початок ідентичний: утворюємо криву перетину (рис. 3.4) площини шпангоута з поверхнею обшивки та будуємо обмережу поверхню по 4 лінія(рис. 3.5). Наступним кроком утворюємо полиці шпангоута. В розділі «Поверхня фланця»

є функція «Продовжувати за законом», яка дозволить утворити поверхню полиці на основі стінки під потрібним кутом та необхідною довжиною (рис. 3.9).

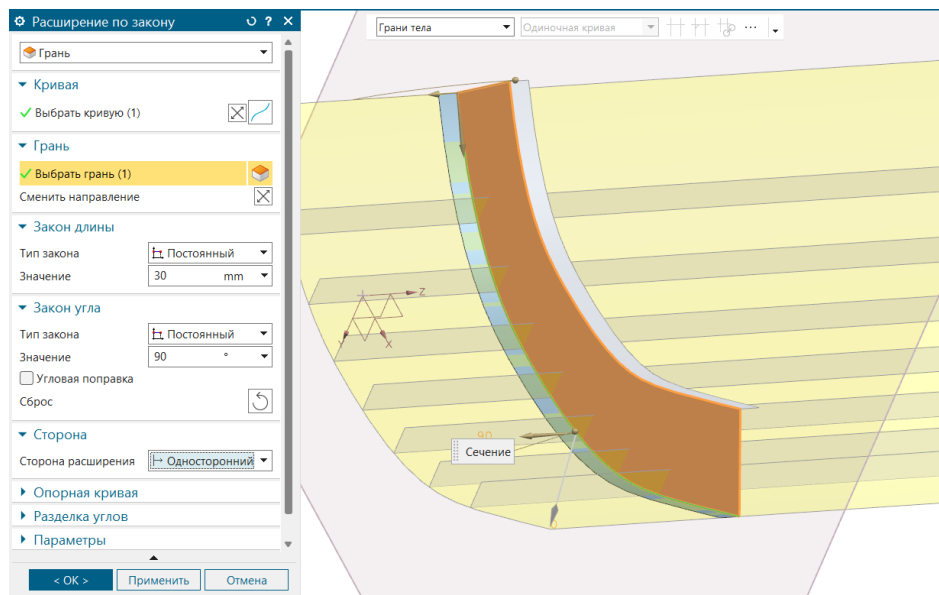


Рис. 3.9. Створення полицки.

Після того як готова поверхня шпангоуту за допомогою функції «Товщина» задаємо товщину поверхонь шпангоуту (рис. 3.10).

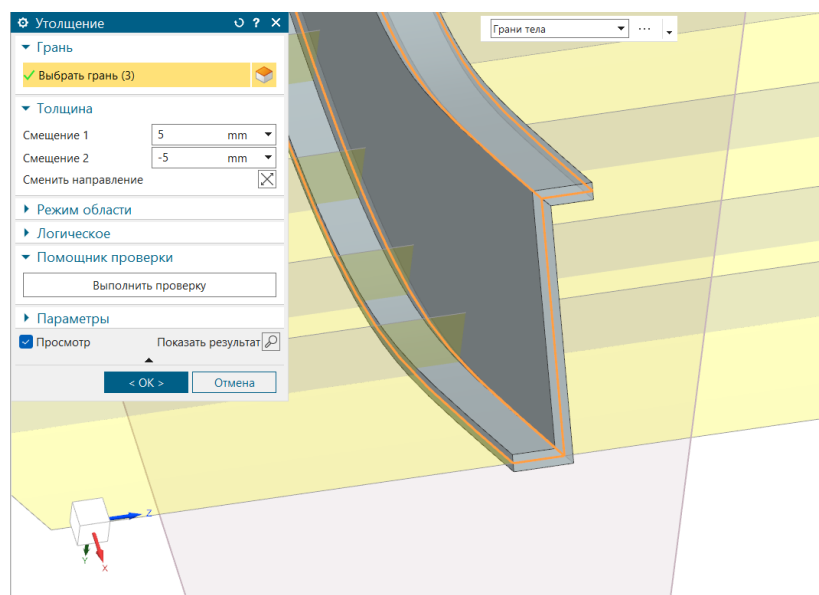


Рис. 3.10. Створення тіла шпангоута.

Далі утворюємо вирізи під стрингер за допомогою функції «Витягування» ескізів вирізів. Останнім кроком буде створення фасок, заокруглень та отворів в шпангоуті.

Третій спосіб найпростіший, але в складних випадках потребуватиме додаткових побудов та доопрацювань над шпангоутом. Спершу отримуємо криву перетину площини шпангоута та поверхні обшивки. Після цього будуємо ескіз профілю шпангоута та користуючись функцією «Замітання» будуємо зразу тіло шпангоута (рис. 3.11).

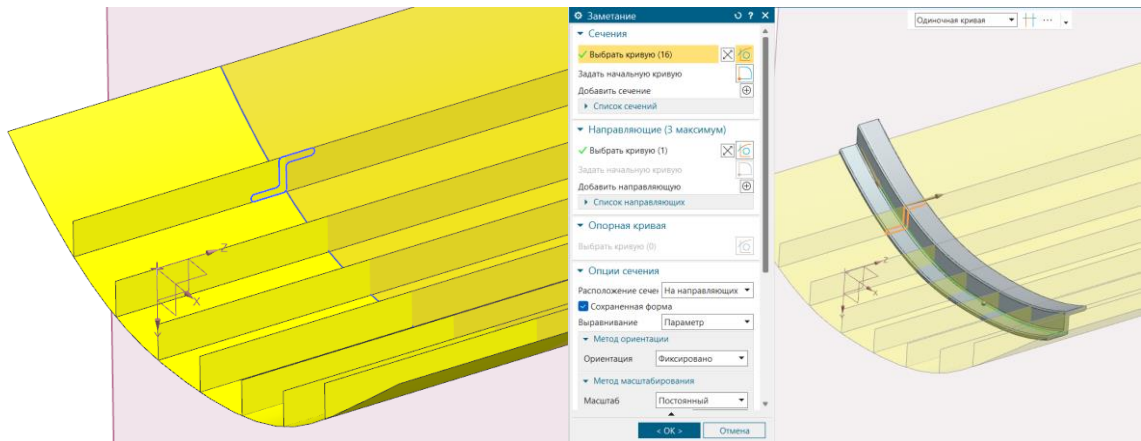


Рис. 3.11. Ескіз профілю шпангоута та створення тіла шпангоута

Далі створюємо ескіз з профілями вирізів під стрингер та утворюємо вирізи за допомогою функції «Витягування». Після того, як основна частина шпангоуту побудова потрібно створити заокруглення, фаски та отвори в шпангоуті.

Останній спосіб відрізняється від попередніх і основна його відмінність полягає в тому що, контур стінки шпангоуту задається ескізом (рис. 3.12) та потім витискається або за допомогою функції «Пластина», або «Товщина».

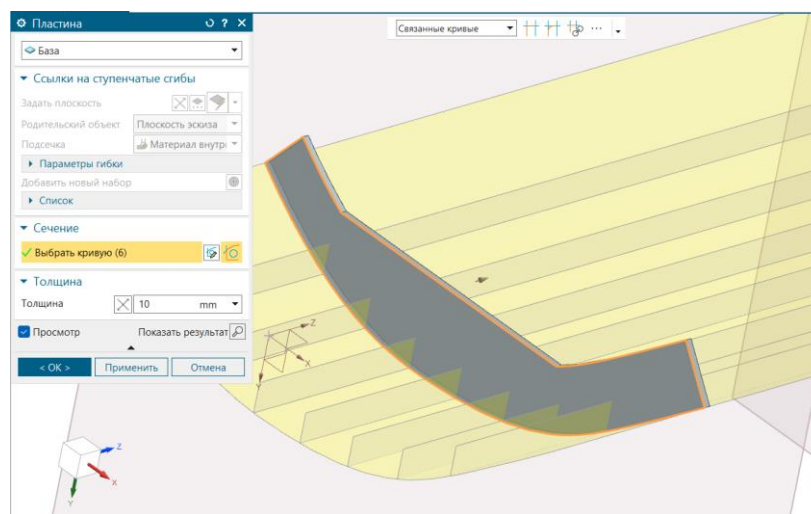


Рис. 3.12. Створення стінки шпангоута.

Це дозволяє задавати складні форми шпангоуту. Після цього краще утворити полиці шпангоуту, а вже потім моделювати вирізи під стрингер способом представленим вище.

Для охоплення більшої кількості можливих варіантів шпангоуту в майбутньому інструменті буде можливість застосувати третій та четвертий метод. Вони є найоптимальнішими та в різних ситуаціях один з них буде мати вигравш перед іншими в швидкості та зручності побудови шпангоута. Коли необхідно побудувати шпангоут з заданим профілем та нескладною формою стінки, то доцільно використовувати третій метод. Якщо ж задача полягає в побудові більш складного та нетипового шпангоуту то четвертий метод дасть більше можливостей

Висновок

В цьому розділі було розглянути методи побудови шпангоута в САД-системі Siemens NX. Побудова шпангоута розпочинається з внесення в робочу деталь допоміжної геометрії до якої має належати поверхня обшивки, площина ескізу, поверхні стрингерів та інші елементи, які будуть визначати форму шпангоута.

Всього було розглянуто чотири методи побудови шпангоута. Вони мають спільні риси, проте можуть використовувати різні інструменти, та в різній послідовності. Також вони можуть відрізнитись фінальними характеристиками моделі шпангоута, наприклад, одні інструменти виконують тіло шпангоута як «тверде тіло», а інші – як «листова тіло».

В результаті було обрано два методи які дадуть можливість побудувати якомога більше варіантів шпангоутів. Один з них використовуватиме базу даних профілів шпангоута та буде простішим у використанні. Інший матиме більше можливостей, проте буде розрахований на досвідченішого працівника.

4. Розробка програми для створення шпангоутів.

Інструмент для створення шпангоутів буде розроблений для CAD системи Siemens NX версії 2212 та буде доступний на панелі інструментів у верхній частині екрану. Цей інструмент використовує бібліотеку NXOpen для виконання функції NX та Microsoft.Office.Interop.Excel для взаємодії з базою даних профілів шпангоута та вирізів під стрингер оформлених у вигляді таблиць MS Excel.

4.1. NXOpen

NXOpen – це набір інструментів API (Application Programming Interface), що дозволяє здійснювати гнучку інтеграцію користувацьких застосунків з NX за допомогою відкритої архітектури, яка може бути задіяна сторонніми розробниками, клієнтами та користувачами. Метою застосування NXOpen є автоматизація складних та одноманітних задач.

Переваги NXOpen:

- Можливість кастомізації NX для конкретних вимог галузі чи технології.
- Зменшення часу виводу продукту на ринок завдяки автоматизації складних та одноманітних процесів.
- Відсутність необхідності спеціального навчання користувачів через повну відповідність інтерфейсу NX.
- Повний функціонал NX.
- Розвивається одночасно з інтерактивними функціями.
- Використовується розробниками Siemens PLM.

NXOpen має широкий вибір мов програмування до яких належать: C#, VB.NET, Java, Python, C++. Записані макроси можна легко подивитися у вигляді програмного коду, що дозволяє використовувати найскладніші функції NX.

4.2. Інтерфейс інструменту

Інтерфейс інструменту розробляється за допомогою додатка NX «Розробка користувацького інтерфейсу» (рис. 4.1).

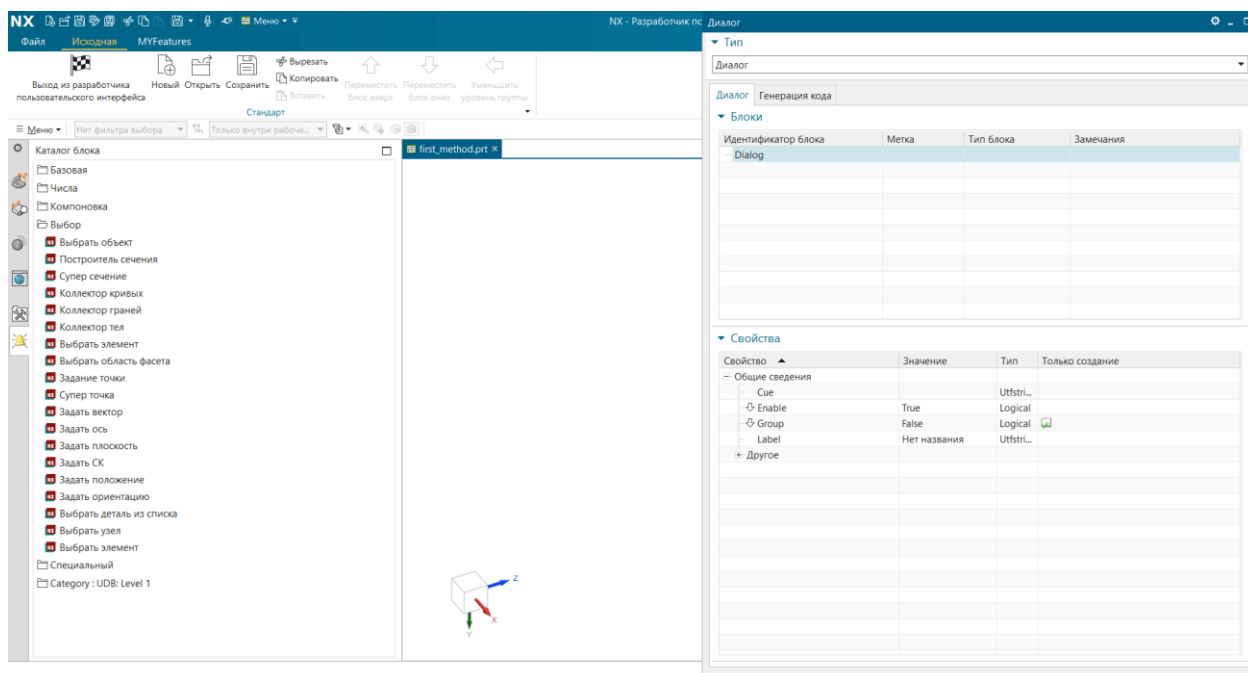


Рис. 4.1. Додаток NX «Розробка користувацького інтерфейсу»

Ліворуч розташовано різноманітні блоки для загальних цілей (кнопка, рядок, перемикач, обрати файл і т. д.), а також спеціальних таких як: «Вибір елемента» «Задати вектор», «Колектор тіл» та багато іншого. Праворуч розміщується дерево обраних блоків, де можна змінювати їх характеристики. Для зручного перемикання між методами побудови шпангоуту я використав спосіб групування «Вкладки».

Перша вкладка (рис. 4.2) містить метод побудови шпангоуту через задавання потрібного профілю шпангоуту. За допомогою груп, я розділив першу вкладку на основні етапи побудови шпангоуту.

Перша група називається «Профіль шпангоуту». Вона містить два блоки «Перечислення».

Наступна група називається «Розташування шпангоуту», в якій є блок «Вибір елемента» з встановленим фільтром, що дозволяє обирати лише грані чи

поверхні та блок «Задати площину». В першому потрібно обрати поверхню обшивки, а в другому є можливість створити площину всіма можливими способами, що представлені в NX або обрати існуючу.

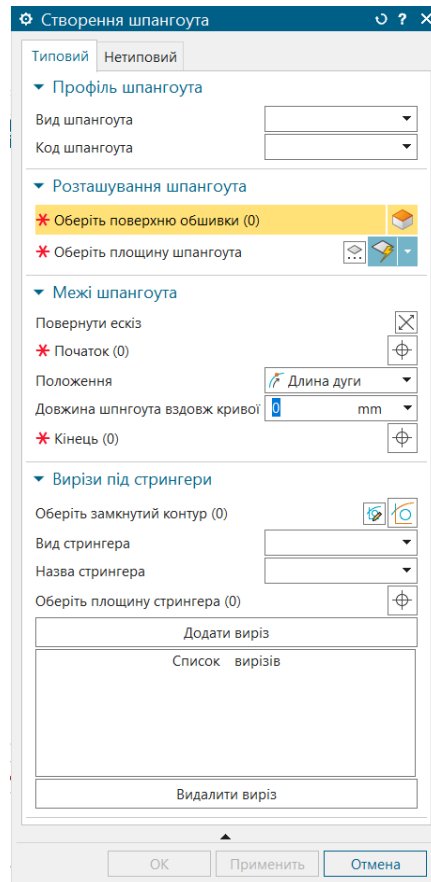


Рис. 4.2. Вкладка №1.

В третій групі, яка називається «Межі шпангоута», розташовано чотири блоки: перший і третій «Вибір елемента», другий блок «Розмір по трасі» та четвертий «Зміна напрямку». Спочатку слід обрати стрингер або іншу поверхню з якої починається шпангоут, а далі є два варіанти: обрати кінцевий стрингер чи задати довжину шпангоуту вздовж траси. В деяких випадках ескіз профілю неправильно розташовується на поверхні тому для таких випадків додано останній блок, який дозволяє повернути ескіз.

В останній групі є можливість створити вирізи під стрингер. Тут є сім блоків, два «Перечислення», «Супер переріз», «Вибір елемента», «Вікно списку» та дві кнопки. Спочатку за допомогою «Перечислення» та «Вибір елемента» обираємо стрингер для якого хочете зробити виріз, а також

поверхню стрингера. Також є можливість обрати або створити ескіз вирізів самостійно. Після вибору ескізу профіля чи профіля вирізу з класифікатора треба натиснути на клавішу «Додати». В блоці «Вікно списку» будуть показані додані вирізи. За допомогою кнопки «Видалити» можна видаляти випадково обрані вирізи.

Друга вкладка має схожість з першою, проте містить більш складну побудову шпангоуту. Ця вкладка містить три групи в яких розташовані блоки.

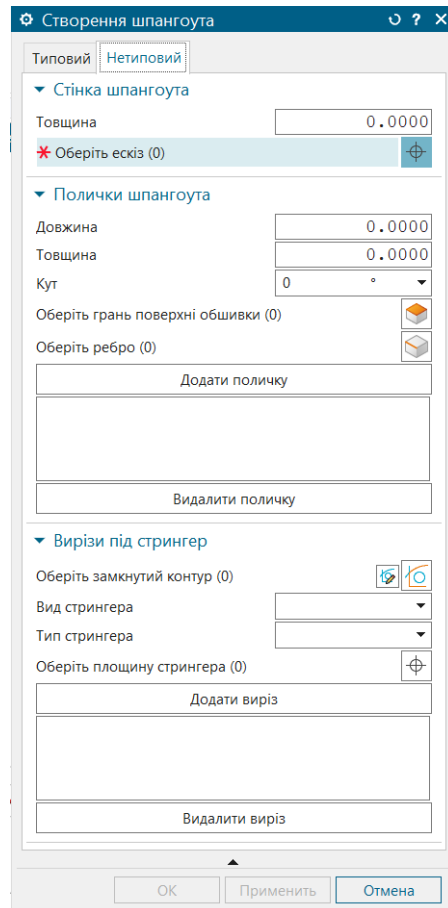


Рис. 4.3. Вкладка №2.

Перша група «Стінка шпангоута» яка містить два блоки «Супер переріз» в якому потрібно задати контур стінки шпангоута та «Число подвійної точності» в якому слід задати товщину стінки. Заданий контур має бути замкнений, а товщина має бути відмінна від 0. Для того щоб змінити напрямок витискання пластини треба змінити знак значення товщини пластини.

Друга група «Полички шпангоута» містить інструменти для побудови полички шпангоута. В цій групі зібрані не абсолютно всі параметри побудови полички, тому що такий інструмент вже є в функціях з широкими можливостями параметризації. Ця група містить вісім блоків такі як: два блоки «Число з подвійною точністю», «Кутовий розмір», «Обрати грань», «Обрати криву», «Вікно списку» та дві кнопки. Спершу слід всі необхідні параметри такі як : довжина, товщина та кут нахилу відносно поверхні стінки, а також сама поверхня стінки та ребро. Задавати ці параметри можна в будь-якому порядку. Після цього треба натиснути кнопку «Додати». Всі додані полички показані в списку. Також за необхідності можна видалити непотрібну поличку за допомогою кнопки «Видалити».

Остання група «Вирізи під стрингер» аналогічна відповідній групі на першій вкладці.

В обох вкладках існують блоки які необхідно заповнити обов'язково, які позначено червоною зірочкою. Неможна заповнювати дві вкладки одночасно. Після того які всі необхідні блоки будуть заповнені з'явиться можливість натиснути кнопки «ОК» та «Применить».

4.3. Розробка інструменту

Отримавши заготовку програми я, за допомогою Visual Studio 2023, створив проект типу «Бібліотека класів (.NET Framework)» та підключив необхідні посилання. Заготовка програми містить в собі структуру за допомогою якої можна виконувати функції в потрібний момент. Наприклад, зразу після запуску програми зчитується файл MS Excel та заповнюється варіанти вибору типів профілів шпангоута. При взаємодії з блоками інтерфейсу також виконуються проміжні функції. Наприклад, після вибору поверхні обшивки, площини шпангоута та початкового стрингера в необхідному місці вставляється ескіз обраного профілю шпангоута.

4.4. Основні функції використовуються в інструменті

Основні функції, що були використані в програмі є так звані будівельники (від англ. builder – будівельник), які застосовуються при побудові всіх елементів в NX. Розглянемо використані в цій програмі «білдери».

Крива перетину.

Функція «Крива перетину» дуже часто використовується при моделюванні деталей. В цій програмі вона буде знаходити криву перетину між площиною шпангоута та обшивкою. Ця крива служитиме направляючою для витягування профілю шпангоута вздовж направляючої.

Оголошення змінної класу «*IntersectionCurveBuilder*» показано на рисунку 4.4. В робочій деталі «*workPart*» ми створюємо цю змінну за допомогою методу *CreateIntersectionCurveBuilder()* (позначено жовтим).

```
NXOpen.Features.IntersectionCurveBuilder intersectionCurveBuilder1 = workPart.Features.CreateIntersectionCurveBuilder(null);
```

Рис. 4.4. Оголошення змінної класу «*IntersectionCurveBuilder*».

Цей метод приймає аргумент типу «*IntersectionCurve*» для редагування існуючого елемента, але так як нам потрібно створити новий, то записуємо «*null*».

Тепер, використовуючи методи та властивості цього класу, слід задати все необхідне для нового елемента, таке як лінійний допуск, перший та другий набір елементів для перетину та інше.

Для того щоб встановити елементи для перетину слід використати властивості показані на рисунку 4.5.

```
intersectionCurveBuilder1.FirstSet.Add(planeSPG.GetSelectedObjects()[0]);  
intersectionCurveBuilder1.SecondSet.Add(casingFace.GetSelectedObjects()[0]);
```

Рис. 4.5. Встановлення елементів для перетину

В перший сет ми додаємо площину шпангоуту, а в другий - поверхню обшивки. В цій властивості є можливість задати об'єкти типу «*TaggedObject*», тобто вибрані користувачем об'єкти.

Часто при використанні класів «*Builder*» значення точності побудови та допусків задається автоматично і не обов'язково їх змінювати, проте краще пересвідчитись та встановити потрібні значення (рис. 4.6).

```
intersectionCurveBuilder1.CurveFitData.Tolerance = 0.01;  
intersectionCurveBuilder1.CurveFitData.AngleTolerance = 0.5;
```

Рис. 4.6. Додаткові значення точності побудови

Після того як ми встановили параметри для побудови елемента нам потрібно перевірити чи достатньо цих параметрів для завершення побудови та у позитивному випадку побудувати елемент (рис. 4.7).

```
if (intersectionCurveBuilder1.Validate())  
    intersectionCurve1 = (NXOpen.Features.IntersectionCurve)intersectionCurveBuilder1.Commit();  
else  
{  
    theUI.NXMessageBox.Show("CreateIntersectionCurve", NXMessageBox.DialogType.Error, "No intersection");  
}  
intersectionCurveBuilder1.Destroy();
```

Рис. 4.7. Завершення побудови.

Метод який перевіряє чи всі необхідні параметри були введені називається *Validate()*. Він не потребує аргументів та повертає змінну типу *bool*.

Метод *Commit()* аналогічній кнопці «Застосувати» та повертає змінну *NXObject*, що після зведення типів присвоюється попередньо створеній змінній типу *NXOpen.Features.IntersectionCurve*.

Метод *Destroy()* звільняє пам'ять після створення елемента.

Ці методи існують у всіх класах «*Builder*», тому частина програмного коду на рисунку 0 є ультимативною та може застосовуватись на будь-якому «білдері». Спочатку перевіряється чи можливо завершити побудову методом *Validate()*. Якщо можливо, то застосовується метод *Commit()* та буде необхідний елемент. В протилежному випадку виводиться стандартне вікно помилки NX. Вкінці в будь-якому випадку звільняється пам'ять змінної «білдера».

Копіювання/вставка.

Функція копіювання/вставка дозволяє переміщати функції (*Feature*) з одної деталі (файл *.prt) в іншу. Вона нам потрібна для того щоб вставляти вибраний користувачем ескіз профілю з існуючого файлу *.prt, де він вже попередньо створений, в робочу деталь, а також додавати нові нестандартні профілі в користувацьку базу даних.

Спочатку треба обрати елементи які потрібно перемістити. Для цього нам треба створити масив типу *NXObject[]* та заповнити його змінними типу *NXOpen.Features.Feature* чи інших «фічерів» (рис. 4.8). В нашому випадку це буде *NXOpen.Features.SketchFeature*.

```
NXOpen.Features.SketchFeature sketch = (NXOpen.Features.SketchFeature)partSK.Features.FindObject(profile);  
NXObject[] features = new NXObject[1] { sketch };
```

Рис. 4.8. Знаходження необхідного профілю та створення масиву

Для того щоб знайти необхідний ескіз профілю потрібно відкрити файл *.prt, де містяться всі типи профілів (змінна «PartSK»), використати метод «FindObject()», аргументом якого є унікальне ім'я ескізу, що відповідає типу профілю обраного користувачем. Після цього створити масив «features» із знайденим об'єктом.

Оголошення змінної класу «CopyPasteBuilder» показано на рисунку 4.9. Метод «CreateCopyPasteBuilder» приймає аргумент типу масиву *NXObject[]*, елементи якого буде переміщено в робочу деталь.

```
NXOpen.Features.CopyPasteBuilder copyPasteBuilder = workPart.Features.CreateCopyPasteBuilder(features);
```

Рис. 4.9. Оголошення змінної класу «CopyPasteBuilder»

Далі зазначаємо додаткові параметри, такі як асоціативність та прив'язка до оригіналу. Це полегшує роботу тим, що нам не потрібно буде мати кожен профіль в базі даних ескізів, а лише типи профілів (двотавр, тавр, кутик та ін.). Це стане можливим лише в тому випадку, якщо ми перенесемо не лише геометрію, а й таблицю виразів прив'язаних до ескізу в якій містяться дані про

відповідні розміри, щоб потім задати потрібні розміри відповідно до профілю, який вибрав користувач (рис. 4.10).

```
copyPasteBuilder.ExpressionOption = NXOpen.Features.CopyPasteBuilder.ExpressionTransferOption.LinkToOriginal;  
copyPasteBuilder.Associative = true;  
sketchSPG = (NXOpen.Features.SketchFeature)copyPasteBuilder.Commit();  
copyPasteBuilder.Destroy();
```

Рис. 4.10. Функція копіювання/вставки.

Після цього, використовуючи методи «Commit()» та «Destroy()» завершуємо виконання функції.

Розміщення ескізу.

Так як функція копіювання/вставка автоматично розміщує ескіз в нулі робочої системи координат то після перенесення ескізу нам потрібно розмістити його в правильному місці. Це відбувається після того як користувач вибере поверхню обшивки, площину шпангоуту та початковий стрингер (поверхню), так як для розміщення ескізу нам потрібно завчасно створити допоміжні елементи. В нашому випадку це буде координатна площина, точка та напрямок, проте існують і інші варіанти прив'язки ескізу.

Площину ми отримаємо від поверхні стрингеру, точку – від перетину цієї площини та кривої перетину, отриманої з поверхні обшивки та площини шпангоута, напрямок – від точки та тієї ж кривої перетину.

Далі на рисунку 4.11 показано програмний код функції розміщення ескізу, який має схожу структуру: оголошення «білдери» в робочій деталі, встановлення необхідних параметрів та елементів прив'язки та завершення програми.

```
SketchInPlaceBuilder sketchInPlaceBuilderNEW = workPart.Sketches.CreateSketchInPlaceBuilder2(sketchFRAME.Sketch);  
sketchInPlaceBuilderNEW.AxisOrientation = AxisOrientation.Vertical;  
sketchInPlaceBuilderNEW.PlaneOption = Sketch.PlaneOption.ExistingPlane;  
sketchInPlaceBuilderNEW.SketchOrigin = point;  
sketchInPlaceBuilderNEW.PlaneReference = plane;  
sketchInPlaceBuilderNEW.AxisReference = direction;  
sketchInPlaceBuilderNEW.Commit();  
sketchInPlaceBuilderNEW.Destroy();
```

Рис. 4.11. Функція розміщення ескізу.

Пластина.

Функція пластини нам знадобиться для створення стінки шпангоута. Вона знаходиться в інструментах листового металу NX. Виконуватись вона буде після того як буде створена поверхня обмежена кривою перетину поверхні обшивки з площиною шпангоуту та кривою в результаті перенесення кривої перетину на висоту шпангоута, а також кривою перетину поверхні початку та кінця шпангоуту (стрингери) з площиною шпангоуту. Також для цієї функції слід задати товщину пластини. На рисунку 4.12 показано програмний код «білдера» пластини.

```
NXOpen.Features.SheetMetal.TabBuilder tabBuilder = workPart.Features.SheetMetalManager.CreateTabFeatureBuilder(null);
tabBuilder.SetApplicationContext(NXOpen.Features.SheetMetal.ApplicationContext.NXSheetMetal);
tabBuilder.MaterialSide = NXOpen.Features.SheetMetal.TabBuilder.SectionSideOptions.Right;
tabBuilder.Section = section;
tabBuilder.SetThickness(thickness);
NXOpen.Features.Tab tab = (NXOpen.Features.Tab)tabBuilder.CommitFeature();
tabBuilder.Destroy();
```

Рис. 4.12. Функція пластини

Як і в попередніх функціях, першим кроком буде оголошення змінної. Оголошуємо змінну класу «*NXOpen.Features.SheetMetal.TabBuilder*» в робочій деталі за допомогою методу «*CreateTabFeatureBuilder()*», аргументом якого є «*null*», так як ми створюємо новий елемент. Далі задаємо додаткові параметри такі як товщина, набір кривих та інші. Після цього завершуємо

Висновок

Цей розділ описує розробку інструменту для побудови шпангоута в системі Siemens NX. Інструмент використовує бібліотеку NXOpen для виконання функції NX та Microsoft.Office.Interop.Excel для взаємодії з базою даних профілів шпангоута та вирізів під стрингер оформлених у вигляді таблиць MS Excel.

NXOpen – це набір інструментів API (Application Programming Interface), що дозволяє здійснювати гнучку інтеграцію користувацьких застосунків з NX за допомогою відкритої архітектури, яка може бути задіяна сторонніми розробниками, клієнтами та користувачами.

Розроблено інтерфейс інструменту за допомогою додатка NX «Розробка користувацького інтерфейсу». Інтерфейс міститиме дві вкладки для кожного методу відповідно. Перша вкладка міститиме метод, що використовує витягування профіль вздовж кривої, а друга метод побудови стінку та полички шпангоута окремо. Можливість побудови вирізів під стрингер буде в обох вкладках.

Розглянуто основні функції NX у вигляді програмного коду, які використовуються в інструменті

5. Розробка допоміжних програм для користувача.

До допоміжних інструментів для користувача належать інструмент для додавання профілів шпангоутів та інструмент для додавання профілів вирізів під стрингер в шпангоуті. Це зменшить час на заповнення бази даних профілів MS Excel та дозволить користувачу додавати стандартні профілі (тавр, двотавр, швелер).

5.1. Інтерфейс допоміжного інструменту для додавання профілів шпангоута

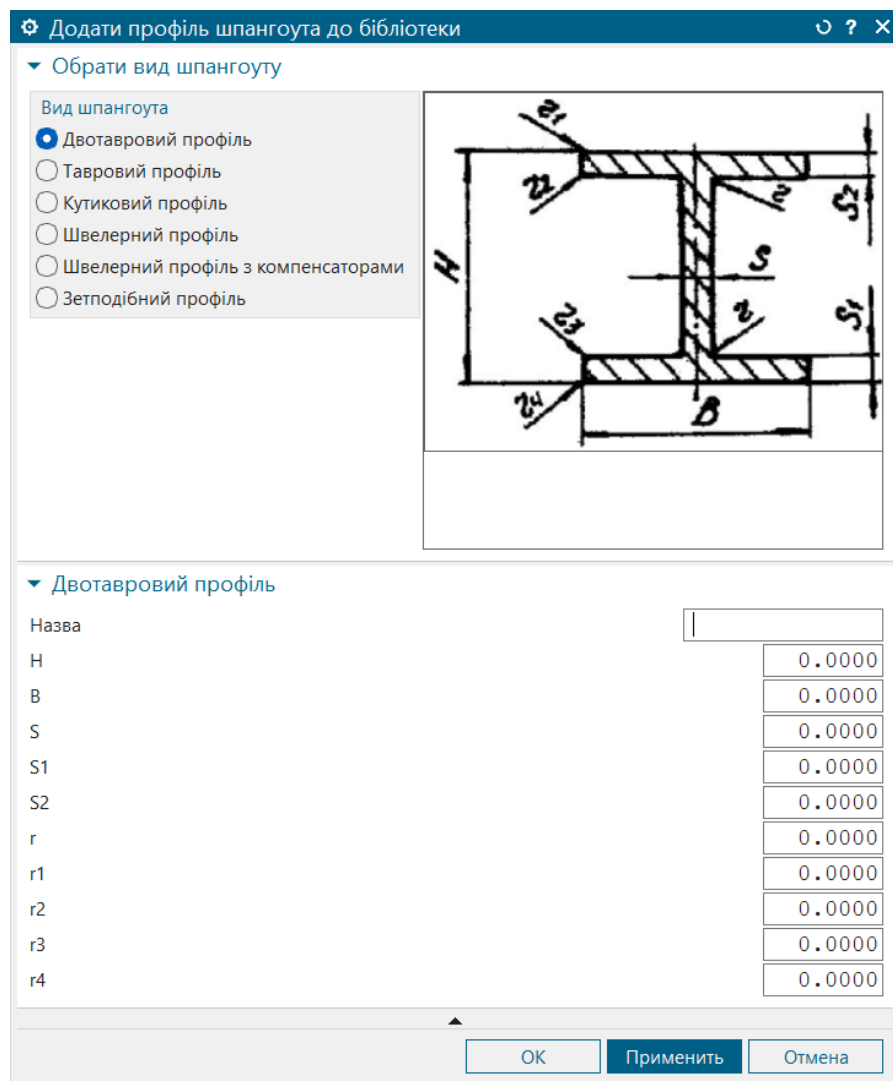


Рис. 5.1. Інтерфейс допоміжного інструменту для додавання профілів шпангоута.

Інтерфейс допоміжного інструменту для додавання нових профілів до бібліотеки дуже простий. Він складається з двох груп. В першій групі вам слід

обрати тип профілю, який ви хочете додати до бібліотеки. Залежно від вашого вибору друга група буде змінюватись. Вона містить поля з розмірами профілями та його іменем, які слід заповнити. Зверху зображена картинка вибраного профілю з позначенням розмірів.

Після заповнення форми слід натиснути кнопку «ОК» або «Применить», щоб записати дані в бібліотеку. Тепер ваш профіль доступний в інструменті побудови шпангоута.

5.2. Інтерфейс допоміжного інструменту для додавання профілів вирізів під стрингери в шпангоуті

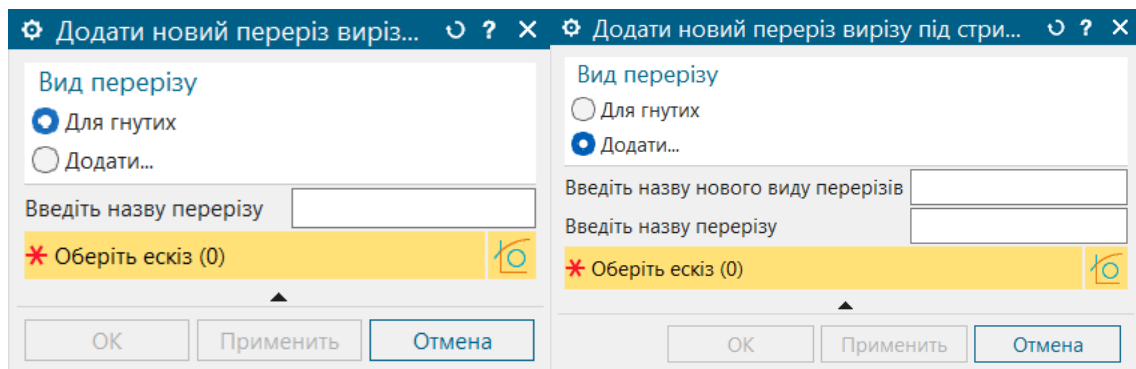


Рис. 5.2. Інтерфейс допоміжного інструменту для додавання профілів вирізів під стрингер в шпангоуті.

Інтерфейс цього допоміжного інструменту також простий. Він складається всього з 4 блоків. Перший блок налічує всі ваші існуючі групи вирізів, з яких ви можете обрати вид вирізу, який хочете додати. Після цього слід дати назву новому профілю, обрати побудований ескіз профілю в робочій деталі. Також є можливість додати нову групу, вибравши останній елемент з назвою «Додати...». В такому випадку з'явиться ще один блок де слід ввести назву нової групи та аналогічно ввести назву профілю та вибрати ескіз. Після натискання клавіш «ОК» чи «Применить»

Висновок

В цьому розділі було розроблено допоміжні інструменти для зручної взаємодії з класифікатором профілів. Вони розроблені для системи Siemens NX за допомогою бібліотеки NXOpen та мають звичний інтерфейс, схожий на інші інструменти NX.

Спершу було описано інтерфейс першого інструменту для користувача призначеного для додавання профілів шпангоуту до класифікатора. Процес додавання нових профілів до класифікатора, який являє собою файл MS Excel, доволі складний тому що в таблиці міститься лише назви профілів та значення відповідних розмірів ескізу. Цей процес полегшується так як в цьому інструменті є картинка профілю з позначеннями розмірами, а також можливість додавати новий профіль шпангоуту не виходячи з системи Siemens NX прискорює цей процес.

Далі описано інтерфейс другого допоміжного інструменту який призначений для додавання профілів вирізів під стрингер. Додавати нові профілі вирізів можливо тільки за допомогою цього інструменту, так як в класифікатор він записує лише назву профілю. Сам профіль зберігається в файлі .prt та має задану користувачем назву, що пов'язує його з класифікатором.

6. Тестування програми.

6.1. Загальне тестування інструменту

Після того як завершено розробку інструменту для побудови шпангоута треба провести тестування для того щоб виявити неточності чи помилки при комбінації нестандартних дій користувача.

Спершу пройдемо звичайний шлях побудови шпангоута покроково, слідуючи інтерфейсу інструменту на першій вкладці. Для початку необхідно побудувати початкову модель, яка буде містити поверхню обшивки, площину шпангоута та поверхні стрингерів (для визначення меж шпангоута та вирізів під стрингери) (рис. 6.1).

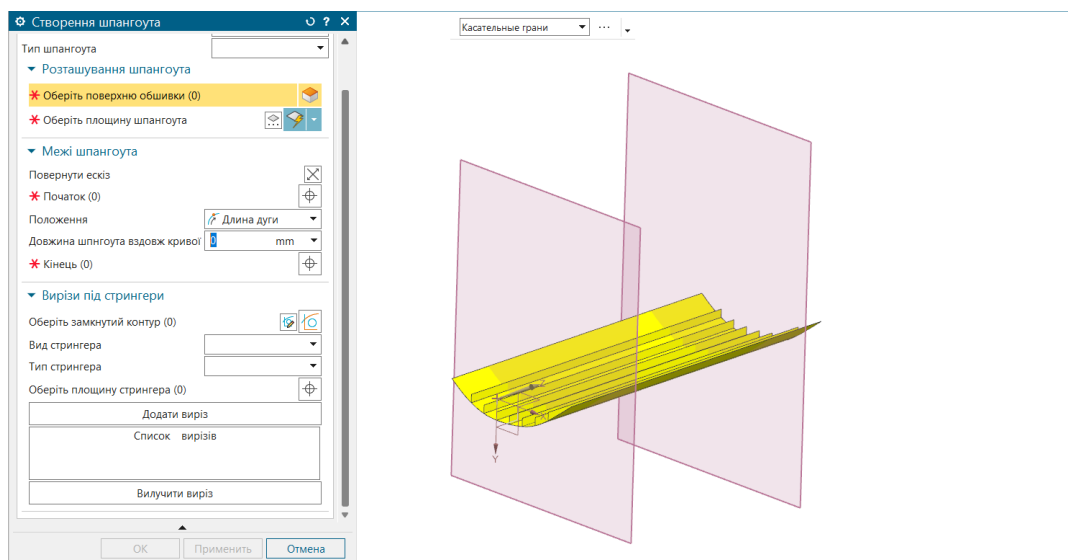


Рис. 6.1. Початкова модель для роботи з інструментом.

Спершу обираємо вид шпангоута, а потім його тип. Обираючи тип шпангоута ви обираєте конкретні розміри профілю шпангоута з бази даних, які відповідають назві цього шпангоута (рис. 6.2).

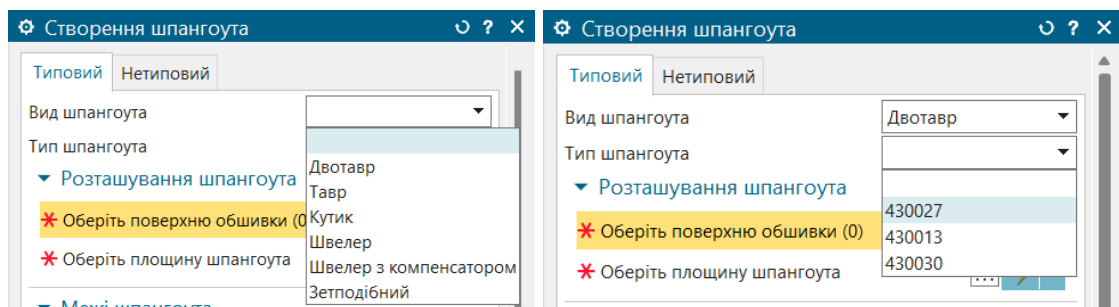


Рис. 6.2. Вибір профілю шпангоута.

Тепер вибираємо поверхню обшивки та площину шпангоута, щоб задати розташування шпангоута (рис. 6.3). Площину шпангоута можна задати багатьма способами. Для цього треба натиснути на значок з трьома крапками праворуч.

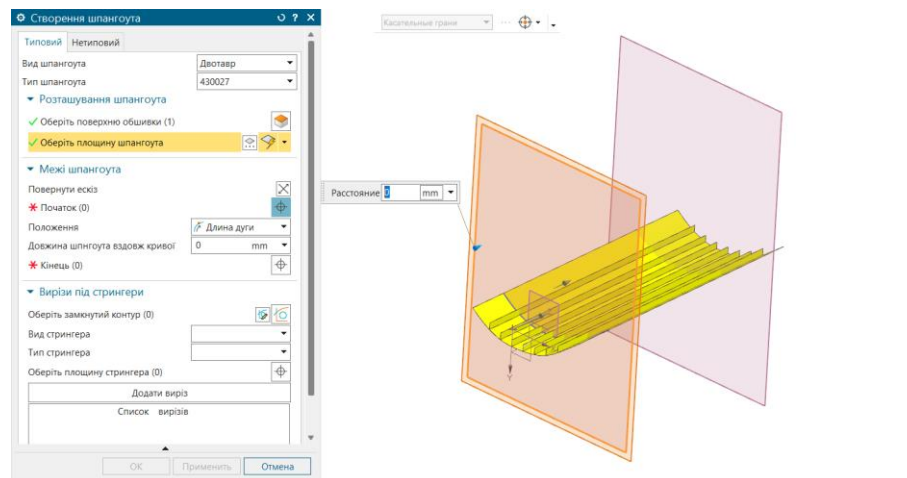


Рис. 6.3. Вибір елементів з групи «Розташування шпангоута»

Наступним кроком буде вибір початку шпангоута. Це може бути поверхня стрингера, чи будь яка інша поверхня на якій розміститься ескіз попередньо обраного профілю шпангоута (рис. 6.4). В процесі розробки інструменту було виявлено, що не в усіх випадках ескіз профілю розміщується правильно. В деяких ситуаціях ескіз розміщується перевернутим (з зовнішньої грані обшивки). Для таких випадків було додано додаткову функцію «Повернути ескіз».

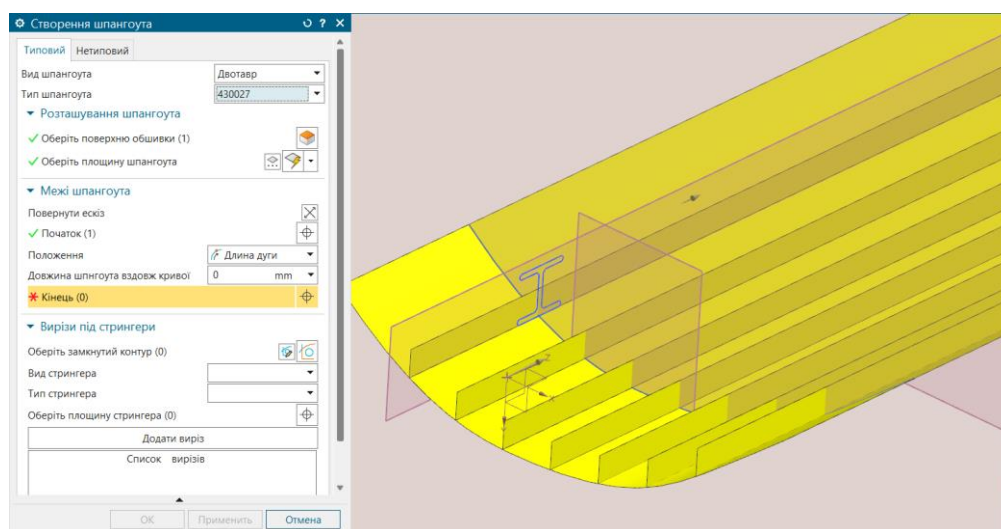


Рис. 6.4. Вибір першої межі шпангоута.

Після цього є два варіанти як обрати другу межу шпангоута. Перший спосіб – це обрати поверхню (поверхню стрингера) та другий спосіб – задати довжину шпангоута вздовж кривої. Після цього кроку буде побудоване тіло шпангоута (рис. 6.5).

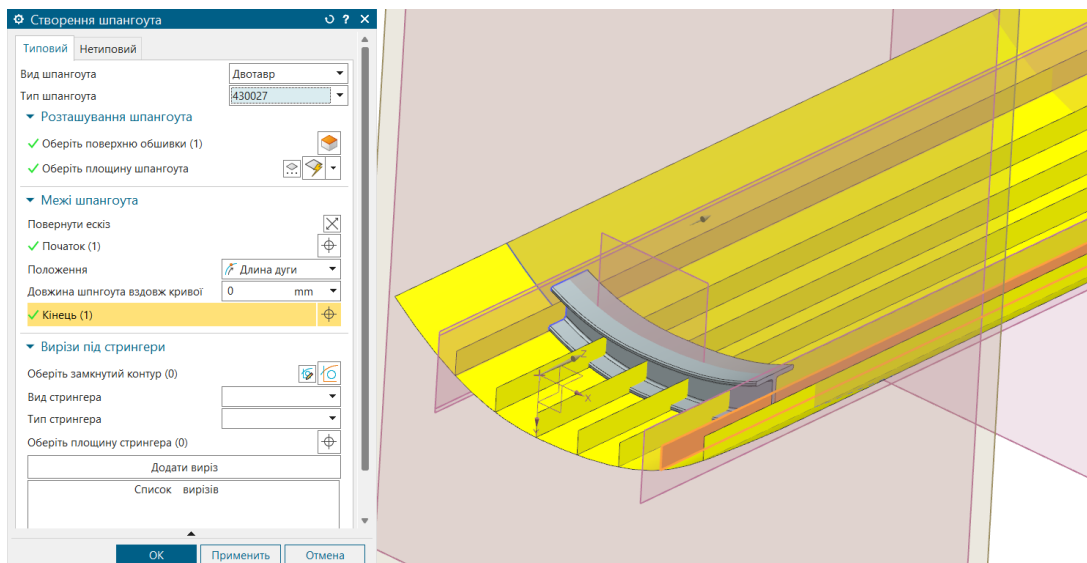


Рис. 6.5. Вибір другої межі шпангоута.

Після побудови тіла з'являється можливість створити вирізи під стрингер. Є можливість обрати з бібліотеки необхідний профіль, обрати вже побудований ескіз в робочій деталі або побудувати новий ескіз. Спробуємо побудувати новий ескіз вирізів під стрингери. Для цього слід скористатись кнопкою праворуч від напису «Оберіть замкнутий контур» (рис. 6.6).

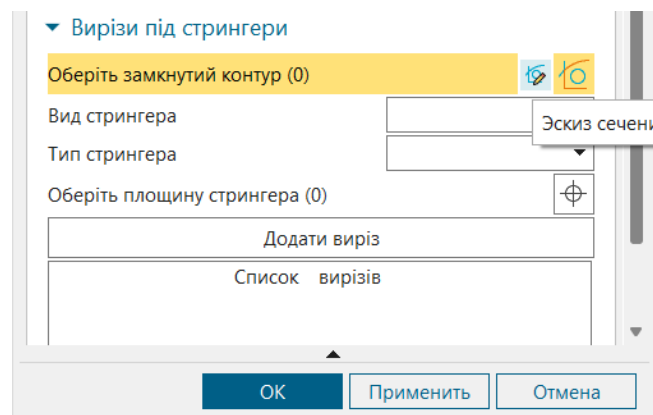


Рис. 6.6. Побудова нового ескізу вирізів під стрингер.

Після натискання на кнопку відбувається перехід на меню створення ескізу, де потрібно «прив'язати» ескіз, тобто обрати площину на якій буде розташований ескіз, точку початку системи координат ескізу та інше (рис. 6.7).

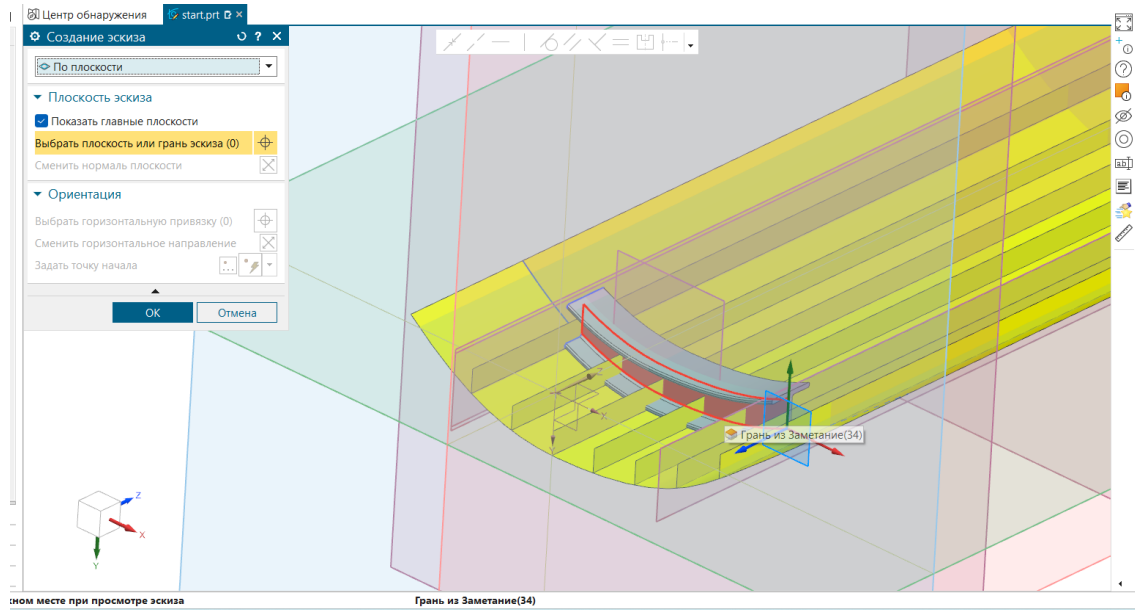


Рис. 6.7. «Прив'язка» ескізу.

Так як є можливість створити лише один ескіз то створюємо вирізи зразу для всіх стрингерів в одному ескізі (рис. 6.8).

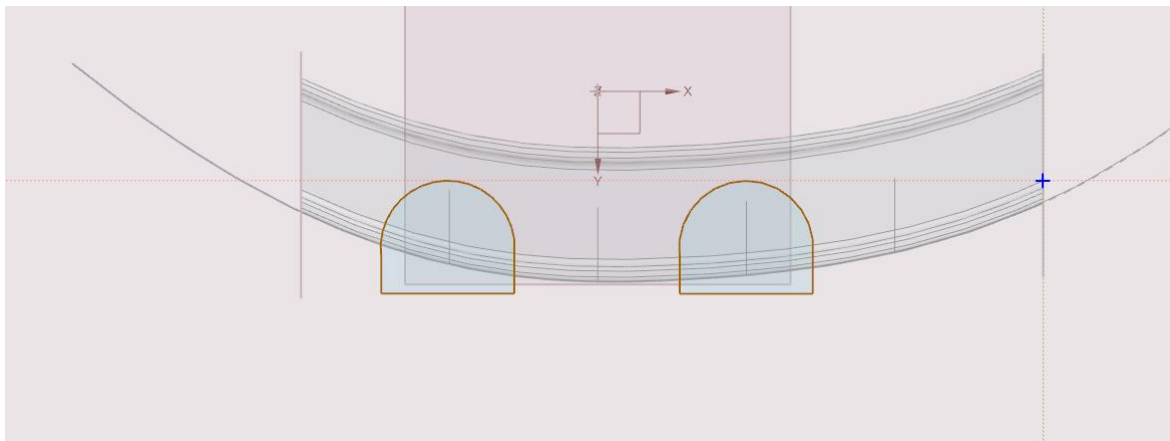


Рис. 6.8. Ескіз вирізів під стрингер.

Для того щоб завершити побудову ескізу та продовжити побудову шпангоуту натискаємо на кнопку в лівому верхньому кутку «Завершити». Далі переконуємось, що наш ескіз побудований в потрібному місці та натискаємо клавішу «Додати виріз» (рис. 6.9). Після цього буде створено виріз під назвою «Власний виріз» та додано в список вирізів. Можливість вставити вже

існуючий профілю вирізу з бази даних зберігається. Виріз відбувається лише в тілі, тільки що побудованого шпангоута.

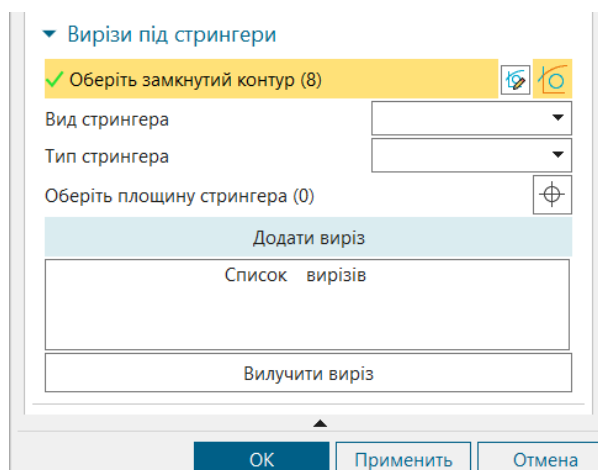


Рис. 6.9. Додавання вирізу.

Список вирізів містить всі існуючі вирізи в створеному шпангоуті. При виборі одного вирізу зі списку він підсвітиться в моделі шпангоута (рис. 6.10), що дозволяє легко перевірити модель. Щоб «погасити» виріз треба натиснути на напис «Список вирізів».

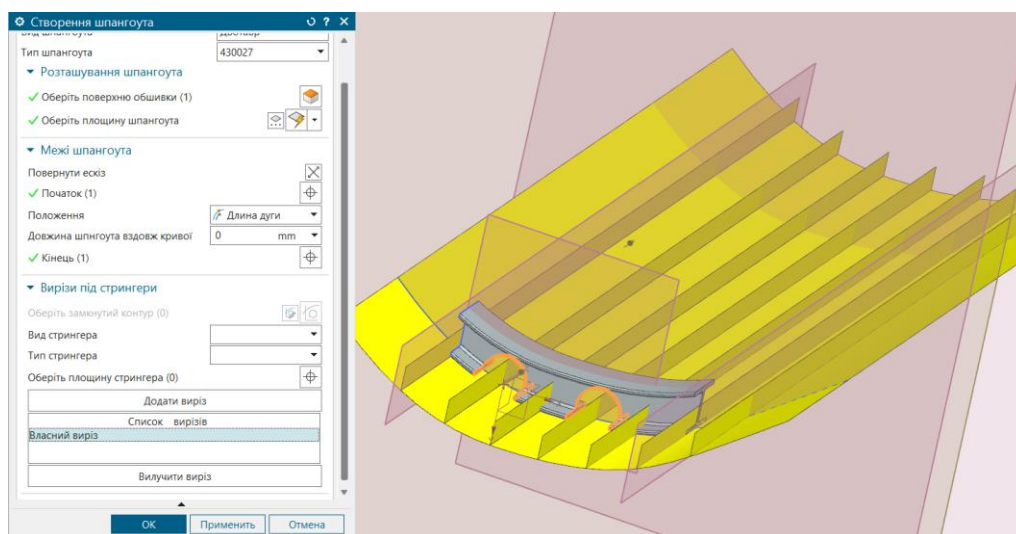


Рис. 6.10. «Власний виріз» в шпангоуті.

Також є можливість видалити виріз. Для цього треба обрати необхідний виріз та натиснути клавішу «Видалити виріз» (рис. 6.11).

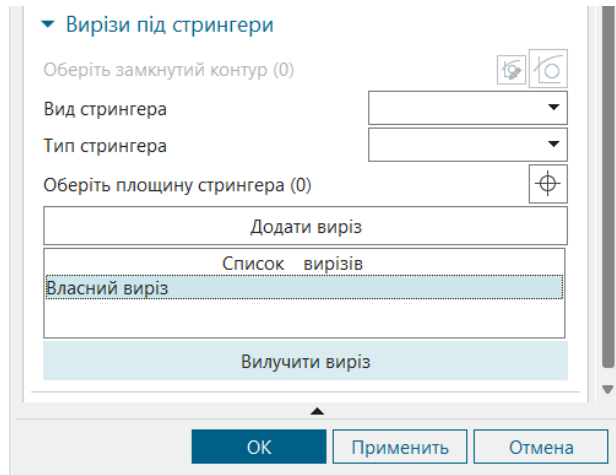


Рис. 6.11. Видалення вирізу.

Він видалиться і в моделі шпангоута і в списку, проте ескіз вирізу залишиться, його за необхідності можна буде видалити після завершення роботи з інструментом (рис. 6.12).

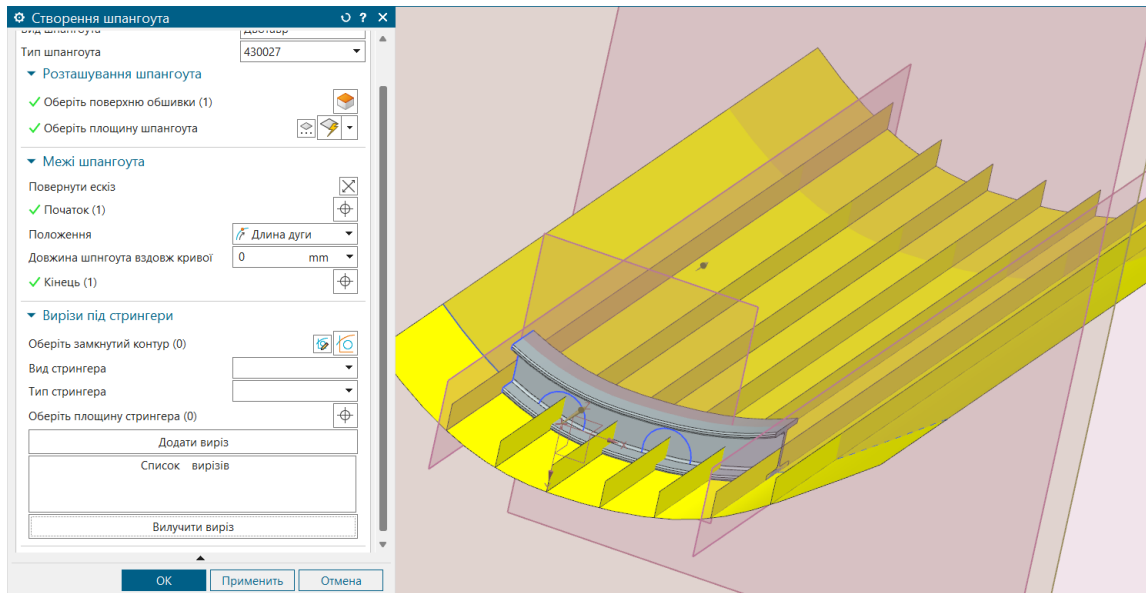


Рис. 6.12. Видалення вирізу.

Натискаємо «ОК», якщо хочемо завершити роботу та «Применить», якщо необхідно використати інструмент ще раз.

Для використання другого методу, який представлений в інструменті слід переключитись на другу вкладку. Для тестування другого методу треба також спершу побудувати початкову модель в якій буде поверхня обшивки, площина шпангоута та поверхні стрингерів (для побудови вирізів під стрингери). Також

для цього методу треба побудувати ескіз контуру стінки шпангоута перед використанням інструменту (рис. 6.13).

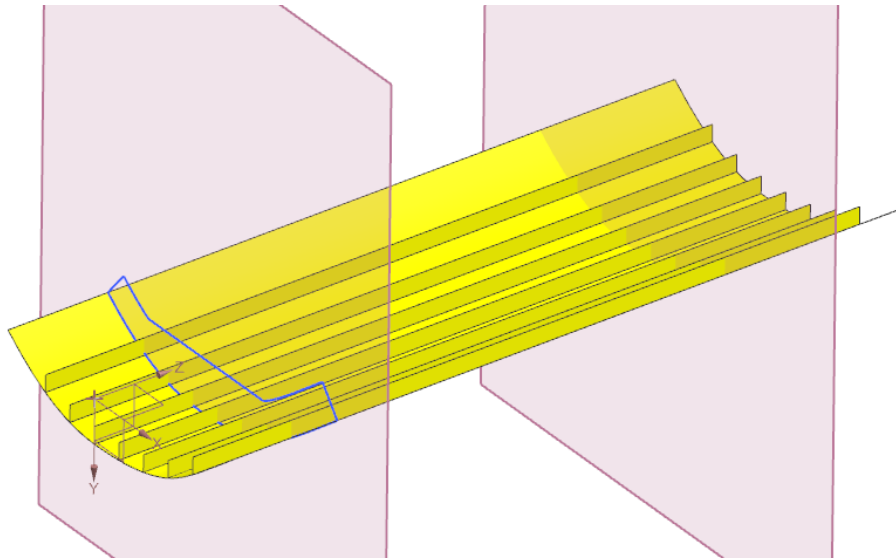


Рис. 6.13. Початкова модель для роботи з інструментом (другим методом).

Після запуску інструменту треба переключитись на другу вкладку інтерфейсу. Першим кроком буде вибір ескізу стінки шпангоута та призначення їй товщини, після чого побудується тіло стінки. Порядок дій немає значення, але це треба зробити обов'язково так як подальші операції будуть доступні лише з створеним тілом. Також є можливість змінювати товщину чи ескіз стінки навіть після її побудови. Можливо побудувати лише одне тіло стінки шпангоута.

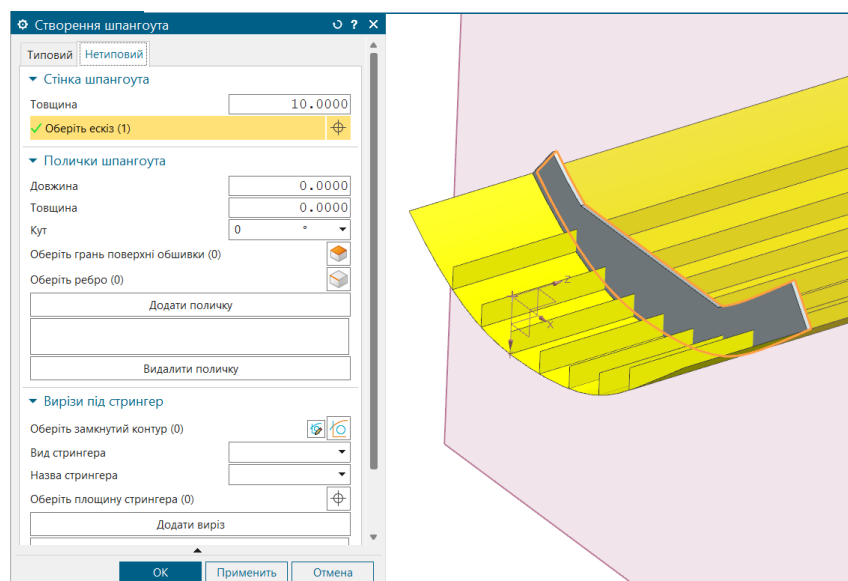


Рис. 6.14. Створення стінки шпангоута.

Наступним кроком буд створення поличок. Для цього слід заповнити всі необхідні поля параметрів полички та додати поличку(рис. 6.15). Створення поличок необов'язкове, тобто ви можете завершити роботу з інструментом без поличок.

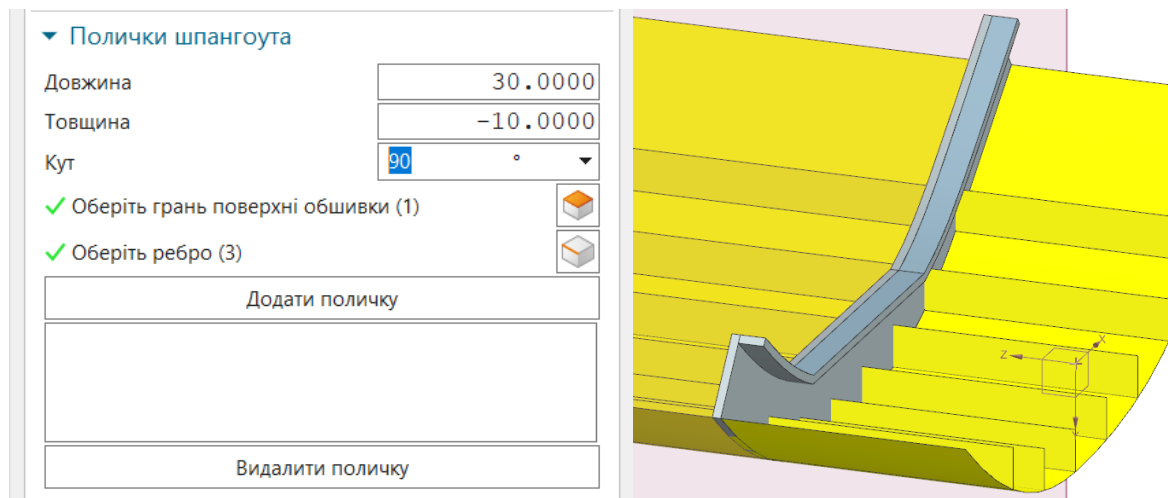


Рис. 6.15. Створення полички шпангоута.

Після того як вказані всі параметри та вибрані всі допоміжні елементи треба натиснути кнопку «Додати». Поличка побудується та відповідний запис з'явиться в списку. Всі вказані параметри дорівнюватимуть нулю та скинуться всі вибрані елементи. Для того щоб видалити поличку слід обрати її в списку та натиснути кнопку «Видалити».

Група з створення вирізів в шпангоуті аналогічна і протестована вище.

6.2. Тестування окремих випадків

Протестуємо незвичні випадки та можливі помилки, які можуть виникнути в користувача. Інструмент має бути захищений від випадкових неправильних виборів елементів, не лінійним (можливість задавати елементи для побудови шпангоута непослідовно).

1. Вибір типу профілю шпангоута не обравши його виду.

Список можливих варіантів вибору типу шпангоута заповнюють після вибору його виду, тому в такому випадку список буде пустий та користувач не зможе обрати конкретний шпангоут (рис. 6.16).

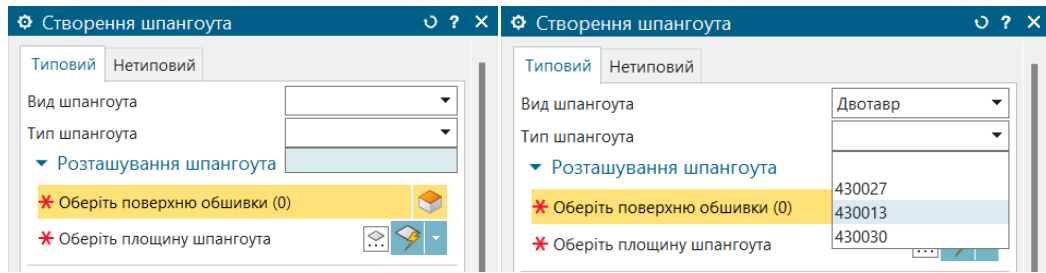


Рис. 6.16. Принцип роботи вибору профіля шпангоута.

2. Поверхня обшивки та площина шпангоута не перетинаються.

В такому випадку буде видано повідомлення користувачу, що лінія перетину не утворена та запропоновано обрати необхідні елементи ще раз. Попередній вибір в блоках інтерфейсу буде видалено, тому користувачу в будь-якому випадку треба буде задати поверхню обшивку та площину шпангоуту ще раз.

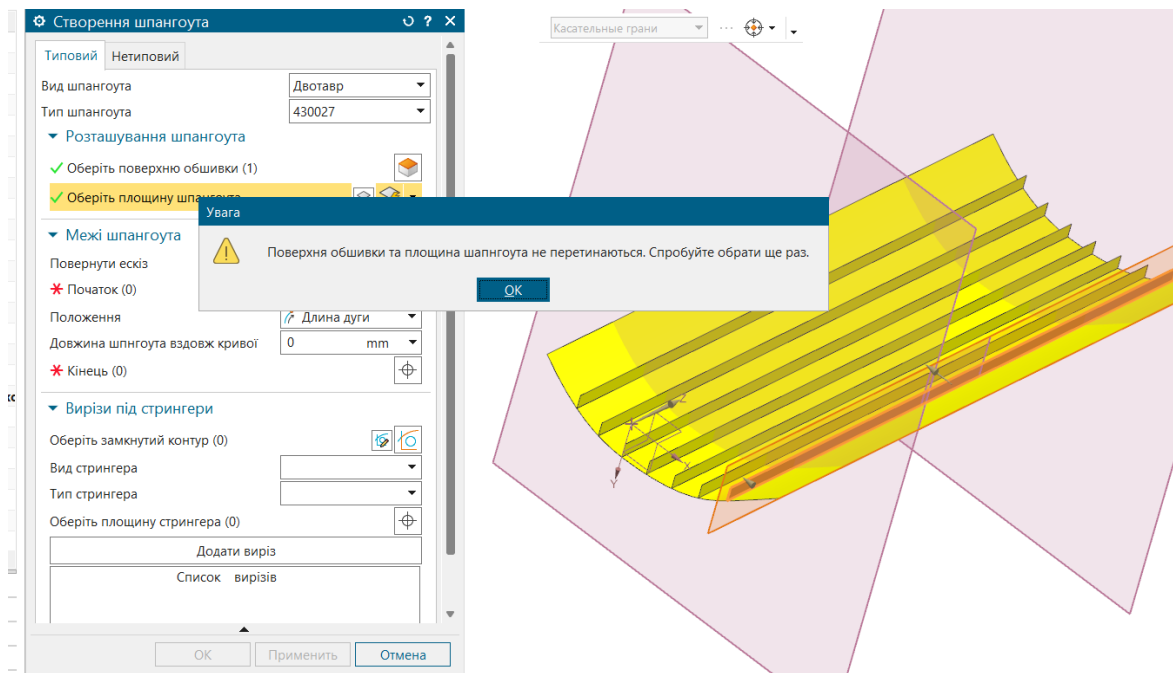


Рис. 6.17. Повідомлення про помилку.

3. Задавання другої межі шпангоута двома способами (площиною та довжиною вздовж кривої).

Такої ситуації не можна буде досягти, так як можливість обрати кінцеву площину чи поверхню буде лише тоді коли значення довжини шпангоуту вздовж кривої дорівнює нулю (рис. 6.18). В іншому випадку можливість обрати

кінцеву межу шпангоуту втрачається (блок стає неактивним та анулюються всі обрані елементи).

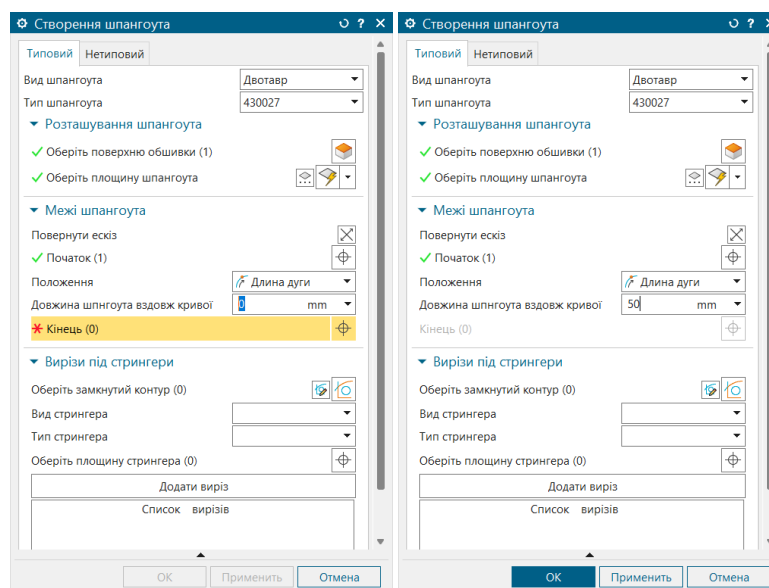


Рис. 6.18. Вибір кінцевої межі шпангоуту.

4. Натискання клавіші «Додати» не вибравши профіль стрингера.

В такому випадку користувачу буде видано повідомлення, що не жоден з перерізів не було обрано та запропонує скористатись одним з них (рис. 6.19).

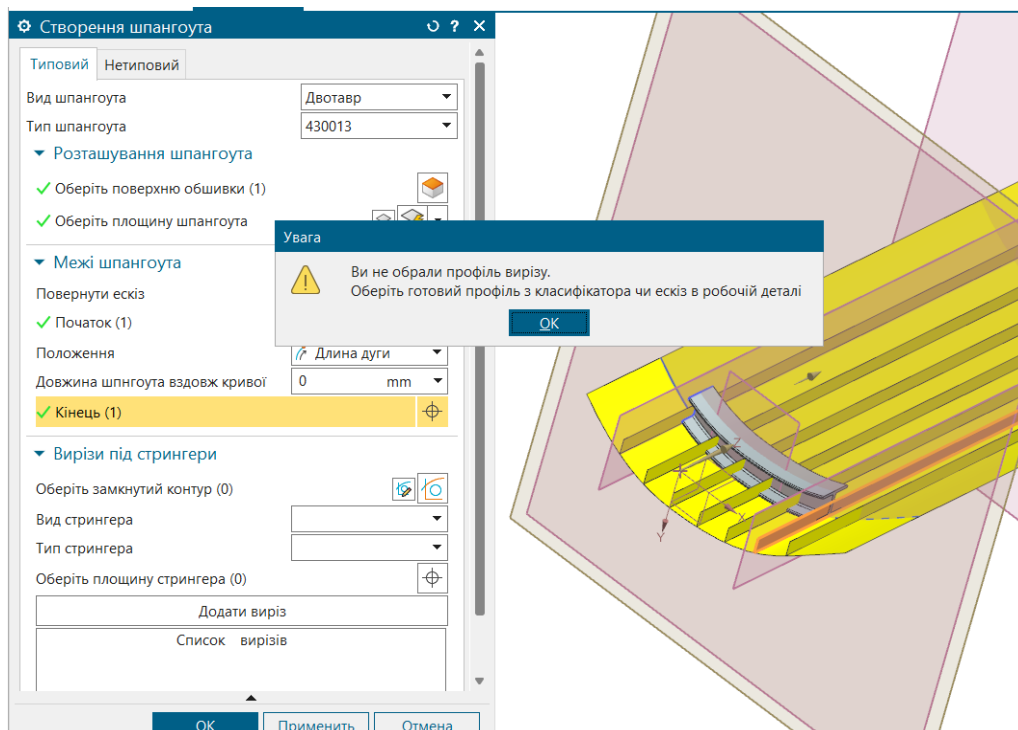


Рис. 6.20. Повідомлення про помилку.

5. Відсутність деяких параметрів при створенні полицок шпангоута.

Якщо не введено параметри довжини, товщини (дорівнюють 0) та не обрано грань та ребро (допоміжні елементи) та натиснута кнопка «Додати» з'явиться відповідне повідомлення (рис. 6.21).

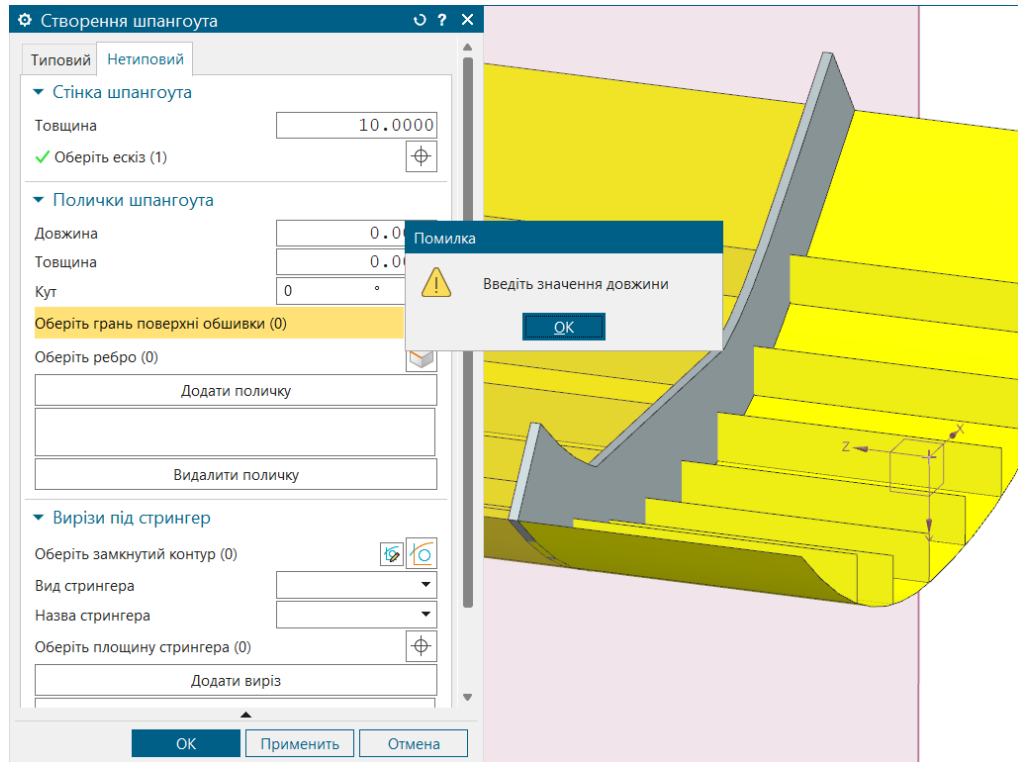


Рис. 6.21. Повідомлення при невведеній довжині шпангоута.

Висновок

В цьому розділі виконано тестування інструменту в лінійному порядку (заповнення блоків інтерфейсу один за одним) та в довільному порядку. При некоректних ситуаціях, які можуть виникнути в користувача при очевидно неправильному виборі допоміжних елементів та параметрів для побудови шпангоуту з'являється вікно помилки, яке пояснює проблему та пропонує змінити вибір елементу чи параметр.

Тестування проведено на завчасно змодельованій моделі обшивки, стрингерів, а також площині шпангоуту. Використовувати обидва методи одночасно не дозволяється, що і продемонстровано в розділі.

Також розглянуто окремі випадки, які можуть виникнути в недосвідченого користувача та пояснено причини їх виникнення та варіанти усунення.

7. Розробка стартап-проекту.

Таблиця 7.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення конструктивних елементів поперечного силового набору фюзеляжу (в деяких випадках крила) літального апарату.	1. Авіабудування (фюзеляж ЛА).	Існуюча база даних профілів, яку можливу змінювати залежно від потреб компанії. Автоматизація проектування шпангоутів різних типів.
	2. Авіабудування (крило ЛА).	Можливе застосування для деяких випадків нервюр.
	3. Розробка CAD-систем (зокрема Siemens NX).	Розробка в існуючій CAD-системі інструменту для створення шпангоутів на основі запропонованого.

Таблиця 7.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту.

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Siemens NX	SolidWorks	Власні розробники інженерних компаній			
1.	Для роботи необхідна інша CAD-система.	+	-	-	+	+		
2.	Наявність бази даних профілів та можливість її налаштування під особливості та потреби виробництва.	Є	Немає	Немає	Можливо			+

Продовження таблиці 7.2.

3.	Необхідність спеціального навчання працівників.	Немає	Повний курс	Повний курс	Немає		+	
4.	Єдиний інструмент, в якому зібрано все необхідне для побудови шпангоута та вирізів під стрингер.	Є	Немає	Немає	Можливо			+
5.	Необхідність спеціалізованих працівників для обслуговування та налаштування інструменту.	-	+	+	+		+	

Таблиця 7. 3. Технологічна здійсненність проекту.

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Інструмент для побудови шпангоутів в системі Siemens NX	NXOpen	Наявні	Доступні разом з ліцензією Siemens NX
2.	База даних профілів шпангоутів	MS Office	Наявні	Доступні

Всі технології необхідні для реалізації проекту доступні авторам проекту.

Таблиця 7.4. Попередня характеристика потенційного ринку стартап-проекту.

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	5
2	Динаміка ринку	Зростає
3	Наявність обмежень для входу	Немає
4	Специфічні вимоги до стандартизації та специфікації	Немає
5	Середня норма рентабельності в галузі, %	54

Таблиця 7.5. Характеристика потенційних клієнтів стартап-проекту.

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних груп клієнтів	Вимоги споживачів до товару
1	Автоматизація проектування шпангоутів	Підприємства, що розробляють ЛА.	Використання різних CAD-систем.	Якісна модель шпангоута.

Таблиця 7.6. Фактори загроз.

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Непрофесійність користувача	Неправильне використання інструменту для побудови шпангоутів	Видача докладної документації (Керівництво користувача) до інструменту
2	Невідповідність CAD-системи користувача.	Користувач не використовує CAD-систему Siemens NX або має застарілу версію.	Доопрацювати інструмент та виправити помилки пов'язані з іншою версією Siemens NX.
3	Ненадійність інструменту	Знайдено випадок шпангоута, який неможливо побудувати за допомогою інструменту.	Запропонувати зміни в способі побудови шпангоуту.

Таблиця 7.7. Фактори можливостей.

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Розширення функціоналу інструменту.	Необхідність в автоматизації проектування інших конструктивних елементів ЛА.	Доопрацювання існуючого чи розробка нового інструменту для побудови інших конструктивних елементів ЛА.
2	Розширення бази даних	Необхідність створення бази даних не лише профілів шпангоутів, але і самих шпангоутів чи деяких їхніх частин.	Створення додаткової бази даних та інструменту для легкої взаємодії з нею.

Таблиця 7.8. Ступеневий аналіз конкуренції на ринку.

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною).
Тип конкуренції – чиста	Відкрита бібліотека NXOpen, за допомогою якої можна створювати власний продукт. Все залежить від вмінь.	Розширювати список розроблених інструментів для Siemens NX, а також для інших CAD-систем.
За рівнем конкурентної боротьби – міжнародний	Розробники (зокрема Siemens NX) знаходяться, як в Україні так і закордоном.	Стати лідером в Україні.
За галузевою ознакою – внутрішньогалузева.	Інструменти розробляються лише для автоматизації проектування і переважно проектування частин ЛА.	Зосередитись на автоматизації проектування ЛА та розробити повноцінний пакет інструментів для потреб проектування ЛА та його частин.

За характером конкурентних переваг – нецінова.	Перевага в першу чергу залежить від якості продукту та способу реалізації.	Покращувати рівень своєї продукції.
--	--	-------------------------------------

Таблиця 7.9. Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Siemens NX.	Власні розробники клієнтів. Бар'єри для входження в ринок: кваліфікований персонал, притаманно великим компаніям з великим об'ємом виготовленої продукції.	Немає.	Компанії, що займаються розробкою документації та виготовленням ЛА.	Інші CAD-системи.
Висновки:	Siemens NX сама заохочує нових розробників для розвитку своєї продукції.	Великі компанії часто мають своїх співробітників, що займаються автоматизацією проектування, проте більшість компаній, що тільки починають масштабувати виробництво потребують автоматизації.		Клієнти не диктують умови роботи на ринку.	

Таблиця 7.10. Обґрунтування факторів конкурентоспроможності.

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Менший масштаб підприємства	Порівнюючи з Siemens, яка є великою компанією з розробки CAD-систем, тому їй складніше реагувати на нагальні потреби всіх клієнтів. Також Siemens розробляє CAD-систему та інструменти для неї для загального проектування. Налаштування ж під свої особливі потреби та технології буде вимагати додаткових домовленостей з Siemens.
2	Представництво в Україні	Більшість розробників розташовані за кордоном та не мають представництва в Україні.
3	Інструмент розроблений для системи Siemens NX.	Не всі клієнти використовують Siemens NX.
4		

Таблиця 7.11. Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Рейтинг товарів-конкурентів у порівнянні з моїм проектом						
		-3	-2	-1	0	+1	+2	+3
1	Менший масштаб підприємства.	-	SolidWorks	Siemens NX	Власні розробники інженерних компаній	-	-	-
2	Представництво в Україні.	Власні розробники інженерних компаній	Siemens NX	SolidWorks				
3	Інструмент розроблений для системи Siemens NX.				Siemens NX	SolidWorks		Власні розробники інженерних компаній

Таблиця 7.12. SWOT-аналіз стартап-проекту.

Сильні сторони: Автоматизація проектування шпангоута та налаштування бази даних профілів шпангоутів та вирізів під стрингер під потреби підприємства.	Слабкі сторони: Інструмент працює лише для системи Siemens NX.
Можливості: Розробка інших інструментів для автоматизації проектування ЛА.	Загрози: Siemens розробить власний інструмент для створення шпангоутів.

Таблиця 7.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів ринкової поведінки)	Ймовірність отримання ресурсів	Строки реалізації
1	Розробка інструменту для створення шпангоутів для іншої CAD-системи.	Збільшення числа клієнтів.	Пів року.

Продовження таблиці 13.

2	Розробка інших інструментів для автоматизації проектування ЛА.	Збільшення числа клієнтів.	Рік.
3	Вдосконалення чи застосування інших принципів в створенні шпангоутів.	Збільшення числа клієнтів.	2 місяці.

Таблиця 7.14. Вибір цільових груп потенційних споживачів.

№ п/п	Опис профілю цільової групи потенційних клієнтів.	Готовність споживачів сприйняти продукт.	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент.
1	Підприємства, що розробляють та проектують ЛА.	Є.	Достатній.	Невеликий.	Середня.
2	Розробники CAD-систем.	Є.	Низький.	Високий.	Середня.
Які цільові групи обрано: Підприємства, що розробляють та проектують ЛА.					

Таблиця 7.15. Визначення базової стратегії розвитку.

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Розробка інструменту для створення шпангоутів для іншої САД-системи.	Широкий вибір потенційних клієнтів.	Інструмент працюватиме незалежно від САД-системи, якою користується клієнт.	Стратегія спеціалізації.
2	Розробка інших інструментів для автоматизації проектування ЛА.	Широкий набір інструментів.	Пакет інструментів для автоматизації проектування ЛА.	Стратегія спеціалізації.
3	Вдосконалення чи застосування інших принципів в створенні шпангоутів.	Унікальний продукт.	Впровадження інструменту в систему підприємства залежно від його специфіки та технології.	Стратегія диференціації.

Таблиця 7.16. Визначення базової конкурентної поведінки.

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента і які?	Стратегія конкурентної поведінки
Так.	Шукати нових.	Так, інтерфейс інструменту буде схожим.	Стратегія заняття конкурентко ніші.

Таблиця 7.17. Визначення стратегії позиціонування.

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових).
1	Автоматизована побудова шпангоута.	Стратегія спеціалізації	Інструмент для побудови шпангоутів.	Автоматизація проектування, шпангоут, інструмент, CAD-система..
2	База даних профілів шпангоутів.	Стратегія спеціалізації	База даних профілів шпангоутів та вирізів під стрингер налаштована під особливості клієнта та його технології, а також допоміжні інструменти для легкої взаємодією з базою даних.	Автоматизація проектування, профіль, шпангоут, база даних.
3	Відсутність необхідності додаткового навчання працівників.	Стратегія спеціалізації	Інтерфейс інструменти відповідає класичному інтерфейсу CAD-системи.	Автоматизація проектування, знайомий інтерфейс, інструмент,

Таблиця 7.18. Визначення ключових переваг потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Автоматизація проектування шпангоута.	Інструмент для побудови шпангоута.	Існують: Після використання одного інструменту буде готове тіло шпангоута та вирізи під стрингери. Простота побудови. Створити. Варіативність побудови.
2	База даних профілів шпангоутів та вирізів під стрингер для використання їх в побудові шпангоута.	База даних профілів шпангоутів та вирізів під стрингер, які вставляються в модель шпангоута при побудові його за допомогою інструменту для побудови шпангоута.	Існують: База даних, яка містить не лише розміри відповідних типів шпангоутів, але й посилання на ескізи цих профілів, які легко вставляються в модель шпангоута при його побудові. Додатковий інструмент для легкої взаємодії з базою даних, що дозволяє додавати розміри профілів певних типів.
3	Інструмент працює для будь-якої CAD-системи.	Інструмент працює для системи Siemens NX.	Створити: Розширити перелік CAD-систем в яких працює інструмент.

Висновок

Розділ "Розробка стартап-проекту" фокусується на аспектах маркетингу у контексті створення стартапу. В цьому розділі розглядаються етапи, такі як відбір ідей, формування концепції продукту, аналіз перспектив ринкової реалізації проекту та розробка маркетингової стратегії. Цей розділ виконується у вигляді оцінки можливостей та розроблення заходів для ринкового впровадження інноваційних інструменту для створення стартапу. В розділі проведено маркетинговий аналіз стартап проекту задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження. Визначено ключові переваги та недоліки стартап-проекту з його можливими та існуючими конкурентами, а також загрози, які можуть виникнути в реалізації цього проекту.

Висновки

У даній роботі викладено результати досліджень актуальної наукової задачі – створення шпангоутів в системі Siemens NX за допомогою розробленого інструменту для побудови шпангоута та класифікатора профілів шпангоутів та вирізів під стрингер, що дозволяє автоматизувати проектування шпангоутів шляхом зменшення часу на побудову ескізів профілів шпангоутів та вирізів вручну та в результаті роботи інструменту отримати тіло шпангоуту з заданими параметрами.

Теоретичне значення отриманих результатів полягає у визначенні типових профілів шпангоутів та вирізів під стрингер, розробці класифікатора профілів та визначення методів побудови шпангоутів в CAD системі Siemens NX.

Практичне значення отриманих результатів полягає в автоматизації проектування шпангоутів, що реалізується завдяки розробленому інструменту для побудови шпангоутів, класифікатору профілів шпангоутів та вирізів під стрингер та допоміжних інструментів для зручного використання класифікатору.

На підставі виконаних досліджень зроблені наступні висновки:

1. Аналіз конструктивно-силового набору літака та огляд типових профілів шпангоутів та вирізів під стрингер дозволив створити класифікатор профілів, який складається з списку профілів та їх розмірів, а також файлу з ескізами кожного типу профілю.

2. Дослідження методології моделювання шпангоуту показало, що існує щонайменше чотири різних способи, щоб побудувати шпангоут в системі Siemens NX. Вони мають спільні риси, проте можуть використовувати різні інструменти NX, та в різній послідовності. Також вони можуть відрізнятися фінальними характеристиками моделі шпангоута, наприклад, одні інструменти виконають тіло шпангоута як «тверде тіло», а інші – як «листова тіло».

3. Siemens NX є інтегрованою системою проектування та виробництва, яка забезпечує повний цикл розробки виробів. Ця система володіє широким

набором інструментів для створення та аналізу складних тривимірних моделей. Також є можливість її кастомізувати відповідно до технологій виробництва та автоматизувати проектувальні процеси за допомогою бібліотеки NXOpen.

4. В результаті дослідження бібліотеки NXOpen та іншого функціоналу NX, що дозволяє створювати користувацькі за стосунку та легко інтегрувати їх в систему NX, було розроблено інструмент для створення шпангоутів та допоміжні інструменти для зручної взаємодії з класифікатором профілів через систему NX.

5. Досліджено можливість та ефективність ринкової комерціалізації проекту, визначено загальні напрями використання та проаналізовано ринкові можливості щодо реалізації стартап-проекту.

За результатами роботи зроблена доповідь на науково-практичній конференції студентів та молодих вчених «Авіа-ракетобудування:Перспективи та напрямки розвитку» <https://arb.kpi.ua/uk/science/conferences> і опубліковані у вигляді тез, результати були опубліковані у фаховому виданні "INTELLIGENCE.INTEGRATION.RELIABILITY" 2023-го року <https://iat.kpi.ua/wp-content/uploads/2023/12/IRR-2023.pdf>.

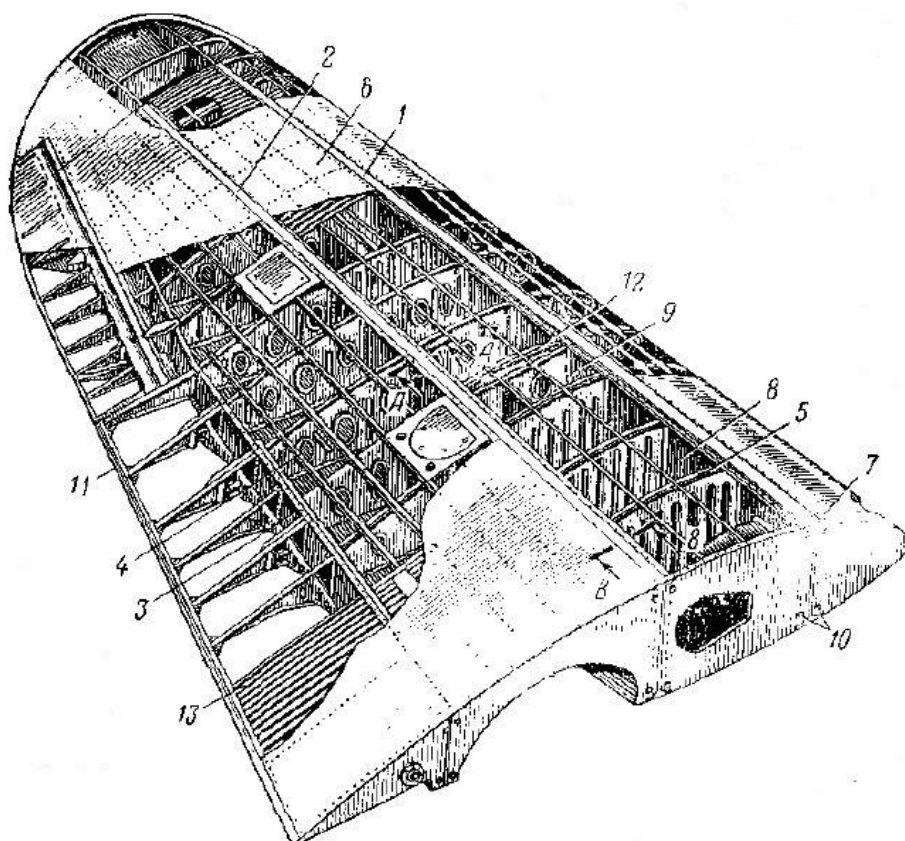
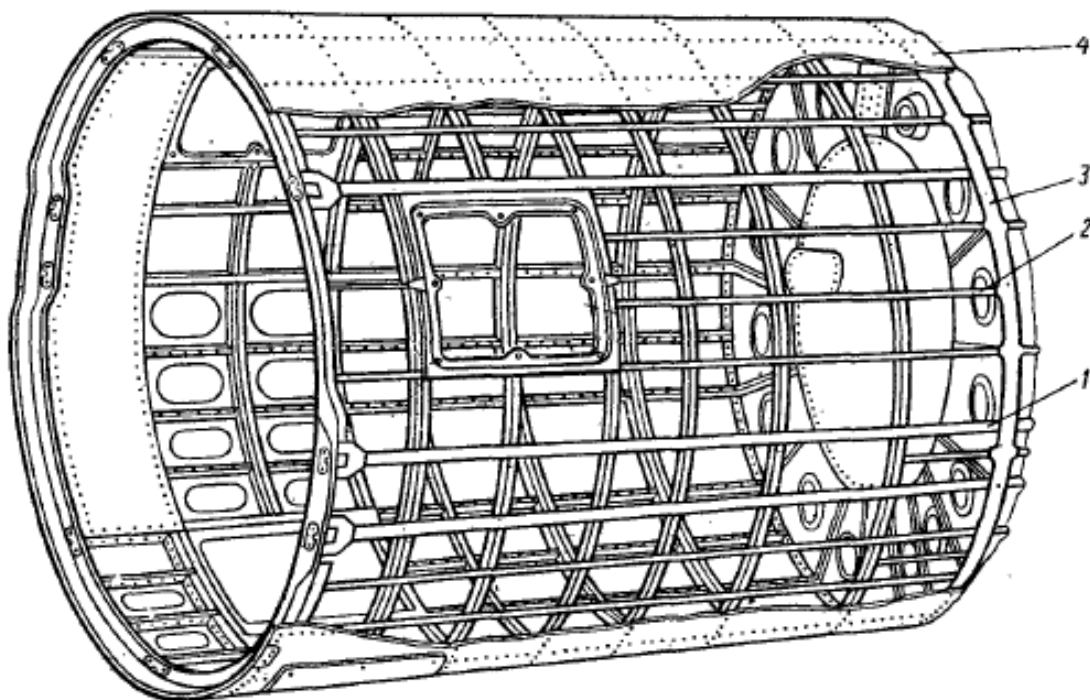
Результати можуть бути використанні при проектуванні шпангоутів літальних апаратів.

Перелік використаних джерел

1. Житомирський Г. И. Конструкция самолетов / Г. И. Житомирський. – москва: Машиностроение, 1995. – 415 с.
2. Шульженко М. Н. Курс конструкции самолетов / М. Н. Шульженко, А. С. Мостовой. – москва: Машиностроение, 1965. – 563 с.
3. Гребеньков О. А. Конструкция самолетов / О. А. Гребеньков., 1984. – 237 с.
4. Шульженко М. Н. Конструкция самолетов / М. Н. Шульженко. – москва: Машиностроение, 1971. – 416 с.
5. Проектирование конструкции самолетов / Е. С.Войт, А. И. Ендогур, З. А. Мелик-Саркисян, И. М. Алявдин. – москва: Машиностроение, 1987. – 416 с.
6. Проектирование самолетов / [С. М. Егер, В. Ф. Мишин, Н. К. Лисейцев та ін.]. – москва: Машиностроение, 1983. – 616 с.
7. Глаголев А. Н. Конструкция самолетов / А. Н. Глаголев, М. Я. Гольдинов, С. М. Григоренко. – москва: Машиностроение, 1975. – 480 с.
8. Ключников Ю. В. АВІАЦІЙНІ МАТЕРІАЛИ ТА ЇХ ТЕХНОЛОГІЇ КОНСПЕКТ ЛЕКЦІЙ [Електронний ресурс] / Ю. В. Ключников, О. Т. Сердітов, В. Л. Дубнюк // КПІ ім. Ігоря Сікорського. – 2022. – Режим доступу до ресурсу: <https://ela.kpi.ua/handle/123456789/48971>.
9. Getting Started with NX Open [Електронний ресурс] // Siemens. – 2019. – Режим доступу до ресурсу: https://docs.plm.automation.siemens.com/data_services/resources/nx/1899/nx_api/custom/en_US/gs_nx_open/NXOpen_Getting_Started.pdf.
10. Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с. Режим доступу до ресурсу: <https://ela.kpi.ua/handle/123456789/35763>

11. Siemens NX Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.sw.siemens.com/en-US/documents/209349590/PL20220512394070742>.
12. Англійсько-український словник авіаційних термінів / Р. О. Гільченко. – К.: НАУ, 2009. 280 с.

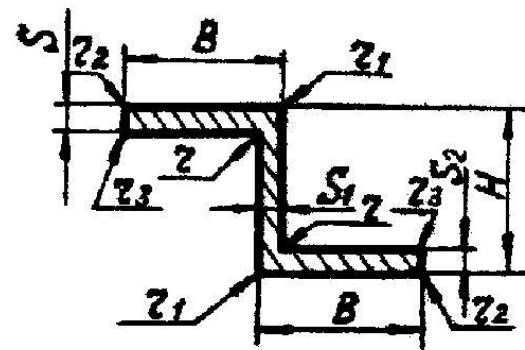
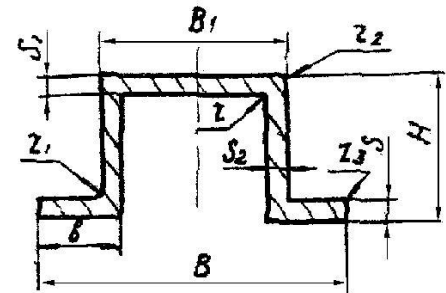
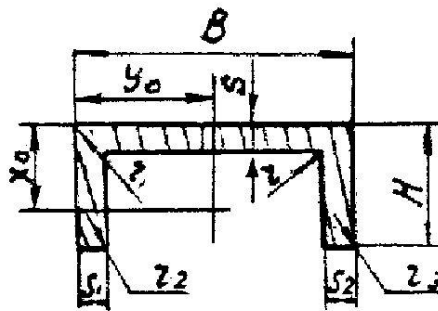
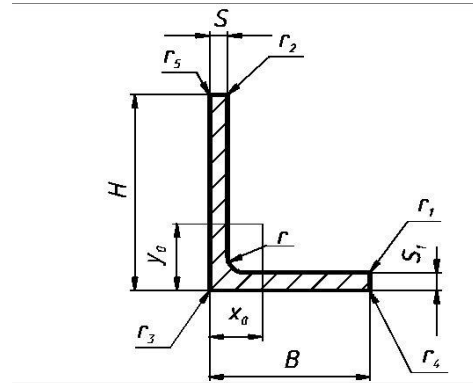
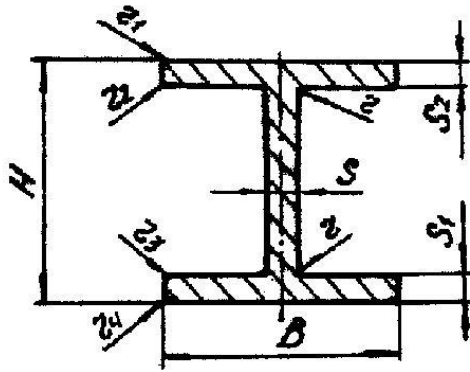
Додаток 1



Додаток 2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Назва																						
2	430027	35	40	4	10	5	0	0	0	0	0												
3	430013	23	38	1,2	1,2	1,2	0	0	0	0	0												
4	430030	36	70	31,5	4,5	4,5	0	0	0	0	0												
5																							
6																							
7																							
8																							
9																							
10																							
11																							
12																							
13																							
14																							
15																							
16																							
17																							
18																							
19																							
20																							
21																							
22																							
23																							
24																							
25																							
26																							
27																							
28																							
29																							

Додаток 3



Додаток 4

Створення шпангоута (Standard Tab):

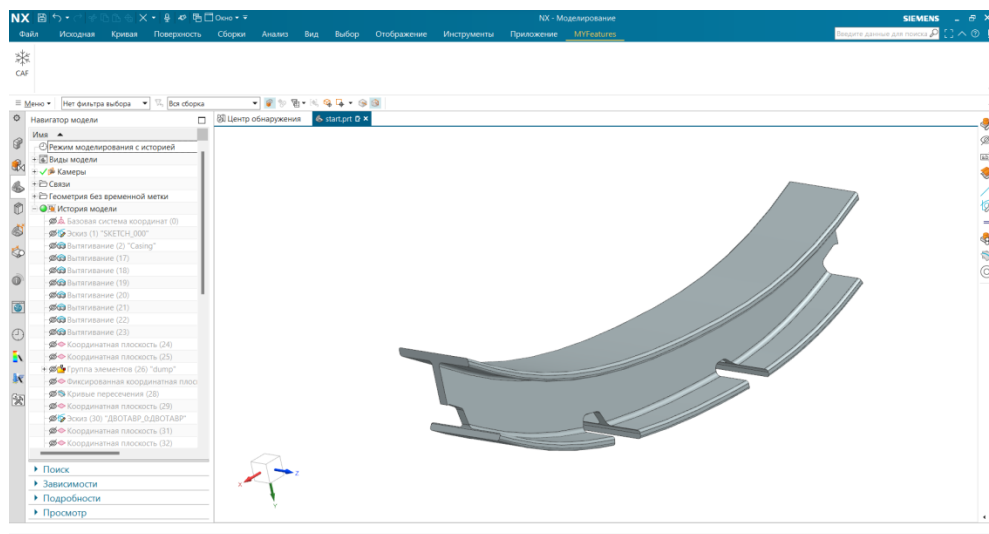
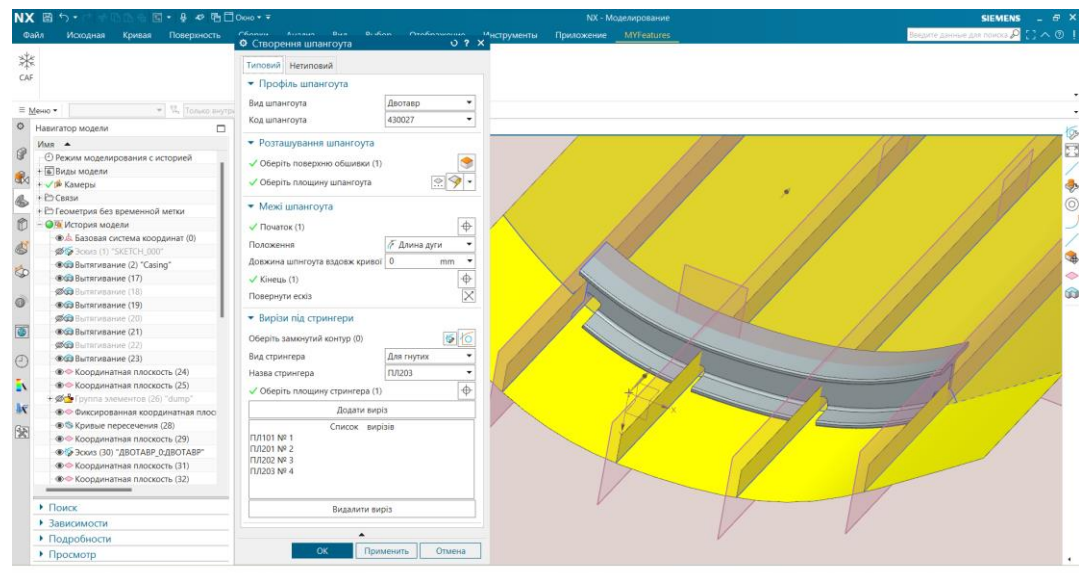
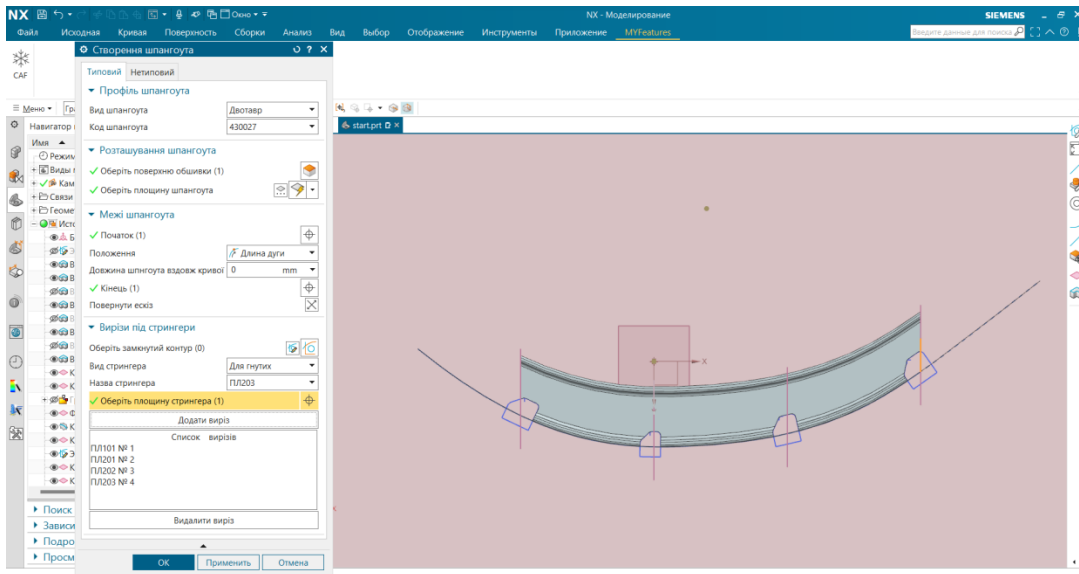
- Типовий** / Нетиповий
- Профіль шпангоута**
 - Вид шпангоута
 - Код шпангоута
- Розташування шпангоута**
 - * Оберіть поверхню обшивки (0)
 - * Оберіть площину шпангоута
- Межі шпангоута**
 - Повернути ескіз
 - * Початок (0)
 - Положення: Длина дуги
 - Довжина шпангоута вздовж кривої: 0 mm
 - * Кінець (0)
- Вирізи під стрингери**
 - Оберіть замкнутий контур (0)
 - Вид стрингера
 - Назва стрингера
 - Оберіть площину стрингера (0)
 - Додати виріз
 - Список вирізів
 - Видалити виріз

Створення шпангоута (Non-standard Tab):

- Типовий** / Нетиповий
- Стінка шпангоута**
 - Товщина: 0.0000
 - * Оберіть ескіз (0)
- Полічки шпангоута**
 - Довжина: 0.0000
 - Товщина: 0.0000
 - Кут: 0 °
 - Оберіть грань поверхні обшивки (0)
 - Оберіть ребро (0)
 - Додати полицку
 - Видалити полицку
- Вирізи під стрингер**
 - Оберіть замкнутий контур (0)
 - Вид стрингера
 - Тип стрингера
 - Оберіть площину стрингера (0)
 - Додати виріз
 - Видалити виріз

Buttons: OK, Применить, Отмена

Додаток 5



Додаток 6

```
// Created by: lytvj
// Version: NX 2212
using System;
using NXOpen;
using NXOpen.BlockStyler;
using NXOpenUI;
using System.IO;
using System.Collections.Generic;
using Excel = Microsoft.Office.Interop.Excel;
using NXOpen.UF;
using NXOpen.Features;
using System.Linq;
using NXOpen.Features.SheetMetal;
using static NXOpen.CAE.PostColorbarValueParameters;

public class BlockStyler2
{
    private static Session theSession = null;
    private static UFSession theUFSession = null;
    private static NXOpen.Part workPart = null;
    private static NXOpen.Part displayPart = null;
    private static UI theUI = null;
    private static ListingWindow lw = null;
    private string theDlxFileName;
    private NXOpen.BlockStyler.BlockDialog theDialog;
    private NXOpen.BlockStyler.TabControl tabControl;// Block type: Tabs Page
    private NXOpen.BlockStyler.Group tabPage1;
    private NXOpen.BlockStyler.Group groupPF;// Block type: Group
    private NXOpen.BlockStyler.Enumeration kindSPG;// Block type: Enumeration
    private NXOpen.BlockStyler.Enumeration typeSPG;// Block type: Enumeration
    private NXOpen.BlockStyler.Group locationGROUP;// Block type: Group
    private NXOpen.BlockStyler.FaceCollector casingFace;// Block type: Face Collector
    private NXOpen.BlockStyler.SpecifyPlane planeSPG;// Block type: Specify Plane
    private NXOpen.BlockStyler.Group limitsGROUP;// Block type: Group
    private NXOpen.BlockStyler.ReverseDirection turn_around;// Block type: Reverse Direction
    private NXOpen.BlockStyler.SelectObject limitStart;// Block type: Selection
    private NXOpen.BlockStyler.OnPathDimension on_path_dim0;// Block type: On Path Dim
    private NXOpen.BlockStyler.SelectObject limitEnd;// Block type: Selection
    private NXOpen.BlockStyler.Group group2;// Block type: Group
    private NXOpen.BlockStyler.SuperSection super_section1;// Block type: Super Section
    private NXOpen.BlockStyler.Enumeration kindStringer;// Block type: Enumeration
    private NXOpen.BlockStyler.Enumeration typeStringer;// Block type: Enumeration
    private NXOpen.BlockStyler.SelectObject planeStringer;// Block type: Selection
    private NXOpen.BlockStyler.Button Addbutton1;// Block type: Button
    private NXOpen.BlockStyler.ListBox list_box0;// Block type: List Box
    private NXOpen.BlockStyler.Button Deletebutton1;// Block type: Button
    private NXOpen.BlockStyler.Group tabPage;
    private NXOpen.BlockStyler.Group group;// Block type: Group
}
```

```

private NXOpen.BlockStyler.DoubleBlock thickness;// Block type: Double
private NXOpen.BlockStyler.SelectObject selectionFrame;// Block type: Selection
private NXOpen.BlockStyler.Group group1;// Block type: Group
private NXOpen.BlockStyler.DoubleBlock length;// Block type: Double
private NXOpen.BlockStyler.DoubleBlock thicknesscab;// Block type: Double
private NXOpen.BlockStyler.AngularDimension angular_dim0;// Block type: Angular Dim
private NXOpen.BlockStyler.FaceCollector face_select0;// Block type: Face Collector
private NXOpen.BlockStyler.CurveCollector edge_select0;// Block type: Curve Collector
private NXOpen.BlockStyler.Button buttonAdd;// Block type: Button
private NXOpen.BlockStyler.ListBox list_box01;// Block type: List Box
private NXOpen.BlockStyler.Button buttonDelete;// Block type: Button
private NXOpen.BlockStyler.Group group3;// Block type: Group
private NXOpen.BlockStyler.SuperSection super_section2;// Block type: Super Section
private NXOpen.BlockStyler.Enumeration kindStringer2;// Block type: Enumeration
private NXOpen.BlockStyler.Enumeration typeStringer2;// Block type: Enumeration
private NXOpen.BlockStyler.SelectObject planeStringer2;// Block type: Selection
private NXOpen.BlockStyler.Button Addbutton2;// Block type: Button
private NXOpen.BlockStyler.ListBox list_box02;// Block type: List Box
private NXOpen.BlockStyler.Button Deletebutton2;// Block type: Button
public static readonly int EntityType_AllowFaces = (1 << 4);
public static readonly int EntityType_AllowDatums = (1 << 5);
public static readonly int EntityType_AllowBodies = (1 << 6);
public static readonly int FaceRules_SingleFace = (1 << 0);
public static readonly int FaceRules_RegionFaces = (1 << 1);
public static readonly int FaceRules_TangentFaces = (1 << 2);
public static readonly int FaceRules_TangentRegionFaces = (1 << 3);
public static readonly int FaceRules_BodyFaces = (1 << 4);
public static readonly int FaceRules_FeatureFaces = (1 << 5);
public static readonly int FaceRules_AdjacentFaces = (1 << 6);
public static readonly int FaceRules_ConnectedBlendFaces = (1 << 7);
public static readonly int FaceRules_AllBlendFaces = (1 << 8);
public static readonly int FaceRules_RibFaces = (1 << 9);
public static readonly int FaceRules_SlotFaces = (1 << 10);
public static readonly int FaceRules_BossandPocketFaces = (1 << 11);
public static readonly int FaceRules_MergedRibFaces = (1 << 12);
public static readonly int FaceRules_RegionBoundaryFaces = (1 << 13);
public static readonly int FaceRules_FaceandAdjacentFaces = (1 << 14);
public static readonly int FaceRules_HoleFaces = (1 << 15);
public static readonly int SnapPointTypesEnabled_UserDefined = (1 << 0);
public static readonly int SnapPointTypesEnabled_Inferred = (1 << 1);
public static readonly int SnapPointTypesEnabled_ScreenPosition = (1 << 2);
public static readonly int SnapPointTypesEnabled_EndPoint = (1 << 3);
public static readonly int SnapPointTypesEnabled_MidPoint = (1 << 4);
public static readonly int SnapPointTypesEnabled_ControlPoint = (1 << 5);
public static readonly int SnapPointTypesEnabled_Intersection = (1 << 6);
public static readonly int SnapPointTypesEnabled_ArcCenter = (1 << 7);
public static readonly int SnapPointTypesEnabled_QuadrantPoint = (1 << 8);
public static readonly int SnapPointTypesEnabled_ExistingPoint = (1 << 9);
public static readonly int SnapPointTypesEnabled_PointonCurve = (1 << 10);
public static readonly int SnapPointTypesEnabled_PointonSurface = (1 << 11);

```

```

public static readonly int SnapPointTypesEnabled_PointConstructor = (1 << 12);
public static readonly int SnapPointTypesEnabled_TwocurveIntersection = (1 << 13);
public static readonly int SnapPointTypesEnabled_TangentPoint = (1 << 14);
public static readonly int SnapPointTypesEnabled_Poles = (1 << 15);
public static readonly int SnapPointTypesEnabled_BoundedGridPoint = (1 << 16);
public static readonly int SnapPointTypesEnabled_FacetVertexPoint = (1 << 17);
public static readonly int SnapPointTypesEnabled_DefiningPoint = (1 << 18);
public static readonly int SnapPointTypesOnByDefault_EndPoint = (1 << 3);
public static readonly int SnapPointTypesOnByDefault_MidPoint = (1 << 4);
public static readonly int SnapPointTypesOnByDefault_ControlPoint = (1 << 5);
public static readonly int SnapPointTypesOnByDefault_Intersection = (1 << 6);
public static readonly int SnapPointTypesOnByDefault_ArcCenter = (1 << 7);
public static readonly int SnapPointTypesOnByDefault_QuadrantPoint = (1 << 8);
public static readonly int SnapPointTypesOnByDefault_ExistingPoint = (1 << 9);
public static readonly int SnapPointTypesOnByDefault_PointonCurve = (1 << 10);
public static readonly int SnapPointTypesOnByDefault_PointonSurface = (1 << 11);
public static readonly int SnapPointTypesOnByDefault_PointConstructor = (1 << 12);
public static readonly int SnapPointTypesOnByDefault_BoundedGridPoint = (1 << 16);
public static readonly int SnapPointTypesOnByDefault_FacetVertexPoint = (1 << 17);
public static readonly int SnapPointTypesOnByDefault_UserDefined = (1 << 0);
public static readonly int SnapPointTypesOnByDefault_Inferred = (1 << 1);
public static readonly int SnapPointTypesOnByDefault_ScreenPosition = (1 << 2);
public static readonly int SnapPointTypesOnByDefault_TwocurveIntersection = (1 << 13);
public static readonly int SnapPointTypesOnByDefault_TangentPoint = (1 << 14);
public static readonly int SnapPointTypesOnByDefault_Poles = (1 << 15);
public static readonly int SnapPointTypesOnByDefault_DefiningPoint = (1 << 18);
public static readonly int OptionMask_Length = (1 << 0);
public static readonly int OptionMask_Percent = (1 << 1);
public static readonly int OptionMask_Parameter = (1 << 2);
public static readonly int OptionMask_ThroughPoint = (1 << 3);
public static readonly int EntityType_AllowEdgesandCurves = (1 << 2);
public static readonly int EntityType_AllowPoints = (1 << 3);
public static readonly int CurveRules_SingleCurve = (1 << 0);
public static readonly int CurveRules_ConnectedCurves = (1 << 1);
public static readonly int CurveRules_TangentCurves = (1 << 2);
public static readonly int CurveRules_FaceEdges = (1 << 3);
public static readonly int CurveRules_SheetEdges = (1 << 5);
public static readonly int CurveRules_FeatureCurves = (1 << 6);
public static readonly int CurveRules_InferCurves = (1 << 7);
public static readonly int CurveRules_RegionBoundaryCurves = (1 << 11);
public static readonly int CurveRules_CurvesinGroup = (1 << 12);
public static readonly int CurveRules_OuterEdgesofFaces = (1 << 13);
public static readonly int CurveRules_RibTopFaceEdges = (1 << 14);
public static readonly int CurveRules_FeatureIntersectionEdges = (1 << 15);
public static readonly int EntityType_AllowEdges = (1 << 0);
public static readonly int EntityType_AllowCurves = (1 << 2);
public static readonly int EntityType_AllowPoint = (1 << 3);
public static readonly int CurveRules_BodyEdges = (1 << 4);
public static readonly int CurveRules_VertexEdges = (1 << 8);
public static readonly int CurveRules_VertexTangentEdges = (1 << 9);

```

```

#region Variavles
public static Excel._Application myEx;
public static Excel.Workbook workbook;
public static Excel.Workbook workbook2;
public static NXOpen.Part partCK;
public static NXOpen.Part partSTR;
List<string> kindsFRAME = new List<string>();
List<int> kinds_visibility = new List<int>();
Dictionary<string, List<string>> typesFRAME = new Dictionary<string, List<string>>();
Dictionary<string, List<int>> types_visibility = new Dictionary<string, List<int>>();
List<string> kindsSTRINGER = new List<string>();
List<int> kinds_visibilitySTR = new List<int>();
Dictionary<string, List<string>> typesSTRINGER = new Dictionary<string, List<string>>();
Dictionary<string, List<int>> types_visibilitySTR = new Dictionary<string, List<int>>();
NXOpen.Features.IntersectionCurve intersectionCurve1;
Point Point;
NXOpen.Features.SketchFeature sketchFRAME;
NXOpen.Features.SketchFeature sketchSTRINGER;
NXOpen.Features.Swept swept;
List<string> cutSTRlist = new List<string>();
List<Extrude> cutEXTlist = new List<Extrude>();
Dictionary<string, Extrude> cutINFO = new Dictionary<string, Extrude>();
Extrude wall = null;
#endregion
public BlockStyler2()
{
    try
    {
        theSession = Session.GetSession();
        theUFSession = UFSession.GetUFSession();
        theUI = UI.GetUI();
        workPart = theSession.Parts.Work;
        displayPart = theSession.Parts.Display;
        lw = theSession.ListingWindow;
        theDlxFileName = "D:\\KPI\\CAF\\DLXs\\6.dlx";
        theDialog = theUI.CreateDialog(theDlxFileName);
        theDialog.AddApplyHandler(new NXOpen.BlockStyler.BlockDialog.Apply(apply_cb));
        theDialog.AddOkHandler(new NXOpen.BlockStyler.BlockDialog.Ok(ok_cb));
        theDialog.AddUpdateHandler(new NXOpen.BlockStyler.BlockDialog.Update(update_cb));
        theDialog.AddInitializeHandler(new
NXOpen.BlockStyler.BlockDialog.Initialize(initialize_cb));
        theDialog.AddDialogShownHandler(new
NXOpen.BlockStyler.BlockDialog.DialogShown(dialogShown_cb));
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
}

```

```

public static void Main()
{
    BlockStyler2 theBlockStyler2 = null;
    try
    {
        theBlockStyler2 = new BlockStyler2();
        // The following method shows the dialog immediately
        theBlockStyler2.Launch();
    }
    catch (Exception ex)
    {
        //---- Enter your exception handling code here ----
        theUI.NXMessageBox.Show("Block Styler", NXMessageBox.DialogType.Error,
ex.ToString());
    }
    finally
    {
        if (theBlockStyler2 != null)
            theBlockStyler2.Dispose();
        theBlockStyler2 = null;
    }
}

public static int GetUnloadOption(string arg)
{
    return System.Convert.ToInt32(Session.LibraryUnloadOption.Immediately);
}

public static void UnloadLibrary(string arg)
{
    try
    {
        //---- Enter your code here ----
    }
    catch (Exception ex)
    {
        //---- Enter your exception handling code here ----
        theUI.NXMessageBox.Show("Block Styler", NXMessageBox.DialogType.Error,
ex.ToString());
    }
}

public NXOpen.BlockStyler.BlockDialog.DialogResponse Launch()
{
    NXOpen.BlockStyler.BlockDialog.DialogResponse dialogResponse =
NXOpen.BlockStyler.BlockDialog.DialogResponse.Invalid;
    try
    {
        dialogResponse = theDialog.Launch();
    }
    catch (Exception ex)
    {

```



```

        //---- Enter your exception handling code here ----
        theUI.NXMessageBox.Show("Block Styler", NXMessageBox.DialogType.Error,
ex.ToString());
    }
    return dialogResponse;
}

//-----
//Method Name: Dispose
//-----
public void Dispose()
{
    if (theDialog != null)
    {
        try
        {
            partCK.Close(BasePart.CloseWholeTree.True, BasePart.CloseModified.UseResponses,
null);
            partSTR.Close(BasePart.CloseWholeTree.True, BasePart.CloseModified.UseResponses,
null);
        }
        catch { }

        theDialog.Dispose();
        theDialog = null;
    }
}

//-----
//-----Block UI Styler Callback Functions-----
//-----

//-----
//Callback Name: initialize_cb
//-----
public void initialize_cb()
{
    try
    {
        tabControl
(NXOpen.BlockStyler.TabControl)theDialog.TopBlock.FindBlock("tabControl");
        tabPage1 = (NXOpen.BlockStyler.Group)theDialog.TopBlock.FindBlock("tabPage1");
        groupPF = (NXOpen.BlockStyler.Group)theDialog.TopBlock.FindBlock("groupPF");
        kindSPG
(NXOpen.BlockStyler.Enumeration)theDialog.TopBlock.FindBlock("kindSPG");
        typeSPG
(NXOpen.BlockStyler.Enumeration)theDialog.TopBlock.FindBlock("typeSPG");
        locationGROUP
(NXOpen.BlockStyler.Group)theDialog.TopBlock.FindBlock("locationGROUP");
    }
}

```

```

    casingFace =
(NXOpen.BlockStyler.FaceCollector)theDialog.TopBlock.FindBlock("casingFace");
    planeSPG =
(NXOpen.BlockStyler.SpecifyPlane)theDialog.TopBlock.FindBlock("planeSPG");
    limitsGROUP =
(NXOpen.BlockStyler.Group)theDialog.TopBlock.FindBlock("limitsGROUP");
    turn_around =
(NXOpen.BlockStyler.ReverseDirection)theDialog.TopBlock.FindBlock("turn_around");
    limitStart =
(NXOpen.BlockStyler.SelectObject)theDialog.TopBlock.FindBlock("limitStart");
    on_path_dim0 =
(NXOpen.BlockStyler.OnPathDimension)theDialog.TopBlock.FindBlock("on_path_dim0");
    limitEnd = (NXOpen.BlockStyler.SelectObject)theDialog.TopBlock.FindBlock("limitEnd");
    group2 = (NXOpen.BlockStyler.Group)theDialog.TopBlock.FindBlock("group2");
    super_section1 =
(NXOpen.BlockStyler.SuperSection)theDialog.TopBlock.FindBlock("super_section1");
    kindStringer =
(NXOpen.BlockStyler.Enumeration)theDialog.TopBlock.FindBlock("kindStringer");
    typeStringer =
(NXOpen.BlockStyler.Enumeration)theDialog.TopBlock.FindBlock("typeStringer");
    planeStringer =
(NXOpen.BlockStyler.SelectObject)theDialog.TopBlock.FindBlock("planeStringer");
    Addbutton1 =
(NXOpen.BlockStyler.Button)theDialog.TopBlock.FindBlock("Addbutton1");
    list_box0 = (NXOpen.BlockStyler.ListBox)theDialog.TopBlock.FindBlock("list_box0");
    Deletebutton1 =
(NXOpen.BlockStyler.Button)theDialog.TopBlock.FindBlock("Deletebutton1");
    tabPage = (NXOpen.BlockStyler.Group)theDialog.TopBlock.FindBlock("tabPage");
    group = (NXOpen.BlockStyler.Group)theDialog.TopBlock.FindBlock("group");
    thickness =
(NXOpen.BlockStyler.DoubleBlock)theDialog.TopBlock.FindBlock("thickness");
    selectionFrame =
(NXOpen.BlockStyler.SelectObject)theDialog.TopBlock.FindBlock("selectionFrame");
    group1 = (NXOpen.BlockStyler.Group)theDialog.TopBlock.FindBlock("group1");
    length = (NXOpen.BlockStyler.DoubleBlock)theDialog.TopBlock.FindBlock("length");
    thicknesscab =
(NXOpen.BlockStyler.DoubleBlock)theDialog.TopBlock.FindBlock("thicknesscab");
    angular_dim0 =
(NXOpen.BlockStyler.AngularDimension)theDialog.TopBlock.FindBlock("angular_dim0");
    face_select0 =
(NXOpen.BlockStyler.FaceCollector)theDialog.TopBlock.FindBlock("face_select0");
    edge_select0 =
(NXOpen.BlockStyler.CurveCollector)theDialog.TopBlock.FindBlock("edge_select0");
    buttonAdd = (NXOpen.BlockStyler.Button)theDialog.TopBlock.FindBlock("buttonAdd");
    list_box01 = (NXOpen.BlockStyler.ListBox)theDialog.TopBlock.FindBlock("list_box01");
    buttonDelete =
(NXOpen.BlockStyler.Button)theDialog.TopBlock.FindBlock("buttonDelete");
    group3 = (NXOpen.BlockStyler.Group)theDialog.TopBlock.FindBlock("group3");
    super_section2 =
(NXOpen.BlockStyler.SuperSection)theDialog.TopBlock.FindBlock("super_section2");

```

```

        kindStringer2                                     =
(NXOpen.BlockStyler.Enumeration)theDialog.TopBlock.FindBlock("kindStringer2");
        typeStringer2                                   =
(NXOpen.BlockStyler.Enumeration)theDialog.TopBlock.FindBlock("typeStringer2");
        planeStringer2                                 =
(NXOpen.BlockStyler.SelectObject)theDialog.TopBlock.FindBlock("planeStringer2");
        Addbutton2                                     =
(NXOpen.BlockStyler.Button)theDialog.TopBlock.FindBlock("Addbutton2");
        list_box02 = (NXOpen.BlockStyler.ListBox)theDialog.TopBlock.FindBlock("list_box02");
        Deletebutton2                                  =
(NXOpen.BlockStyler.Button)theDialog.TopBlock.FindBlock("Deletebutton2");
        //-----
        //Registration of ListBox specific callbacks
        //-----
        //list_box0.SetAddHandler(new
NXOpen.BlockStyler.ListBox.AddCallback(AddCallback));

        //list_box0.SetDeleteHandler(new
NXOpen.BlockStyler.ListBox.DeleteCallback(DeleteCallback0));

        //list_box01.SetAddHandler(new
NXOpen.BlockStyler.ListBox.AddCallback(AddCallback));

        //list_box01.SetDeleteHandler(new
NXOpen.BlockStyler.ListBox.DeleteCallback(DeleteCallback));

        //list_box02.SetAddHandler(new
NXOpen.BlockStyler.ListBox.AddCallback(AddCallback));

        //list_box02.SetDeleteHandler(new
NXOpen.BlockStyler.ListBox.DeleteCallback(DeleteCallback));

        //-----
    }
    catch (Exception ex)
    {
        //---- Enter your exception handling code here ----
        theUI.NXMessageBox.Show("Block Styler", NXMessageBox.DialogType.Error,
ex.ToString());
    }
}

//-----
//Callback Name: dialogShown_cb
//This callback is executed just before the dialog launch. Thus any value set
//here will take precedence and dialog will be launched showing that value.
//-----
public void dialogShown_cb()
{
    try

```

```

{
//---- Enter your callback code here ----
lw.Open();
#region Tab1
tabControl.ActivePage = 0;
kindsFRAME.Clear();
kinds_visibility.Clear();
typesFRAME.Clear();
types_visibility.Clear();
limitStart.AddFilter(2);
limitEnd.AddFilter(2);
planeStringer.AddFilter(2);
selectionFrame.AddFilter(NXOpen.UF.UFConstants.UF_curve_group_type);
list_box0.SetListItems(new string[1] { "                Список вирізів " });
myEx = new Excel.Application();
workbook
myEx.Workbooks.Open(@"D:\KPI\CAF\CAF(v.0.0.1)\bin\Debug\ProfilesSPG.xlsx");
kindsFRAME.Add("");
kinds_visibility.Add(1);
for (int i = 1; i <= workbook.Worksheets.Count; i++)
{
    kindsFRAME.Add(workbook.Worksheets[i].Name);
    kinds_visibility.Add(1);
    Excel.Worksheet worksheet = workbook.Worksheets[i];
    List<string> temp_name = new List<string>();
    List<int> temp_visab = new List<int>();
    temp_name.Add("");
    temp_visab.Add(1);
    for (int r = 2; r < worksheet.Rows.Count; r++)
    {
        if (worksheet.Cells[r, 1].Text.ToString() != "")
        {
            temp_name.Add(worksheet.Cells[r, 1].Text.ToString());
            temp_visab.Add(1);
        }
        else
        {
            break;
        }
    }
    typesFRAME.Add(workbook.Worksheets[i].Name, temp_name);
    types_visibility.Add(workbook.Worksheets[i].Name, temp_visab);
}

kindSPG.SetEnumMembers(kindsFRAME.ToArray());
kindSPG.ValueAsString = "";
PartLoadStatus partLoadStatus;
partCK = theSession.Parts.Open(@"D:\KPI\CAF\CAF(v.0.0.1)\bin\Debug\ProfilesSPG.prt",
out partLoadStatus);
workbook.Close(false);

```

```

System.Runtime.InteropServices.Marshal.ReleaseComObject(workbook);
System.Runtime.InteropServices.Marshal.FinalReleaseComObject(workbook);
workbook2
myEx.Workbooks.Open(@"D:\KPI\CAF\CAF(v.0.0.1)\bin\Debug\ProfilesSTR.xlsx");
kindsSTRINGER.Add("");
kinds_visibilitySTR.Add(1);
for (int i = 1; i <= workbook2.Worksheets.Count; i++)
{
    kindsSTRINGER.Add(workbook2.Worksheets[i].Name);
    kinds_visibilitySTR.Add(1);
    Excel.Worksheet worksheet = workbook2.Worksheets[i];
    List<string> temp_name = new List<string>();
    List<int> temp_visab = new List<int>();
    temp_name.Add("");
    temp_visab.Add(1);
    for (int r = 1; r < worksheet.Rows.Count; r++)
    {
        if (worksheet.Cells[r, 1].Text.ToString() != "")
        {
            temp_name.Add(worksheet.Cells[r, 1].Text.ToString());
            temp_visab.Add(1);
        }
        else
        {
            break;
        }
    }
    typesSTRINGER.Add(workbook2.Worksheets[i].Name, temp_name);
    types_visibilitySTR.Add(workbook2.Worksheets[i].Name, temp_visab);
}
kindStringer.SetEnumMembers(kindsSTRINGER.ToArray());
kindStringer.ValueAsString = "";
partSTR = theSession.Parts.Open(@"D:\KPI\CAF\CAF(v.0.0.1)\bin\Debug\PROFFIN.prt",
out partLoadStatus);
//workbook.Save();
workbook2.Close(false);

myEx.Application.Quit();
myEx.Quit();
System.Runtime.InteropServices.Marshal.ReleaseComObject(workbook2);
System.Runtime.InteropServices.Marshal.ReleaseComObject(myEx);
System.Runtime.InteropServices.Marshal.FinalReleaseComObject(workbook2);
System.Runtime.InteropServices.Marshal.FinalReleaseComObject(myEx);
#endregion

thickness.Value = 0;
thicknesscab.Value = 0; ;
length.Value = 0;

```

```

angular_dim0.Value = 0;

#region Enable = false
thickness.Enable = false;
selectionFrame.Enable = false;
length.Enable = false;
thicknesscab.Enable = false;
angular_dim0.Enable = false;
face_select0.Enable = false;
edge_select0.Enable = false;
buttonAdd.Enable = false;
list_box01.Enable = false;
super_section2.Enable = false;
kindStringer2.Enable = false;
typeStringer2.Enable = false;
planeStringer2.Enable = false;
list_box02.Enable = false;
#endregion
}
catch (Exception ex)
{
    //---- Enter your exception handling code here ----
    theUI.NXMessageBox.Show("Block Styler", NXMessageBox.DialogType.Error,
ex.ToString());
}
}

//-----
//Callback Name: apply_cb
//-----
public int apply_cb()
{
    int errorCode = 0;
    try
    {
        //---- Enter your callback code here ----
        partCK.Close(BasePart.CloseWholeTree.True, BasePart.CloseModified.UseResponses,
null);
        partSTR.Close(BasePart.CloseWholeTree.True, BasePart.CloseModified.UseResponses,
null);
    }
    catch (Exception ex)
    {
        //---- Enter your exception handling code here ----
        errorCode = 1;
        theUI.NXMessageBox.Show("Block Styler", NXMessageBox.DialogType.Error,
ex.ToString());
    }
    return errorCode;
}
}

```

```

//-----
//Callback Name: update_cb
//-----
public int update_cb(NXOpen.BlockStyler.UIBlock block)
{
    try
    {
        if (block == tabControl)
        {
            if (tabControl.ActivePage == 0)
            {
                kindSPG.Enable = true;
                typeSPG.Enable = true;
                casingFace.Enable = true;
                planeSPG.Enable = true;
                turn_around.Enable = true;
                limitStart.Enable = true;
                on_path_dim0.Enable = true;
                limitEnd.Enable = true;
                super_section1.Enable = true;
                kindStringer.Enable = true;
                typeStringer.Enable = true;
                planeStringer.Enable = true;
                Addbutton1.Enable = true;
                list_box0.Enable = true;
                Deletebutton1.Enable = true;
                //-----//
                thickness.Enable = false;
                selectionFrame.Enable = false;
                length.Enable = false;
                thicknesscab.Enable = false;
                angular_dim0.Enable = false;
                face_select0.Enable = false;
                edge_select0.Enable = false;
                buttonAdd.Enable = false;
                list_box01.Enable = false;
                buttonDelete.Enable = false;
                super_section2.Enable = false;
                kindStringer2.Enable = false;
                typeStringer2.Enable = false;
                planeStringer2.Enable = false;
                Addbutton2.Enable = false;
                list_box02.Enable = false;
                Deletebutton2.Enable = false;
            }
            else if (tabControl.ActivePage == 1)
            {
                kindSPG.Enable = false;
                typeSPG.Enable = false;
                casingFace.Enable = false;
            }
        }
    }
}

```

```

planeSPG.Enable = false;
turn_around.Enable = false;
limitStart.Enable = false;
on_path_dim0.Enable = false;
limitEnd.Enable = false;
super_section1.Enable = false;
kindStringer.Enable = false;
typeStringer.Enable = false;
planeStringer.Enable = false;
Addbutton1.Enable = false;
list_box0.Enable = false;
Deletebutton1.Enable = false;
//-----//
thickness.Enable = true;
selectionFrame.Enable = true;
length.Enable = true;
thicknesscab.Enable = true;
angular_dim0.Enable = true;
face_select0.Enable = true;
edge_select0.Enable = true;
buttonAdd.Enable = true;
list_box01.Enable = true;
buttonDelete.Enable = true;
super_section2.Enable = true;
kindStringer2.Enable = true;
typeStringer2.Enable = true;
planeStringer2.Enable = true;
Addbutton2.Enable = true;
list_box02.Enable = true;
Deletebutton2.Enable = true;
}
}
else if (block == kindSPG)
{
    foreach (string item in typesFRAME.Keys)
    {
        if (kindSPG.ValueAsString == item)
            typeSPG.SetEnumMembers(typesFRAME[item].ToArray());
    }
    types_visibility[kindSPG.ValueAsString][0] = 1;
    typeSPG.ValueAsString = "";
    kinds_visibility[0] = 0;
    kindSPG.SetEnumVisibility(kinds_visibility.ToArray());
    #region

    #endregion
}
else if (block == typeSPG)
{
    types_visibility[kindSPG.ValueAsString][0] = 0;

```



```

    typeSPG.SetEnumVisibility(types_visibility[kindSPG.ValueAsString].ToArray());
}
else if (block == casingFace)
{
    planeSPG.Focus();

}
else if (block == planeSPG)
{
    try
    {
        NXOpen.Features.IntersectionCurveBuilder intersectionCurveBuilder1 =
workPart.Features.CreateIntersectionCurveBuilder(null);
        intersectionCurveBuilder1.CurveFitData.Tolerance = 0.01;
        intersectionCurveBuilder1.CurveFitData.AngleTolerance = 0.5;
        Plane plane = (Plane)planeSPG.GetSelectedObjects()[0];
        DatumPlane datumPlane = workPart.Datums.CreateFixedDatumPlane(plane.Origin,
plane.Matrix);
        DatumPlane[] planes1 = new DatumPlane[1] { datumPlane };
        FaceDumbRule faceDumbRule1;
        faceDumbRule1 = workPart.ScRuleFactory.CreateRuleFaceDatum(planes1);
        SelectionIntentRule[] rules1 = new SelectionIntentRule[1];
        rules1[0] = faceDumbRule1;
        intersectionCurveBuilder1.FirstFace.ReplaceRules(rules1, false);
        intersectionCurveBuilder1.FirstSet.Add(planeSPG.GetSelectedObjects()[0]);

        Face[] faces1 = new Face[1] { (Face)casingFace.GetSelectedObjects()[0] };
        FaceDumbRule faceDumbRule2;
        faceDumbRule2 = workPart.ScRuleFactory.CreateRuleFaceDumb(faces1);
        SelectionIntentRule[] rules2 = new SelectionIntentRule[1];
        rules2[0] = faceDumbRule2;
        intersectionCurveBuilder1.SecondFace.ReplaceRules(rules2, false);
        intersectionCurveBuilder1.SecondSet.Add(casingFace.GetSelectedObjects()[0]);
        intersectionCurve1 =
(NXOpen.Features.IntersectionCurve)intersectionCurveBuilder1.Commit();
        intersectionCurveBuilder1.Destroy();
    }
    catch
    {
        theUI.NXMessageBox.Show("Увага", NXMessageBox.DialogType.Warning,
"Поверхня обшивки та площина шапнгоута не перетинаються. Спробуйте обрати ще раз.");
        casingFace.SetSelectedObjects(new TaggedObject[0] { });
        planeSPG.SetSelectedObjects(new TaggedObject[0] { });
    }
    limitStart.Focus();
}
else if (block == turn_around)
{
    if (sketchFRAME != null)
    {

```

```

SketchInPlaceBuilder          sketchInPlaceBuilderNEW          =
workPart.Sketches.CreateSketchInPlaceBuilder2(sketchFRAME.Sketch);
    if (sketchInPlaceBuilderNEW.ReverseAxis)
    {
        sketchInPlaceBuilderNEW.ReverseAxis = false;
    }
    else
    {
        sketchInPlaceBuilderNEW.ReverseAxis = true;
    }
    sketchInPlaceBuilderNEW.Commit();
    sketchInPlaceBuilderNEW.Destroy();
}
}
else if (block == limitStart)
{
    Face face = (Face)limitStart.GetSelectedObjects()[0];
    DatumPlaneBuilder          datumPlaneBuilder          =
workPart.Features.CreateDatumPlaneBuilder(null);
    datumPlaneBuilder.SetFaceAndOffset(face, 0.05, "face");
    DatumPlaneFeature          datumPlaneFeature          =
(DatumPlaneFeature)datumPlaneBuilder.Commit();
    Plane plane = workPart.Planes.CreatePlane(datumPlaneFeature);
    Curve curve = (Curve)intersectionCurve1.GetEntities()[0];
    Spline spline;
    Line line;
    Direction direction;
    try
    {
        spline = (Spline)intersectionCurve1.GetEntities()[0];
        IParameterizedSurface          parameterizedSurface          =
(IPParameterizedSurface)limitStart.GetSelectedObjects()[0];
        Point hpoint1 = workPart.Points.CreatePoint(spline.Get3DPoles()[0]);
        Point hpoint2 = workPart.Points.CreatePoint(spline.Get3DPoles()[0]);
        Point = workPart.Points.CreatePoint(parameterizedSurface, spline, hpoint1, hpoint2,
SmartObject.UpdateOption.WithinModeling);
        direction          =          workPart.Directions.CreateDirection(spline,          Point,
Direction.OnCurveOption.Tangent, Sense.Reverse, SmartObject.UpdateOption.WithinModeling);
    }
    catch
    {
        line = (Line)intersectionCurve1.GetEntities()[0];
        IParameterizedSurface          parameterizedSurface          =
(IPParameterizedSurface)limitStart.GetSelectedObjects()[0];
        Point hpoint1 = workPart.Points.CreatePoint(line.StartPoint);
        Point hpoint2 = workPart.Points.CreatePoint(line.StartPoint);
        Point = workPart.Points.CreatePoint(parameterizedSurface, line, hpoint1, hpoint2,
SmartObject.UpdateOption.WithinModeling);
        direction          =          workPart.Directions.CreateDirection(line,          Point,
Direction.OnCurveOption.Tangent, Sense.Reverse, SmartObject.UpdateOption.WithinModeling);

```

```

    }
    for (int i = 0; i < partCK.Features.ToArray().Length; i++)
    {
        if (kindSPG.ValueAsString.ToUpper() == partCK.Features.ToArray()[i].Name &&
kindSPG.ValueAsString.ToUpper() != "")
        {
            NXOpen.Features.SketchFeature          sketch          =
(NXOpen.Features.SketchFeature)partCK.Features.ToArray()[i];
            NXObject[] features = new NXObject[1] { sketch };
            NXOpen.Features.CopyPasteBuilder      copyPasteBuilder =
workPart.Features.CreateCopyPasteBuilder(features);

copyPasteBuilder.SetBuilderVersion((NXOpen.Features.CopyPasteBuilder.BuilderVersion)7);
            NXOpen.Features.FeatureReferencesBuilder featureReferencesBuilder1 =
copyPasteBuilder.GetFeatureReferences();
            featureReferencesBuilder1.AutomaticMatch(true);
            copyPasteBuilder.ExpressionOption    =
NXOpen.Features.CopyPasteBuilder.ExpressionTransferOption.LinkToOriginal;
            copyPasteBuilder.CopyResolveGeometry = false;
            copyPasteBuilder.Associative = true;
            sketchFRAME = (NXOpen.Features.SketchFeature)copyPasteBuilder.Commit();
            copyPasteBuilder.Destroy();
        }
    }

    SketchInPlaceBuilder          sketchInPlaceBuilderNEW          =
workPart.Sketches.CreateSketchInPlaceBuilder2(sketchFRAME.Sketch);
    sketchInPlaceBuilderNEW.AxisOrientation = AxisOrientation.Vertical;
    sketchInPlaceBuilderNEW.PlaneOption = Sketch.PlaneOption.ExistingPlane;
    sketchInPlaceBuilderNEW.SketchOrigin = Point;
    sketchInPlaceBuilderNEW.PlaneReference = plane;
    sketchInPlaceBuilderNEW.AxisReference = direction;
    sketchInPlaceBuilderNEW.Commit();
    sketchInPlaceBuilderNEW.Destroy();
}
else if (block == on_path_dim0)
{
    if (on_path_dim0.Value != 0)
    {
        limitEnd.Enable = false;
    }
    else
    {
        limitEnd.Enable = true;
    }
}
else if (block == limitEnd)
{
    Face faceEnd = (Face)limitEnd.GetSelectedObjects()[0];

```

```

DatumPlaneBuilder datumPlaneBuilderEnd =
workPart.Features.CreateDatumPlaneBuilder(null);
datumPlaneBuilderEnd.SetFaceAndOffset(faceEnd, 0.05, "face");
DatumPlaneFeature datumPlaneFeatureEnd =
(DatumPlaneFeature)datumPlaneBuilderEnd.Commit();
Plane planeEnd = workPart.Planes.CreatePlane(datumPlaneFeatureEnd);
Face faceStart = (Face)limitStart.GetSelectedObjects()[0];
DatumPlaneBuilder datumPlaneBuilderStart =
workPart.Features.CreateDatumPlaneBuilder(null);
datumPlaneBuilderStart.SetFaceAndOffset(faceStart, 0.05, "face");
DatumPlaneFeature datumPlaneFeatureStart =
(DatumPlaneFeature)datumPlaneBuilderStart.Commit();
Plane planeStart = workPart.Planes.CreatePlane(datumPlaneFeatureStart);
#region TrimCurve
Spline spline_null = null;
NXOpen.Features.TrimCurveBuilder trimCurveBuilder =
workPart.Features.CreateTrimCurveBuilder(spline_null);
trimCurveBuilder.FirstBoundingObject.Add(planeEnd);
trimCurveBuilder.FirstBoundingPlane = planeEnd;
NXOpen.SelectionIntentRuleOptions selectionIntentRuleOptions1 =
workPart.ScRuleFactory.CreateRuleOptions();
IBaseCurve[] baseCurves = new IBaseCurve[1] {
(Curve)intersectionCurve1.GetEntities()[0] };
NXOpen.CurveDumbRule curveDumbRule1 =
workPart.ScRuleFactory.CreateRuleBaseCurveDumb(baseCurves, selectionIntentRuleOptions1);
NXOpen.SelectionIntentRule[] rules1 = new NXOpen.SelectionIntentRule[1] {
curveDumbRule1 };
trimCurveBuilder.CurvesToTrim.AddToSection(rules1,
intersectionCurve1.GetEntities()[0], null, null, Point.Coordinates, Section.Mode.Create);
trimCurveBuilder.CurveEndOption = TrimCurveBuilder.CurveEndOptions.End;
NXOpen.Features.TrimCurve trimCurve =
(TrimCurve)trimCurveBuilder.CommitFeature();
trimCurveBuilder.Destroy();
#endregion
#region Swept
NXOpen.Features.SweptBuilder sweptBuilder =
workPart.Features.CreateSweptBuilder(null);
sweptBuilder.G0Tolerance = 0.01;
sweptBuilder.G1Tolerance = 0.5;
IBaseCurve[] baseCurves2 = new IBaseCurve[1] { (Curve)trimCurve.GetEntities()[0] };
NXOpen.CurveDumbRule curveDumbRule2 =
workPart.ScRuleFactory.CreateRuleBaseCurveDumb(baseCurves2, selectionIntentRuleOptions1);
NXOpen.SelectionIntentRule[] rules2 = new NXOpen.SelectionIntentRule[1] {
curveDumbRule2 };
Section section = workPart.Sections.CreateSection();
section.AddToSection(rules2, trimCurve.GetEntities()[0], null, null, Point.Coordinates,
Section.Mode.Create);
sweptBuilder.GuideList.Append(section);
sweptBuilder.BodyPreference.BodyType =
NXOpen.GeometricUtilities.FeatureOptions.BodyStyle.Solid;

```

```

List<IBaseCurve> baseCurves1 = new List<IBaseCurve>();
foreach (NXObject item in sketchFRAME.Sketch.GetAllGeometry())
{
    if (item is Curve)
    {
        baseCurves1.Add(item as Curve);
    }
}
NXOpen.CurveDumbRule curveDumbRule3 =
workPart.ScRuleFactory.CreateRuleBaseCurveDumb(baseCurves1.ToArray(),
selectionIntentRuleOptions1);
NXOpen.SelectionIntentRule[] rules3 = new NXOpen.SelectionIntentRule[1] {
curveDumbRule3 };
Section section2 = workPart.Sections.CreateSection();
section2.AddToSection(rules3, sketchFRAME.Sketch.GetAllGeometry()[0], null, null,
Point.Coordinates, Section.Mode.Create);
sweptBuilder.SectionList.Append(section2);
swept = (NXOpen.Features.Swept)sweptBuilder.CommitFeature();
sweptBuilder.Destroy();
#endregion
NXOpen.Features.TrimBodyBuilder trimBodyBuilder =
workPart.Features.CreateTrimBodyBuilder(null);
trimBodyBuilder.AddTarget(swept.GetBodies()[0]);
trimBodyBuilder.Tool = planeStart;
trimBodyBuilder.TrimDirection = TrimBodyBuilder.DirectionType.NegativeNormal;
TrimBody trimBody = (TrimBody)trimBodyBuilder.CommitFeature();
trimBodyBuilder.Destroy();

}
else if (block == super_section1)
{
}
else if (block == kindStringer)
{
    foreach (string item in typesSTRINGER.Keys)
    {
        if (kindStringer.ValueAsString == item)
            typeStringer.SetEnumMembers(typesSTRINGER[item].ToArray());
    }
    types_visibilitySTR[kindStringer.ValueAsString][0] = 1;
    typeStringer.ValueAsString = "";
    kinds_visibilitySTR[0] = 0;
    kindStringer.SetEnumVisibility(kinds_visibilitySTR.ToArray());
}
else if (block == typeStringer)
{
}
else if (block == planeStringer)
{
}
}

```

```

else if (block == Addbutton1)
{
    if (planeStringer.GetSelectedObjects().Length > 0 && typeStringer.ValueAsString != "")
    {
        NXOpen.Features.IntersectionCurveBuilder      intersectionCurveBuilder1      =
workPart.Features.CreateIntersectionCurveBuilder(null);
        intersectionCurveBuilder1.CurveFitData.Tolerance = 0.01;
        intersectionCurveBuilder1.CurveFitData.AngleTolerance = 0.5;
        Face face = (Face)planeStringer.GetSelectedObjects()[0];
        DatumPlaneBuilder      datumPlaneBuilder      =
workPart.Features.CreateDatumPlaneBuilder(null);
        datumPlaneBuilder.SetFaceAndOffset(face, 0.05, "face");
        DatumPlaneFeature      datumPlaneFeature      =
(DatumPlaneFeature)datumPlaneBuilder.Commit();
        Plane plane = workPart.Planes.CreatePlane(datumPlaneFeature);
        DatumPlane datumPlane = workPart.Datums.CreateFixedDatumPlane(plane.Origin,
plane.Matrix);
        DatumPlane[] planes1 = new DatumPlane[1] { datumPlane };
        FaceDumbRule faceDumbRule1;
        faceDumbRule1 = workPart.ScRuleFactory.CreateRuleFaceDatum(planes1);
        SelectionIntentRule[] rules1 = new SelectionIntentRule[1];
        rules1[0] = faceDumbRule1;
        intersectionCurveBuilder1.FirstFace.ReplaceRules(rules1, false);
        intersectionCurveBuilder1.FirstSet.Add(planeSPG.GetSelectedObjects()[0]);
        Face[] faces1 = new Face[1] { (Face)casingFace.GetSelectedObjects()[0] };
        FaceDumbRule faceDumbRule2;
        faceDumbRule2 = workPart.ScRuleFactory.CreateRuleFaceDumb(faces1);
        SelectionIntentRule[] rules2 = new SelectionIntentRule[1];
        rules2[0] = faceDumbRule2;
        intersectionCurveBuilder1.SecondFace.ReplaceRules(rules2, false);
        intersectionCurveBuilder1.SecondSet.Add(casingFace.GetSelectedObjects()[0]);
        IntersectionCurve      intersectionCurve      =
(NXOpen.Features.IntersectionCurve)intersectionCurveBuilder1.Commit();
        intersectionCurveBuilder1.Destroy();
        Plane planeFrame = (Plane)planeSPG.GetSelectedObjects()[0];
        Curve curve = (Curve)intersectionCurve1.GetEntities()[0];
        Spline spline;
        Line line;
        Direction direction;
        Direction      direction1      =      workPart.Directions.CreateDirection(planeFrame,
Sense.Forward, SmartObject.UpdateOption.WithinModeling);
        try
        {
            spline = (Spline)intersectionCurve1.GetEntities()[0];
            IParameterizedSurface      parameterizedSurface      =
(IPParameterizedSurface)planeStringer.GetSelectedObjects()[0];
            Point hpoint1 = workPart.Points.CreatePoint(spline.Get3DPoles()[0]);
            Point hpoint2 = workPart.Points.CreatePoint(spline.Get3DPoles()[0]);
            Point = workPart.Points.CreatePoint(parameterizedSurface, spline, hpoint1, hpoint2,
SmartObject.UpdateOption.WithinModeling);

```

```

        direction = workPart.Directions.CreateDirection(spline, Point,
Direction.OnCurveOption.Tangent, Sense.Reverse, SmartObject.UpdateOption.WithinModeling);
    }
    catch
    {
        line = (Line)intersectionCurve1.GetEntities()[0];
        IParameterizedSurface parameterizedSurface =
(IPParameterizedSurface)planeStringer.GetSelectedObjects()[0];
        Point hpoint1 = workPart.Points.CreatePoint(line.StartPoint);
        Point hpoint2 = workPart.Points.CreatePoint(line.StartPoint);
        Point = workPart.Points.CreatePoint(parameterizedSurface, line, hpoint1, hpoint2,
SmartObject.UpdateOption.WithinModeling);
        direction = workPart.Directions.CreateDirection(line, Point,
Direction.OnCurveOption.Tangent, Sense.Reverse, SmartObject.UpdateOption.WithinModeling);
    }
    for (int i = 0; i < partSTR.Features.ToArray().Length; i++)
    {
        if (typeStringer.ValueAsString.ToUpper() == partSTR.Features.ToArray()[i].Name
&& typeStringer.ValueAsString.ToUpper() != "")
        {
            NXOpen.Features.SketchFeature sketch =
(NXOpen.Features.SketchFeature)partSTR.Features.ToArray()[i];
            NXObject[] features = new NXObject[1] { sketch };
            NXOpen.Features.CopyPasteBuilder copyPasteBuilder =
workPart.Features.CreateCopyPasteBuilder(features);

            copyPasteBuilder.SetBuilderVersion((NXOpen.Features.CopyPasteBuilder.BuilderVersion)7);
            NXOpen.Features.FeatureReferencesBuilder featureReferencesBuilder1 =
copyPasteBuilder.GetFeatureReferences();
            featureReferencesBuilder1.AutomaticMatch(true);
            copyPasteBuilder.ExpressionOption =
NXOpen.Features.CopyPasteBuilder.ExpressionTransferOption.CreateNew();
            copyPasteBuilder.CopyResolveGeometry = false;
            copyPasteBuilder.Associative = true;
            sketchSTRINGER =
(NXOpen.Features.SketchFeature)copyPasteBuilder.Commit();
            copyPasteBuilder.Destroy();
        }
    }
    SketchInPlaceBuilder sketchInPlaceBuilderNEW =
workPart.Sketches.CreateSketchInPlaceBuilder2(sketchSTRINGER.Sketch);
    sketchInPlaceBuilderNEW.AxisOrientation = AxisOrientation.Horizontal;
    sketchInPlaceBuilderNEW.PlaneOption = Sketch.PlaneOption.ExistingPlane;
    sketchInPlaceBuilderNEW.SketchOrigin = Point;
    sketchInPlaceBuilderNEW.PlaneReference = planeFrame;
    sketchInPlaceBuilderNEW.AxisReference = direction;
    sketchInPlaceBuilderNEW.Commit();
    sketchInPlaceBuilderNEW.Destroy();

```

```

        NXOpen.SelectionIntentRuleOptions          selectionIntentRuleOptions1          =
workPart.ScRuleFactory.CreateRuleOptions();
        List<IBaseCurve> baseCurves1 = new List<IBaseCurve>();
        foreach (NXObject item in sketchSTRINGER.Sketch.GetAllGeometry())
        {
            if (item is Curve)
            {
                baseCurves1.Add(item as Curve);
            }
        }
        NXOpen.CurveDumbRule                    curveDumbRule3                    =
workPart.ScRuleFactory.CreateRuleBaseCurveDumb(baseCurves1.ToArray(),
selectionIntentRuleOptions1);
        NXOpen.SelectionIntentRule[] rules3 = new NXOpen.SelectionIntentRule[1] {
curveDumbRule3 };
        Section section = workPart.Sections.CreateSection();
        section.AddToSection(rules3, sketchSTRINGER.Sketch.GetAllGeometry()[0], null,
null, Point.Coordinates, Section.Mode.Create);
        NXOpen.Features.ExtrudeBuilder          extrudeBuilder                    =
workPart.Features.CreateExtrudeBuilder(null);
        extrudeBuilder.Section = section;
        extrudeBuilder.BooleanOperation.SetTargetBodies(swept.GetBodies());
        extrudeBuilder.BooleanOperation.Type    =
NXOpen.GeometricUtilities.BooleanOperation.BooleanType.Subtract;
        extrudeBuilder.Limits.StartExtend.TrimType =
NXOpen.GeometricUtilities.Extend.ExtendType.ThroughAll;
        extrudeBuilder.Limits.EndExtend.TrimType =
NXOpen.GeometricUtilities.Extend.ExtendType.ThroughAll;
        extrudeBuilder.Direction = direction1;
        Extrude extrude = (Extrude)extrudeBuilder.Commit();
        extrudeBuilder.Destroy();
        cutINFO.Add(typeStringer.ValueAsString + " № " + (cutINFO.Count + 1), extrude);
        List<string> listItems = list_box0.GetListItems().ToList();
        listItems.Add(cutINFO.Keys.Last());
        list_box0.SetListItems(listItems.ToArray());
    }
    else
    {
        theUI.NXMessageBox.Show("Увага", NXMessageBox.DialogType.Warning, "Ви не
обрали профіль вирізу.\nОберіть готовий профіль з класифікатора чи ескіз в робочій деталі");
    }
}
else if (block == list_box0)
{
    string cut_name = list_box0.GetSelectedItemStrings()[0];
    if (cut_name != "                Список вирізів ")
    {
        cutINFO[cut_name].Highlight();
    }
}
else

```



```

    {
        foreach (string item in cutINFO.Keys)
        {
            cutINFO[item].Unhighlight();
        }
    }
}
else if (block == Deletebutton1)
{
    string cut_name = list_box0.GetSelectedItemStrings()[0];
    if (cut_name != "                Список  вирізів ")
    {
        if (cut_name == "Власний виріз")
        {
            super_section1.Enable = true;
        }
        List<string> listItems = list_box0.GetListItems().ToList();
        listItems.Remove(cut_name);
        list_box0.SetListItems(listItems.ToArray());
        theSession.UpdateManager.AddToDeleteList(cutINFO[cut_name]);
        cutINFO.Remove(cut_name);
        NXOpen.Session.UndoMarkId markId1;
        markId1 = theSession.SetUndoMark(NXOpen.Session.MarkVisibility.Invisible,
"delete");
        theSession.UpdateManager.DoUpdate(markId1);
    }
}
else if (block == thickness)
{
    if (selectionFrame.GetSelectedObjects().Length != 0 && thickness.Value != 0)
    {
        double thickdel2 = thickness.Value / 2;
        string thicknessstr = thickdel2.ToString();
        SketchFeature sketch = (SketchFeature)selectionFrame.GetSelectedObjects()[0];
        NXOpen.SelectionIntentRuleOptions selectionIntentRuleOptions1 =
workPart.ScRuleFactory.CreateRuleOptions();
        List<IBaseCurve> curves = new List<IBaseCurve>();
        foreach (NXObject obj in sketch.Sketch.GetAllGeometry())
        {
            if (obj is Curve)
            {
                curves.Add((Curve)obj);
            }
        }
        NXOpen.CurveDumbRule curveDumbRule1 =
workPart.ScRuleFactory.CreateRuleBaseCurveDumb(curves.ToArray(),
selectionIntentRuleOptions1);
        NXOpen.SelectionIntentRule[] rules1 = new NXOpen.SelectionIntentRule[1] {
curveDumbRule1 };
        Section section = workPart.Sections.CreateSection();

```

```

        section.AddToSection(rules1, sketch, null, null, sketch.Sketch.Origin,
Section.Mode.Create);
        Direction direction = workPart.Directions.CreateDirection(sketch.Sketch,
Sense.Forward, SmartObject.UpdateOption.WithinModeling);
        if (thickness.Value > 0)
        {
            NXOpen.Features.SheetMetal.TabBuilder tabBuilder1 =
workPart.Features.SheetmetalManager.CreateTabFeatureBuilder(null);
            tabBuilder1.SetApplicationContext(ApplicationContext.NxSheetMetal);
            tabBuilder1.MaterialSide = TabBuilder.SectionSideOptions.Right;
            tabBuilder1.Section = section;
            tabBuilder1.ThicknessSide = TabBuilder.ThicknessSideOptions.SectionNormalSide;
            tabBuilder1.Thickness.Value = thickness.Value;
            tabBuilder1.Commit();
            tabBuilder1.Destroy();
        }
        else if (thickness.Value < 0)
        {
            NXOpen.Features.SheetMetal.TabBuilder tabBuilder1 =
workPart.Features.SheetmetalManager.CreateTabFeatureBuilder(null);
            tabBuilder1.SetApplicationContext(ApplicationContext.NxSheetMetal);
            tabBuilder1.MaterialSide = TabBuilder.SectionSideOptions.Left;
            tabBuilder1.Section = section;
            tabBuilder1.ThicknessSide = TabBuilder.ThicknessSideOptions.SectionNormalSide;
            tabBuilder1.Thickness.Value = Math.Abs(thickness.Value);
            tabBuilder1.Commit();
            tabBuilder1.Destroy();
        }
    }
}
else if (block == selectionFrame)
{
    if (selectionFrame.GetSelectedObjects()[0] is SketchFeature && thickness.Value != 0)
    {
        double thickdel2 = thickness.Value / 2;
        string thicknessstr = thickdel2.ToString();
        SketchFeature sketch = (SketchFeature)selectionFrame.GetSelectedObjects()[0];
        NXOpen.SelectionIntentRuleOptions selectionIntentRuleOptions1 =
workPart.ScRuleFactory.CreateRuleOptions();
        List<IBaseCurve> curves = new List<IBaseCurve>();
        foreach (NXObject obj in sketch.Sketch.GetAllGeometry())
        {
            if (obj is Curve)
            {
                curves.Add((Curve)obj);
            }
        }
        NXOpen.CurveDumbRule curveDumbRule1 =
workPart.ScRuleFactory.CreateRuleBaseCurveDumb(curves.ToArray(),
selectionIntentRuleOptions1);

```

```

        NXOpen.SelectionIntentRule[] rules1 = new NXOpen.SelectionIntentRule[1] {
curveDumbRule1 };
        Section section = workPart.Sections.CreateSection();
        section.AddToSection(rules1, sketch, null, null, sketch.Sketch.Origin,
Section.Mode.Create);
        Direction direction = workPart.Directions.CreateDirection(sketch.Sketch,
Sense.Forward, SmartObject.UpdateOption.WithinModeling);
        if (thickness.Value > 0)
        {
            NXOpen.Features.SheetMetal.TabBuilder tabBuilder1 =
workPart.Features.SheetmetalManager.CreateTabFeatureBuilder(null);
            tabBuilder1.SetApplicationContext(ApplicationContext.NxSheetMetal);
            tabBuilder1.MaterialSide = TabBuilder.SectionSideOptions.Right;
            tabBuilder1.Section = section;
            tabBuilder1.ThicknessSide =
TabBuilder.ThicknessSideOptions.SectionReverseNormalSide;
            tabBuilder1.Thickness.Value = thickness.Value;
            tabBuilder1.Commit();
            tabBuilder1.Destroy();
        }
        else if (thickness.Value < 0)
        {
            NXOpen.Features.SheetMetal.TabBuilder tabBuilder1 =
workPart.Features.SheetmetalManager.CreateTabFeatureBuilder(null);
            tabBuilder1.SetApplicationContext(ApplicationContext.NxSheetMetal);
            tabBuilder1.MaterialSide = TabBuilder.SectionSideOptions.Left;
            tabBuilder1.Section = section;
            tabBuilder1.ThicknessSide = TabBuilder.ThicknessSideOptions.SectionNormalSide;
            tabBuilder1.Thickness.Value = Math.Abs(thickness.Value);
            tabBuilder1.Commit();
            tabBuilder1.Destroy();
        }
    }
}
else if (block == length)
{
}
else if (block == thicknesscab)
{
}
else if (block == angular_dim0)
{
}
else if (block == face_select0)
{
}
else if (block == edge_select0)
{
}
}
}

```

```

else if (block == buttonAdd)
{
    if (length.Value == 0)
    {
        theUI.NXMessageBox.Show("Помилка",      NXMessageBox.DialogType.Warning,
"Введіть значення довжини");
        return 0;
    }
    if (thicknesscab.Value == 0)
    {
        theUI.NXMessageBox.Show("Помилка",      NXMessageBox.DialogType.Warning,
"Введіть значення товщини");
        return 0;
    }
    if (face_select0.GetSelectedObjects().Length == 0)
    {
        theUI.NXMessageBox.Show("Помилка",      NXMessageBox.DialogType.Warning,
"Оберіть грань");
        return 0;
    }
    if (edge_select0.GetSelectedObjects().Length == 0)
    {
        theUI.NXMessageBox.Show("Помилка",      NXMessageBox.DialogType.Warning,
"Оберіть ребро");
        return 0;
    }
    Face face = (Face)face_select0.GetSelectedObjects()[0];
    Edge[] edges = new Edge[edge_select0.GetSelectedObjects().Length];
    for (int i = 0; i < edges.Length; i++)
    {
        edges[i] = (Edge)edge_select0.GetSelectedObjects()[i];
    }
    FaceDumbRule faceDumbRule = workPart.ScRuleFactory.CreateRuleFaceDumb(new
Face[1] { face });
    NXOpen.SelectionIntentRule[] rules1 = new NXOpen.SelectionIntentRule[1] {
faceDumbRule };
    EdgeDumbRule edgeDumbRule =
workPart.ScRuleFactory.CreateRuleEdgeDumb(edges);
    NXOpen.SelectionIntentRule[] rules2 = new NXOpen.SelectionIntentRule[1] {
edgeDumbRule };

    NXOpen.Features.LawExtensionBuilderEx lawExtensionBuilderEx =
workPart.Features.CreateLawExtensionBuilderEx(null);
    lawExtensionBuilderEx.CurveFitData.Tolerance = 0.01;
    lawExtensionBuilderEx.CurveFitData.AngleTolerance = 0.5;
    lawExtensionBuilderEx.Faces.ReplaceRules(rules1, false);
    lawExtensionBuilderEx.BaseProfile.AddToSection(rules2, edges[0], null, null, new
Point3d(), Section.Mode.Create);
    lawExtensionBuilderEx.AngleLaw.LawType =
NXOpen.GeometricUtilities.LawBuilder.Type.Constant;

```

```

        lawExtensionBuilderEx.AngleLaw.Value.Value = angular_dim0.Value;
        lawExtensionBuilderEx.LengthLaw.LawType
NXOpen.GeometricUtilities.LawBuilder.Type.Constant;
        lawExtensionBuilderEx.LengthLaw.Value.Value = length.Value;
        NXOpen.Features.LawExtensionEx          lawExtensionEx
(NXOpen.Features.LawExtensionEx)lawExtensionBuilderEx.Commit();
        lawExtensionBuilderEx.Destroy();
        FaceDumbRule          faceDumbRule3
workPart.ScRuleFactory.CreateRuleFaceDumb(lawExtensionEx.GetFaces());
        NXOpen.SelectionIntentRule[] rules3 = new NXOpen.SelectionIntentRule[1] {
faceDumbRule3 };
        NXOpen.Features.ThickenBuilder          thickenBuilder
workPart.Features.CreateThickenBuilder(null);
        thickenBuilder.Tolerance = 0.025;
        thickenBuilder.BooleanOperation.Type
NXOpen.GeometricUtilities.BooleanOperation.BooleanType.Create;
        thickenBuilder.FaceCollector.ReplaceRules(rules3, false);
        thickenBuilder.FirstOffset.Value = thicknesscab.Value;
        //thickenBuilder.SecondOffset.Value = 0;
        thickenBuilder.Commit();
        thickenBuilder.Destroy();
        length.Value = 0;
        thicknesscab.Value = 0;
        angular_dim0.Value = 0;
        TaggedObject[] nullobjects = new TaggedObject[0];
        face_select0.SetSelectedObjects(nullobjects);
        edge_select0.SetSelectedObjects(nullobjects);
    }
    else if (block == list_box01)
    {
    }
    else if (block == buttonDelete)
    {
    }
    else if (block == super_section2)
    {
    }
    else if (block == kindStringer2)
    {
    }
    else if (block == typeStringer2)
    {
    }
    else if (block == planeStringer2)
    {
    }
    else if (block == Addbutton2)
    {
    }
    else if (block == list_box02)

```

```

        {
            //-----Enter your code here-----
        }
        else if (block == Deletebutton2)
        {
            //-----Enter your code here-----
        }
    }
    catch (Exception ex)
    {
        theUI.NXMessageBox.Show("Block Styler", NXMessageBox.DialogType.Error,
ex.ToString());
    }
    return 0;
}

//-----
//Callback Name: ok_cb
//-----
public int ok_cb()
{
    int errorCode = 0;
    try
    {
        errorCode = apply_cb();
    }
    catch (Exception ex)
    {
        errorCode = 1;
        theUI.NXMessageBox.Show("Block Styler", NXMessageBox.DialogType.Error,
ex.ToString());
    }
    return errorCode;
}
public PropertyList GetBlockProperties(string blockID)
{
    PropertyList plist = null;
    try
    {
        plist = theDialog.GetBlockProperties(blockID);
    }
    catch (Exception ex)
    {
        theUI.NXMessageBox.Show("Block Styler", NXMessageBox.DialogType.Error,
ex.ToString());
    }
    return plist;
}
}

```