

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Навчально-науковий інститут цифрових технологій, дизайну та транспорту

Методичні вказівки до виконання лабораторних робіт по дисципліні
«Програмне забезпечення комп'ютерного проектування» модуль 2
для студентів всіх форм навчання спеціальності 122 «Комп'ютерні науки»

ОДЕСА: 2025

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Навчально-науковий інститут цифрових технологій, дизайну та транспорту

Методичні вказівки до виконання лабораторних робіт по дисципліні
«Програмне забезпечення комп'ютерного проектування» модуль 2
для студентів всіх форм навчання спеціальності 122 «Комп'ютерні науки»

Затверджено
на засіданні кафедри
Інформаційних технологій
проектування та дизайну
Протокол № 1 від 26.08.2025

ОДЕСА: 2025

Методичні вказівки до виконання лабораторних робіт по дисципліні «Програмне забезпечення комп'ютерного проектування» модуль 2 для студентів всіх форм навчання спеціальності 122 «Комп'ютерні науки» / Укл.: В.М. Тігарєв., О.С. Лопаков., Д.С. Солодкий– Одеса: 2025. – 67 с.

Укладачі: В.М. Тігарєв, канд. техн. наук, доцент
О.С. Лопаков, ст. викладач.
Д.С. Солодкий, асистент.

	Зміст	Стор
Лабораторна робота № 4		6
Лабораторна робота № 5		38
Лабораторна робота № 6		52
Лабораторна робота № 7		58
ПЕРЕЛІК ЛІТЕРАТУРИ		67

ВСТУП

iLogic надає користувачеві Inventor можливість реалізації проектів, створюваних, модифікованих, керованих та/або конфігурованих за допомогою програмних правил або користувацьких розмірів. iLogic додає до Inventor дуже потужні функціональні можливості. Ці можливості дозволяють користувачеві за допомогою правил автоматизувати завдання, що повторюються, і тим самим створювати інтелектуальні конфігурації компонентів. Мова iLogic є спрощеною версією Visual Basic, яка, тим не менш, містить кілька чудових покращень з точки зору продуктивності, що дозволяють використовувати мову безпосередньо – без необхідності писати і рядки коду.

У цих методичних вказівках розглядається робота з системою Autodesk Inventor з використанням можливостей iLogic. Впровадження правил безпосередньо в документи деталі, складання або креслення. Ці правила можуть керувати зміною параметрів або атрибутів документа при виникненні різних подій (програмних або викликаних користувачем), які, у свою чергу, можуть керувати будь-яким геометричним чи атрибутивним змістом документа. Механізм автоматичного управління моделлю в iLogic відкриває вам можливість внесення інтелектуальної складової в будь-який ваш документ - ви можете керувати формою компонентів, виконанням збірок, приводити в рух складні кінематичні системи і навіть вносити в Inventor нові інструменти, що автоматизують необхідні вам проектні операції. Вся ця інтелектуальна складова після створення зберігатиметься в документі поряд із його геометричними елементами. Крім того, правила iLogic можна зберігати і на рівні шаблонів документів, отримуючи можливість використовувати їх надалі у документах, що створюються, без необхідності повторного введення. За допомогою правил iLogic, крім звичайних однозначних чисельних параметрів, ви можете використовувати і нові типи параметрів користувача в Autodesk Inventor, такі як текст, істина/брехня, а також параметри-списки.

Методичні вказівки призначені для студентів всіх форм навчання спеціальності 122 «Комп'ютерні науки» по дисципліні «Програмне забезпечення комп'ютерного проектування».

Лабораторна робота № 4

Частина 1 «Інтерфейс iLogic в САПР Autodesk Inventor»

Частина 2 «Робота з правилами у iLogic у САПР Autodesk Inventor»

Частина 3 «Створення правила активації та придушення елемента з використанням iLogic»

Частина 4 «Створення правил для розмірів та робота з формами»

Мета частини 1: Вивчити інтерфейс інтерфейс iLogic в САПР Autodesk Inventor. Навчитися створювати різні типи параметрів.

Теоретична частина:

Технологія iLogic забезпечує проектування на основі правил, що дозволяє легко зберігати і повторно використовувати свою роботу. Використовуємо iLogic, щоб стандартизувати і автоматизувати процеси проектування і налаштовувати свої віртуальні продукти.

Можливо керувати видимістю браузера iLogic на вкладці стрічки "Вид" > панель "Вікна" > "Користувальницький інтерфейс" > "Браузер iLogic"

ФУНКЦІЇ ILOGIC

iLogic впроваджує правила у вигляді об'єктів безпосередньо в деталь, складання і документи креслень. Ці правила визначають значення параметрів і атрибутів для проекту, а також керують ними. Керуючи цими значеннями, можна визначати поведінку атрибутів, елементів і компонентів моделі. Знання зберігаються безпосередньо в документі і зберігаються там аналогічна тій, як зберігаються елементи геометричного проекту.

У правилах iLogic можна використовувати призначені для користувача типи параметрів, які доступні в Inventor (наприклад, текст, значення "істина / неправда" і списки, що містять кілька значень). Ці типи параметрів можна використовувати для запису правил, де в якості вхідних значень використовуються не тільки чисельні значення.

Ці спеціалізовані параметри підтримуються в діалоговому вікні "Параметри Inventor", де містяться розширені функції фільтрації, що спрощують визначення та редагування параметрів, а також управління ними.

Крім звичайних наявних в Autodesk Inventor числових параметрів можна створювати текстові та логічні параметри, які згодом можуть використовуватися для управління моделлю. Створені додаткові параметри моделі в подальшому можуть бути застосовані в правилах iLogic.

Створення числового параметра

1. Виберіть у спадному меню в нижній частині діалогового вікна команду Додати числовий параметр.

У нижній частині списку параметрів буде створено новий рядок, при цьому курсор буде встановлений в осередку "Ім'я параметра" цього рядка.

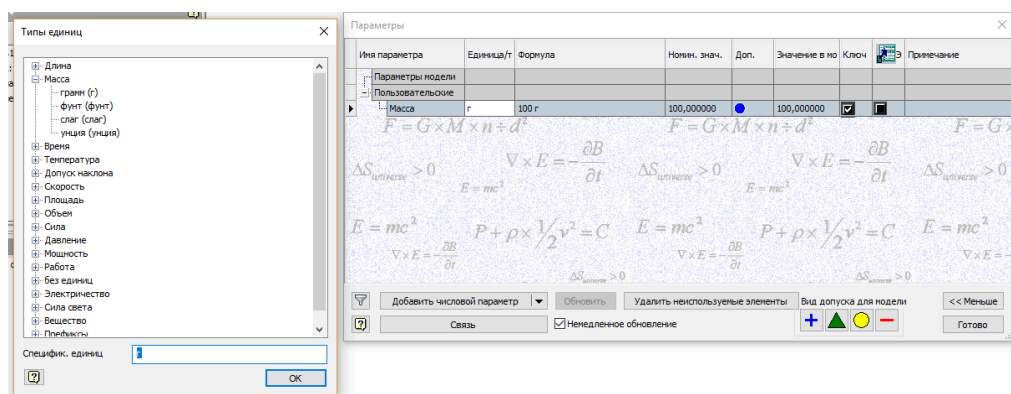
2. Введіть ім'я **маси** і клацніть осередок Одиниця, щоб відкрити діалогове вікно "Тип одиниць".

3. Розкрийте вузол Маса і виберіть Грам.

Імена параметрів в iLogic чутливі до регістру. При створенні правил слід зберігати регістр, який використовували в діалоговому вікні "Параметри".

4. Введіть 100 в осередок Формула, потім клацніть іншу клітинку в рядку, при цьому значення поля Номінальне значення має стати одно 100,000000.

5. Встановіть прапорець в клітині Ключовий цього параметра, щоб зробити цей параметр ключовим.



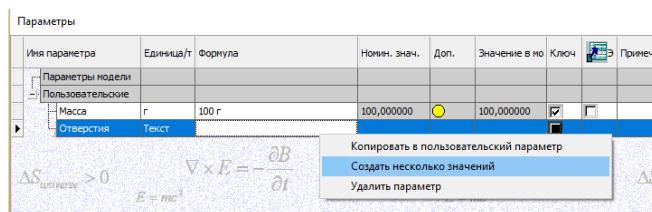
Створення текстового параметра

При відкритому діалоговому вікні "Параметри":

1. Виберіть у спадному меню в нижній частині діалогового вікна команду Додати текстовий параметр.

2. Введіть Отвори в порожню осередок Ім'я параметра, розташовану в нижній частині списку параметрів.

3. Клацніть правою кнопкою одну з осередків ряду, при цьому відкриється контекстно-залежне меню з варіантами Створити кілька значень і Видалити параметр.

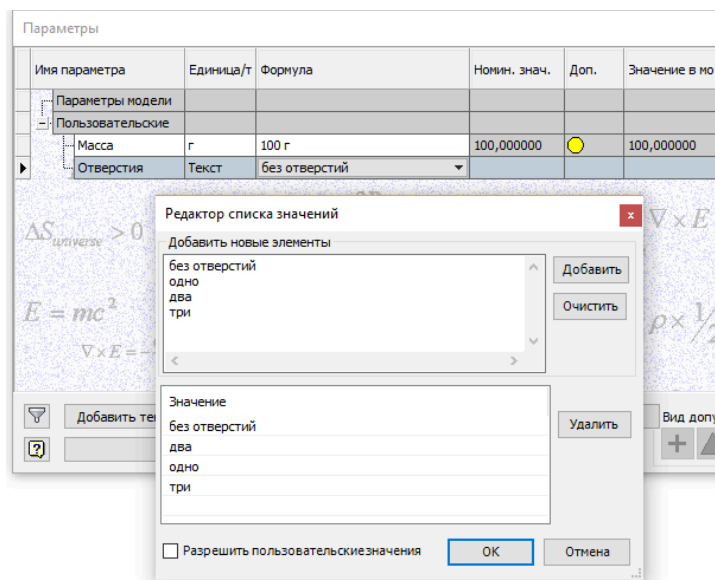


4. Щоб відкрити діалогове вікно "Редактор списка значений", виберіть Створити кілька значень.

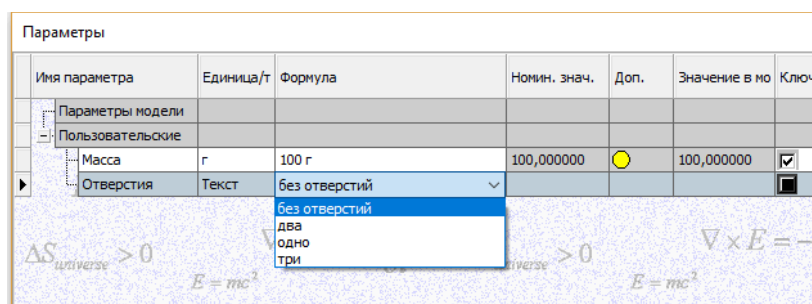
5. У верхній частині діалогового вікна в полі Додати нові елементи введіть кілька значень. Не забувайте натискати клавішу Enter після того, як кожен елемент поміщається в свою рядок.

6. Щоб перемістити нові елементи в поле Значення внизу діалогового вікна, натисніть кнопку Додати.

7. Щоб прийняти ці значення і закрити діалогове вікно "Редактор списка значений", натисніть кнопку ОК.



Щоб побачити введені рядкові значення, натисніть кнопку зі стрілкою в осередку Формула параметра Holes.



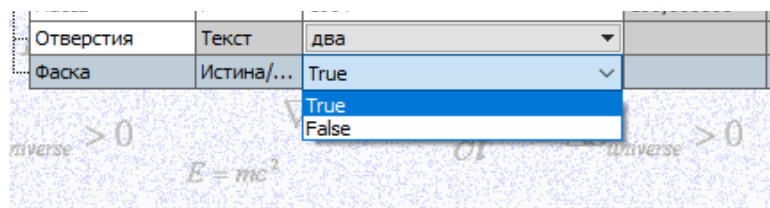
8. Щоб зробити цей параметр ключовим, встановіть для цього параметра прапорець Ключовий.

Створення логічного параметра

1. Виберіть у спадному меню в нижній частині діалогового вікна команду Додати "істина / неправда".

2. Введіть ім'я в порожню осередок "Ім'я параметра", розташовану в нижній частині списку параметрів.

3. Клацніть клітинку Формула і зверніть увагу на спливаюче меню з параметрами Істина і Брехня.



4. Встановіть прапорець "Ключ", щоб зробити даний параметр ключовим.

ФУНКЦІЇ АВТОМАТИЗАЦІЇ В ILOGIC

Конструктори і проектувальники можуть реалізовувати функції автоматизації iLogic, навіть якщо вони не мають досвід програмування. Використовуйте функції автоматизації, щоб:

- автоматично шукати і змінювати конфігурації параметричних деталей або збірок на основі умовного оператора, визначеного в правилах на рівні збірки,

- активізувати елементи деталей або збірки, а також компоненти і залежно збірки на основі правил з використанням умовних аргументів,

- автоматично оновлювати і застосовувати визначення різьблення при зміні розміру отвори або стрижня,

- читати і записувати властивості матеріалу або подання в документах проекту, масу або обсяг деталей і параметри проекту, а також реагувати на них,

- оновлювати відомості специфікації, якщо зміни моделі призводять до нової конфігурації,

- обмежувати або автоматично виправляти вводяться користувачем значення таким чином, щоб підсумкові конфігурації були допустимі і відповідали специфікаціям і стандартам проекту,

- виконувати читання і запис в таблиці Excel.

Функції роботи з вікнами повідомлень і введення дозволяють отримувати зворотний зв'язок, параметри та інформацію при запуску правил.

Створення призначеного для користувача інтерфейсу з використанням правил iLogic

Використовуємо правила iLogic, щоб безпосередньо підключати призначені для користувача форми до параметрів проекту для деталі або збірки. Ця можливість дозволяє створити інтерфейс користувача для шаблону проекту на базі правил.

ПАРАМЕТРИ В ILOGIC

Крім стандартних параметрів в моделі Inventor можна створювати і редагувати текстові параметри та параметри типу "істина / неправда". Текстові параметри - це символні параметри, а параметри типу "істина / неправда" містять логічне значення True або False. Параметри обох типів можна використовувати в правилах. Таким чином забезпечується застосування параметрів Inventor або будь-яких інших параметрів, що застосовуються за рахунок правил. У правилах ці параметри представляють стандартні типи VB.NET String і Boolean.

У VB.NET строкові значення полягають в подвійні лапки. наприклад:

```
string1 = "Left Side"
```

Багато функцій iLogic містять рядкові аргументи, які також полягають в лапки. Це, наприклад, імена компонентів і елементів в моделі Autodesk Inventor.

Логічні параметри можуть приймати значення True або False. наприклад:

```
boolean1 = True
```

```
boolean2 = False
```

Логічні значення True і False в лапки не полягають.

Строкові (текстові) параметри можна призначити властивостями Inventor типу Text. Логічні (істина / неправда) параметри можна призначити властивостями Inventor типу "Так чи ні".

Далі наводиться приклад синтаксису для запису правила:

```
iProperties.Value ( "Project", "Description") = string1  
iProperties.Value ( "Custom", "Passed") = boolean1
```

Можна задати значення властивостей Inventor на підставі текстових і логічних параметрів:

```
string1 = iProperties.Value ( "Project", "Description")  
boolean1 = iProperties.Value ( "Custom", "Passed")
```

Багатозначність ПАРАМЕТРІВ В ILOGIC

Багатозначний параметр - це параметр Autodesk Inventor, що містить список можливих значень. Однак, подібно до інших параметрах, багатозначні параметри в кожному конкретному випадку можуть приймати одне єдине значення. Значення такого варіанту не буде автоматично прив'язано тільки до одного з значень багатозначного списку, хоча це можна реалізувати за допомогою правила.

Багатозначні параметри можна створювати і редагувати в діалоговому вікні "Параметри". Кладніть правою кнопкою миші по параметру в будь-якому стовпці і виберіть "Створити кілька значень", щоб вказати, що цей параметр може приймати кілька значень. При цьому відображається діалогове вікно, де можна ввести список значень. Можна також вставити список значень з текстового редактора або іншої програми, наприклад, з Microsoft Excel.

Список значень для числових параметрів може містити дробу або формули.
наприклад:

4/9
10,25 мм
d1 * 0,75

У разі багатозначних або логічних параметрів в осередку "Формула" міститься список, що розкривається, з якого можна вибрати поточне значення (або формулу для числових параметрів).

У правилі можна виконувати читання і запис списку значень для параметра, використовуючи функції роботи з декількома значеннями.

У браузері iLogic перераховані правила, форми, глобальні форми і зовнішні правила iLogic.

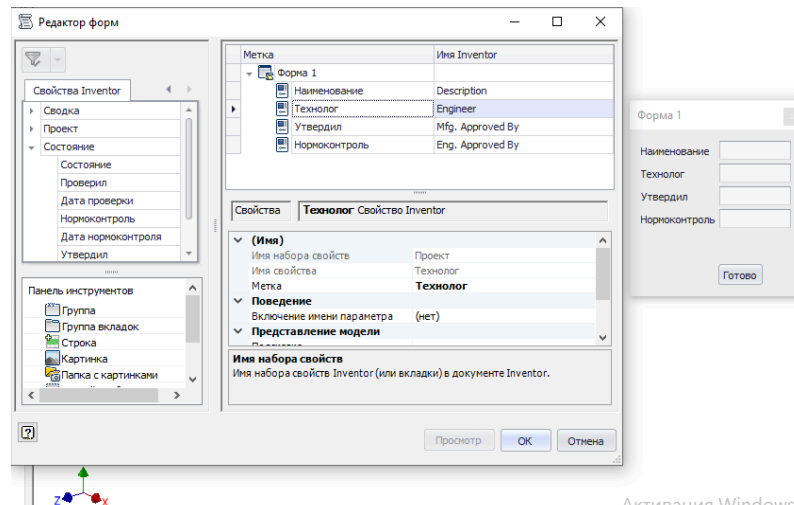
ПРАВИЛА

Правило - це невелика програма на Visual Basic (VB.NET), яку можна використовувати для контролю за іншими параметрами, елементами або компонентами Inventor.

iLogic впроваджує правила у вигляді об'єктів безпосередньо в деталь, складання і документи креслень. Правило визначає і застосовує значення параметра проекту та атрибутів. Регулюючи ці значення, можна визначити поведінку атрибутів, елементів і компонентів моделі.

Знання зберігаються безпосередньо в документі і зберігаються там аналогічно тій, як зберігаються елементи геометричного проекту.

ФОРМИ



Можливо створювати та підключати призначений для користувача інтерфейс для параметрів, властивостей і правил деталі, збірки або документа креслення Inventor.

Операції перетягування дозволяють проектувати інтерфейс користувача без необхідності програмування. Для управління змінами існуючого проекту використовуються елементи управління.

МАЙСТЕР

iLogic містить кілька майстрів, які можна використовувати для створення складних правил для певних завдань. Ці майстри доступні на вкладці "Майстра" в діалоговому вікні "Редагувати правило":

Створити правило для діалогового вікна - використовувати для користувача діалогове вікно для застосування значень параметрів.

Запам'ятати поточний вид - запам'ятати орієнтацію виду і значення масштабу відображення.

Вікно повідомлення - створити правило для відображення вікна повідомлення Windows.

Обмеження параметрів - задати мінімальні і максимальні значення для параметрів.

Середовище браузерів ILOGIC

У браузері iLogic правила впорядковані на двох вкладках.

На вкладці "Правила" перераховані всі правила, збережені в документі.

На вкладці "Зовнішні правила" перераховані всі правила, які зберігаються поза Inventor (на диску), незалежно від того, який документ відкритий.

Форми впорядковані двох вкладках:

На вкладці "Форми" перераховані всі форми і кнопки для ініціювання правил, збережених в поточному документі.

На вкладці "Глобальні форми" перераховані всі форми і кнопки для ініціювання правил, збережених поза середовищем Inventor, незалежно від того, який документ відкритий.

ПАРАМЕТРИ В ПРАВИЛАХ

В якості змінних Visual Basic в правилах можна використовувати імена стандартних параметрів Inventor. При запуску правила значення параметра присвоюється змінної. У разі числових параметрів значення - це дійсне число, виражене в одиницях виміру документа, заданих у вікні "Сервіс" > "Процес моделювання". Текстові параметри стають в правилі рядковими змінними. Параметри типу "Істина / брехня" стають логічними змінними.

Параметри можуть бути отримані двома різними способами:

Змінні VB.NET, пов'язані з числовими або нечисловими параметрами. Змінні виділяються синім кольором:

```
d1 = 1.25 in  
d2 = 12 mm
```

При запуску правила iLogic читає значення для цих параметрів з моделі і передає їх правилом. Після завершення виконання правила iLogic зберігає змінні значення назад в модель.

Щоб зберегти значення до завершення виконання правила, використовуйте функцію RuleParametersOutput. Ця функція корисна, коли потрібно негайно змінити модель.

Параметри, отримані за допомогою функції Parameter, наприклад:

```
Parameter("d1") = 2.4 in  
Parameter("Pin1:1", "d2") = 34.3 mm
```

При призначенні значень цим способом значення негайно зберігаються в моделі до запуску наступного оператора в правилі. Коли значення читається за допомогою функції Parameter, воно негайно читається з моделі Inventor.

Імена об'єктів за замовчуванням в правилах

До використання в правилі одного з наступних імен об'єктів Autodesk Inventor змініть його ім'я, якщо воно було призначено як ім'я за замовчуванням при його створенні:

- елемент
- ескіз
- робочий елемент
- шаблон
- залежність
- конструктивна пара

Використовуємо значущі імена. Зміна імені дозволяє краще зрозуміти, для чого призначене правило.

ЗАПУСК ПРАВИЛА

Правила зазвичай запускаються автоматично у відповідь на зміну значень параметрів, якщо параметри використовуються безпосередньо як змінні правила. Правила можна також запускати в результаті події.

Правило запускається відразу ж після його редагування. Щоб цього не відбувалося, правило необхідно придушити. При запуску правила зміни параметрів не призводять до оновлення моделі до завершення виконання правила. При необхідності можна примусово застосувати зміна параметрів негайно. Використовуємо для цього в правилі:

функцію `Parameter`;

функцію `RuleParametersOutput ()`.

ОБРОБКА ПРАВИЛ

При створенні або редагуванні правила текст правила перетворюється в допустимий код VB.NET, який потім компілюється. Деякі імена параметрів Inventor можна використовувати як імена змінних VB.NET. Щоб використовувати весь діапазон імен параметрів Inventor, iLogic підставляє замість параметрів Inventor нові імена змінних.

Ключові слова, які використовуються в якості імен параметрів

Хоча ключові слова Visual Basic можна використовувати як імена параметрів, це не рекомендується робити в нових документах. Однак при додаванні правил в існуючі документи, імена параметрів зазвичай змінювати не доводиться. Якщо ім'я збігається з ключовим словом, і при цьому це ключове слово потрібно використовувати в правилі, можна:

змінити по можливості ім'я параметра Inventor (рекомендується);

використовувати це ключове слово в правилі, змінивши регістр. Наприклад, якщо параметр Inventor називається `class`, то використовуємо ключове слово в формі `Class`. Імена параметрів Inventor чутливі до регістру, проте ключові слова VB до регістру не чутливі. У нашому прикладі `class` буде змінено на інше ім'я до того, як VB побачить код, проте `Class` залишається незмінним.

Одиниці виміру в правилах

Одиниці виміру Inventor можна використовувати в правилах точно так же, як вони використовуються в стандартних параметричних формулах Inventor.

Для правил одиниці вимірів не потрібні. Одиниці виміру документа задаються у вікні "Сервіс" > "Процес моделювання" > "Одиниці" в Inventor.

Імена одиниць вимірювання можна використовувати тільки відразу ж після чисел в тексті правила. Їх не можна використовувати після параметрів, змінних або виразів.

Приклад - змішані одиниці виміру документа

У цьому прикладі в файлі деталі використовуються одиниці виміру документа в дюймах. У редакторі параметрів iLogic:

параметру `x` задається значення 1 дюйм

параметру `y` задається значення 12.5 мм

Правило записується в такий спосіб:

```
result = x + y
```

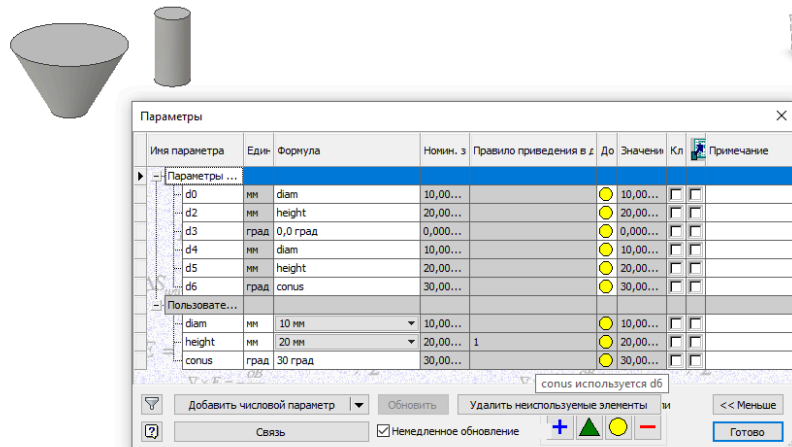
```
MessageBox.Show (result, "Mixed Units Defined in Parameter Editor")
```

При виконанні додавання iLogic перетворює значення параметра у з 12.5 мм в 0.4921 дюйма. У вікні повідомлення відображається значення +1.49212598425197 і зазначено, що в редакторі параметрів визначені змішані одиниці виміру.

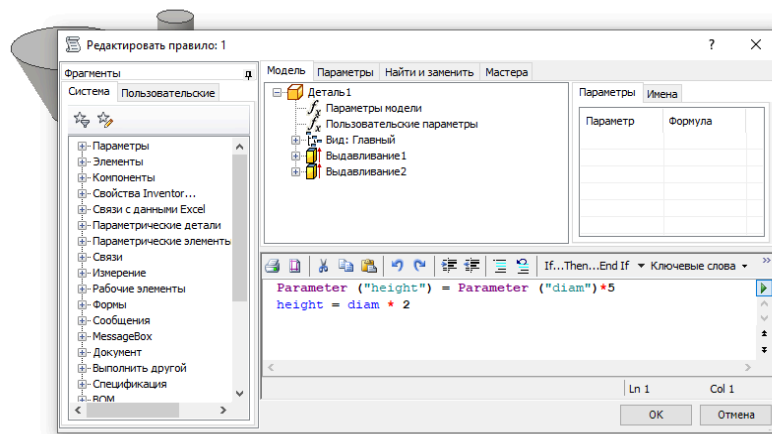
І навпаки, якщо в якості одиниць виміру документа вказані міліметри, то iLogic перетворює значення параметра х з 1 дюйм в 25.4 мм. У вікні повідомлення відображається значення 37.9 та вказано, що в редакторі параметрів визначені змішані одиниці виміру.

Практична частина:

1. Будуємо два параметричних примітиви



2. Вказуємо в новому правилі параметри, які необхідно перерахувати та отримуємо необхідний результат.



Висновок частини 1: Під час виконання лабораторної роботи необхідно ознайомитися з інтерфейсом iLogic в САПР Autodesk Inventor та з варіантами створення різних типів параметрів.

Запитання до частини 1.

1. Які типи параметрів існують у iLogic?
2. Що таке правило у iLogic?
3. Для чого використовуються форми у iLogic?
4. Як задавати параметри у правилах iLogic?

5. Як запускати та обробляти правила у iLogic?

Частина 2 «Робота з правилами у iLogic у САПР Autodesk Inventor»

Мета частини 2: Навчитися створювати правила та керувати параметрами деталі. Використовувати мастера з повідомленнями фрагменти коду для швидкого програмування.

Теоретична частина:

РОБОТА З ПРАВИЛАМИ У ILOGIC

Фрагменти можна створювати, редагувати, перейменовувати, запускати і вставляти в правила.

Створення правила

До створення правила створюємо всі параметри, які слід включити в це правило.

Обираємо на стрічці вкладки "Управління" ► панель iLogic ► Додати правило . У діалоговому вікні "Ім'я правила" вводимо ім'я нового правила і клацаємо кнопку "ОК".

Використовуємо діалогове вікно "Edit Rule" ("Редагувати правило") для визначення правила.

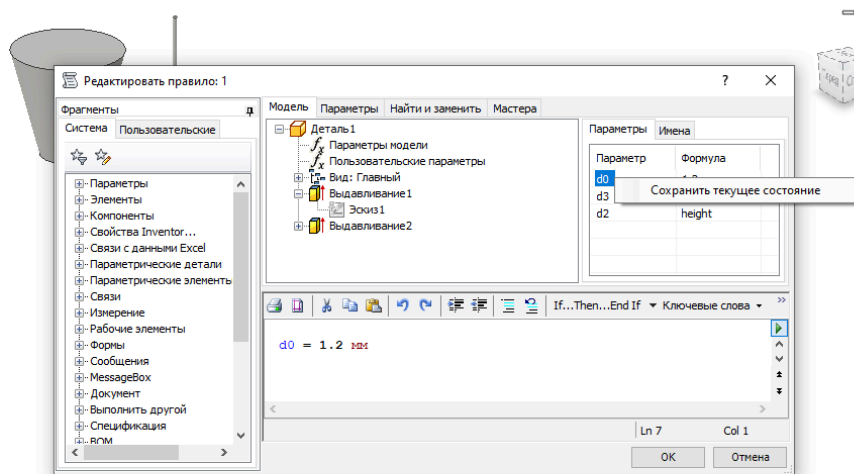
У документі деталі або креслення обираємо групу параметрів, щоб відобразити цей список параметрів на вкладці "Параметри" справа. Потім двічі обираємо в списку параметр, щоб вставити це ім'я в текст правила.

У документі збірки виберіть вкладку "3D модель" для відображення компонентів збірки або вкладку "Дерево файлів" для відображення файлів вузлів і деталей. Розгорніть елемент файлу деталі для відображення параметрів і елементів усередині цього файлу.

Якщо в дереві моделі клацнути непараметрический елемент, його ім'я наводиться на вкладці "Імена" (поряд з вкладкою "Параметри"). Щоб вставити ім'я елемента в правила, двічі клацніть цей елемент в списку. Для багатьох з фрагментів креслень слід вказати ім'я виду. Ім'я виду з поточного креслення для його включення в правило можна отримати, двічі клацнувши відповідний елемент у списку.

Клацніть елемент правою кнопкою миші і виберіть Зберегти поточний стан, щоб вставити текст в правило, призначають значення елементу.

Наприклад, збережіть поточний стан параметра, щоб вставити параметр і його поточне значення (наприклад, $d0 = 1.2 \text{ мм}$). Параметри завжди зберігаються разом з одиницями вимірювання, але при записі нових формул одиниці виміру не потрібні. Щоб зберегти стан всіх параметрів певного типу, можна натиснути правою кнопкою миші групу параметрів. Можна також клацнути елемент або компонент правою кнопкою миші, щоб зберегти його поточний стан Активний (пригнічений або неподавленной).



Використовуємо фрагменти, перераховані на вкладці "Системні і призначені для користувача" в області "Фрагменти", для додавання функцій, які згодом можна налаштувати для правила. Двічі клацніть фрагмент для додавання його до правила.

Використовуйте варіанти, запропоновані на панелі інструментів редагування над областю тексту правила, щоб управляти поданням правила. Також можна використовувати пропонувані варіанти, щоб вставити умовні оператори, ключові слова і оператори. Клацніть "ОК" для збереження правила.

Редагування правила

Виберіть на стрічці вкладку "Управління" > панель iLogic > Оглядач iLogic,

якщо оглядач в даний час не відображається.

У браузері iLogic перейдіть на вкладку "Правила", щоб редагувати правило у відкритому документі. Можна також перейти на вкладку "Зовнішні правила", щоб редагувати правило для компонента, що зберігається в зовнішньому файлі.

Двічі клацніть правило або клацніть його правою кнопкою миші і виберіть в контекстному меню команду "Редагувати правило".

Використовуйте діалогове вікно "Редагувати правило" для редагування правила. Під час введення тексту до тексту застосовується відповідний колірний код для підвищення зручності читання. Закінчивши, натисніть ОК.

Перейменування правила

Щоб перейменувати правило, двічі клацніть (два окремих клацання) його ім'я в браузері iLogic для переходу в режим редагування, а потім змініть існуюче ім'я.

Вставка синтаксису правила або фрагмента

У діалоговому вікні "Редагувати правило" помістіть курсор в задане положення в правилі.

Виконайте одну з таких дій:

Для вставки рядка синтаксису використовуйте списки, що розкриваються на панелі інструментів над областю тексту правила.

Для вставки фрагмента коду виберіть вкладку "Система" або "Інші" в області "Фрагменти" і двічі клацніть фрагмент.

Відредагуйте вставлену рядок або фрагмент.

Робота зі згортанням ділянками в правилах

Для визначення згорнути ділянки коду на початку коду введіть одинарні лапки і ліву квадратну дужку, а в кінці - одинарні лапки і праву квадратну дужку.

```
'[
If Duty = "Light" Then
    TubeHeight = 1.5
    TubeRad = 0.25
    TubeThick = 0.125
ElseIf Duty = "Heavy" Then
    TubeHeight = 2
    TubeRad = 0.5
    TubeThick = 0.25
End If
']
```

Для визначення згорнути пояснення вставте три одинарних лапки на початку кожного рядка пояснення.

```
'''Comment line 1
'''Comment line 2
'''Comment line 3
```

Щоб згорнути код або пояснення, клацніть значок вузла (-).



Щоб увімкнути або вимкнути розширення ділянки коду або пояснення в положенні курсору, клацніть правою кнопкою миші і виберіть Структурування ➤ Включити розширення структурування.

Щоб увімкнути або вимкнути розширення всіх згортаються ділянок коду і пояснень в правилі, клацніть правою кнопкою миші і виберіть Структурування ➤ Включити всі структурування.

Щоб відключити всі можливості структурування і розгорнути всі згорнуті ділянки коду або пояснення, клацніть правою кнопкою миші і виберіть Структурування ➤ Зупинити структурування. Щоб заново включити можливості структурування, клацніть правою кнопкою миші і виберіть Структурування ➤ Запустити автоматичне структурування.

Запуск правил

Щоб вручну запустити правило, клацніть його в браузері iLogic правою кнопкою миші і виберіть "Запустити правило".

Щоб забезпечити синхронізацію параметрів в правилі з моделлю, перед запуском правила клацніть правою кнопкою "Повторно створити правило". Для синхронізації параметрів у всіх правилах виберіть "Повторно створити всі правила".

Щоб явно запустити інше правило всередині правила, використовуйте функцію iLogicVb.RunRule.

Управління запуском правил

Відкрийте правило в редакторі правил.

На вкладці "Параметри" виберіть потрібні параметри:

Пригнічений. Заборона на автоматичний запуск правила. (Пригнічений правило можна запускати вручну. Придушити правило можна також в браузері iLogic.). Операція не в діалоговому режимі. Запуск правила без відображення діалогових вікон Inventor.

Негайно запустити залежні правила. Запуск залежних правил при зміні значення пов'язаних параметрів. При негайному запуску залежних правил можна включити явні поновлення моделі після завдання параметрів. У момент поновлення параметри вже оновлені і залежні правила вже виконані.

Чи не виконувати автоматично. Запобігання запуску правила при зміні параметрів. Якщо вибрати цей параметр, можна запустити правило вручну або використовувати ініціатори подій або iLogicVb.RunRule для запуску правила.

Спільне використання коду в правилах

Відкрийте правило в редакторі правил.

Виберіть вкладку "Параметри".

Встановіть прапорець Прямий код VB.

Клацніть ОК.

iLogic не виконує попередню обробку коду в правилі, якщо він позначений як прямий код VB; отже, імена параметрів Inventor розпізнаються в правилі. Правило повинне містити класи або модулі, які можна викликати з інших правил. Використовуйте цей параметр, щоб кілька правил могли спільно використовувати код, або ж щоб підтримувати модульність коду. Однак пам'ятайте, що код спільного доступу створити складніше.

Параметр можна вставити в правило, клацнувши його ім'я під час створення правил на рівні збірки. Якщо згодом змінити ім'я компонента, щоб воно не збігалось з вихідним ім'ям файлу, використання параметричного синтаксису дозволяє легше знайти компонент.

Видалення правил

Виберіть на стрічці вкладку "Управління". На панелі "iLogic" виконайте одну з таких дій.

Для видалення правила відкрийте оглядач iLogic, потім клацніть правило правою кнопкою миші і виберіть "Видалити".

Для видалення всіх правил в документі виберіть команду "Видалити всі правила" в спадному меню панелі iLogic.

Якщо використовувати команду "Видалити всі правила" в збірці, всі правила в вузлах і деталях також видаляються.

Практична частина:

Використання майстрів правил

За допомогою майстрів iLogic можна створювати для певних завдань складні правила.

Збереження виду поточного екрану

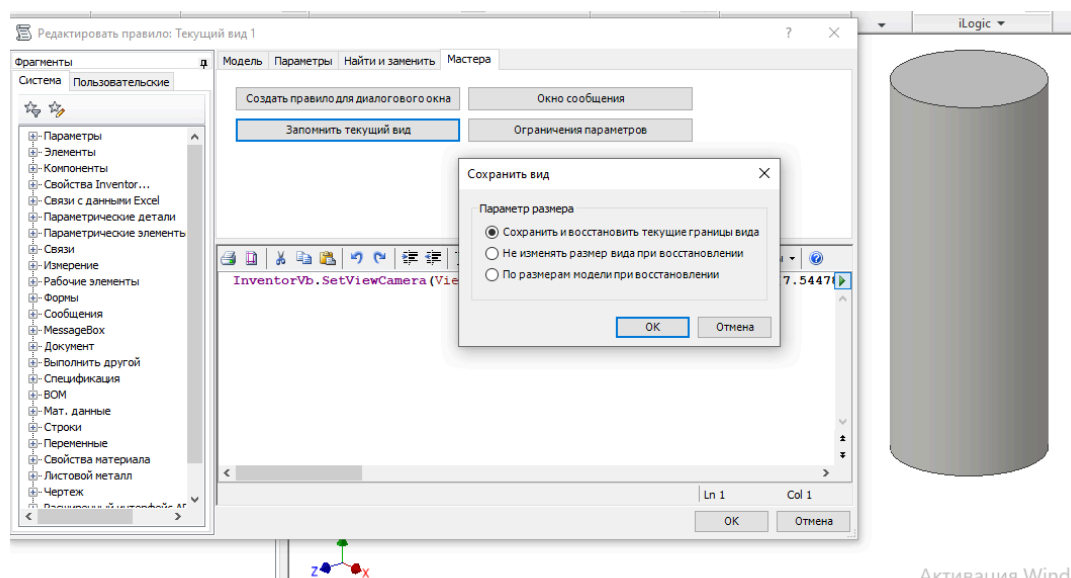
Клацніть Запам'ятати поточний вид на вкладці "Майстра" в діалоговому вікні "Редагувати правило". Виберіть "Параметр розміру" в діалоговому вікні "Запам'ятати вид".

Зберегти та відновити поточні кордону виду- відновлення виду з використанням налаштувань орієнтації і масштабу в момент збереження виду за допомогою правила.

Не змінювати розмір виду при відновленні- переміщення моделі до центру дисплея і завдання орієнтації без зміни масштабу.

За розмірами моделі при відновленні- переміщення моделі в центр дисплея і заповнення всього дисплея (аналогічно команді "Показати всі").

Клацніть ОК, щоб створити код.



Створення вікна повідомлення

Використовуйте цей майстер для створення правила, яке генерує вікно повідомлення.

Клацніть Вікно повідомлення на вкладці "Майстра" в діалоговому вікні "Редагувати правило".

Введіть заголовок, який буде відображатися у верхній частині вікна повідомлення.

Виберіть, які кнопки повинні відображатися у вікні повідомлення.

Вкажіть кнопку за замовчуванням, вибравши номер, відповідний її відносного положення по порядку на основі вибору кнопок.

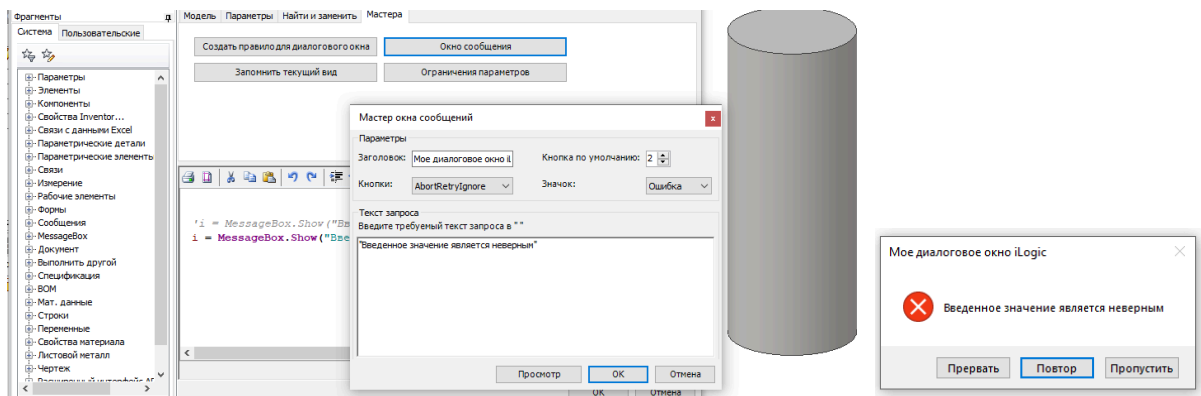
Наприклад, якщо в якості схеми кнопок вибрати Да Нет Отмена, а кнопкою за замовчуванням повинна бути кнопка "Ні", то виберіть 2.

Виберіть значок статусу, який Ви бажаєте бачити у вікні повідомлення.

Введіть текст, що відображається у вікні повідомлення, уклавши його в лапки.

Клацніть "Перегляд" для відображення діалогового вікна з обраними параметрами.

Клацніть ОК, щоб створити код.



Відображення діагностичних відомостей, коли значення параметрів виходять за вказаний діапазон

Клацніть Обмеження параметров на вкладці "Майстра" в диалоговом окне "Редагувати правило".

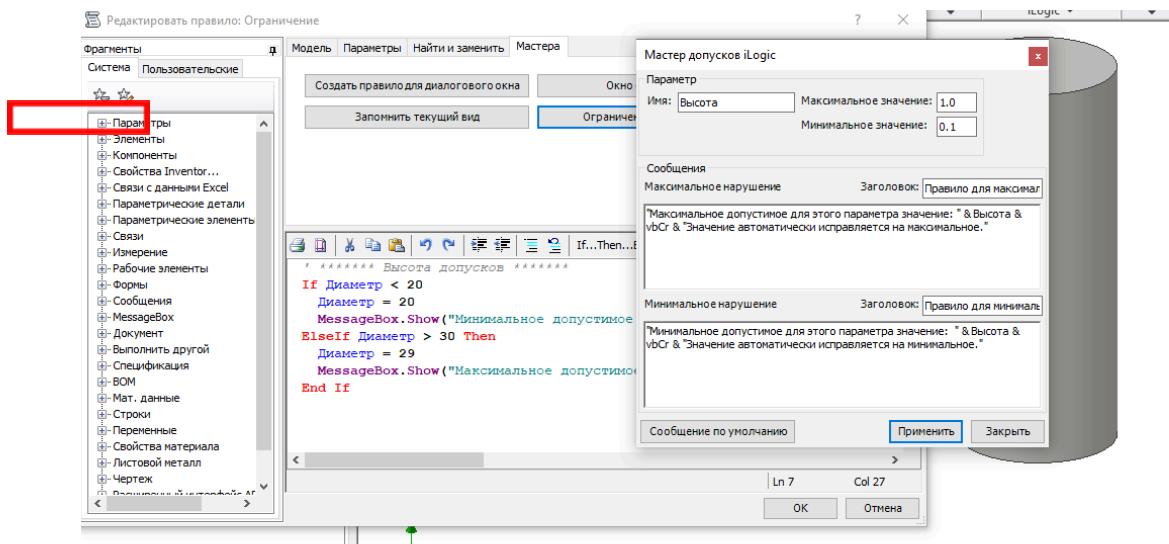
Введіть ім'я параметра.

Задайте максимальне і мінімальне значення.

Введіть заголовок і текст повідомлення, яке буде відображатися при порушенні мінімального і максимального значень.

Клацніть "Застосувати" для створення коду або "Повідомлення за замовчуванням", щоб скинути повідомлення до тексту за замовчуванням.

Щоб перейти до функцій Parameter, розгорніть вузол "Parameters" на вкладці "Система" в області "Фрагменти".



PARAMETER

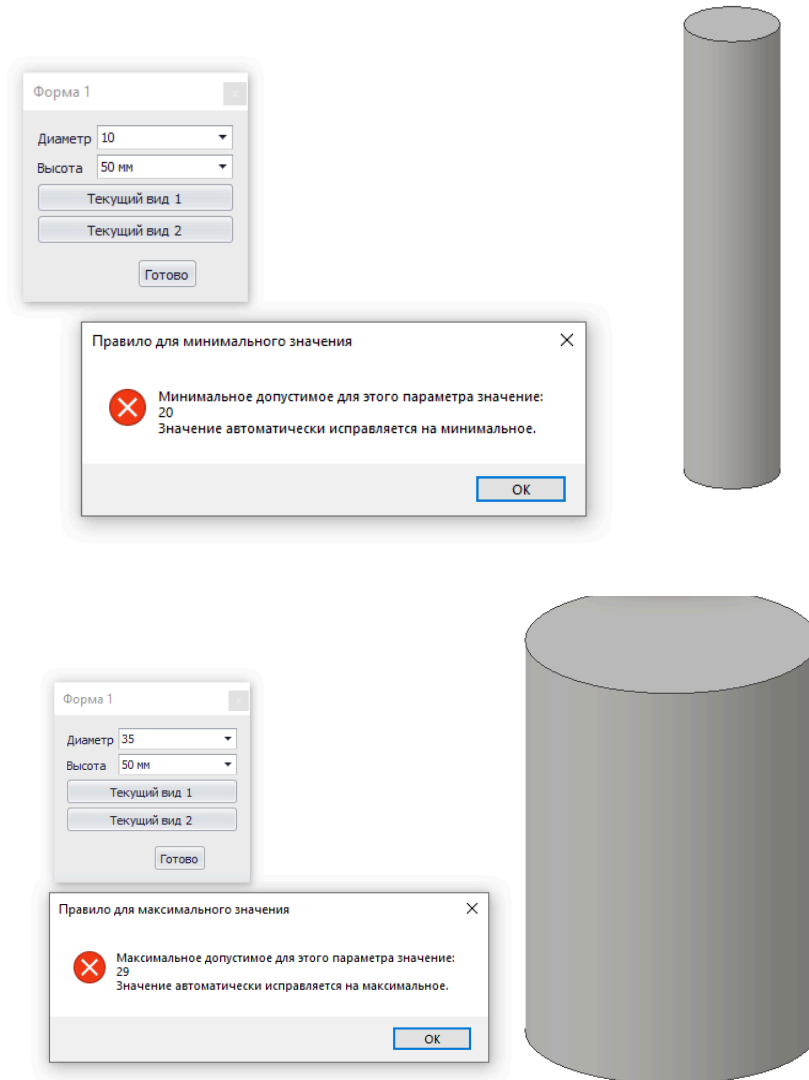
Parameter ("parameterName")

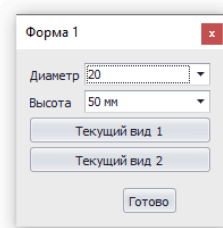
Ця функція безпосередньо змінює значення параметрів.

У збірках можна отримати доступ до параметрів в пригнічених компонентах.

Parameter ("PartA: 1", "D12") = 6.3

Зверніть увагу, що замість імені файлу вказано ім'я компонента. Хоча функції присвоюється номер певного компонента, зміна параметра впливає на всі екземпляри компонента. Ім'я компонента потрібно лише для того, щоб функція могла визначити файл.





Для імені компонента можна також використовувати функцію MakePath:

`Parameter (MakePath ("SubAssem1: 1", "Part1: 1"), "D12") = 6.3`

Щоб прочитати поточне значення параметра, використовуйте один з наступних варіантів синтаксису для оператора

`param_val = Parameter ("d0")`

`param_val = Parameter ("PartA: 1", "d0")`

Як показано в цих прикладах, параметр може перебувати в поточному документі. Можна використовувати ім'я будь-якого параметра, до якого можливий доступ з правила.

ФУНКЦІЇ MULTIVALUE

Функції MultiValue використовуються для отримання доступу до списку значень, що зберігаються для багатозначного параметра, і зміни цих значень. Їх використання можна проілюструвати такими прикладами:

MultiValue.SetList ("d0", 0,5, 0,75, 1,0, 1,25)

Задає список доступних значень для параметра d0.

MultiValue.SetList ("d0", "3/8", "d1 * 2", "d1 * 3")

Задає, чи не значення, а формули. У список можна включати як формули, так і значення.

MultiValue.SetList ("ім'я_файла.ipt.str0", "значення1", "значення2")

Задає список значень для текстового параметра деталі.

MultiValue.SetList ("деталь: 1", "d0", 0,5, 0,75, 1,0, 1,25)

Задає список значень параметра компонента.

values = MultiValue.List ("d0")

Витягує поточний список значень, де змінна values - це об'єкт VB.NET типу ArrayList.

MultiValue.UpdateAfterChange = True

Як і настройка функції Parameter.UpdateAfterChange, якій встановлено значення True, цей вислів змушує модель Inventor оновлюватися при кожній зміні значення параметра функцією MultiValue.List або MultiValue.SetList.

Установка значення True призведе до того, що модель (документ) буде оновлюватися після зміни параметра. Це має місце лише у разі зміни параметрів за допомогою функції Parameter:

Parameter.UpdateAfterChange= True

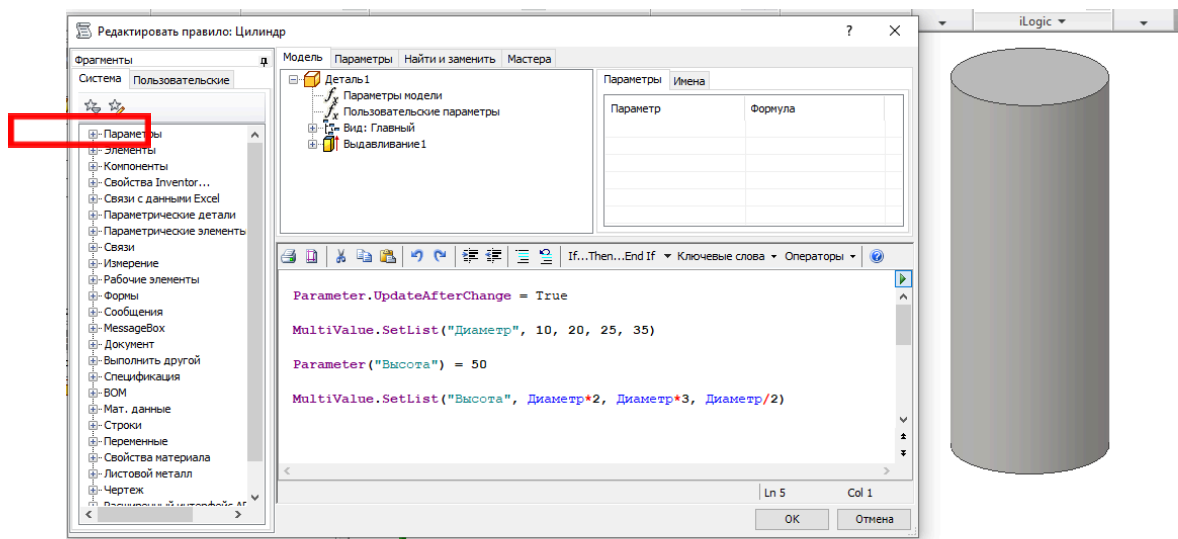
Установка значення True призведе до того, що модель (документ) буде оновлюватися при зміні параметра функцією Parameter. Це має місце лише у разі зміни параметрів за допомогою функції Parameter. наприклад:

Parameter.UpdateAfterChange= True

Parameter("D0") = 2,5

Це дозволить оновити модель.

Прим .: Вираз ("d0") = 2,5 НЕ буде оновлювати модель.



Висновок частина 2. Під час виконання лабораторної роботи необхідно навчитися створювати правила та керувати параметрами деталі. Засвоити використання мастера з повідомленнями фрагменти коду для швидкого програмування.

Запитання до частини 2.

1. Як можливо створити правило у iLogic?
2. Як проводити редагування правил у iLogic?
3. Як проводити запуск та управління правилом iLogic?
4. Як використовувати майстер правил у iLogic?
5. Як можливо спільне використання коду у правилах?

Частина 3 «Створення правила активації та придушення елемента з використанням iLogic»

Мета частини 3: Навчитися створювати правила придушення елемента з використанням iLogic.

Практична частина:

Правила можуть бути визначені для самих різних дій, включаючи установку значень параметрів і активацію придушення елементів. Перше вводиться правило пригнічує елемент деталі.

Редактор правил

1. Виберіть на стрічці вкладку "Управління" ► панель iLogic ► Додати правило

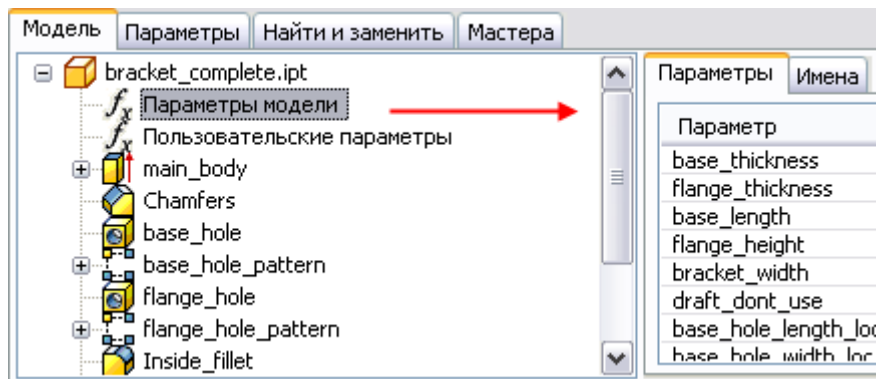


2. Введіть Modify Feature в поле "Ім'я" діалогового вікна "Ім'я правила" і натисніть ОК для відображення діалогового вікна "Редагувати правило".

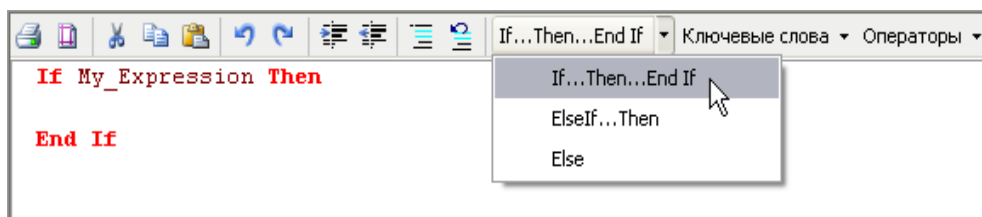
У діалоговому вікні "Редагувати правило" - це основа функції iLogic. Це діалогове вікно застосовується для створення і зміни правил iLogic.

3. Перейдіть на вкладку Модель. У лівій верхній панелі цього вікна присутній вид дерева моделі. Виберіть в дереві вузол Параметри моделі. Зверніть увагу, що тепер на верхній правій панелі відображаються тільки параметри моделі.

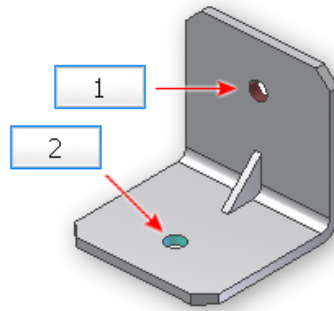
4. Щоб побачити інші набори параметрів можна клацнути вузол Власні параметри на дереві моделі і побачити тільки параметри, створені вручну.



Правила записуються в текстовій області правила, розташованій на нижній панелі діалогового вікна "Редагувати правило". Ключові слова правила можна вводити вручну, набравши їх у поле введення тексту. Або можна вибрати базові вирази на панелі інструментів над цим полем, а потім відредагувати затвердження.



5. Для відображення параметрів користувача клацніть вузол Власні параметри. У моделі скоби є два отвори: одне на підставі, друге на загнутої частини.



(1) отвір на загнутої частини (2) отвір на підставі

Створюване правило включатиме (або вимикати) отвір на підставі, отвір на загнутої частини або обидва. У попередньому уроці створювався параметр з декількома значеннями, названий holes. Йому було присвоєно три значення: base, flange і none. Це правило включає отвір фланця при виборі значення flange. При виборі base включається отвір в підставі, а при виборі none обидва отвори відключаються

Додавання параметрів правилом.

Спробуємо створити правило. Почнемо з установки значення flange для параметра holes.

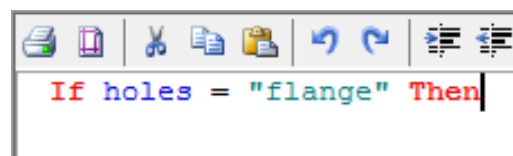
1. У текстовому полі введіть If і натисніть пробіл.

Зверніть увагу на те, що текст ключового слова If став напівжирним і змінив колір на червоний. Червоним кольором позначаються розпізнані мовні елементи (в даному випадку - ключове слово).

2. У дереві моделі клацніть вузол "Власні параметри" і двічі клацніть holes, щоб ввести в редактор ім'я параметра holes.

3. Введіть=, Пробіл, а потім "Flange" (Не забудьте про лапки). Додайте ще один пробіл і введіть Then, Щоб завершити цей рядок.

Зверніть увагу, що різним елементам мови в певному тільки що вираженні надані різні кольори. Таке колірне кодування істотно полегшує читання правил, дозволяє швидко розуміти їх сенс і виявляти помилки введення.

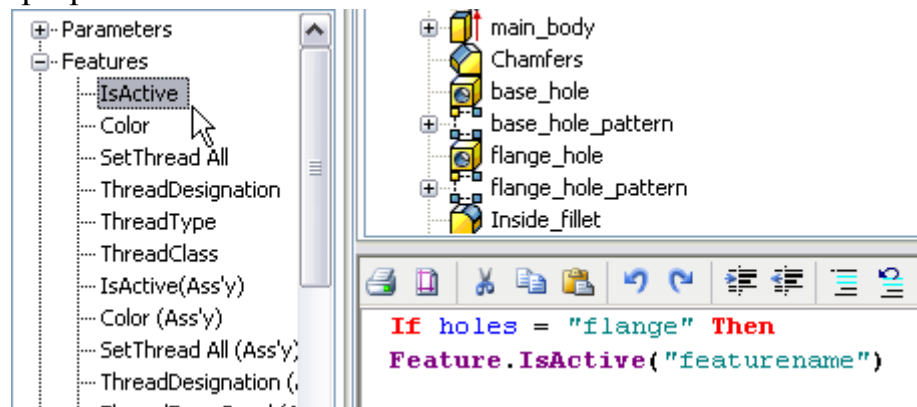


4. Для переходу до наступного рядка натисніть клавішу Enter.

За допомогою функції iLogic отвір на загнутої частини можна зробити активним.

Вставка фрагментів програми

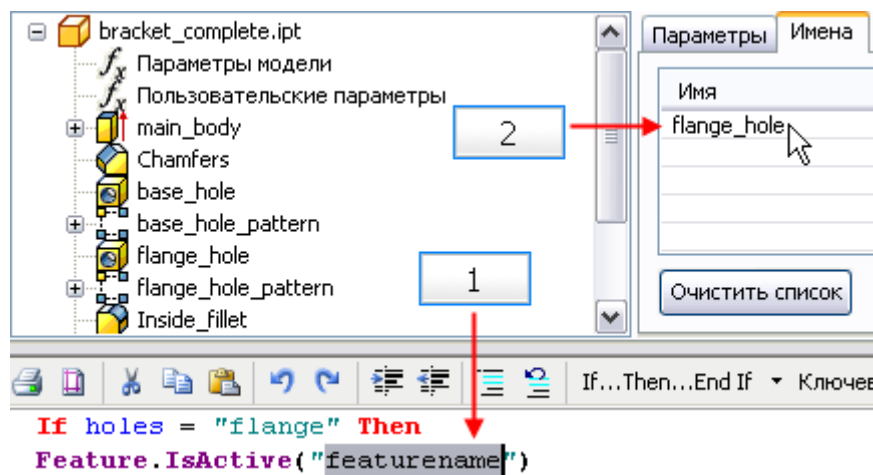
1. В області Фрагменти в лівій частині редактора перейдіть на вкладку Система. Розкрийте вузол Елементи і двічі клацніть IsActive, щоб вставити його текст в редактор правил.



2. Перейдіть на вкладку "Модель" вгорі діалогового вікна "Редагувати правило" і натисніть елемент flange_hole в дереві моделі.

3. У верхньому правому куті діалогового вікна натисніть "Імена" і зверніть увагу на відобразився текст: flange_hole.

4. Виділіть featurename в тексті правила, а потім двічі клацніть flange_hole на вкладці "Імена", щоб замінити ім'я featurename ім'ям flange_hole.



(1) Виділіть типовий текст. (2) Для заміни виділеного тексту двічі клацніть ім'я.

Функція Feature.IsActive встановлює активний стан (стан придушення) елемента, визначеного в лапках всередині круглих дужок.

5. Щоб призначити значення True, спочатку вставте пропуск в кінці виразу. Після пропуску введіть=, Потім знову пробіл, після чого словоTrue.

Присвоєне значення True означає, що отвір фланця активно (не пригнічуючи). Якщо для параметра holes вибирається варіант flange, це означає, що активним має бути тільки отвір фланця. Слід включити команду, деактивує отвір в підставі.

6. В кінці тексту правила натисніть клавішу `Enter` для переходу на наступний рядок і потім вставте ще одну функцію `Feature.IsActive ( "featurename")`.

7. Виділіть рядок `featurename` і клацніть `base_hole` в дереві моделі. Потім двічі клацніть `base_hole` на вкладці "Ім'я", щоб замінити ім'я `featurename` ім'ям `base_hole`, і надайте значення `False`.

Ці два рядки включають отвір на загнутої частини і вимикають отвір на підставі. Тепер правило містить три рядки.

```
If holes = "flange" Then  
Feature.IsActive ( "flange_hole") = True  
Feature.IsActive ( "base_hole") = False
```

Для варіантів з активним отвором в підставі застосовуються схожі дії. Слід активізувати отвір в підставі і деактивізувати отвір на загнутої частини.

Повторне використання блоків коду

Для створення наступної частини правила слід скопіювати і вставити повторно використовувану частину попередніх виразів. Далі, при необхідності, слід змінити текст.

1. Натисніть `Enter`, щоб перейти на новий рядок і потім введіть `Else If`.

2. Виділіть повторно використовуваний текст, а саме все після слова `If`, і натисніть `CTRL+C`, В результаті чого текст скопіюється в буфер обміну. Тепер встановіть курсор після слова **`Elseif`** і натисніть **`CTRL+V`**, Щоб вставити текст.

Прим.: Операції вирізання, копіювання і вставки можна також виконати, клацнувши правою кнопкою виділений текст і вибравши потрібну команду в контекстному меню. У ньому також містяться і інші команди редагування. Крім того, можна використовувати значки в рядку інструментів редагування над текстовою областю правила.

3. У щойно вставленому тексті змініть `flange` на `base` і переведіть умови `True` і `False`.

```
If holes = "flange" Then  
Feature.IsActive ( "flange_hole") = True  
Feature.IsActive ( "base_hole") = False  
ElseIf holes = "base" Then  
Feature.IsActive ( "flange_hole") = False  
Feature.IsActive ( "base_hole") = True
```

4. Додайте ще один вислів `Else if` за допомогою того ж методу копіювання і вставки створіть третю частину цього правила, не показує жодного отвору. Змініть тільки що вставлений текст так, щоб придушити обидва елементи отвори у випадку монтажу для параметри **`holes`** значення **`none`**.

5. Завершіть вираз, набравши `End If` (Або натиснувши відповідну копку ключового слова).

Правило завершено.

```
If holes = "flange" Then
```

```

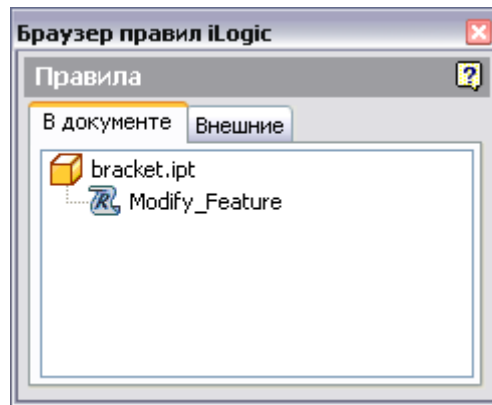
Feature.IsActive ( "flange_hole") = True
Feature.IsActive ( "base_hole") = False
ElseIf holes = "base" Then
Feature.IsActive ( "flange_hole") = False
Feature.IsActive ( "base_hole") = True
ElseIf holes = "none" Then
Feature.IsActive ( "flange_hole") = False
Feature.IsActive ( "base_hole") = False
End If

```

6. У діалоговому вікні "Редагувати правило" натисніть кнопку **ОК**.

При відсутності помилок діалогове вікно закриється без повідомлення про помилку. У браузері правил з'явиться новий значок, що позначає нове правило.

7.  Для перевірки нового правила виберіть вкладку "Управління" ► панель iLogic ► Оглядач iLogic і перегляньте дерево.



Браузер правил дозволяє переглянути правила для даної моделі. Браузер правил досліджується далі в цьому навчальному посібнику.

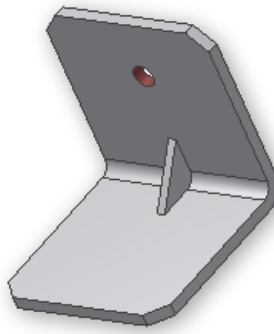
Перевірка правила придушення елемента

1. Виберіть на стрічці вкладку "Управління" ► панель "Параметри" ► Параметри, щоб відкрити діалогове вікно "Параметри".

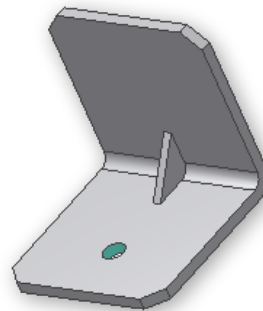
2. Натисніть на значок вузла зліва від області "**Параметри моделі**", В результаті чого список параметрів моделі закриється.

3. клацніть поле **Формула** параметра **holes**, Щоб відкрити список з кількома значеннями. Потім клацніть стрілку і виберіть **flange** в спадному меню.

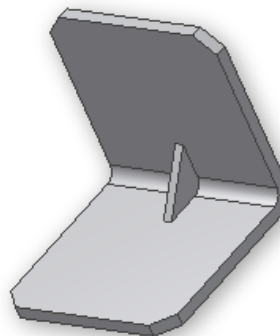
4. Виберіть будь-яку іншу клітинку і подивіться на скобу. Єдине показане отвір - це отвір на загнутої частини.



5. Змініть вибір з декількома значеннями на **base** і клацніть іншу клітинку. Видимим є лише отвір в підставі.



6. Змініть вибір з декількома значеннями на **none** і клацніть іншу клітинку. Жодне отвір не є видимим.



7. Натисніть кнопку **завершити**, Щоб закрити діалогове вікно "Параметри".

Перейменування правила придушення елемента

Створеному правилу слід привласнити більш зрозуміле ім'я.

1.  Виберіть на стрічці вкладки "Управління" ► панель iLogic ► Оглядач iLogic.
2. Лише один раз клацніть в дереві **Modify_Feature**, Щоб виділити правило, а потім натисніть ще раз для переходу в режим редагування.
3. Перейменуйте правило в `Hole_Rule` і натисніть клавішу Enter.
4. Закрийте Браузер правил.

СТВОРЕННЯ ПРАВИЛА АКТИВІЗАЦІЇ ЕЛЕМЕНТА

Спробуємо створити друге правило, управляє активізацією фасок скоби. Раніше ми створили логічний параметр з ім'ям `chamfers`. У логічного параметра є

два можливих значення: True і False. Ці значення будуть застосовуватися для включення або виключення фасок.



1. Виберіть на стрічці вкладки "Управління" ► панель iLogic ► Додати правило.

2. Назвіть нове правило Chamfer_Rule і натисніть ОК, щоб відкрити діалогове вікно "Редагувати правило".

У першій частині правила затверджується, що якщо значення логічного параметра chamfers істинно, елемент фасок активізується.

3. Введіть для цього правила вираз If.

If chamfers = True Then

4. З області Фрагменти вставте в правило копію фрагмента IsActive (Feature.IsActive). Замініть у вставленому фрагменті featurename на Chamfers і встановіть для вираження значення True.

If chamfers = True Then

Feature.IsActive ("Chamfers") = True

У другій частині правила стверджується, що коли значення параметра chamfers одно False, фаски деактивуються.

5. Додайте вираз Else і використовуйте для створення другої частини правила команди Копіювати і Вставити. Завершіть правило за допомогою виразу End If.

If chamfers = True Then

Feature.IsActive ("Chamfers") = True

Else Feature.IsActive ("Chamfers") = False

End If

Створення правила завершено.

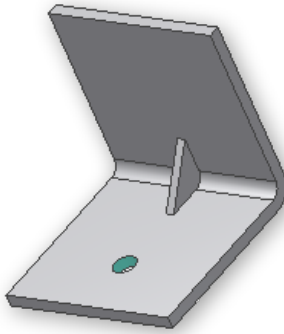
6. Натисніть кнопку ОК, щоб прийняти правило. Якщо повідомлень про помилки немає, правило можна перевірити.

Перевірка правила активізації елемента



1. Клацніть на стрічці вкладки "Управління" ► панель "Параметри" ► Параметри.

2. У рядку chamfers клацніть поле "Формула", що активізує розкривається декількох значень, потім натисніть на стрілку і виберіть False. Зверніть увагу на те, що всі фаски деактивізують.



3. Тепер встановіть для обраного варіанту "Формула" значення True. Фаски активізовані.

Висновок: частина 3. Під час виконання лабораторної роботи необхідно навчитися створювати правила придушення елемента з використанням iLogic.

Запитання до частини 2.

1. Як створити правило пригнічення елемента?
2. Як проводити додавання параметрів правилом?
3. Як налаштувати повторне використання блоків коду?
4. Як створити правило активізації елемента?
5. Як проводити перевірку створених правил?

Частина 4 «Створення правил для розмірів та робота з формами»

Мета частини 4: Навчитися створювати правила для розмірів з використанням можливостей iLogic та створювати форми.

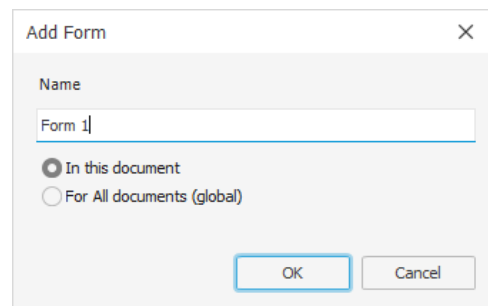
Теоретична частина:

Можна створювати, копіювати і вставляти форми користувальницького інтерфейсу, робити їх модальними, додавати, редагувати елементи управління в браузері iLogic і змінювати їх розмір, а також використовувати різні зображення.

Можна створити власний користувальницький інтерфейс і зв'язати його з параметрами, властивостями і правилами з документа деталі, збірки або креслення, створеного в Inventor. Можна додавати елементи керування на форму і редагувати їх, а також переглядати проміжні результати під час створення форми. Крім того, можна змінити уявлення вкладки "Форми iLogic" (або "Глобальні форми"), наприклад, розбивши елементи управління по групах або додавши зображення або підказку. Форму також можна скопіювати і вставити в тому ж документі, в іншому документі або на вкладці "Глобальні форми".

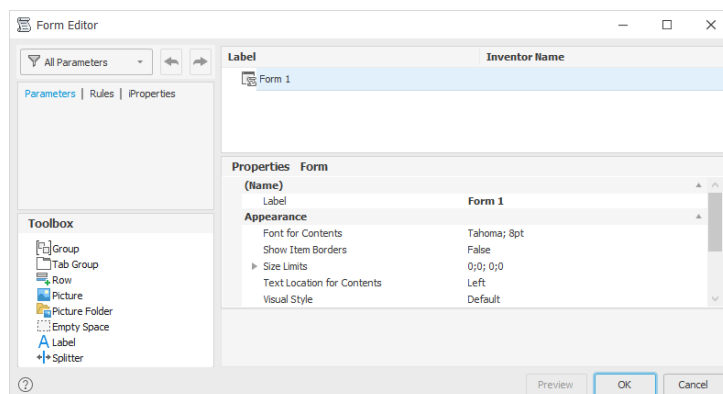
Створення форми користувальницького інтерфейсу

1. У спадному меню iLogic (Управління ➤ iLogic) виберіть "Додати форму".
2. Вкажіть ім'я форми.

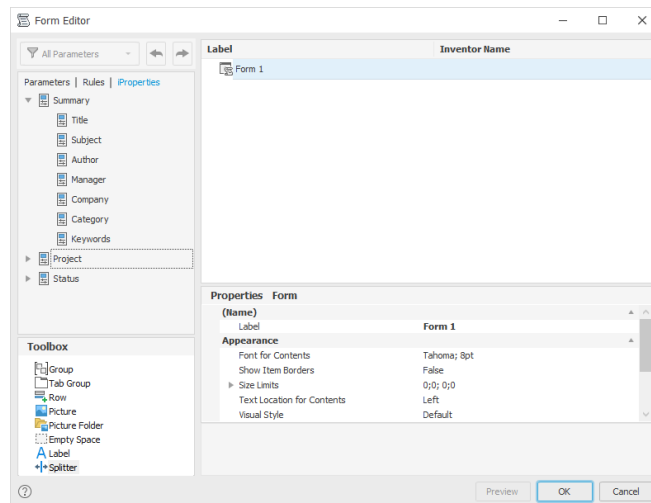


Якщо потрібно зробити форму доступною для декількох документів, виберіть параметр "Для всіх документів (глобальна)".

3. Створіть компоновку форми шляхом перетягування елементів з панелі інструментів.



4. Елементи управління можна додати шляхом перетягування з вкладок "Параметри", "Правила" і "Властивості Inventor".



5. Щоб скористатися формою, натисніть кнопку, додану в браузері iLogic.

Створення модальної форми

За замовчуванням форми не є модальними: на них немає кнопок "ОК" або "Застосувати", а при відображенні форми користувач може взаємодіяти з моделлю і іншими інструментами. Щоб зробити немодальну форму модальною, її потрібно відредагувати.

1. У браузері iLogic (Управління > iLogic > Оглядач iLogic) клацніть кнопку форми правою кнопкою миші і виберіть команду "Редагувати".

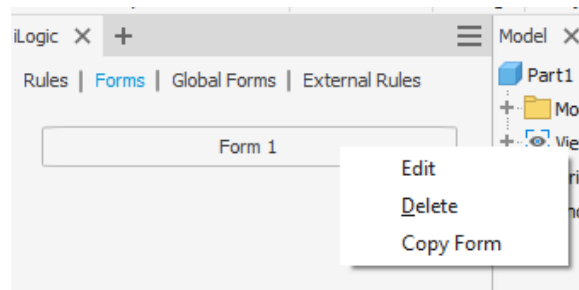
2. В області "Властивості" вкажіть значення "Істина" для властивості модальності і виберіть кнопки у властивості "Попередньо певні кнопки".

Редагування форми або вкладок форм

1. Відкрийте браузер iLogic (Управління > iLogic > Оглядач iLogic).

2. Виконайте одну з таких дій:

Для редагування форми клацніть кнопку форми правою кнопкою миші в браузері iLogic і виберіть "Редагувати".



Для редагування елементів управління в браузері iLogic клацніть правою кнопкою миші пусте місце на вкладці "Форми" або "Глобальні форми" і виберіть команду "Редагувати".

Виконайте одну з таких дій.

Клацніть мітку, щоб відредагувати її.

Перетягніть інструменти з панелі інструментів в дерево проекту або перетягніть елементи між різними компонентами дерева проекту.

Поверніть розділи дерева проекту, щоб полегшити їх перетягування.

Щоб змінити елемент управління або інструмент, виберіть його в дереві проекту і внесіть необхідні зміни в області "Властивості".

Щоб видалити елемент, клацніть його правою кнопкою миші в дереві проекту і виберіть "Видалити". Перед видаленням елемента перетягніть вкладені елементи, розташовані під ним, в інше місце, якщо їх необхідно зберегти.

Для додавання кнопки ініціювання правила перетягніть його зі списку правил.

Щоб елементи керування відображалися завжди, перетягніть їх за межі області вкладки в дереві проекту.

Копіювання і вставка форми

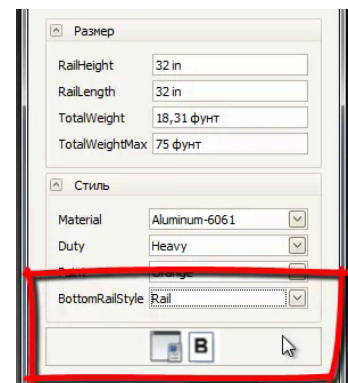
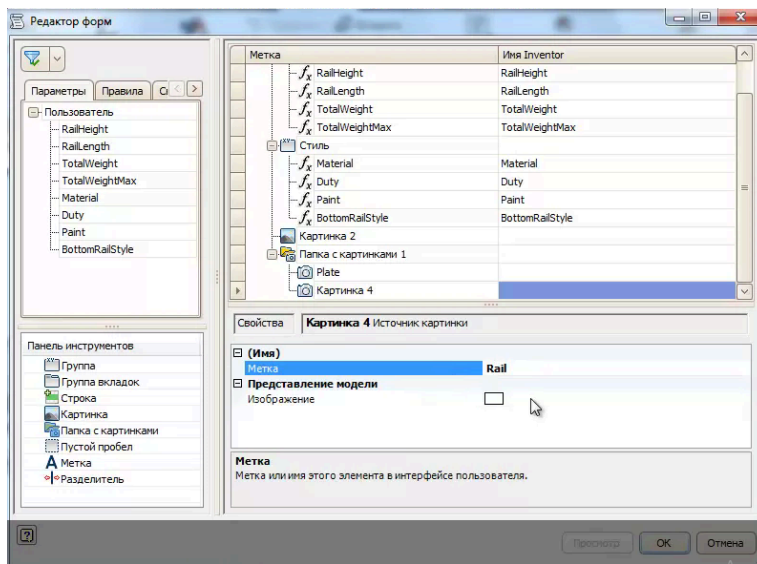
1. У браузері iLogic (Управління > iLogic > Оглядач iLogic) клацніть кнопку форми правою кнопкою миші і виберіть команду "Копіювати форму".

2. Щоб створити нову форму, перейдіть до документа і на вкладку "Форми".

3. Клацніть правою кнопкою миші в порожній області на вкладці "Форми" (або "Глобальні форми") і виберіть команду "Вставити форму".

Використання зображень в формі

Можна додати в форму статичні зображення, а також налаштувати зміна зображень відповідно до значення параметра.



Зміна розміру елементів управління

1. Запустіть форму в браузері iLogic (Управління > iLogic > Оглядач iLogic).

2. Клацніть правою кнопкою миші в будь-якому місці на формі і виберіть команду "Змінити розмір елементів управління". Перетягніть межі полів поруч з порожнім простором, потім перетягніть кордону кнопки.

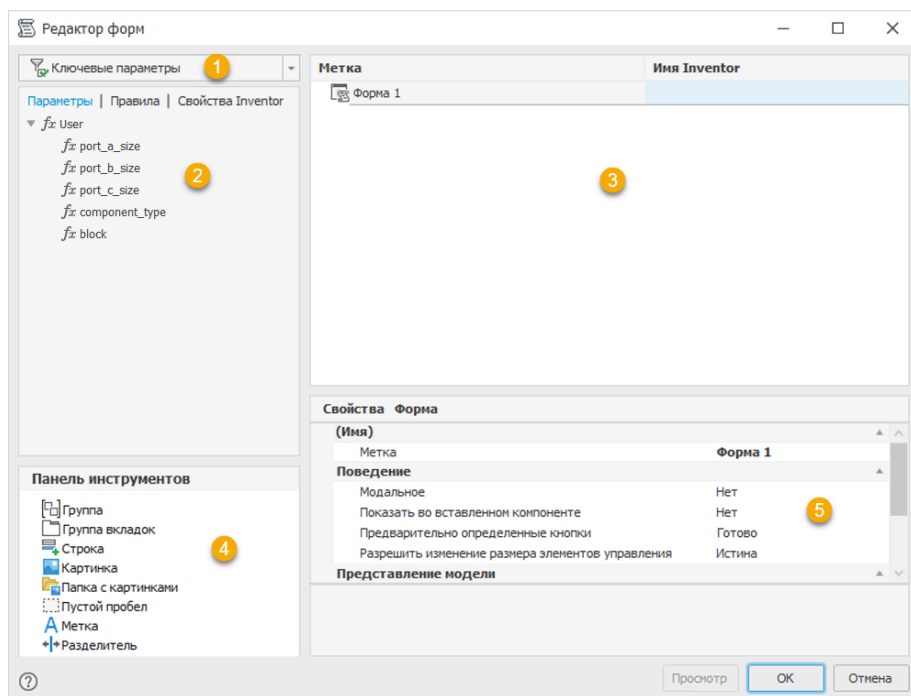
3. Щоб завершити редагування, клацніть правою кнопкою миші і виберіть команду "Вийти з режиму зміни розміру".

4. Щоб відновити вихідні розміри елементів управління, запустіть форму, а потім клацніть правою кнопкою миші і виберіть команду "Відновити розміри елементів управління за замовчуванням".

При подальшому редагуванні форми елементи управління повертаються до розмірів за замовчуванням.

РОБОТА З РЕДАКТОРОМ ФОРМ

Створення або редагування спеціальних форм призначеного для користувача інтерфейсу.



1. Фільтр

Фільтрація списку на вкладках «Параметри» для наступних елементів:
"Всі"

клавіша

перейменований

2. Виберіть

Перетягніть елементи з вкладок в дерево розробки форм для додавання елементів управління в інтерфейс користувача. Тільки існуючі параметри і правила відображаються на вкладках.

3. Дерево розробки форм

Розробка інтерфейсу користувача в області дерева розробки форм.

Перетягніть параметри, правила, властивості Inventor і елементи панелі інструментів в дерево розробки форм для розробки і організації інтерфейсу користувача.

Перетягніть елементи в дерево для організації елементів управління.

Клацніть мітку для редагування тексту.

Виділіть елемент для визначення властивостей в області властивостей.

4. Toolbox

Перетягніть елементи з вкладок в дерево розробки форм для додавання елементів управління в інтерфейс користувача. Тільки існуючі параметри і правила відображаються на вкладках.

Перетягніть елемент в дерево розробки форм.

	згрупувати	Угрупування елементів в інтерфейсі. Групи поміщаються в рамки і згортаються.
	Група з вкладками	Створення вкладки в інтерфейсі користувача.
	рядок	Створення рядка для впорядкування елементів управління по горизонталі. За замовчуванням елементи управління впорядковані по вертикалі.
	зображення	Додавання зображення в інтерфейс користувача. Зображення додаються тільки в інформаційних цілях. Виберіть елемент зображення в дереві розробки форм і задайте файл зображення у властивостях.
	папка зображень	Додавання папки для альтернативних зображень.
	порожня область	Додавання порожньої області в форму інтерфейсу користувача.
	мітка	Додавання текстової мітки в форму інтерфейсу користувача.
	роздільник	Додавання масштабовуваної розділової смуги в інтерфейс користувача.

5. Властивості

Перетягніть елементи з вкладок в дерево розробки форм для додавання елементів управління в інтерфейс користувача. Тільки існуючі параметри і правила відображаються на вкладках.

Визначення властивостей для виділеного елемента в дереві розробки форм. Властивості розрізняються залежно від обраного елемента.

Практична частина:

СТВОРЕННЯ ПРАВИЛА ДЛЯ РОЗМІРІВ

Третє створюване правило буде управляти розмірами скоби. Раніше був створений призначений для користувача параметр з ім'ям mass. Нове правило буде визначати ширину скоби в залежності від значення цього параметра. У першому сценарії ширина скоби змінюється відповідно до наступних значень:

Маса	Ширина скоби
1	10 мм
2	20 мм
3	30 мм
4	40 мм

Додавання значень

Спочатку буде додано набір можливих значень параметра mass. Для відображення в списку тільки ключових параметрів використовуйте меню в області Фільтри. Ці фільтри дозволяють зосередити увагу на параметрі маси.

1. Клацніть правою кнопкою будь-яку вільну позицію рядка mass і виберіть в контекстному меню команду Створити кілька значень.

Відкривається "Редактор списку значень".

2. В поле Додати новий елемент введіть значення 200,300 і 400 (значення 100 вже має бути в списку Значення).

3. Щоб заповнити список Значення, натисніть кнопку Додати, а потім - ОК, щоб прийняти список і повернутися до діалогового вікна "Параметри".

Щоб переглянути список значень можна відкрити спливаюче меню в поле багатозначний рядки mass редактора параметрів.

4. Натисніть кнопку Завершити, щоб завершити реорганізацію параметра mass.

Додавання правила

Спробуємо створити правило, управляє шириною скоби.

1.  Виберіть на стрічці вкладки "Управління" > панель iLogic > Додати правило.

2. Призначте новим правилом ім'я Width_Rule.

У першій частині правила затверджується, що якщо маса дорівнює 100, то ширина скоби дорівнює 10 мм.

3. У текстовій області діалогового вікна "Редагування правила" почніть правило з виразу If.

4. Клацніть вузол Параметри моделі в дереві моделі, потім знайдіть параметр з ім'ям bracket_width на вкладці Параметри праворуч від дерева.

5. Двічі клацніть bracket_width, щоб вставити ім'я параметра в текст правила. Хоча імена параметрів можна набирати безпосередньо в правилі, введення за допомогою подвійного клацання виключає помилки правопису.

6. Встановіть для bracket_width значення 10 мм.

```
If mass = 1 Then  
  bracket_width = 10  
ElseIf mass = 2 Then  
  bracket_width = 20  
ElseIf mass = 3 Then  
  bracket_width = 30  
ElseIf mass = 4 Then  
  bracket_width = 40  
End If
```

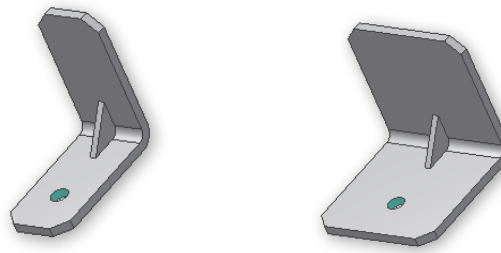
7. Щоб зберегти це правило, натисніть ОК.

Перевірка правила

1.  Відкрийте діалогове вікно "Параметри".

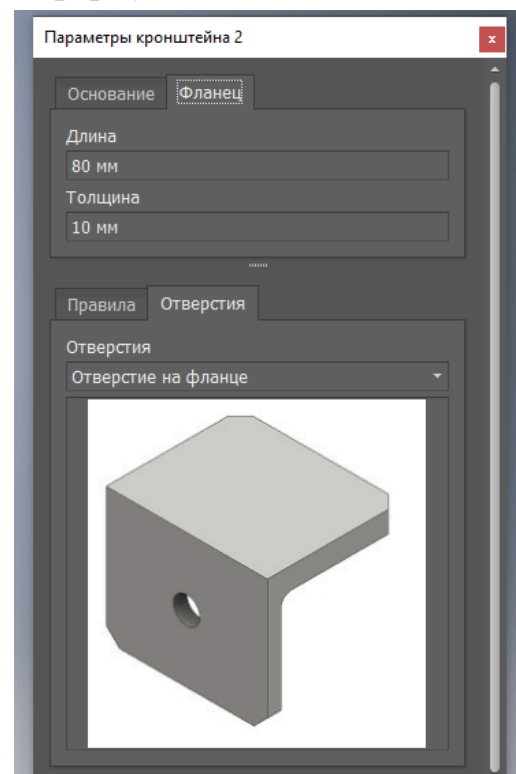
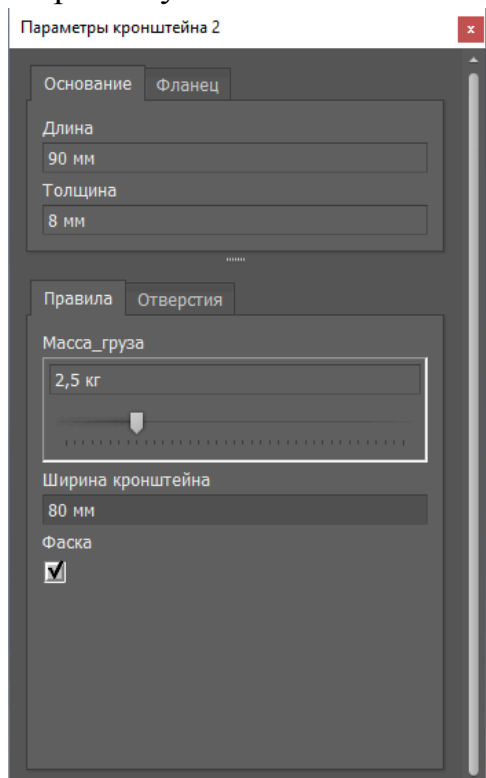
2. Встановіть для параметра mass значення 100. Зверніть увагу, що для параметра bracket_width встановлено значення 10 мм.

3. Змініть значення параметра mass на 200 і зверніть увагу на те, що ширина скоби знову змінилася.



Натисніть кнопку "Оновити" для параметра "Ширина", щоб застосувати зміни до графіку.

Використовуючи можливості iLogic створюємо форму



Висновок частина 4. Під час виконання лабораторної роботи необхідно навчитися створювати правила для розмірів з використанням можливостей iLogic, створювати та редагувати форми.

Запитання до частини 4.

1. Як створити форму користувальницького інтерфейсу?
2. Як створити модальну форму користувальницького інтерфейсу?
3. Як можливо проводити редагування форми або вкладок форм?
4. Як можливо використовувати зображення в формі?
5. Як проводити роботу з редактором форм?

Лабораторна робота № 5

Частина 1 «Робота зі складальними одиницями з використанням можливостей iLogic»

Частина 2 «Керування присвоєнням матеріалів компонентам складальної одиниці з використанням можливостей iLogic»

Частина 3 «Заміна групи компонентів в складальній одиниці з використанням можливостей iLogic»

Мета роботи частини 1: Навчитися створювати та редагувати параметри в складальній одиниці з використанням можливостей iLogic та використати на практиці здобуті знання.

Теоретична частина:

Імпорт та експорт параметрів

Різні параметри можна імпортувати з файлу XML та експортувати до нього. Також параметри з файлу XML можна експортувати як компонент.

Імпорт параметрів із файлу XML та їх експорт у файл XML

1. Відкрийте модель, яка містить параметри.
2. Виберіть на стрічці вкладки "Керування" > панель "Параметри" >

"Експорт у файл XML"  або Імпорт із файлу XML .

Відобразяться діалогові вікна "Експорт у файл XML" (або "Імпорт із файлу XML").

3. Для експорту необхідно встановити розташування XML та його ім'я або прийняти запропоновані значення.

Натисніть кнопку "Параметри", виберіть "Тільки ключові параметри" або "Всі параметри", а потім натисніть кнопку "ОК".

Натисніть кнопку "ОК", щоб експортувати параметри.

Для імпорту необхідно знайти файл XML, який містить параметри, експортовані з іншої моделі за допомогою діалогового вікна.

Якщо потрібно, натисніть кнопку "Пошук", щоб відкрити діалогове вікно "Пошук" і знайдіть файл XML.

Щоб імпортувати установки, натисніть ОК.

Експорт моделі та параметрів користувача у властивості компонента

1. Виберіть на стрічці вкладки "Керування" > панель "Параметри" >

Параметри .

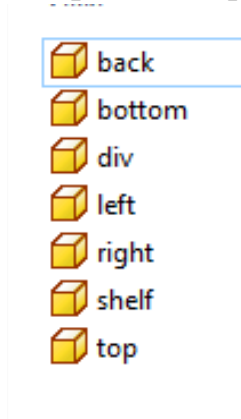
2. У діалоговому вікні "Параметри" у стовпці "Експортувати параметри" виберіть параметр, який потрібно експортувати.

Щоб задати формат для експортованого параметра, клацніть правою кнопкою миші рядок параметрів та виберіть пункт "Формат додаткових властивостей".

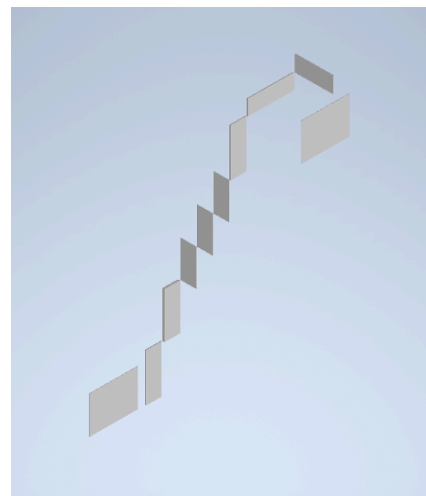
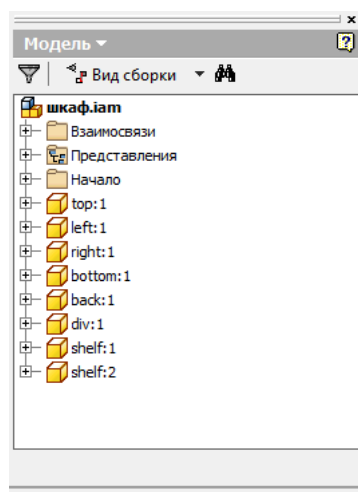
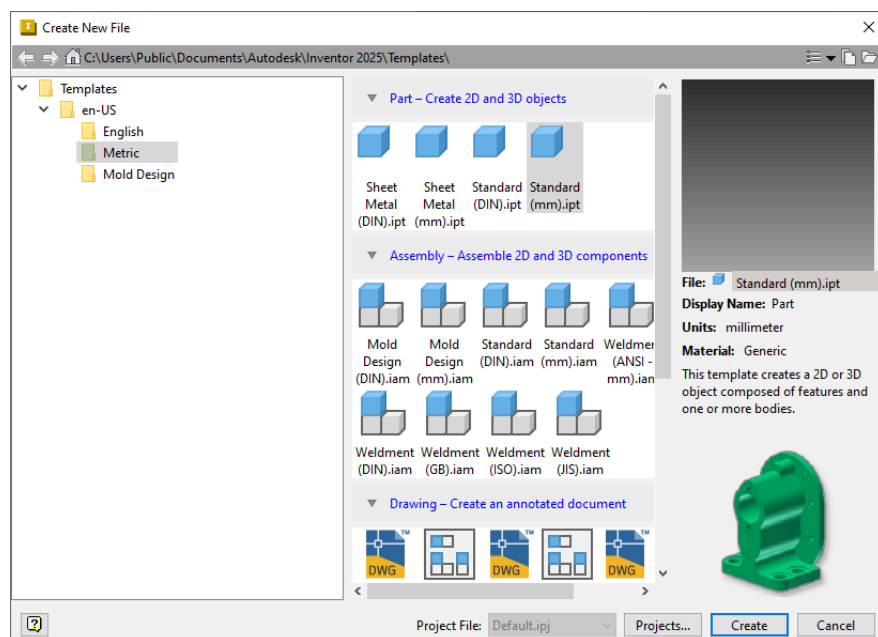
3. Натисніть кнопку "Завершити".

Практична частина:

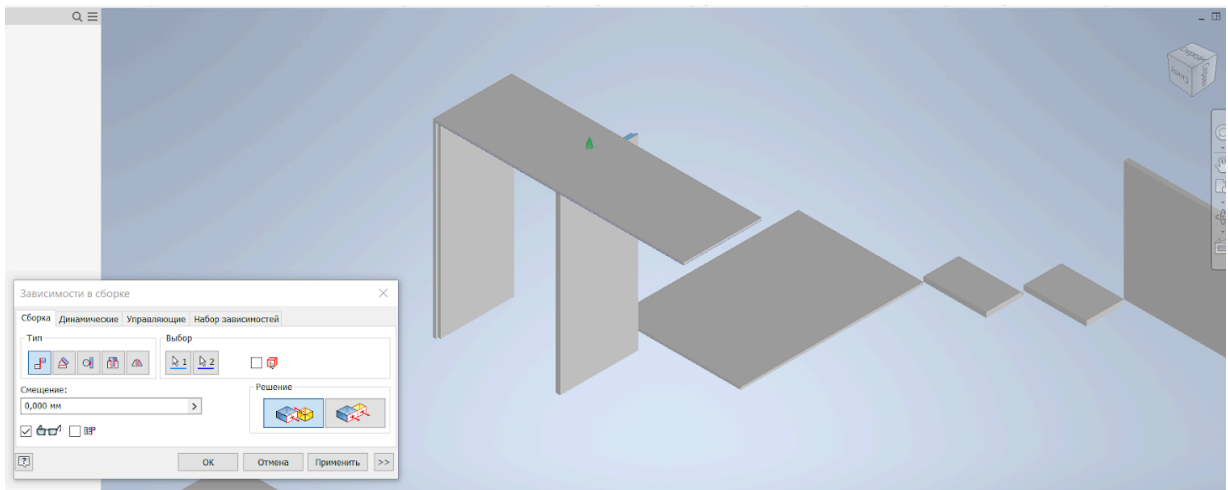
1. Створити всі елементи меблі «Шафи» та зберегти їх в окрему папку.



2. Створення файлу складальної одиниці.



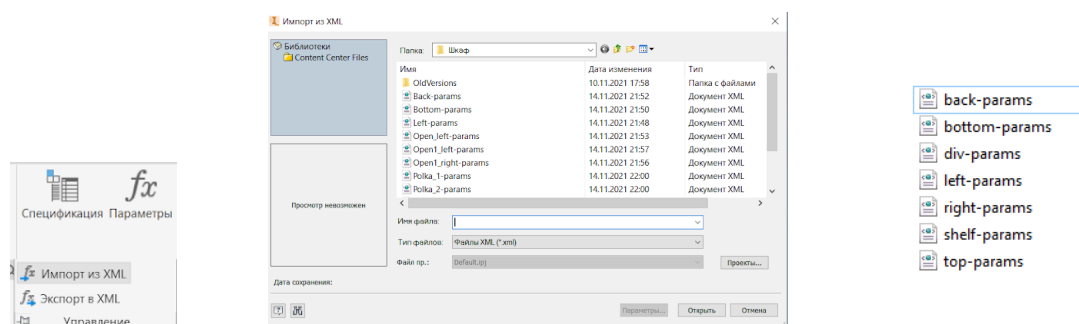
3. Позиціонування всіх елементи шафи у правильному положенні. Для цього використовуємо команду «Залежність».



Результат:



4. Імпортування параметрів створених елементів для зручності роботи зі збіркою.



Тепер у користувацьких параметрах з'явилися всі імпортовані значення.

Имя параметра	Используется	Единица	Формула	Норм. знач.	Доп.	Значение в мм, Косм.	Вид	Примечание
d36	Совмещает...	мм	0,00 мм	0,000000	●	0,000000	■	
d37	Совмещает...	мм	0,00 мм	0,000000	●	0,000000	■	
Пользовательские								
width_top		мм	600 мм	600,000000	●	600,000000	■	
deep_top		мм	250 мм	250,000000	●	250,000000	■	
d21		мм	16 мм	16,000000	●	16,000000	■	
height_back		мм	750 мм	750,000000	●	750,000000	■	
width_back		мм	584 мм	584,000000	●	584,000000	■	
d211		мм	8 мм	8,000000	●	8,000000	■	
d312		град	0,00 град	0,000000	●	0,000000	■	
deep_bottom		мм	250 мм	250,000000	●	250,000000	■	
width_bottom		мм	568 мм	568,000000	●	568,000000	■	
d310		град	0,00 град	0,000000	●	0,000000	■	
d03		мм	16 мм	16,000000	●	16,000000	■	
deep_left		мм	250 мм	250,000000	●	250,000000	■	
d24		мм	8 мм	8,000000	●	8,000000	■	
height_left		мм	750 мм	750,000000	●	750,000000	■	
d45		град	0,00 град	0,000000	●	0,000000	■	
d023		мм	10 мм	10,000000	●	10,000000	■	
deep_openleft_1		мм	285 мм	285,000000	●	285,000000	■	
width_openleft_1		мм	302 мм	302,000000	●	302,000000	■	
d324		град	0,00 град	0,000000	●	0,000000	■	
d021		мм	16 мм	16,000000	●	16,000000	■	
deep_openleft		мм	250 мм	250,000000	●	250,000000	■	
height_openleft		мм	734 мм	734,000000	●	734,000000	■	

5. Створення правила «ParametersLink», в якому виконується звернення до параметрів кожної деталі та присвоюємо їм значення для подальшої роботи.

```

Parameter("Top:1", "deep_top")=deep_top
Parameter("Left:1", "deep_left")=deep_left
Parameter("Left:1", "height_left")=height_left
Parameter("Right:1", "deep_right")=deep_right
Parameter("Right:1", "height_right")=height_right
Parameter("Bottom:1", "deep_bottom")=deep_bottom
Parameter("Bottom:1", "width_bottom")=width_bottom
Parameter("Back:1", "height_back")=height_back
Parameter("Back:1", "width_back")=width_back
Parameter("Openleft:1", "deep_openleft")=deep_openleft
Parameter("Openleft:1", "height_openleft")=height_openleft
Parameter("Stenka:1", "height_stenka")=height_stenka
Parameter("Stenka:1", "deep_stenka")=deep_stenka
Parameter("Polka1:1", "deep_polka1")=deep_polka1
Parameter("Polka1:1", "width_polka1")=width_polka1
Parameter("Polka2:1", "deep_polka2")=deep_polka2
Parameter("Polka2:1", "width_polka2")=width_polka2

```

6. Створюємо ще правило «Parameters». До цього створюємо два числових параметри Width, Height зі списком значень.

Имя параметра	Используется	Единица	Формула	Норм. знач.	Правило приведения в	Доп.	Значение в мм, Косм.	Вид	Примечание
d215		мм	10 мм	10,000000		●	10,000000	■	
d316		град	0,00 град	0,000000		●	0,000000	■	
deep_polka3		мм	242 мм	242,000000	Parameters	●	242,000000	■	
width_polka3		мм	302 мм	302,000000	Parameters	●	302,000000	■	
d213		мм	10 мм	10,000000		●	10,000000	■	
d314		град	0,00 град	0,000000		●	0,000000	■	
d06		мм	16 мм	16,000000		●	16,000000	■	
deep_right		мм	250 мм	250,000000	Parameters	●	250,000000	■	
height_right		мм	750 мм	750,000000	Parameters	●	750,000000	■	
d48		град	0,00 град	0,000000		●	0,000000	■	
d019		мм	16 мм	16,000000		●	16,000000	■	
height_stenka		мм	735 мм	735,000000	Parameters	●	735,000000	■	
deep_stenka		мм	242 мм	242,000000	Parameters	●	242,000000	■	
d220		град	0,00 град	0,000000		●	0,000000	■	
Height		мм	750 мм	750,000000		●	750,000000	■	
Width		мм	600 мм	600,000000		●	600,000000	■	
Deep		мм	250 мм	250,000000		●	250,000000	■	

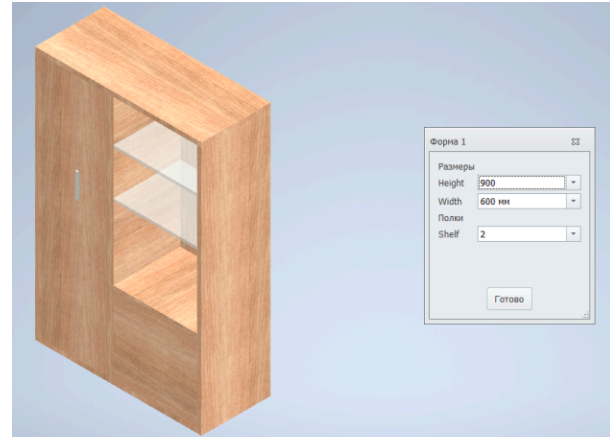
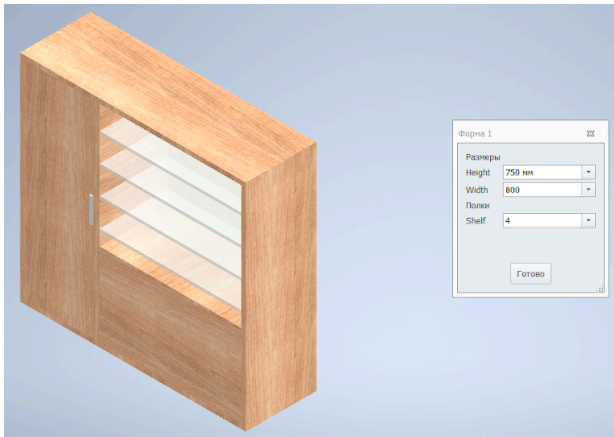
```

deep_bottom=Deep
deep_right=Deep
deep_openleft = Deep

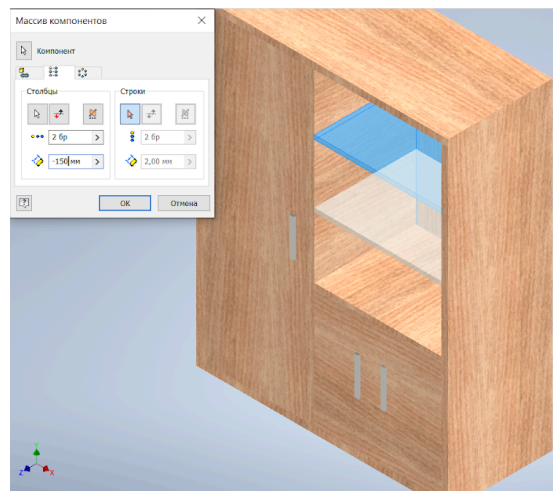
width_back = Width - 16
width_bottom = Width - 32
deep_openleft_1 = Deep + 35
width_openleft_1 = Width - 298
height_openleft = Height - 16
deep_polka1 = Deep - 8
width_polka1 = Width - 298
deep_polka2 = Deep - 8
width_polka2 = Width - 298
deep_polka3 = Deep - 8
width_polka3 = Width - 298
height_stenka = Height - 15
deep_stenka = Deep - 8

```

7. Створюємо форму.

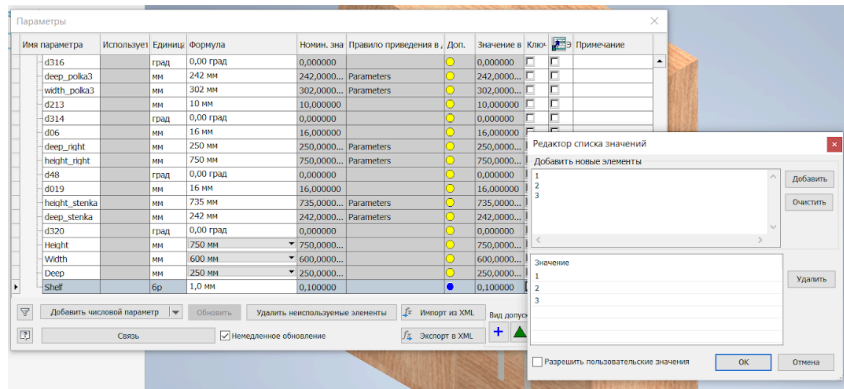


8. Створюємо масив для полиць.

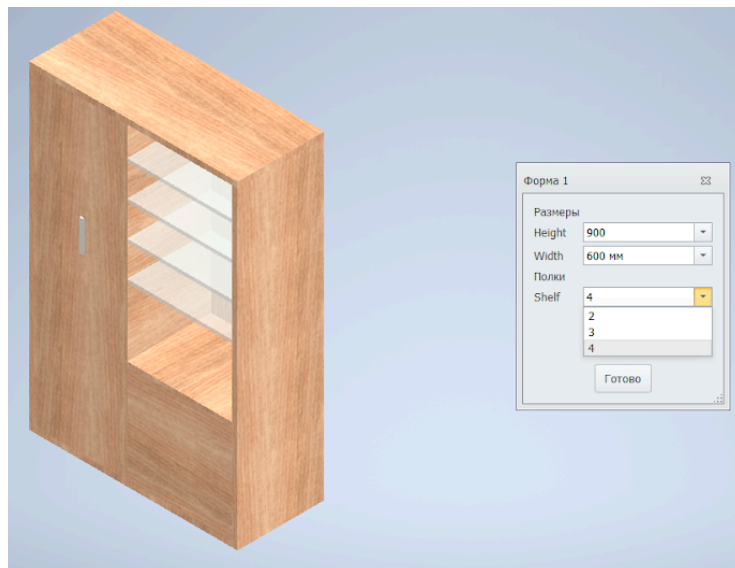


9. Створюємо правило, яке керує кількістю полиць.

Для цього попередньо створюємо числовий параметр «Shelf» зі списком список значень.



Результат:



Висновок частини 1.

Під час виконання лабораторної роботи навчитися створювати та редагування параметри в складальній одиниці з використанням можливостей iLogic та використати на практиці здобуті знання.

Запитання до частини 1.

1. Який тип файлу використовується для імпорту та експорту параметрів?
2. Як проводити імпорт параметрів моделі?
3. Як проводити експорт параметрів моделі?
4. Як виконувати експорт моделі у властивості компонента?
5. Як виконувати експорт параметрів користувача у властивості компонента?

Частина 2 «Керування присвоєнням матеріалів компонентам складальної одиниці з використанням можливостей iLogic»

Мета частини 2. Навчитися створювати та керувати присвоєнням матеріалів компонентам складальної одиниці з використанням можливостей iLogic та використати на практиці здобуті знання.

Теоретична частина:

В iLogic містяться функції для правил, що дозволяють задавати або вилучати властивості Inventor для документів деталей, складання та креслення Inventor. У міру використання правил, щоб застосувати до проектів нові конфігурації, оновлюйте властивості Inventor у документах проекту. Щоб забезпечити правильність специфікації, необхідно використовувати останні властивості Inventor. Правила також можна використовувати, щоб читати значення властивостей Inventor з документа і реагувати на них. Наприклад, можна використовувати правило зміни товщини деталі при зміні властивості матеріалу. Правило можна застосовувати для внесення змін, коли маса деталі виходить за граничні значення. Будь-яка властивість Inventor, яка задається вручну, можна також задати або прочитати за допомогою правила.

Щоб перейти до функцій iProperties, розгорніть вузол "iProperties" на вкладці "Система" в області "Фрагменти".

iProperties.Material

Задає чи читає матеріал документа.

При заданні матеріалу використовуйте точне найменування матеріалу (чутливо до регістру) у вигляді, як воно включено до активного стандартного списку матеріалів. Значення матеріалу — це текстові рядки, укладені в лапки.

Приклади

Щоб задати матеріал:

iProperties.Material = "Gold"

Щоб прочитати матеріал:

MyStringParam = iProperties.Material

iProperties.Material("filename.ipt")

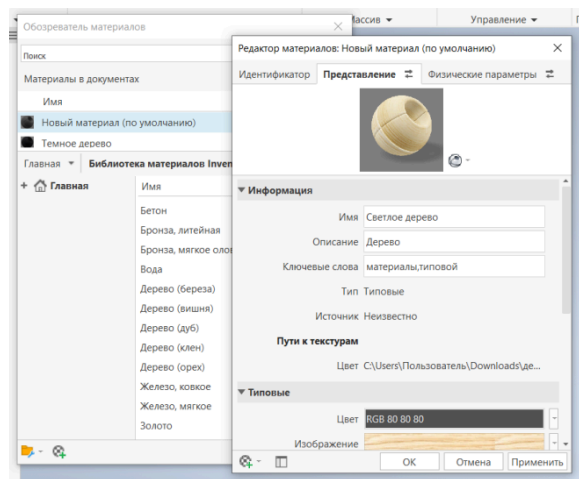
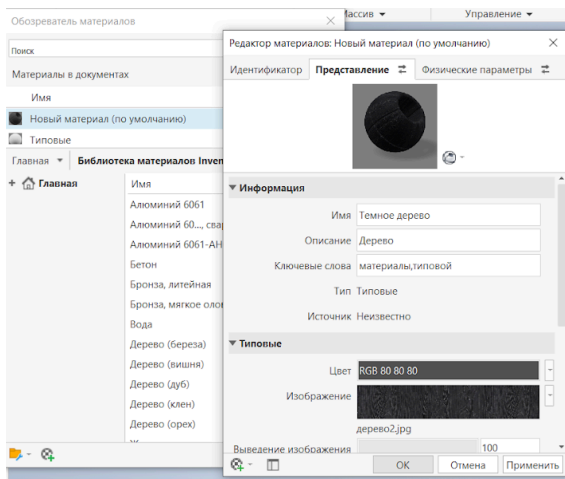
Задає або читає матеріал деталі всередині складання за заданим ім'ям файлу.

iProperties.Material("componentName:1")

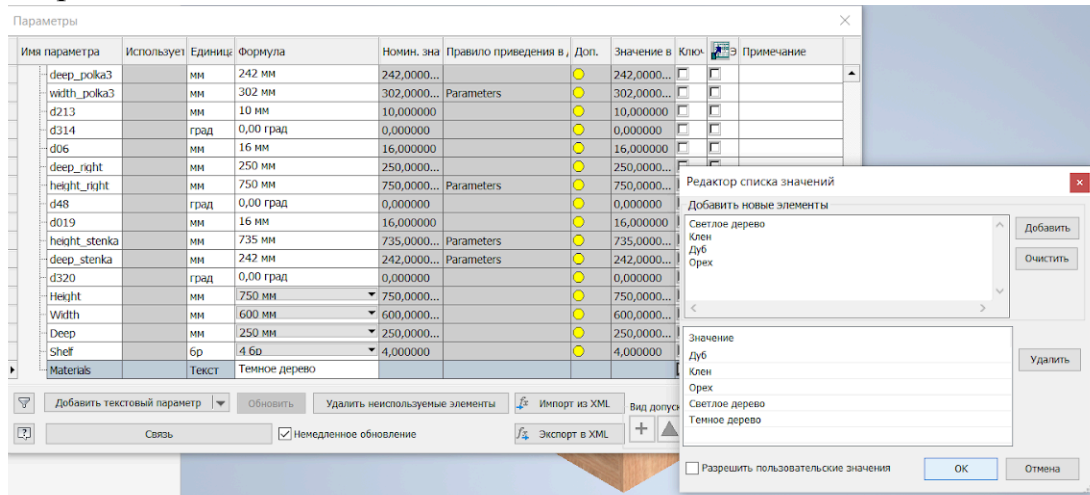
Використовується в правилі на рівні складання для отримання доступу до матеріалу деталі у складання за заданим ім'ям компонента.

Практична частина:

1. Створюємо необхідні матеріали. Для цього натискаємо на плюс та починаємо створювати два матеріали.



2. Створюємо правило «Materials», а перед цим як текстовий параметр зі списком матеріалів



3. Фрагмент коду цього правила представлений на зображенні.

```

If Material = "Дуб" Then

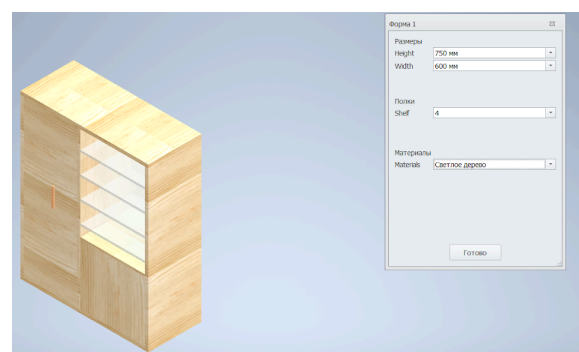
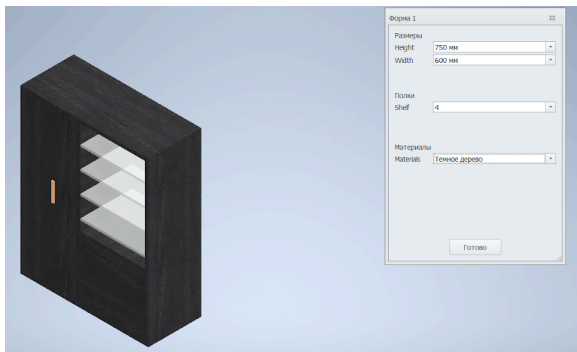
Material = iProperties.Material ("top:1")
iProperties.Material("top:1") = "Дерево (дуб)"
Material = iProperties.Material ("left:1")
iProperties.Material("left:1") = "Дерево (дуб)"
Material = iProperties.Material ("right:1")
iProperties.Material("right:1") = "Дерево (дуб)"
Material = iProperties.Material ("bottom:1")
iProperties.Material("bottom:1") = "Дерево (дуб)"
Material = iProperties.Material ("div:1")
iProperties.Material("div:1") = "Дерево (дуб)"
Material = iProperties.Material ("shelf:1")
iProperties.Material("shelf:1") = "Дерево (дуб)"
Material = iProperties.Material ("shelf:2")
iProperties.Material("shelf:2") = "Дерево (дуб)"

ElseIf Material = "Береза" Then

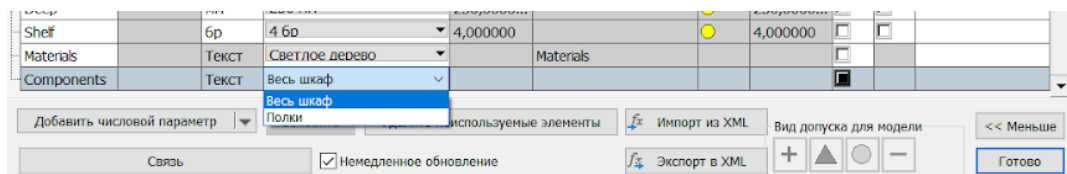
Material = iProperties.Material ("top:1")
iProperties.Material("top:1") = "Дерево (береза)"
Material = iProperties.Material ("left:1")
iProperties.Material("left:1") = "Дерево (береза)"
Material = iProperties.Material ("right:1")
iProperties.Material("right:1") = "Дерево (береза)"
Material = iProperties.Material ("bottom:1")

```

Результат:



4. Створюємо ще один текстовий параметр зі списком компонентів збірки.



5. Після чого доповнюємо правило


```

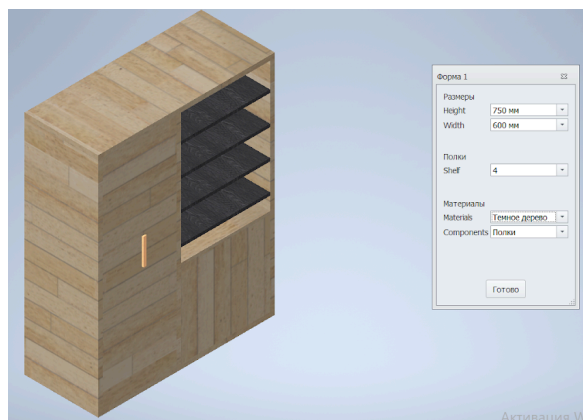
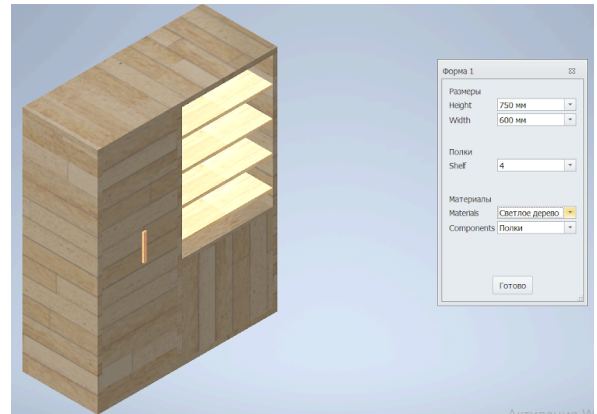
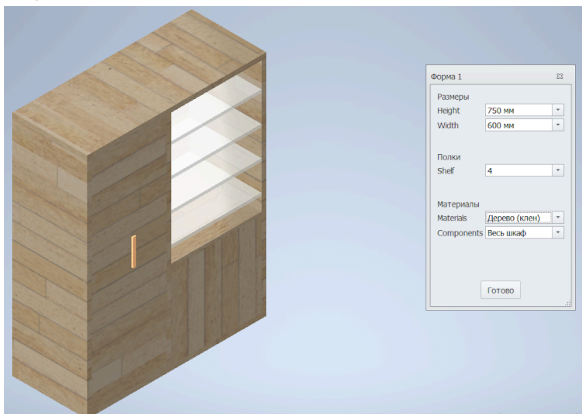
1
If Components = "Вся модель" Then

    If Material = "Дуб" Then
        ' [
        Material = iProperties.Material ("top:1")
        iProperties.Material("top:1") = "Дерево (дуб) "
        Material = iProperties.Material ("left:1")
        iProperties.Material("left:1") = "Дерево (дуб) "
        Material = iProperties.Material ("right:1")
        iProperties.Material("right:1") = "Дерево (дуб) "
        Material = iProperties.Material ("bottom:1")
        iProperties.Material("bottom:1") = "Дерево (дуб) "
        Material = iProperties.Material ("div:1")
        iProperties.Material("div:1") = "Дерево (дуб) "
        Material = iProperties.Material ("shelf:1")
        iProperties.Material("shelf:1") = "Дерево (дуб) "
        Material = iProperties.Material ("shelf:2")
        iProperties.Material("shelf:2") = "Дерево (дуб) "
        ' ]

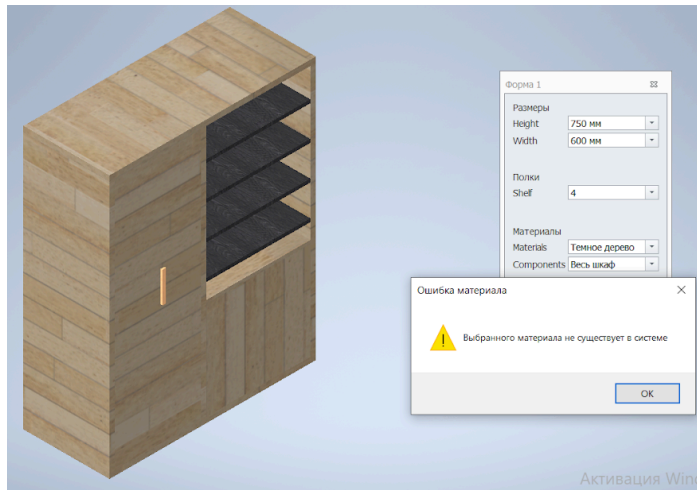
    ElseIf Material = "Береза" Then
        ' [
        Material = iProperties.Material ("top:1")
        iProperties.Material("top:1") = "Дерево (береза) "
        Material = iProperties.Material ("left:1")

```

Результат:



Якщо матеріал не відповідає компоненту з'являється попередження



Висновок частина 2. Під час виконання лабораторної роботи необхідно навчитися створювати та керувати присвоєнням матеріалів компонентам складальної одиниці з використанням можливостей iLogic та використати на практиці при створенні шафи.

Запитання до частини 2.

1. Як завдати властивості матеріалу?
2. Як можливо прочитати властивості матеріалу?
3. У якому вигляді представлені значення матеріалу?
4. Можливо задати або зчитати матеріал деталі всередині складання?
5. Як створити новий матеріал?

Частина 3 «Заміна групи компонентів в складальній одиниці з використанням можливостей ILogic»

Мета частини 3. Навчитися замінювати чи керувати видимістю груп компонентів в складальній одиниці з використанням можливостей ILogic та використати на практиці здобуті знання.

Теоретична частина:

Component.Visible

Задає чи читає видимість компонента. Ця функція не змінює структури специфікації компонента.

Синтаксис:

Component.Visible("componentName")

Приклади:

Задати видимість:

Component.Visible("Plate") = true

Component.Visible("Bearing") = false

Читати видимість:

parameter = Component.Visible("Plate")

parameter = Component.Visible("Bearing")

If Component.Visible("PartE") Then

do something

End If

Component.Replace

Замінює одну деталь на інший або один вузол іншим. Цю функцію також можна використовувати для заміни масивів компонентів.

У замінених компонентах використовуйте iMates, щоб залишити незмінними залежності збирання. Можна замінити одну деталь іншою, деталь складання або складання деталлю.

Функція шукає файл, який слід використовувати для заміни, у кількох папках:

- у папці, в якій знаходиться документ документа, що замінюється;

- у папці, де знаходиться документ складання;

- у каталозі робочого простору поточного проекту.

Як ім'я файлу можна використовувати відносний шлях (щодо будь-якого з цих каталогів пошуку).

До цих функцій "стабілізуйте" ім'я компонента, щоб не змінювати його після заміни. Щоб стабілізувати ім'я компонента, змініть його. Його можна стабілізувати, навіть якщо змінити його, а потім повернути назад вихідне ім'я. Якщо ім'я не стабілізується, операція заміни співвідносить компонент з іншим ім'ям. Після цього, при наступному запуску правила, воно не може знайти компонент.

Синтаксис:

`Component.Replace("ComponentToReplaceName", "OtherPartfilename.ipt",
<replaceAll>)`

ComponentToReplaceName

Ім'я деталі, що замінюється, або вузла, що замінюється.

OtherPartfilename

Деталь або вузол, які слід використовувати як заміну.

`<replaceAll>`

Надайте логічному змінному значення Правда заміни всіх екземплярів компонента. Надайте їй значення помилковий, щоб замінити лише один іменований екземпляр.

Приклад:

`If PartType = "Type A" Then`

`Component.Replace("Widget", "PartA.ipt", True)`

`ElseIf PartType = "Type B" Then`

`Component.Replace("Widget", "PartB.ipt", True)`

`End If`

Component.ReplacePart

Ця функція необхідна для параметричних деталей з параметрами користувача, але вона також рекомендується для стандартних параметричних деталей. Якщо компонент є параметричною деталлю, використовуйте її замість Компонент.Замінити. Щоб змінити конфігурацію певної параметричної деталі, після заміни можна використовувати `iPart.ChangeRow` або `iPart.FindRow`.

Для параметричної деталі з параметрами користувача наведіть список значень параметрів користувача після аргументу `rowNumber`. Значення необхідно перерахувати у порядку, у якому включені в таблицю.

Синтаксис:

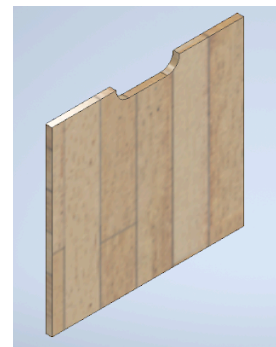
`Component.ReplacePart("iPart1:1", "OtherPartfilename.ipt", True, rowNumber)`

Використовуйте `rowNumber`, щоб замінити компонент та одночасно вибрати рядок параметричних деталей.

Практична частина:

1. Створюємо правило для заміни дверцят шафи. Перед цим створюємо текстовий параметр зі списком та будуємо другий варіант компонента для заміни.

Ім'я параметра	Использует	Единица	Формула	Номин. зна	Правило приведения в	Доп.	Значение в	Класс	Примечание
deep_sterka		мм	242 мм	242,0000...	Parameters		242,0000...		
d320		град	0,00 град	0,000000			0,000000		
Height		мм	750 мм	750,0000...			750,0000...		
Width		мм	600 мм	600,0000...			600,0000...		
Deep		мм	250 мм	250,0000...			250,0000...		
Shelf		бр	4 бр	4,000000			4,000000		
Materials	Текст	Дерево (lvb)			Materials				
Components	Текст	Весы шкафа							
sd023		мм	10 мм	10,000000			10,000000		
deep_openi...		мм	285 мм	285,0000...			285,0000...		
width_openi...		мм	302 мм	302,0000...			302,0000...		
sd324		град	0,00 град	0,000000			0,000000		
sd6		мм	20 мм	20,000000			20,000000		
openleft_bo...		мм	60,000 мм	60,000000			60,000000		
sd9		мм	30 мм	30,000000			30,000000		
sd10		град	0,00 град	0,000000			0,000000		
Variant	Текст	С ручкой							
		Без							



2. Фрагмент коду цього правила представлений на зображенні.

```

If Variant = "С ручкой" Then
|
Component.Visible("Openleft_box") = False
Component.Visible("Openleft_1") = True

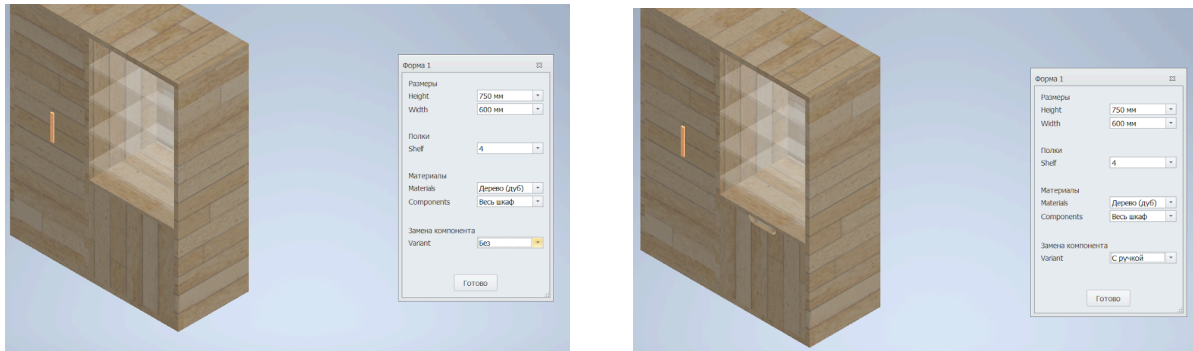
ElseIf Variant = "Без" Then

Component.Visible("Openleft_1") = False
Component.Visible("Openleft_box") = True

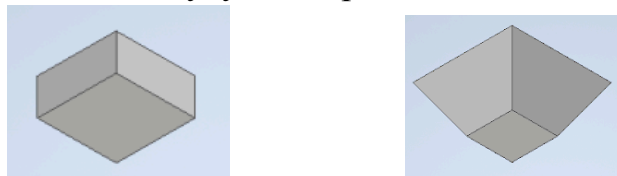
End If

```

Результат:



3. Створюємо правило для заміни ніжок шафи. Перед цим створюємо текстовий параметр зі списком та будуємо варіанти компонентів для заміни



4. Фрагмент коду цього правила представлений на зображенні.

```

If VariantFoot = "Конусные" Then

Component.Replace("Foot:1", "Foot_1.ipt", True)

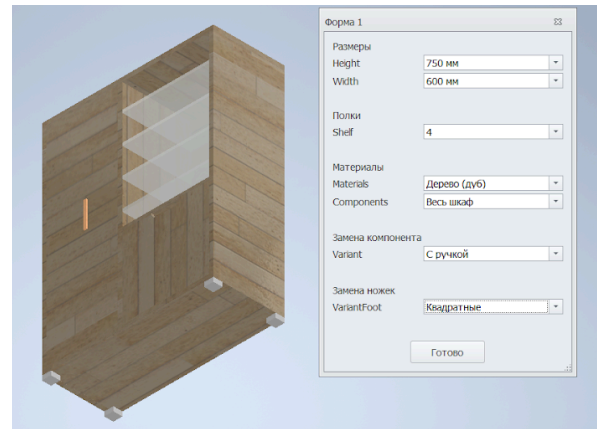
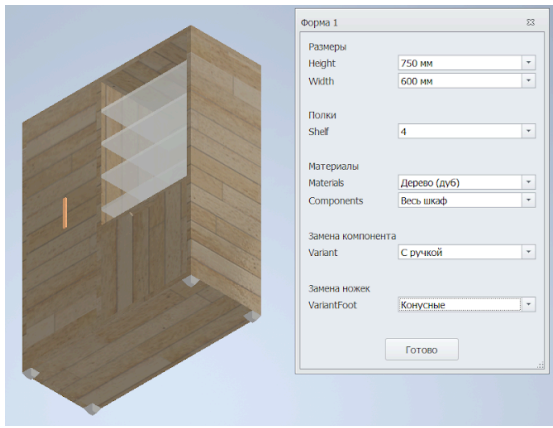
ElseIf VariantFoot = "Квадратные" Then

Component.Replace("Foot_1:1", "Foot.ipt", True)

End If

```

Результат:



Висновок частини 3. Під час виконання лабораторної роботи необхідно навчитися замінювати чи керувати видимістю груп компонентів в складальній одиниці з використанням можливостей ILogic та використати на практиці здобуті знання при виконанні індивідуального завдання.

Запитання до частини 3.

1. Як задати видимість компонента?
2. Як зчитати видимість компонента?
3. Як налаштувати заміну однієї деталі на іншу?
4. Яка функція необхідна для параметричних деталей з параметрами користувача при замінах?
5. Можливо ли для параметричної деталі з параметрами користувача наводити список значень?

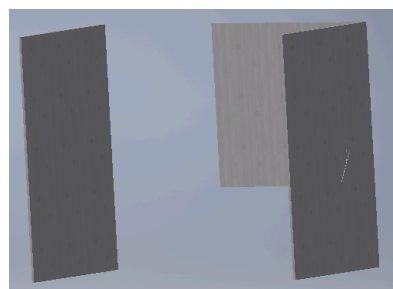
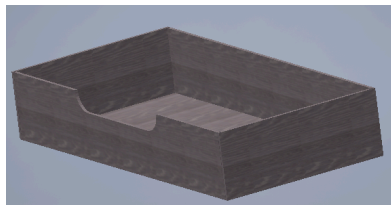
Лабораторна робота № 6

Тема: «Побудова складального вузла зі складальних одиниць з використанням можливостей ILogic»

Мета роботи: з використанням можливостей iLogic створити програмний модуль для керування розмірами та конструкцією складного складального вузла.

Практична частина

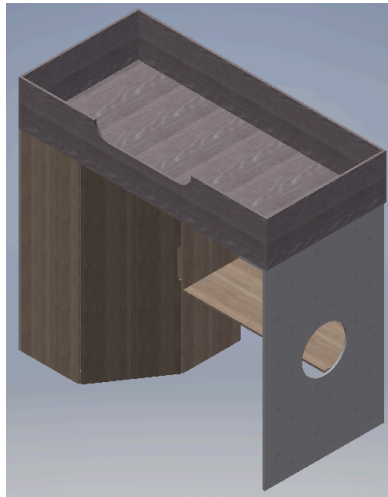
1. Створюємо необхідні складальні одиниці "Шафа", "Стіл", "Ліжко" та "База". Та позиціонуємо всі елементи тумби, стола, панелі у правильному положенні. Для цього використовуємо команду «Залежність».



2. Експортуємо параметри столу, тумби та панелі та імпортуємо їх в основний файл складального вузла

Имя параметра	Исполн	Ед	Формула	Норм	Правило приведения к	Значен	Кл	Примечание
-214_кровати	мм	0,000 мм	0,00...			0,00...		
-215_кровати	мм	0,000 мм	0,00...			0,00...		
высота_опрокидывания_кровати	мм	350 мм	350...	Размеры		350...		
ширина_боковой_панели_кровати	мм	932 мм	932...	Параметры		932...		
длина_стола	мм	1100 мм	1100...	Параметры		1100...		
ширина_стола	мм	434 мм	434...	Параметры		434...		
тип_шкафа	Т...	угловой						
перекладина_панель_кровати	Т...	волна						
тип_боковой_панели_базы	Т...	круглое окно						
возраст_ребенка	Т...	старше 12						
материал	Т...	Дуб синяя						
Комплектация	Т...	Кровать						
S1	бр	2,4 бр	2,40...			2,40...		
S2	н	0 н	0,00...			0,00...		
S3	н	0 н	0,00...			0,00...		
площадь_материала	н	0 н	0,00...			0,00...		
площадь_углового_шкафа	бр	0,4 бр	0,40...			0,40...		
площадь_стен_углового_шкафа	бр	1,35 бр	1,35...	Площадь		1,35...		
площадь_задней_панели_шкафа	бр	1,15 бр	1,15...	Площадь		1,15...		
-231_шкаф	мм	0,000 мм	0,00...			0,00...		
-232_шкаф	мм	0,000 мм	0,00...			0,00...		
-233_шкаф	мм	0,000 мм	0,00...			0,00...		
-234_шкаф	мм	0,000 мм	0,00...			0,00...		
площадь_боковой_панели_базы	бр	1,35 бр	1,35...	Площадь		1,35...		
площадь_дверей_прямого_шкафа	бр	1,15 бр	1,15...	Площадь		1,15...		
площадь_дн_прямого_шкафа	бр	0,44 бр	0,44...	Площадь		0,44...		
площадь_дн_углового_шкафа	бр	0,4 бр	0,40...	Площадь		0,40...		
площадь_шкафа	бр	4,25 бр	4,25...	Площадь		4,25...		

3. В основной файл складального вузла вставляем складальні одиниці "Шафа", "Стіл", "Ліжко" та "База". Встановлюємо між ними залежності



6. Створюємо правило «Link», в якому реалізується зв'язок між імпортованими параметрами в основному файлі та параметрами складальних одиниць для подальшої роботи.

```
[  Параметры ШКАФА

Parameter("шкаф:1", "высота_шкафа")=высота_шкафа
Parameter("шкаф:1", "доска")=доска
Parameter("шкаф:1", "d3")=d3
Parameter("шкаф:1", "ручка_шкафа")=ручка_шкафа
Parameter("шкаф:1", "ручка_шкафа_диам")=ручка_шкафа_диам
Parameter("шкаф:1", "d4")=d4
Parameter("шкаф:1", "ширина_шкафа")=ширина_шкафа
Parameter("шкаф:1", "ширина_дна_шкафа")=ширина_дна_шкафа
Parameter("шкаф:1", "ширина_двери_прямого_шкафа")=ширина_двери_прямого_шкафа
Parameter("шкаф:1", "ширина_дна_шкафа")=ширина_дна_шкафа
Parameter("шкаф:1", "глубина_прямого_шкафа")=глубина_прямого_шкафа
Parameter("шкаф:1", "глубина_дна_шкафа")=глубина_дна_шкафа
Parameter("шкаф:1", "ширина_кровати")=ширина_кровати
Parameter("шкаф:1", "ширина_стойки_углового_шкафа")=ширина_стойки_углового_шкафа
Parameter("шкаф:1", "ширина_стойки_углового_шкафа2")=ширина_стойки_углового_шкафа2
Parameter("шкаф:1", "высота_дна_шкафа")=высота_дна_шкафа
Parameter("шкаф:1", "доска2")=доска2
Parameter("шкаф:1", "d5")=d5
Parameter("шкаф:1", "ширина_задней_панели_шкафа")=ширина_задней_панели_шкафа
Parameter("шкаф:1", "высота_плоскости_дна_шкафа")=высота_плоскости_дна_шкафа
Parameter("шкаф:1", "глубина_прямого_шкафа")=глубина_прямого_шкафа
']

[  Параметры БАЗЫ
Parameter("База:1", "высота_задней_стенки")=высота_задней_стенки
Parameter("База:1", "длина_кровати")=длина_кровати
Parameter("База:1", "ширина_шкафа")=ширина_шкафа
Parameter("База:1", "доска")=доска
Parameter("База:1", "d3")=d3
Parameter("База:1", "длина_задней_стенки")=длина_задней_стенки
Parameter("База:1", "ширина_кровати")=ширина_кровати
Parameter("База:1", "высота_шкафа")=высота_шкафа
Parameter("База:1", "высота_базы")=высота_базы
```

7. Були створені інші правила для складального вузла, такі як:
Зміна типу шафи:

```

If тип_шкафа = "угловой" Then
Component.Visible("Дно углового шкафа:1") = 1
Component.Visible("Стойка углового шкафа:1") = 1
Component.Visible("Стойка углового шкафа:2") = 1
Component.Visible("Дно прямого шкафа:1") = 0
Component.Visible("Стойка прямого шкафа:1") = 0
Component.Visible("Дверь прямого шкафа Л:1") = 0
Component.Visible("Дверь прямого шкафа П:1") = 0

ElseIf тип_шкафа = "прямой" Then
Component.Visible("Дно углового шкафа:1") = 0
Component.Visible("Стойка углового шкафа:1") = 0
Component.Visible("Стойка углового шкафа:2") = 0
Component.Visible("Дно прямого шкафа:1") = 1
Component.Visible("Стойка прямого шкафа:1") = 1
Component.Visible("Дверь прямого шкафа Л:1") = 1
Component.Visible("Дверь прямого шкафа П:1") = 1

End If

```

Призначення матеріалів:

```

If Комплектация = "Вся модель шкафа" Then
3  '[
    If Материал = "Дуб сонома" Then
iProperties.Material ("Дно углового шкафа:1") = "Дуб сонома"
iProperties.Material ("Задняя панель шкафа:1") = "Дуб сонома"
iProperties.Material ("Стойка углового шкафа:1") = "Дуб сонома"
iProperties.Material ("Стойка углового шкафа:2") = "Дуб сонома"
iProperties.Material ("Дно прямого шкафа:1") = "Дуб сонома"
iProperties.Material ("Стойка прямого шкафа:1") = "Дуб сонома"
iProperties.Material ("Дверь прямого шкафа Л:1") = "Дуб сонома"
iProperties.Material ("Дверь прямого шкафа П:1") = "Дуб сонома"

    ElseIf Материал = "Вермут" Then
iProperties.Material ("Дно углового шкафа:1") = "Вермут"
iProperties.Material ("Задняя панель шкафа:1") = "Вермут"
iProperties.Material ("Стойка углового шкафа:1") = "Вермут"
iProperties.Material ("Стойка углового шкафа:2") = "Вермут"
iProperties.Material ("Дно прямого шкафа:1") = "Вермут"
iProperties.Material ("Стойка прямого шкафа:1") = "Вермут"
iProperties.Material ("Дверь прямого шкафа Л:1") = "Вермут"
iProperties.Material ("Дверь прямого шкафа П:1") = "Вермут"

    ElseIf Материал = "Венге" Then
iProperties.Material ("Дно углового шкафа:1") = "Венге"
iProperties.Material ("Задняя панель шкафа:1") = "Венге"
iProperties.Material ("Стойка углового шкафа:1") = "Венге"
iProperties.Material ("Стойка углового шкафа:2") = "Венге"
iProperties.Material ("Дно прямого шкафа:1") = "Венге"
iProperties.Material ("Стойка прямого шкафа:1") = "Венге"
iProperties.Material ("Дверь прямого шкафа Л:1") = "Венге"
iProperties.Material ("Дверь прямого шкафа П:1") = "Венге"

    ElseIf Материал = "Нимфея" Then
iProperties.Material ("Дно углового шкафа:1") = "Нимфея"

```

Зміна типу ліжка:


```

If передня_панель_кровати = "волна" Then
Component.Visible("Передняя панель кровати волна:1") = 1
Component.Visible("Передняя панель кровати:1") = 0

ElseIf передня_панель_кровати = "обычная" Then
Component.Visible("Передняя панель кровати волна:1") = 0
Component.Visible("Передняя панель кровати:1") = 1
End If

```

Зміна типу бокової панелі бази:

```

If тип_боковой_панели_базы = "глухая" Then

Component.Visible("Боковая панель П:1")=1
Component.Visible("Боковая панель П с круглым отверстием:1")=0
Component.Visible("Боковая панель П с фигурным отверстием:1")=0

ElseIf тип_боковой_панели_базы = "круглое окно" Then
Component.Visible("Боковая панель П:1")=0
Component.Visible("Боковая панель П с круглым отверстием:1")=1
Component.Visible("Боковая панель П с фигурным отверстием:1")=0

ElseIf тип_боковой_панели_базы = "фигурное окно" Then
Component.Visible("Боковая панель П:1")=0
Component.Visible("Боковая панель П с круглым отверстием:1")=0
Component.Visible("Боковая панель П с фигурным отверстием:1")=1
End If

```

Керування загальними розмірами складального вузла залежно від віку дитини:

```

If возраст_ребенка = "до 3 лет" Then
длина_кровати = 1200
ширина_кровати = 550
высота_шкафа = 1200
высота_отраждения_кровати = 450

ElseIf возраст_ребенка = "от 3 до 6 лет" Then
длина_кровати = 1400
ширина_кровати = 600
высота_шкафа = 1200
высота_отраждения_кровати = 450

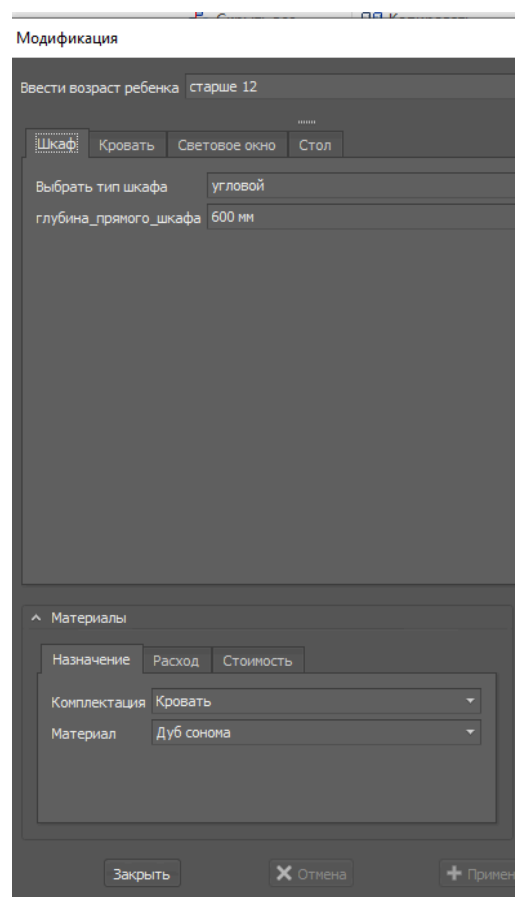
ElseIf возраст_ребенка = "от 6 до 12 лет" Then
длина_кровати = 1600
ширина_кровати = 800
высота_шкафа = 1400
высота_отраждения_кровати = 400

ElseIf возраст_ребенка = "старше 12" Then
длина_кровати = 1900
ширина_кровати = 900
высота_шкафа = 1500
высота_отраждения_кровати = 350

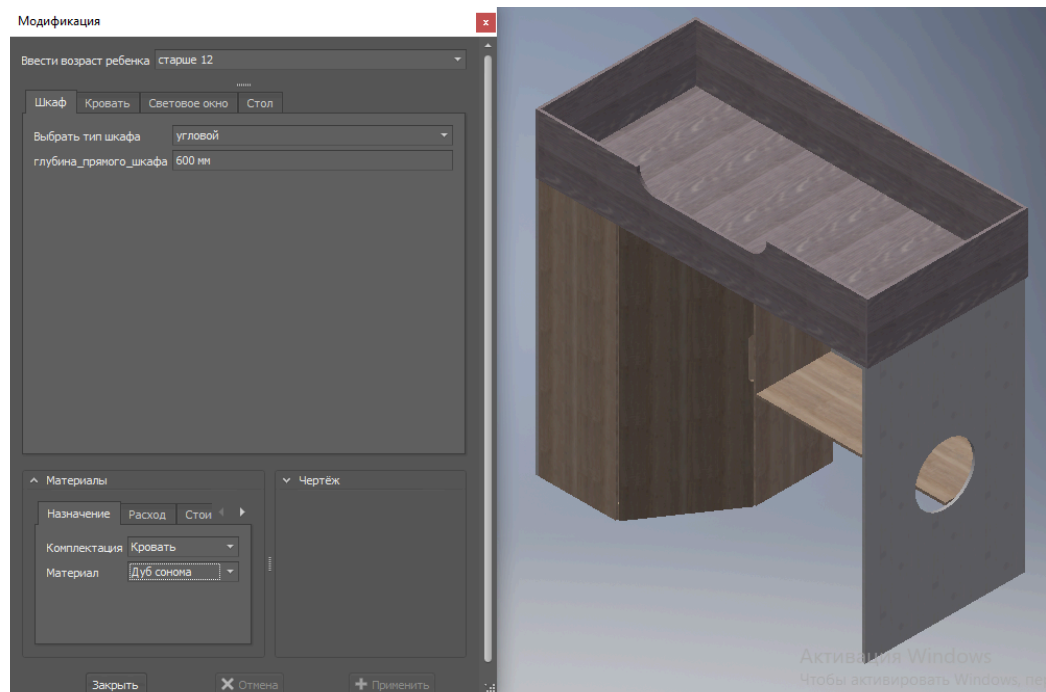
End If

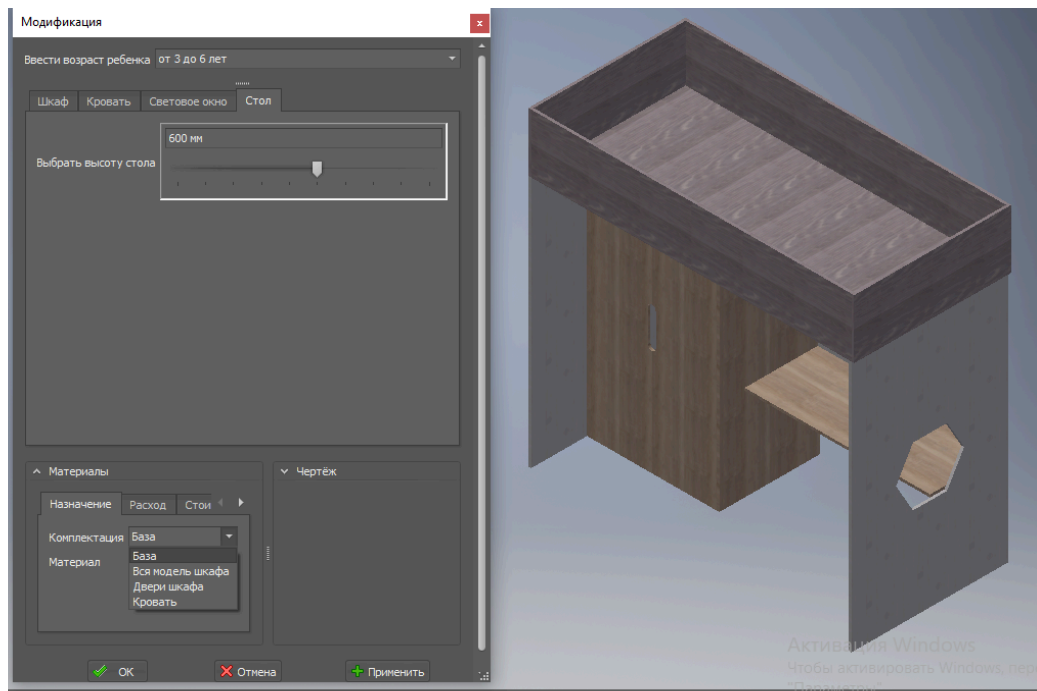
```

8. Створити програмний модуль для керування розмірами та конструкцією складального вузла



9. Результат:





Висновок: Під час виконання лабораторної роботи необхідно навчитися створювати програмний модуль для керування розмірами та конструкцією складного складального вузла з використанням можливостей iLogic.

Запитання до лабораторної роботи 6.

1. Які види залежностей використано при створенні складання?
2. Які складальні одиниці додаємо у файл складального вузла?
3. Як створити правило «Link», в якому реалізується зв'язок між імпортованими параметрами в основному файлі та параметрами складальних одиниць?
4. Які додаткові правила були створені для складального вузла?
5. Як створити програмний модуль для керування розмірами та конструкцією складального вузла?

Лабораторна робота № 7

Частина 1 «Підрахунок площі компонентів, використаних для виготовлення складального вузла з використанням можливостей ILogic»

Частина 2 «Підрахунок витрат використаних матеріалів для виготовлення складального вузла та їх загальної вартості з використанням можливостей ILogic»

Частина 3 «Робота з кресленнями з використанням можливостей ILogic»

Мета частини 1: з використанням можливостей iLogic реалізувати інтерактивний підрахунок площі компонентів складального вузла.

Теоретична частина:

Round() - Округлює аргумент до цілого числа або до заданого числа цифр після коми.

Синтаксис : Round (аргумент, при необхідності число знаків після коми)

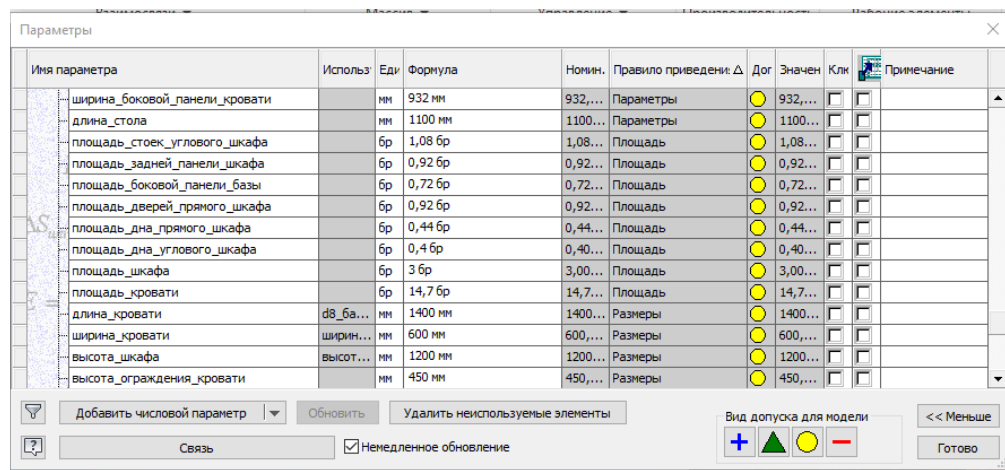
$\text{Round}(2.55689) = 3$

$\text{Round}(2.55689, 1) = 2.6$

$\text{Round}(2.55689, 3) = 2.557$

Практична частина:

1. Для підрахунку загальної площі компонентів створюємо необхідні числові параметри для кожної складальної одиниці.
- 2.



Имя параметра	Используй	Ед	Формула	Номин	Правило приведения	Дог	Значен	Кли	Примечание
ширина_боковой_панели_кровати		мм	932 мм	932...	Параметры	●	932,...		
длина_стола		мм	1100 мм	1100...	Параметры	●	1100...		
площадь_стоек_углового_шкафа		бр	1,08 бр	1,08...	Площадь	●	1,08...		
площадь_задней_панели_шкафа		бр	0,92 бр	0,92...	Площадь	●	0,92...		
площадь_боковой_панели_базы		бр	0,72 бр	0,72...	Площадь	●	0,72...		
площадь_дверей_прямого_шкафа		бр	0,92 бр	0,92...	Площадь	●	0,92...		
площадь_дна_прямого_шкафа		бр	0,44 бр	0,44...	Площадь	●	0,44...		
площадь_дна_углового_шкафа		бр	0,4 бр	0,40...	Площадь	●	0,40...		
площадь_шкафа		бр	3 бр	3,00...	Площадь	●	3,00...		
площадь_кровати		бр	14,7 бр	14,7...	Площадь	●	14,7...		
длина_кровати		мм	1400 мм	1400...	Размеры	●	1400...		
ширина_кровати		мм	600 мм	600...	Размеры	●	600...		
высота_шкафа		мм	1200 мм	1200...	Размеры	●	1200...		
высота_ограждения_кровати		мм	450 мм	450...	Размеры	●	450...		

3. Программный код правила для підрахунку площі:

```

Parameter.UpdateAfterChange = True

площадь_задней_панели_шкафа = Round( ширина_задней_панели_шкафа*высота_шкафа / 1000000, 2)
площадь_боковой_панели_базы = Round( ширина_кровати*высота_шкафа / 1000000, 2)
площадь_дверей_прямого_шкафа = Round( 2*ширина_двери_прямого_шкафа*высота_шкафа / 1000000, 2)
площадь_стоек_углового_шкафа = Round(2*ширина_стойки_углового_шкафа* высота_шкафа / 1000000, 2)
площадь_дна_прямого_шкафа = Round(ширина_дна_шкафа*глубина_дна_шкафа / 1000000, 2)
площадь_дна_углового_шкафа = Parameter("шкаф:1", "Площадь_углового_шкафа")

If тип_шкафа = "прямой" Then
площадь_шкафа = площадь_задней_панели_шкафа + площадь_боковой_панели_базы + площадь_дверей_прямого_шкафа + площадь_дна_прямого_шкафа

ElseIf тип_шкафа = "угловой" Then
площадь_шкафа = площадь_задней_панели_шкафа + площадь_боковой_панели_базы + площадь_стоек_углового_шкафа + площадь_дна_углового_шкафа

End If

площадь_кровати = площадь_дна_кровати + площадь_задней_панели_кровати + 2 * площадь_боковой_панели_кровати + площадь_передней_панели_кровати

```

4. Доповнюємо програмний модуль вкладкою «Площа», де буде показана площа матеріалів, використаних при виготовленні складального вузла.

Висновок частина1. Під час виконання лабораторної роботи необхідно навчитися реалізувати інтерактивний підрахунок витрат матеріалів для виготовлення складального вузла з використанням можливостей iLogic.

Запитання до частини 1.

1. Як проводити округлення аргументу до цілого числа?
2. Як проводимо підрахунок площі компоненту?
3. Як проводимо підрахунок загальної площі всіх компонентів?
4. Як створити та налаштувати вкладку з параметрами «Площа»?

Частина 2 «Підрахунок витрат використаних матеріалів для виготовлення складального вузла та їх загальної вартості з використанням можливостей ILogic»

Мета частини 2: з використанням можливостей iLogic реалізувати інтерактивний підрахунок витрат матеріалів для виготовлення складального вузла та їх загальної вартості.

Практична частина

1. Для підрахунку витрат матеріалів створюємо необхідні числові параметри для кожного варіанту матеріалу.

The screenshot shows the 'Parameters' window in ILogic. It contains a table with the following columns: 'Имя параметра' (Parameter Name), 'Используй' (Use), 'Едини' (Unit), 'Формула' (Formula), 'Номин.' (Nominal), 'Правило приведения: Δ' (Reduction Rule: Δ), 'Дог' (Tolerance), 'Значен' (Value), 'Клк' (Click), and 'Примечание' (Note). The table lists 12 parameters for different materials and their formulas. Below the table are buttons for 'Добавить числовой параметр' (Add numeric parameter), 'Обновить' (Update), 'Удалить неиспользуемые элементы' (Delete unused elements), and a 'Связь' (Link) section with a 'Немедленное обновление' (Immediate update) checkbox. There are also buttons for 'Вид допуска для модели' (Model tolerance view) and 'Готово' (Done).

Имя параметра	Используй	Едини	Формула	Номин.	Правило приведения: Δ	Дог	Значен	Клк	Примечание
расход_ДСП_дуб_сонома_шкаф		бр	4,09 бр	4,09...	Расход материалов	0,00...	4,09...		
расход_ДСП_вермут_шкаф		бр	0 бр	0,00...	Расход материалов	0,00...	0,00...		
расход_ДСП_венге_шкаф		бр	0 бр	0,00...	Расход материалов	0,00...	0,00...		
расход_ДСП_нимфея_шкаф		бр	0 бр	0,00...	Расход материалов	0,00...	0,00...		
расход_ДСП_дуб_сонома_кровать		бр	0 бр	0,00...	Расход материалов	0,00...	0,00...		
расход_ДСП_вермут_кровать		бр	0 бр	0,00...	Расход материалов	0,00...	0,00...		
расход_ДСП_венге_кровать		бр	14,7 бр	14,7...	Расход материалов	0,00...	14,7...		
расход_ДСП_нимфея_кровать		бр	0 бр	0,00...	Расход материалов	0,00...	0,00...		
расход_ДСП_дуб_сонома		бр	4,09 бр	4,09...	Расход материалов	0,00...	4,09...		
расход_ДСП_вермут		бр	0 бр	0,00...	Расход материалов	0,00...	0,00...		
расход_ДСП_нимфея		бр	0 бр	0,00...	Расход материалов	0,00...	0,00...		
расход_ДСП_венге		бр	14,7 бр	14,7...	Расход материалов	0,00...	14,7...		

2. Створюємо відповідне правило для підрахунку витрат кожного з матеріалів:

```
расход_ДСП_дуб_сонома = расход_ДСП_дуб_сонома_шкаф + расход_ДСП_дуб_сонома_кровать
расход_ДСП_вермут = расход_ДСП_вермут_шкаф + расход_ДСП_вермут_кровать
расход_ДСП_венге = расход_ДСП_венге_шкаф + расход_ДСП_венге_кровать
расход_ДСП_нимфея = расход_ДСП_нимфея_шкаф + расход_ДСП_нимфея_кровать
```

3. Доповнюємо правило, програмним кодом, який активує тільки використаний матеріал:

```

If Комплектация = "Вся модель шкафа" Then

    If iProperties.Material ("Задняя панель шкафа:1") = "Дуб сонома" Then
        расход_ДСП_дуб_сонома_шкаф = площадь_шкафа
        расход_ДСП_вермут_шкаф = 0
        расход_ДСП_венге_шкаф = 0
        расход_ДСП_нимфея_шкаф = 0

    ElseIf iProperties.Material ("Задняя панель шкафа:1") = "Вермут" Then
        расход_ДСП_вермут_шкаф = площадь_шкафа
        расход_ДСП_дуб_сонома_шкаф = 0
        расход_ДСП_венге_шкаф = 0
        расход_ДСП_нимфея_шкаф = 0

    ElseIf iProperties.Material ("Задняя панель шкафа:1") = "Венге" Then
        расход_ДСП_венге_шкаф = площадь_шкафа
        расход_ДСП_дуб_сонома_шкаф = 0
        расход_ДСП_вермут_шкаф = 0
        расход_ДСП_нимфея_шкаф = 0

    ElseIf iProperties.Material ("Задняя панель шкафа:1") = "Нимфея"
        расход_ДСП_нимфея_шкаф = площадь_шкафа
        расход_ДСП_дуб_сонома_шкаф = 0
        расход_ДСП_вермут_шкаф = 0
        расход_ДСП_венге_шкаф = 0
    End If

ElseIf Комплектация = "Кровать" Then

    If iProperties.Material ("Дно кровати:1") = "Дуб сонома" Then
        расход_ДСП_дуб_сонома_кровать = площадь_кровати
        расход_ДСП_вермут_кровать = 0
        расход_ДСП_венге_кровать = 0
        расход_ДСП_нимфея_кровать = 0

    ElseIf iProperties.Material ("Дно кровати:1") = "Вермут" Then

        расход_ДСП_дуб_сонома_кровать = 0
        расход_ДСП_вермут_кровать = площадь_кровати
        расход_ДСП_венге_кровать = 0
        расход_ДСП_нимфея_кровать = 0

    ElseIf iProperties.Material ("Дно кровати:1") = "Венге" Then

        расход_ДСП_дуб_сонома_кровать = 0
        расход_ДСП_вермут_кровать = 0
        расход_ДСП_венге_кровать = площадь_кровати
        расход_ДСП_нимфея_кровать = 0

    ElseIf iProperties.Material ("Дно кровати:1") = "Нимфея" Then

        расход_ДСП_дуб_сонома_кровать = 0
        расход_ДСП_вермут_кровать = 0
        расход_ДСП_венге_кровать = 0
        расход_ДСП_нимфея_кровать = площадь_кровати
    End If

```

```

ElseIf Комплектация = "База" Then

    If iProperties.Material ("Задняя стенка базы:1") = "Дуб сонома" Then
        расход_ДСП_дуб_сонома_кровать = площадь_кровати
        расход_ДСП_вермут_кровать = 0
        расход_ДСП_венге_кровать = 0
        расход_ДСП_нимфея_кровать = 0

    ElseIf iProperties.Material ("Задняя стенка базы:1") = "Вермут" Then

        расход_ДСП_дуб_сонома_кровать = 0
        расход_ДСП_вермут_кровать = площадь_кровати
        расход_ДСП_венге_кровать = 0
        расход_ДСП_нимфея_кровать = 0

    ElseIf iProperties.Material ("Задняя стенка базы:1") = "Венге" Then

        расход_ДСП_дуб_сонома_кровать = 0
        расход_ДСП_вермут_кровать = 0
        расход_ДСП_венге_кровать = площадь_кровати
        расход_ДСП_нимфея_кровать = 0

    ElseIf iProperties.Material ("Задняя стенка базы:1") = "Нимфея" Then

        расход_ДСП_дуб_сонома_кровать = 0
        расход_ДСП_вермут_кровать = 0
        расход_ДСП_венге_кровать = 0
        расход_ДСП_нимфея_кровать = площадь_кровати
    End If

End If

```

4. Доповнюємо програмний модуль вкладкою «Площа», де буде показана площа матеріалів, використаних при виготовленні складального вузла.

Назначение	Расход	Стоимость
расход_ДСП_дуб_сонома	4,09	
расход_ДСП_вермут	0	
расход_ДСП_венге	14,7	
расход_ДСП_нимфея	0	

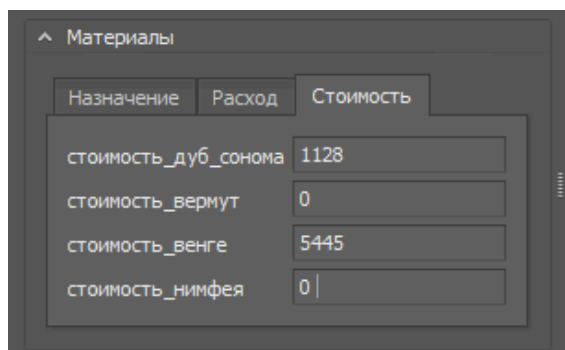
5. Для підрахунку вартості використаних матеріалів створюємо необхідні числові параметри для кожного варіанту матеріалу.

Имя параметра	Использует	Ед.	Формула	Номинал	Правило приведения	Доп.	Значение	Кл.	Примечание
Параметры модели									
Пользовательские									
стоимость_дуб_сонома		бр	1128 бр	1128...	Стоимость	●	1128...		
стоимость_вермут		бр	0 бр	0,00...	Стоимость	●	0,00...		
стоимость_венге		бр	5445 бр	5445...	Стоимость	●	5445...		
стоимость_нимфея		бр	0 бр	0,00...	Стоимость	●	0,00...		

6. Створюємо відповідне правило для підрахунку вартості кожного з матеріалів:

```
стоимость_дуб_сонома = Round (расход_ДСП_дуб_сонома, 0) * цена_дуб_сонома  
стоимость_вермут = Round (расход_ДСП_вермут, 0) * цена_вермут  
стоимость_венге = Round (расход_ДСП_венге, 0) * цена_венге  
стоимость_нимфея = Round (расход_ДСП_нимфея, 0) * цена_нимфея
```

7. Доповнюємо програмний модуль вкладкою «Вартість», де буде показана загальна сума вартості матеріалів, використаних при виготовленні складального вузла.



The screenshot shows a window titled "Материалы" (Materials) with a table containing the following data:

Назначение	Расход	Стоимость
стоимость_дуб_сонома	1128	
стоимость_вермут	0	
стоимость_венге	5445	
стоимость_нимфея	0	

Висновок частини 2. Під час виконання лабораторної роботи необхідно навчитися реалізувати інтерактивний підрахунок витрат матеріалів для виготовлення складального вузла та їх загальної вартості з використанням можливостей iLogic.

Запитання до частини 2.

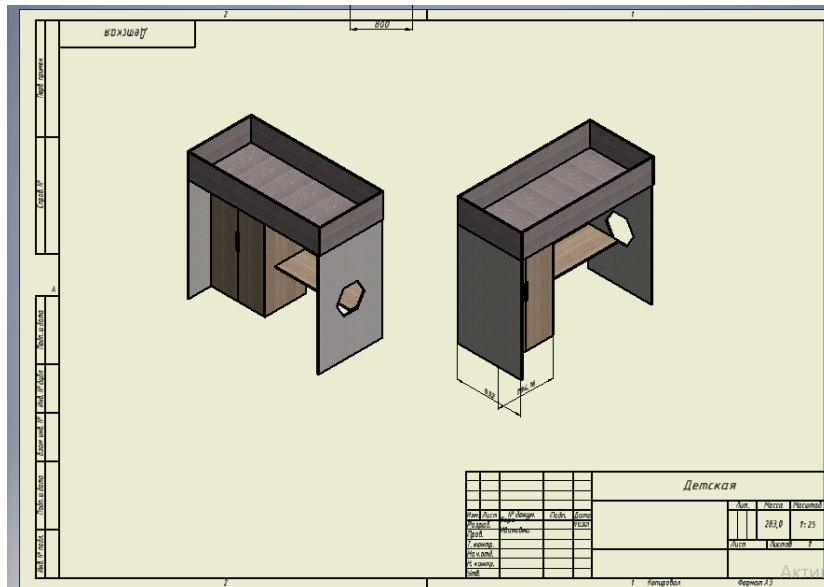
1. Як створити числові параметри для кожного варіанту матеріалу?
2. Як створюється правило для підрахунку витрат кожного з матеріалів?
3. Як створити правило активизації тільки використаного матеріалу?
4. Як доповнити програмний модуль вкладкою «Площа»?
5. Як створити правило для підрахунку вартості кожного з матеріалів

Частина 3 «Робота з кресленнями з використанням можливостей ILogic»

Мета частини 3: з використанням можливостей iLogic реалізувати інтерактивну роботу з кресленнями.

Практична частина:

1. Створюємо двовимірний кресленик з тривимірної моделі за допомогою команди «Базовий».



2. Створюємо правило для відкриття файлу кресленнику:

Try

`oDWG = ThisDoc.FileName(False) & ".dwg"`

`ThisDoc.Launch(oDWG)`

Catch

`Goto IDW`

End Try

Return

IDW :

Try

`oDWG = ThisDoc.FileName(False) & ".idw"`

`ThisDoc.Launch(oDWG)`

Catch

`MessageBox.Show("У даної моделі немає креслення", "Відкрити креслення")`

End Try

3. Створюємо правило для повернення до файлу моделі:

Try

```

oDWG = ThisDoc.FileName(False) & ".iam"
ThisDoc.Launch(oDWG)
Catch
Goto IDW
End Try

```

```
Return
```

```
IDW :
Try
```

```
Catch
MessageBox.Show(("У даного креслення немає моделі", "Відкрити модель")
End Try
```

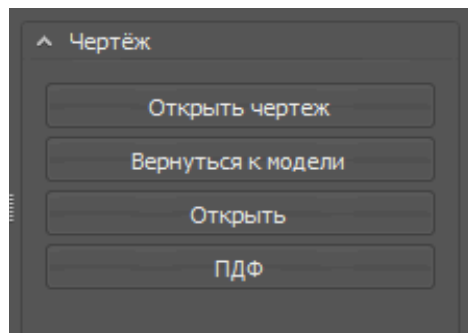
4. Створюємо правило для повернення до файлу моделі:

```

i = MessageBox.Show("Відкрити креслення?", "@ClintBrown3D",
MessageBoxButtons.YesNo)
If i = vbYes Then : launchviewer = 1 : Else : launchviewer = 0 : End If
If launchviewer = 1 Then
Try
ThisDoc.Launch(ThisDoc.PathAndFileName(False) + ".dwg")
Catch
Try
ThisDoc.Launch(ThisDoc.PathAndFileName(False) + ".idw")
Catch
MessageBox.Show("Для даної моделі креслення не знайдено", "Відкрити
креслення")
End Try
End Try
End If

```

5. Доповнюємо програмний модуль вкладкою «Креслення», де будуть відображені кнопки «Відкрити креслення», «Повернутися до моделі» та інші.



Висновок: Під час виконання лабораторної роботи необхідно навчитися реалізувати інтерактивну роботу з кресленниками з використанням можливостей iLogic.

Запитання до частини 3.

1. Як створити двовимірний кресленик з тривимірної моделі?
2. Як створити правило для відкриття файлу кресленнику?
3. Як створити правило для повернення до файлу моделі?
4. Як доповнити програмний модуль вкладкою «Креслення»?

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Функціональність Autodesk Inventor – Режим доступу:
<https://www.pointcad.ua/product/autodesk-inventor/podrobnoe-opisanie-autodesk-inventor>
2. Зінов'єв Д.В. - Основи проектування в Autodesk Inventor 2024
3. Правила та форми в iLogic - Режим доступу:
<https://knowledge.autodesk.com/ru/support/inventor/learn-explore/caas/CloudHelp/cloudhelp/2024/Inventor-Help/files/GUID-EF53484C-D750-41F8-9AB1-032B73BB071F-htm.html>
4. Створення параметричної деталі в Autodesk Inventor: порожнечі та параметризація - Режим доступу:
<https://infars.ua/blog/sozдание-parametricheskoy-detali-pustoty-i-parametrizatsiya/>
5. Autodesk Inventor - iLogic Tutorial : Furniture Model Configurator Режим доступу: https://www.youtube.com/watch?v=aP_TC4lzgjo&t=93s
6. Autodesk Inventor Tutorial Режим доступу:
<https://www.youtube.com/watch?v=Bydrjcmz64Q>
7. Inventor iLogic Best Practices and Fundamentals for Success. URL:
<https://www.autodesk.com/autodesk-university/article/iLogic-Best-Practices-and-Fundamentals-for-Success>.
8. Munford P. Autodesk Inventor iLogic forms to communicate design intent. URL:
<https://blogs.autodesk.com/inventor/five-autodesk-inventor-illogic-productivity-hacks-forms/>
9. About Rules and Forms in iLogic. URL:
<https://help.autodesk.com/view/INVNTOR/2023/ENU/?guid=GUID-EF53484C-D750-41F8-9AB1-032B73BB071F>
10. To Work with Forms. URL:
<https://help.autodesk.com/view/INVNTOR/2023/ENU/?guid=GUID-223E9090-5118-406D-9D80-EC247997A7F4>