

Національний лісотехнічний університет України

(повне найменування вищого навчального закладу)

Навчально-науковий інститут деревообробних та
комп'ютерних технологій і дизайну

(повне найменування інституту, назва факультету (відділення))

Кафедра інформаційних технологій

(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до дипломної роботи

другий (магістерський)

(освітньо-кваліфікаційний рівень)

на тему: Розроблення САПР меблів для вітальні з використанням
AutoCAD.NET API.

Виконав: студент б курсу групи КН-61м
спеціальності

122 “Комп’ютерні науки”

(шифр і назва напрямку підготовки, спеціальності)

Пацута В.В.

(прізвище та ініціали)

Керівник Сторожук О.Л.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Львів – 2021

Національний лісотехнічний університет України

(повне найменування вищого навчального закладу)

ННІ деревообробних та комп'ютерних технологій і дизайну

Кафедра інформаційних технологій

Рівень вищої освіти другий (магістерський)

Спеціальність 122 "Комп'ютерні науки"

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Крошній І.М.

" " " 2021 року

**З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Пацуті Владиславі Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення САПР меблів для вітальні з використанням AutoCAD.NET API.»

керівник роботи *Сторожук Олександр Леонідович, к.т.н., доцент*,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "31"12 2020 р. № C-593

2. Термін подання студентом роботи 10.12.2021р.

3. Вихідні дані до роботи: Постановка задачі та формалізація. Основні параметри та вимоги до проектування модульних меблів (вітальні).

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити): Вступ. Розділ 1. Стан проблемної області. Розділ 2. Інформаційне забезпечення. Розділ 3. Математичне забезпечення. Розділ 4. Програмне забезпечення. Розділ 5. Розроблення стартап-проекту. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Підготовка матеріалу до доповіді.

6. Дата видачі завдання 18 грудня 2020 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури за вказаною тематикою.</i>	<i>18.12.2020р. – 10.01.2021р.</i>	
2.	<i>Аналіз поставленої задачі та написання вступу.</i>	<i>11.01.2021р. – 31.01.2021р.</i>	
3.	<i>Вибір та обґрунтування методів і технологій для створення програмного забезпечення у вигляді додатка до AutoCAD.</i>	<i>01.02.2021р. – 28.02.2021р.</i>	
4.	<i>Пошук аналогів та виявлення переваг майбутнього додатка. Написання першого розділу.</i>	<i>01.03.2021р. – 31.03.2021р.</i>	
5.	<i>Розробка користувачького інтерфейсу. Написання другого розділу.</i>	<i>01.04.2021р. – 30.04.2021р.</i>	
6.	<i>Програмна реалізація системи. Написання третього розділу.</i>	<i>01.05.2020р. – 31.08.2021р.</i>	
7.	<i>Написання четвертого розділу.</i>	<i>01.09.2021р. – 30.09.2021р.</i>	
8.	<i>Аналіз перспектив старту та план його подальшого розвитку. Написання п'ятого розділу.</i>	<i>01.10.2021р. – 31.10.2021р.</i>	
9.	<i>Аналіз отриманих результатів та написання висновків.</i>	<i>01.11.2021р. – 09.12.2021р.</i>	
10.	<i>Здача пояснювальної записки на перевірку керівнику, виправлення помилок та здача роботи рецензенту.</i>	<i>10.12.2021 р.</i>	

Студент

_____ (підпис)

Пацута В.В.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Сторожук О.Л.

_____ (прізвище та ініціали)

РЕФЕРАТ

Дипломна робота містить 60 сторінок пояснювальної записки, 5 рисунків, 3 таблиці, 1 додаток, 23 джерела.

Завданням даної дипломної роботи є розробка програмного забезпечення у вигляді додатка до AutoCAD, що дозволяє проектувати модульні меблі на прикладі вітальні.

У роботі було використано AutoCAD .NET API який дозволяє використовувати широкий ряд класів, структур, методів і подій, що забезпечують доступ до об'єктів файлу креслення в додатку AutoCAD. Передбачено виведення усіх результатів геометричних розрахунків вітальні д у файл для подальшого створення технічної документації.

Програмне забезпечення реалізовано за допомогою мови програмування C# в середовищі Microsoft Visual Studio.

Ключові слова: AutoCAD, C#, AutoCAD .NET API, Visual Studio, IDE. меблева промисловість, вітальня.

ABSTRACT

Diploma work contains 60 pages of explanatory note, 5 figures, 3 tables, 1 appendix, 23 sources.

The task of this diploma work is to develop software in the form of an application to AutoCAD, which allows you to design modular furniture on the example of the living room.

The AutoCAD .NET API was used, which allows the use of a wide range of classes, structures, methods and events that provide access to drawing file objects in the AutoCAD application. It is planned to output all the results of geometric calculations of the living room to a file for further creation of technical documentation.

The software is implemented using the C # programming language in Microsoft Visual Studio.

Keywords: AutoCAD, C#, AutoCAD .NET API, Visual Studio, IDE. furniture industry, living room.

ТЕХНІЧНЕ ЗАВДАННЯ

Необхідно розробити програмне забезпечення проектування модульних меблів (вітальні) на базі AutoCAD. Програмне забезпечення має бути у вигляді додатка для AutoCAD, написане мовою програмування C# в середовищі Microsoft Visual Studio із використанням AutoCAD .NET API який дозволяє використовувати широкий ряд класів, структур, методів і подій, що забезпечують доступ до об'єктів файлу креслення в додатку AutoCAD.

Передбачити виведення усіх результатів геометричних розрахунків у файл для подальшого створення необхідної документації.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ	11
1.1. Аналіз сфери меблевого виробництва	11
1.2. Характеристика проектування	12
1.3. Аналіз розробок САПР в області меблевого проектування	13
Висновки до розділу	17
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	18
2.1. Аналіз системи проектної документації	18
2.2 Мова програмування C#	19
2.3 AutoCAD .NET API	21
Висновки до розділу	23
РОЗДІЛ 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	24
3.1. Моделі та їхні параметри в САПР	24
3.2. Геометричне моделювання та машинна графіка	25
Висновки до розділу	31
РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	32
4.1. Система AutoCAD	32
4.2. Реалізація програмного забезпечення	34
Висновки до розділу	39
РОЗДІЛ 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ	40
5.1 Опис ідеї проекту	40
5.2 Маркетингова програма стартап-проекту	41
5.3 Аналіз технологічних можливостей реалізації ідей проекту	41
Висновки до розділу	42
ВИСНОВКИ	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	44
ДОДАТОК А	46

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

C# – об'єктно орієнтована мова програмування;

AutoCAD – автоматизована система для двовимірного та тривимірного проектування;

AutoCAD .NET API – керований інтерфейс програмування прикладних програм на базі AutoCAD;

САПР – система автоматизованого проектування і розрахунку;

UI (User Interface) – інтерфейс користувача;

UX (User experience) – дизайн інтерфейсу користувача;

IDE (Integrated Development Environment) – інтегроване середовище розробки.

ВСТУП

Меблева промисловість знаходиться в процесі постійного розвитку і вдосконалення. Меблі здавна вважають предметом першої необхідності, а як відомо, найбільш надійним є заробіток саме на предметах першої необхідності. В зв'язку з динамічним розвитком меблевого ринку відбувається постійне зростання попиту, меблеві підприємства розширюють об'єми і асортимент випускаючої продукції, а також дають можливість придбати меблі, виготовлені на замовлення.

Лідером серед систем автоматизованого проектування в світі є AutoCAD, це зумовлено досить розвиненими засобами розробки та адаптації під нього. Усього у світі налічується понад 10000 комерційних продуктів заснованих на AutoCAD.

Програма AutoCAD включає в себе повний набір засобів які забезпечують комплексне тривимірне моделювання, у тому числі роботу з довільними формами, створення і редагування 3D-моделей та ефективні засоби випуску робочої документації.

Технологія AutoCAD .NET API допомагає створювати, ефективні і компактні програми, використовуючи переваги відкритої архітектури AutoCAD. Що дає можливість пристосовувати AutoCAD до вирішення нестандартних завдань проектування.

Актуальність проблеми. Меблі оточують людину всюди, як вдома, так і на роботі під час виконання своїх обов'язків, навіть під час купування всього необхідного у супермаркеті. Створити комфортну атмосферу у приміщенні не застосовуючи меблеві вироби на даний момент неможливо. На даний час, актуальним є питання швидкого проектування і виготовлення якісних меблів.

Об'єктом дослідження – засоби створення САПР на базі AutoCAD.

Предмет дослідження – процес реалізації проектування модульних меблів за допомогою мови C# та технології AutoCAD .NET API в AutoCAD.

Мета роботи – розробити програмне забезпечення у вигляді додатка до AutoCAD, що дозволяє проектувати модульні меблі (на прикладі вітальні).

Новизна роботи полягає у застосуванні об’єктно-орієнтованої мови C# та технології AutoCAD .NET API для розробки додатка до AutoCAD, що дозволяє проектувати модульні меблі (на прикладі вітальні).

Практичне значення полягає в інтуїтивно зрозумілому інтерфейсі, легкій доступності для використання початківцями та інформативності.

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1. Аналіз сфери меблевого виробництва

Меблева промисловість лідирує серед найпотужніших галузей у сфері деревообробного виду діяльності. Ця галузь включає в собі половину від усіх робочих місць з усієї промисловості. Для оформлення житлових та нежитлових будівель меблями використовуються потужності меблевої промисловості. Предмет виробництва даної промисловості – меблі поділяються на різні групи [22]:

- меблі для побуту, які використовують споживачі у своєму повсякденному житті;
- меблі для офісного використання, для яких споживачами є державні, приватні підприємства, а також держслужбовці та високопосадовці;
- корпусні меблі, наприклад шкільні меблі для забезпечення навчальних закладів необхідним меблевим устаткуванням;
- м'які меблі, які включають в себе крісла, дивани а також кутки;
- меблі які виготовляються з натурального дерева, та мають притаманний класиці дизайн для функціонування у офісах та житлових будівлях;
- скляні меблі.

З метою економії житлової площі у приміщенні було розроблено і поставлено на виробництво корпусні меблі для яких використовуються модульні конструкції. Проблему заповнення порожнього простору прекрасно вирішило рішення використовувати корпусні меблі з раціональним оформленням інтер'єру. З цим підходом всі найнеобхідніші у побуті речі завжди на своєму місці та знаходяться у легкій доступності. Основні критерії модульних меблів є безсумнівно легкий інтер'єр, а також функціональна зручність. Адже для цих меблів допустимі безліч варіантів як висоти конструкцій, так і їх глибини вкладення і різні способи відкривання дверей

шаф та стулок. У квартирах та кімнатах з маленькою площею житлового приміщення є проблема що звичані побутові меблі можуть створити враження хаосу, тому модульні меблі це незамінна альтернатива яка вирішує це питання

У таких випадках використовують комбінування житлового приміщення яке розділене на певні зони, наприклад коли кімната служить як для сну, так і кабінету або вітальня разом із модульною кухнею, тобто планування меблевого забезпечення на рівні багатофункціонального гібридного модулю. Будь порожній куточок квартири або кімнати може бути заповнений оригінально виконаним елементом меблів, шафою або тумбою, при цьому можна все замовити на свій розсуд під даний тип інтер'єру. Для оригінального заповнення житлового приміщення може бути використана абсолютно будь-яка незадіяна площа житлового приміщення, для цього є можливість замовити під потрібний тип інтер'єру на свій розсуд усе необхідне. Окрім цього є можливість найняти дизайнера та реалізувати нові меблі на свій смак [23].

1.2. Характеристика проектування

Естетичні вимоги є у глибокому взаємозв'язку разом з функціональним призначенням меблевих виробів. Адже саме меблі як ніщо інше оточують споживача у повсякденному житті, тому вони повинні бути досконалі у своїй естетиці.

Меблеві вироби мають перебувати у гармонії з іншими частинами житлового приміщення, так як знаходяться у спільній експлуатації. Всі предмети меблевих виробів мають відповідати сучасному інтер'єру, не порушуючи гармонійність. Окрім цього меблі повинні не порушувати існуючі санітарні та гігієнічні вимоги. Якщо брати до уваги якість а також комфортність меблевих виробів, їх вартість (собівартість) по можливості має бути не захмарною. Виконання цієї вимоги залежить від багатьох чинників,

деякі з них це організація виробництва, степінь ефективності підприємства, технологічні конструкції, а також матеріаломісткість. Матеріаломісткість – це безсумнівно частина одного з найважливіших показників, так як остаточна вартість меблевих виробів для кінцевого споживача на 50–80 відсотків визначається собівартістю матеріалів затрачених на виробництво. Тому дуже важливим фактором являється завдання оптимального використання початкових матеріалів, а також оптимальний розкрій пиломатеріалів.

Меблевий виріб зобов'язаний бути технологічним в умовах існуючого виробництва. Тобто проектувальний виріб повинен відповідати мінімальним трудовим, а також матеріальним витратам, виробництво має легко його освоїти без необхідності корінної перебудови технології і додаткового придбання устаткування або оснащення. Також під час виробництва цього виробу не потрібно закуповувати матеріали які знаходяться у дефіциті або займатись небезпечним прийомом робіт. Під час проектування є необхідним передбачення і забезпечення взаємозамінності матеріалів та складних одиниць які знаходяться в межах мінімальної точності для забезпечення виготовлення даного виробу. Потрібно добитись мінімальної кількості з'єднань, які належать до високих ступенів точності обробки. Також потрібно зазначити що при виробництві необхідно використовувати стандартні та уніфіковані елементи, уникати оригінальності під час кріплення деталей, а також складних з'єднань. Під час проектування нового виду меблів серед необхідного потрібно використовувати вже освоєні виробництвом вузли та деталі. На стадії конструювання потрібно враховувати як естетичні так і функціональні вимоги для кінцевого виробу.

1.3. Аналіз розробок САПР в області меблевого проектування

Для створення тривимірної графіки у підсистемах використовують сучасні комп'ютери, які це дозволяють. Беручи за увагу специфіку цієї представленої області, стандартне використання тривимірних редакторів не

являється достатнім, так як складність та вимоги до ресурсів обробної машини є великими впливовими факторами. Тому повстає питання необхідності певної підсистеми для тривимірної візуалізації, яка була б для роботи дизайнера зрозуміла і проста, але і з іншого боку мала можливість точно і повно відповідати специфіці поставленого завдання у проектуванні меблевих виробів.

Якщо врахувати величезну кількість підприємств представленої області, необхідність для розробки систем що дозволяють автоматизоване проектування і максимально ефективно використання усіх ресурсів провідної комп'ютерної техніки різко зростає. Саме в області систем автоматизованого проектування меблевих виробів існує величезна кількість розробок як створених на території нашої країни, так і за її межами. Серед найбільш популярних можна виділити наступні системи [5, 6]:

- **САПР "KitchenDraw"**

KitchenDraw - програма для створення 3D дизайну кухні. Це програмне забезпечення французької розробки, кінцевими споживачами якої є професійні дизайнери, продавці кухонних меблевих виробів, а також архітектори, виробники меблів на корпусній основі, будівельники. Особливістю даного програмного забезпечення є те, що плата стягується не за саму програму, а лише за час який було затрачено на роботу в ній, іншими словами, програма здається в оренду на певний необхідний час. Під час будь-якого етапу роботи є можливість відтворити приміщення яке знаходиться у проектуванні у тривимірному представленні, а також можна проаналізувати у перспективі, в розрізі а також зробити анімаційне відео. Абсолютно всі складові проекту існують одночасно (кошторис, тривимірна перспектива, піднесення і план). Якщо зміни були внесені хоча б в одному із них, вона буде відображена в інших автоматично.

Представлена програма легка в освоєнні для недосвідчених споживачів, за умови ознайомлення технічної документації даного програмного

забезпечення, а також комфортна для роботи професійних дизайнерів, а також архітекторів.

Давайте розглянемо можливості цієї програми:

- програмне забезпечення повністю перекладене багатьма мовами;
- є можливість обліку всіх продавців та продажів які пройшли реєстрацію;
- можна обмежити доступ до певного функціонування програми;
- версії для мережевого та локального використання;
- декілька або одне приміщення може бути одночасно в роботі, з можливістю встановлення всього необхідного в них;
- можливість встановлення "drag / drop" виробів які можна знайти в перспективі, плані або ж в каталозі;
- встановлення стін з точно виміряними кутами.

- **САПР "Базис-Мебельщик"**

Ця програма є основним з модулів БАЗИС системи. Основним призначенням його є проектування корпусних меблевих виробів незалежно від складності, а також автоматизоване отримання комплекту певної специфікації та креслень в повному обсязі. Застосовуючи цей модуль є можливість у 10 або 15 разів скоротити підготовку виробництва та проектування якщо порівнювати цей самий обсяг з ручною роботою, а також скоротити так званий "людський фактор" і кількість можливо допущених суб'єктивних помилок.

Великим плюсом є простий і легко зрозумілий модульний інтерфейс. Представлення моделі проектного виробу виконується у площинах в трьох видах. Спосіб представлений вище являється легко зрозумілим та наочним для всіх конструкторів. Якщо говорити про процес проектування, він являється створенням скандально просторового креслення. Виконання формулювання креслень на базі тривимірної проектованої моделі діє автоматично.

Щодо можливостей даного модуля:

- графічний редактор професійного рівня;
- набір певних команд дозволяє створити високоякісні панелі будь-якої форми;
- Встановлення об'єктів з встановленими відступами, зазорами, а також прив'язкою вже створених об'єктів яка відбувається автоматично;
- можливість створення власних бібліотек;
- можливість візуалізації проектованої структури виробу із швидким пошуком, переглядом і навіть редагування довільного елемента;
- формування певних специфікацій на автоматичному рівні із конкретними умовами налаштування виробництва алгоритму;
- можливість отримання практично реалістичної візуалізації виробу з типом і кольором різних світлових джерел, з урахуванням прозорості, розташування та дзеркальності;
- можливість групових змін для параметрів а також властивостей деталей;

- **САПР "bCAD Мебель"**

Даний комплекс програм використовується з метою автоматизації проектування меблів на базі корпусної структури. Представлений комплекс вирішує всі завдання від дизайну та конструювання, до технологічної підготовки виробництва. З метою оформлення дизайну для офісних інтер'єрів і заповнених меблями квартир цей комплекс також успішно використовується серед широких мас спеціалістів даної предметної області.

"bCAD Мебель" прекрасно підходить для проектування стелажів, барних стійок, офісних меблів, шаф-купе. Також він має можливість отримати карту розкрою а також робочі креслення в автоматичному режимі. Цей пакет працює за принципом проектування об'ємної моделі виробу, модифікування вже створених проектів, збирання моделі з окремих частин (панелей). Також включає в себе все необхідне, таке як облік всіх необхідних матеріалів, робочі креслення, повний витратний кошторис, а також реалістично привабливу візуалізацію моделі.

- **САПР "Woody"**

Представлена програма виконує моделювання меблів на тривимірному рівні, налає повний зазорний контроль для деталей в моделі виробу з наданням габаритів та певного нахилу виробів з будь-яким кутом.

Ця програма наділена такими властивостями та можливостями:

- дуже зручно та легко використовувати;
- проекти дуже легко піддаються тривимірній візуалізації;
- існує інтерактивно добре організована база даних потрібних фурнітурних матеріалів [1, 2].

Висновки до розділу

У даному розділі проаналізовано виробничий процес меблевої промисловості, а також актуальність проблеми розробки САПР в меблевій галузі. На основі цієї інформації було виявлено максимально економічний та продуктивний процес на всіх етапах виробництва меблевого виробу. Проаналізовано найвідоміші програми для розробки САПР з метою визначення оптимального варіанта для використання як досвідченому спеціалісту, так і початківцю у даній галузі.

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Аналіз системи проектної документації

Програмне забезпечення SPDS GraphiCS функціонально контролює, чи відповідає документ стандартам, які використовуються компанією, і чи відповідає він специфікаціям Системи документів архітектурного проектування (SPDS). Оскільки унікальна функція GraphiCS SPDS може створювати керовані користувачські об'єкти з автоматичним пошуком документів форми, що дозволяє автоматизувати проектну документацію. По суті, SPDS GraphiCS не є автоматизованою системою проектування. Це доповнення до графічної платформи, розробленої Autodesk. SPDS GraphiCS – це програма AutoCAD (ADT), яку можна запускати на її платформі незалежно від версії.

Стандартна поставка продуктів AutoCAD не включає українські національні стандарти проектування. Для ефективнішої роботи необхідно встановлювати (або розробляти) спеціалізовані програми. Зазвичай при розробці та публікації проектних документів організації створюють власну бібліотеку блоків (загальних елементів), основним недоліком якої є складність створення та редагування. Відповідно до спеціальних досліджень і висновків дизайнерського агентства CSoft Development було прийнято рішення про розробку програмного продукту, який працює на платформі AutoCAD і відповідає вимогам SPDS (Project File System) України. Програмне забезпечення GraphiCS SPDS було створено з використанням об'єктно-орієнтованої технології Autodesk ObjectARX, яка дозволяє керувати розумними та простими в управлінні об'єктами [14].

Під час свого розвитку (з 2000 року) SPDS GraphiCS доповнила всі необхідні інструменти для підготовки проектної документації відповідно до специфікацій системи проектної документації. Починаючи з версії 3, GraphiCS розробила деякі інструменти для автоматичного генерування

табличних форм (специфікацій, декларацій тощо) та створення керованих, зручних для користувача об'єктів. Завдання адаптації атрибутів графічних елементів до даних, зазначених у звіті проектної документації, є трудомістким і трудомістким. Тому вибір напряму функціонального розвитку є дуже розумним, і це ще один крок до автоматизації роботи дизайнера. Інструмент, який використовується для створення спеціальної бібліотеки (майстра об'єктів) поточної комерційної версії SPDS GraphiCS, дозволяє реалізувати креативність будь-якого користувача щодо створення власної бази даних об'єктів і контролювати їх поведінку на кресленні.

Переваги SPDS GraphiCS:

- простий та інтуїтивно зрозумілий інтерфейс;
- сертифікат відповідності в наявності;
- використання бази даних параметрів для швидкого отримання графіки та зображень;
- інструменти можна використовувати для доповнення бази параметрів керованих об'єктів;
- можливість автоматично вказувати елементи креслення за допомогою загальних тегів, атрибутів блоку AutoCAD, об'єктів SPDS та параметричних елементів бази даних;
- випущений стандарт документації, в єдиному вигляді [15].

2.2 Мова програмування C#

Основною особливістю мови C# є те, що вона зосереджена на платформі Microsoft .NET – природний спосіб доступу до всіх функцій платформи .NET. Очевидно, це рішення вважалось більш-менш вимушеним, оскільки спочатку платформа .NET забезпечувала більше можливостей, ніж будь-яка мова програмування, яка існувала на той час.

Крім того, розробники C# сподіваються якомога менше приховувати від користувачів деякі незначні технічні деталі, зокрема

пакування/розпакування типів, ініціалізацію змінних та операції зі збирання сміття. Це дозволяє програмістам C# краще зосередитися на змісті завдання. Щоб вирішити цю проблему, дизайнери C# намагалися вчитися на досвіді та уроках Visual Basic. Він був досить успішним у приховуванні деталей реалізації, але цього недостатньо для ефективного написання 3D промислових систем: розробники C# стверджують, що ця мова має переваги як C++, так і Visual Simplicity Basic.

Таким чином, C# – це нова мова програмування, яка зосереджена на розробці платформи .NET, придатної для швидкого створення прототипів додатків і розробки великомасштабних додатків.

Синтаксис і структура багатьох існуючих мов програмування дуже заплутані, а семантика неочевидна – просто запам'ятайте перевантажені дужки в C++, використовуйте ключові слова як ідентифікатори в P/i або розрізняйте дескриптори типів і оперативні проблеми. Алгол 68. Всі ці особливості мови ускладнюють компілятор і є джерелом серйозних помилок при створенні програм. Іншою крайністю цієї проблеми є мова Pascal, яка вирішила пожертвувати навіть очевидними можливостями для спрощення для програмістів.

C# зосереджено: найбільш дратівлива і неоднозначна функція C++ вилучена зі стандарту мови, але в той же час мова зберігає потужні можливості виразу таких мов, як C++, Java або VB.

Перерахуємо певні особливості мови C++, які не будуть підтримуватись C#:

- за замовчуванням C# забороняє прямі операції з пам'яттю, надаючи систему багатих типів і збір сміття. Пряме керування пам'яттю все ще доступне в спеціальному режимі «небезпечного» коду (див. нижче), але вимагає явного оголошення. Тому в C# активно використовується лише один оператор доступу.

- перетворення типів у C# набагато суворіші, ніж у C++, особливо тому, що більшість перетворень можна виконати лише явно. Крім того, усі

скорочення мають бути безпечними (тобто заборонено неявне переповнення, використання цілочисельних змінних як покажчиків тощо). Природно, це значно спрощує аналіз типу компіляції.

Однією з поширених помилок в C++ є відсутність оператора `break` під час обробки однієї з гілок оператора `switch`. Проблема «пропуску» в C# просто вирішується: компілятору потрібен явний оператор перетворення (`break` або `goto case <name>`) у будь-якій гілці.

Як і Java, C# не має множинного успадкування, але рекомендується використовувати кілька реалізацій інтерфейсів. Хоча думки щодо множинного успадкування дуже різняться, відсутність цього механізму в C# має принаймні покращити розробку компілятора [11, 12, 13].

2.3 AutoCAD .NET API

AutoCAD. NET API дозволяє керувати програмами AutoCAD і файлами креслень на рівні програмного забезпечення, використовуючи доступні колекції або бібліотеки. Ці об'єкти можуть бути надані багатьма різними мовами програмування та різними середовищами розробки програмного забезпечення.

Деякі з переваг, одержуваних від застосування. NET API в AutoCAD:

- програмне забезпечення з відкритим вихідним кодом для доступу до графіки AutoCAD з різних середовищ розробки додатків. До API .NET розробники були обмежені ActiveX і мовами, які підтримували COM, AutoLISP і C++ з ObjectARX;

- при використанні «рідної» інтеграції з іншими програмами Windows (наприклад, MS Excel і Word) вона стає ефективнішою та простішою. NET API або бібліотека ActiveX/COM;

- бібліотека .NET Framework розроблена для 32-розрядних і 64-розрядних версій Windows. VBA доступний лише в 32-розрядній версії;

- надає доступ до інших налаштувань API, які не вважаються традиційними мовами, наприклад C+.

Об'єкти є основними будівельними блоками AutoCAD. NET API. Кожен з цих об'єктів точно відповідає конкретним типам примітивів, доступних в AutoCAD. Класи цих об'єктів згруповані в просторах імені і різних бібліотеках.

AutoCAD. NET API складається з різноманітних DLL. Ці DLL містять широкий спектр класів, структур, методів і подій і можуть надавати доступ до графічних файлів AutoCAD. Кожна DLL визначає інший простір імен для організації розміщення компонентів бібліотеки, скомпільованих відповідно до її функцій.

Розділяють 3 основних DLL-файлу в AutoCAD. NET API:

1. AcDbMgd.dll. Використовують з об'єктами файлів креслення.
2. AcMgd.dll. Використовують з самим додатком AutoCAD.
3. AcCui.dll. Використовують з файлами користувача налаштувань [7, 17,18].

Перш ніж ви зможете використовувати класи, структури, методи та події AutoCAD .NET API, ви повинні спочатку встановити відповідну DLL в проєкті, а потім встановити посилання на необхідний простір імен. Після вказівки посилання на файл DLL, що дозволяє використовувати AutoCAD .NET API у вашому проєкті, вам потрібно встановити властивість «локальна копія» для підключення до бібліотеки зі значенням «False». Ця властивість відповідає за визначення того, чи створює MS Visual Studio копію DLL-файлу, на який посилаються, і під час компіляції його в CIL звертатися до копії, розміщуючи її в каталозі, який є частиною збірки проєкту. Оскільки ці DLL є частиною AutoCAD, створення їх копій під час завантаження файлів деталей до AutoCAD може призвести до несподіваних результатів.

Висновки до розділу

В цьому розділі був проведений аналіз системи проектної документації на основі програмного забезпечення SPDS GraphiCS. Було визначено що представлене програмне забезпечення існує як доповнення до графічної платформи програми AutoCAD і приносить із собою багато нових функцій з метою покращення роботи спеціаліста. Було розглянуто позитивні сторони даного доповнення і визначено раціональність його використання у меблевому проектуванні.

Також наведені основні особливості мови C# та її переваги і недоліки щодо інших мов.

AutoCAD .NET API використовується для адаптації та розширення функціональних можливостей AutoCAD та продуктів на його основі. Дана технологія забезпечує безпосередній доступ до структур бази даних AutoCAD, визначення вбудованих команд та інших внутрішніх програмних елементів.

РОЗДІЛ 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Моделі та їхні параметри в САПР

У процесі автоматизованого проектування замість використання неіснуючих проєкційних об'єктів ми маніпулюємо деякими квазіоб'єктами - моделями, які відображають певні властивості об'єктів. Моделі можуть бути фізичними об'єктами (макети, дужки) або специфікаціями. У нормативній моделі розрізняють функціональні моделі, моделі поведінки, інформаційні моделі та структурні моделі. Якщо ці моделі формалізовані за допомогою математичних інструментів і мови, їх називають математичними моделями.

У свою чергу, математичні моделі можуть бути геометричними, топологічними, динамічними, логічними й т.п., якщо вони відбивають відповідні властивості об'єктів. На рівні з математичними моделями під час проектування використовують функціональні моделі, а також інформаційні моделі у представленні діаграм сутність - відношення, геометричні моделі (креслення). Надалі, якщо немає спеціального застереження, під словом "модель" будемо мати на увазі математичну модель. Своєю чергою, математичні моделі можуть бути представлені логічними, геометричними, динамічними, топологічними, якщо вони відображають відповідні властивості об'єкта. Разом з математичною моделлю при проектуванні використовують функціональну модель, інформаційну модель у вигляді діаграм суттєвих зв'язків та геометричну модель (креслення). Відтепер, якщо не вказано інше, слово «модель» представлятимомо математичну модель.

Математична функціональна модель у загальному випадку являє собою алгоритм обчислення вектора вихідних параметрів Y при заданих векторах параметрів елементів (внутрішніх параметрів) X і зовнішніх параметрів Q .

Математичні моделі бувають символічні та чисельні. Під час використання моделей символічних, в основному використовують не

значення величин, а певні символічні позначення (ідентифікатори). Моделі чисельні моделі бувають аналітичні, іншими словами є можливість їх представити у вигляді явно виражених залежностей параметрів на виході Y від внутрішніх параметрів X та зовнішніх Q , або моделями алгоритмічними, у яких зв'язок Y , X та Q заданий неявно і його можна предствити у вигляді певного алгоритму моделювання.

Імітаційні моделі – один з найважливіших окремих випадків моделей алгоритмічних, під час наявності зовнішніх впливів на систему, будуть відображатись системні процеси. Іншими словами, поведінково-алгоритмічна модель це імітаційна модель.

По ряду інших ознак виконують математичну класифікацію моделей.

Тому розрізняють системні моделі, функціонально-логічні моделі, макро (централізовані) моделі та мікро (розподілені) моделі, які належать до ієрархії. За характером математичних засобів, що використовуються для опису, розрізняють мовні моделі, моделі теорії множин, моделі абстрактної алгебри, нечіткі моделі, автоматичні моделі тощо. Наприклад, особливе положення займає геометрична модель, яка використовується в системі проектування.

3.2. Геометричне моделювання та машинна графіка

Центральне місце САПР машинного виробництва займають підсистеми машинної графіки та геометричного моделювання. При використанні геометричної моделі проектування виробу зазвичай виконується інтерактивно, тобто математичний об'єкт, що відображає форму деталі, склад компонента та деякі можливі додаткові параметри (масу, момент інерції, поверхню). У цьому типі підсистеми типовий маршрут обробки даних включає отримання плану проектування в програмі, подання плану проектування з використанням геометричної моделі (геометричне моделювання) і підготовку плану проектування для візуалізації, візуалізації

робочої станції та взаємодії на вимогу. Останні дві операції є машинними графічними пристроями. Коли ми говоримо про математичне програмне забезпечення, то в основному мають на увазі моделі, методи та алгоритми, які використовуються для геометричного моделювання та підготовки візуалізації.

Існує математичне забезпечення 2D (двовимірне) і 3D (тривимірне) моделювання. Головне застосування двовимірної графіки – топологічне проектування друкованих плату САПР промисловості електронної галузі, підготовка креслярської документації в машинобудівних САПР [3].

Під час процесу тривимірного моделювання створюються моделі геометричні, тобто такі моделі, що показують геометричні властивості виробів. Також розрізняють геометричні моделі каркасні (дротові), поверхневі, об'ємні (твердотільні).

У даній роботі здійснюється моделювання вітальні разом із отриманням усіх необхідних розмірів потрібних частин, які необхідні з метою подальшого отримання документації та формування карт листових матеріалів із яких виготовляється вітальня. З метою розуміння всіх відмінностей між різними моделями, буде доцільно описати і інші моделі.

Рамна модель представляє певну форму деталі у вигляді набору скінченних ліній на поверхності деталі. Для кожної лінії відомі координати кінцевих точок і вказується їх зв'язок з ребром або поверхнею. Використовувати каркасну модель в подальших операціях маршруту проектування незручно, тому модель каркаса на даний час використовується рідко.

Модель поверхні відображає форму деталі та визначає її граничну поверхню, наприклад, у вигляді набору даних про грані, ребра та вершини.

Детальна модель з поверхнею складної форми займає особливе положення, яке називають поверхнею скульптури. До таких частин належать корпуси багатьох транспортних засобів (наприклад, кораблів та автомобілів) і частини (лопаті турбіни, крила літака), які обтікають рідини та газу.

Відмінність 3D-моделей полягає в тому, що вони чітко зберігають інформацію про атрибуцію елементів відносно внутрішнього або зовнішнього простору деталі.

Ці моделі показують об'єкти із замкнутими об'ємами, які називаються manifold. Деякі системи геометричного моделювання дозволяють маніпулювати нерізноманітними моделями, такими як моделі об'єктів, які торкаються одна одної в одній точці або вздовж прямої лінії. Маленька модель дуже зручна в процесі проектування, на проміжному етапі можна одночасно використовувати тривимірні і двовимірні моделі без уточнення товщини стінки конструкції.

Використовуються операції перенесення, масштабування та повороту для перетворення графічних об'єктів.

Переніс точки з положення P в положення C можливо виконувати за такими формулами

$$C_{xi} = P_{xi} + \Delta x_i \quad (3.1)$$

де: Δx_i – приріст по координаті x_i . Проте буде зручніше операції перетворення уявляти в матричній формі.

$$C = PT \quad (3.2)$$

де: T – матриця яка перетворюється. Точки C та P у 2D випадку зображують векторами-рядками 1×3 , у яких окрім значень 2 координат, що є при даному поданні однорідними, ще вказують масштабний множник W. Тоді перенос для двовимірного випадку можна виразити у вигляді (3.2), де T береться із табл. 3.1, а $W = 1$.

1	0	0
0	1	0
Δx_1	Δx_2	1

Таблиця 3.1

Поворот матриці T , разом із пераціями масштабування а також повороту матриці T будуть представлені в табл. 3.2 і табл. 3.3, де m_x, m_y – масштабні множники, φ – кут повороту.

Формула переміщення в свою чергу разом із одночасним поворотом буде зображатись наступним чином

$$C = P T_{сд} T_{пов} = PT \quad (3.3)$$

де: $T = T_{сд} T_{пов}$, $T_{сд}$ – зображує певну матрицю переміщення, а також $T_{пов}$ – зображує матрицю повороту.

m_x	0	0
0	m_y	0
0	0	1

Таблиця 3.2

$\cos \varphi$	$\sin \varphi$	0
$-\sin \varphi$	$\cos \varphi$	0
0	0	1

Таблиця 3.3

Деяке зображення певних графічних елементів представлених в даній формі необхідний для візуалізації представлених елементів на карту бітового представлення відеосистеми. Уявімо що є потреба розгорнути певний відрізок AB прямої $y = ax + b$, нехай $1 \geq a \geq 0$ (під час інших значеннях a розглянутий нижче алгоритм залишається справедливим після певних модифікацій). Введімо такі позначення: $A = (x_a, y_a), B = (x_b, y_b)$; за величність пікселя візьмемо одиницю. В алгоритмі номери стовпців а також

рядків бітової карти, при перетині зазначених мають існувати точки відрізка, які будуть визначатись в наступний спосіб:

1. $\Delta x = xb - xa; \Delta y = yb - ya; x = xa; y = ya;$

2. $d = 2\Delta y - \Delta x;$

3. якщо $d \geq 0$, то $\{y = y + 1; d = d + 2(\Delta y - \Delta x)\}$,
інакше $d = d + 2\Delta y;$

4. $x = x + 1$

5. перехід до пункту 3, поки не досягнута точка **B**.

Відсутність довгих арифметичних операцій як наприклад множення сприяє економічності даного алгоритму. Потрібно виділити вікно для визначення частини сцени яка повинна бути виведена на екран дисплею.

Припустимо вікно обмежене цими лініями $x = x_1, x = x_2, y = y_1, y = y_2$, (рис. 3.1). Почергово для кожного багатокутника буде перевірятися розташування його вершин та ребер щодо крайніх границь вікна. Так, для багатокутника **ABCD** (див. рис. 3.1) при відсіканні по крайній границі $x = x_2$, утворюється певна множина вершин в порядку обходу по годинниковій стрілці. Можливі чотири ситуації для наступних послідовних вершин **P i R**:

- припустимо що $x_P > x_2$ та $x_R > x_2$, тоді дані дві вершини та рівне їм ребро існують поза межами вікна і будуть виключатись з подальшого аналізу;
- припустимо що $x_P \leq x_2$ та $x_R \leq x_2$, тоді дві вершини та рівне їм ребро будуть залишатись для подальшого аналізу;
- припустимо що $x_P \leq x_2$ та $x_R > x_2$, тоді вершина **P** залишається в списку вершин, а в свою чергу **R** буде замінитись новою вершиною з

наступними координатами $x = x_2, y = y_P + (y_R - y_P)(x_2 - x_P)/(x_R - x_P)$ в даному прикладі такою новою вершиною стане E ;

- припустимо що $x_P > x_2$ та $x_R \leq x_2$, тоді вершина P буде замінитись новою вершиною з наступними координатами $x = x_2, y = y_R + (y_P - y_R)(x_2 - x_R)/(x_P - x_R)$ а вершина R буде залишатись в списку вершин; у даному прикладі новою вершиною буде F .

Після закінчення перегляду щодо всіх границь у вікні з'являються вершини, що залишились у списку.

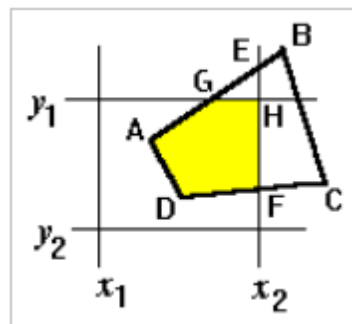


Рис. 3.1 Виділення вікна

За допомогою цих правил, одержимо багатокутник $AEFD$, а також провівши відсікання граничуючи верхньою границею – багатокутник $AGFD$ (див. рис. 3.1). Однак правильні розрахунки будуть інші, багатокутник $AGHFD$. Алгоритми перебудови проєкцій перетворюють 3D зображення у 2D. Якщо брати до уваги випадок побудови центральної проєкції, тоді кожна точка 3D зображення відображається на 2D поверхню методом перерахування координат x та y (рис. 3.2). Так, координату x'_a точки A' обчислюють за формулою

$$x'_a = \frac{x_a d}{z} \quad (3.4)$$

Також аналогічно вираховується координата y'_a точки A' .

У паралельних проєкціях $d \rightarrow \infty$ і координати x та y точок A' і A збігаються. Тому побудова паралельних проєкцій приводиться до виділення вікна, при необхідності до перевероту зображення та можливо до видалення прихованих ліній.

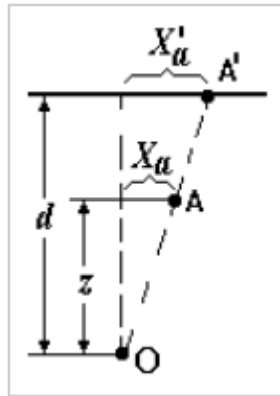


Рис. 3.2. Побудова центральної проєкції точки A

Висновки до розділу

У цьому розділі був проведений аналіз використовуваного математичного забезпечення для двовимірного та тривимірного моделювання САПР. На основі цього було описано процес 2D та 3D моделювання меблевого виробу з використанням різних підходів та формул, а також у різних умовах. Отже, можна підбити підсумки і сказати що певні можливості геометричного моделювання набагато краще представлені у програмному забезпеченні AutoCAD ніж було описано в розділі 3. В нашому випадку була спроба описати найбільш необхідні речі математичного забезпечення для AutoCAD.

РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

4.1. Система AutoCAD

AutoCAD – це автоматизована система проектування та креслення від Autodesk. Продукти серії AutoCAD є однією з найпоширеніших систем САПР у світі. Виробник AutoCAD Autodesk розробляє AutoCAD CAD більше 25 років. За цей час сторонні компанії та самі Autodesk створили тисячі доповнень і професійних рішень.

У першій версії AutoCAD використовував примітивні об'єкти (лінії, кола, дуги, текст). З середини 1990-х років AutoCAD підтримує об'єкти інтерфейсу користувача C++. Сучасний AutoCAD включає в себе повний набір інструментів для моделювання та 3D-графіки. AutoCAD підтримує кілька інтерфейсів: AutoLisp, VisualLisp, VBA_.NET і ObjectARX.

В останні роки AutoCAD надає підтримку формату зберігання САПР DWF, розробленому та просуваному Autodesk.

AutoCAD LT – це «вбудована» версія AutoCAD. Ви також можете придбати версії AutoCAD LT на комп'ютерах і в пов'язаних магазинах, а AutoCAD продається виключно авторизованими посередниками Autodesk.

AutoCAD 2020 є лідером у додатках для малювання, деталізації та концептуального проектування. AutoCAD 2020 спеціально розроблений для задоволення потреб дизайнерів. Його функції можуть прискорити щоденну роботу з кресленням, тим самим підвищуючи швидкість і точність виконання та заощаджуючи час.

Можливість масштабувати пояснення шарів та атрибути безпосередньо у вікні попереднього перегляду заощадить ваш час; розширені функції для обробки текстів і таблиць, а також можливість створювати комбіновані виноски для задоволення естетичних та професійних вимог. Інструменти для концептуального проектування та візуалізації AutoCAD завжди інноваційні і

можуть підвищити ефективність майже відразу, роблячи ефективність чудовим супутником вашої роботи [14, 15].

Можливості

AutoCAD 2020 знайомить користувачів з новим стандартом продуктивності САПР. Менеджер підшивки, наданий у цій версії, дозволяє логічно поєднувати креслення, відкриваючи шлях для ефективного створення, спільного використання та керування всіма проектними документами.

У AutoCAD 2020 до цієї міцної основи було додано багато твердих інструментів для малювання, включаючи покращені інструменти для обробки таблиць, багаторядкового тексту та штрихування. Крім того, покращено інтерфейс та додана можливість використання динамічних блоків.

AutoCAD 2020 надає потужні інструменти концептуального проектування, візуалізації та документації, включаючи моделювання твердих поверхонь і поверхонь, анімацію, поперечний переріз, тривимірну геометрію площини тощо.

Нові функції в AutoCAD 2020 допомагають спростити щоденні завдання проектування. Масштабування вкладок, окреме керування шарами в кожному вікні перегляду, покращення тексту та електронних таблиць і комбінована візуалізація можуть досягти високого рівня дизайну та документації.

Новий AutoCAD 2020 може прискорити створення креслень, покращуючи швидкість і точність, заощаджуючи час на масштабування та пояснення елементів, підтримуючи використання шарів і таблиць, а також поєднуючи функції візуалізації – усе це може досягти розширеного проектування [8, 9].

4.2. Реалізація програмного забезпечення

Проектування корпусних меблів реалізовано на прикладі вітальні.

Дане програмне забезпечення працює у вигляді додатку для AutoCAD і реалізує побудову корпусних меблів із модульних конструкцій на прикладі вітальні.

На першому етапі створюємо функцію (mybox) на мові C#, і обробляємо функцію в середовищі AutoCAD. Функцію створюємо у середовищі Microsoft Visual Studio використовуючи .NET API.

Приклад вікна додатку для побудови вітальні створеного у Microsoft Visual Studio наведений на рис.4.1. Одне з дизайнерських рішень розробленої вітальні наведено на рис.4.2.

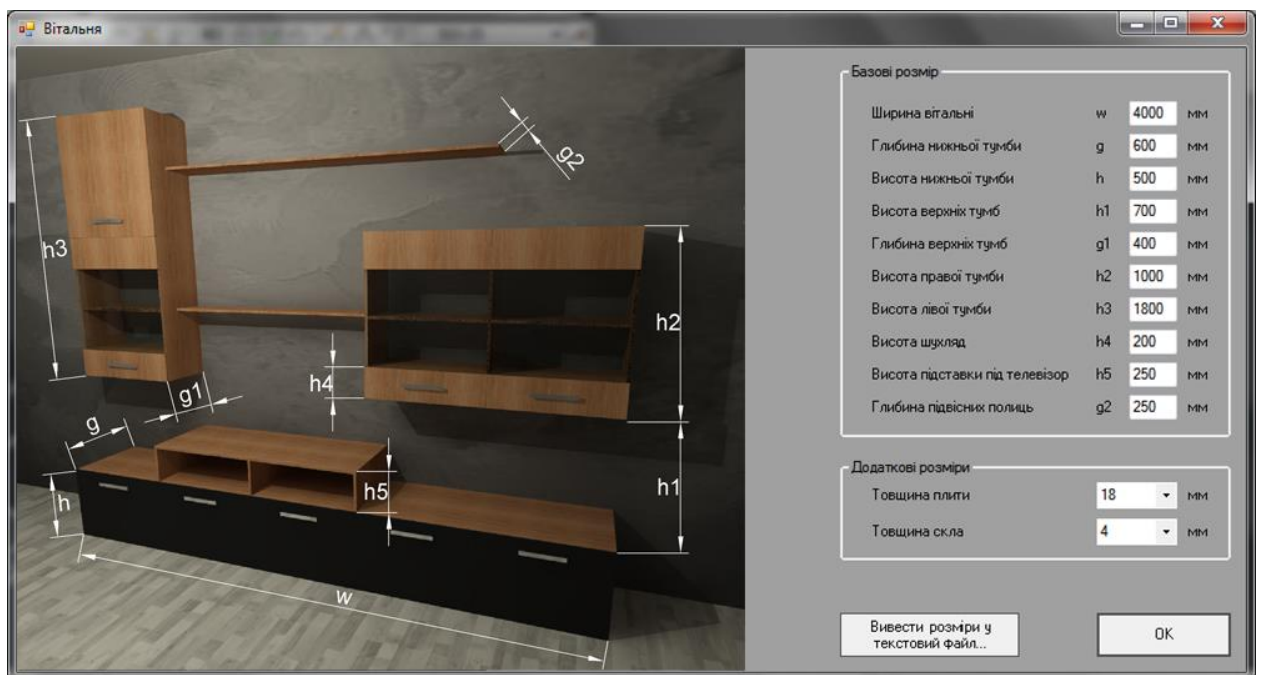


Рис. 4.1. Діалогове вікно для побудови вітальні



Рис. 4.2. Дизайнерське рішення розробленої вітальні

```

gozmir.txt - Блокнот
Файл Редагування Формат Вигляд Довідка
Розміри заготовок плит:
Розміри заготовок нижньої тумби:
1. Розмір нижньої стінки:-----1шт.----(582 4000)
2. Розмір верхньої стінки:-----1шт.----(620 4000)
3. Розмір перегородки:-----6шт.----(582 464)
4. Розмір дверки:-----5шт.----(799.25 482)
Розміри заготовок правої верхньої тумби:
5. Розмір нижньої стінки:-----1шт.----(382 1600)
6. Розмір верхньої стінки:-----1шт.----(420 1600)
7. Розмір перегородки:-----3шт.----(382 964)
8. Розмір полиці над шухлядою:-----2шт.----(382 773)
9. Розмір шухляди:-----2шт.----(799.25 200)
10. Розмір накладки:-----2шт.----(799.25 200)
Розміри заготовок лівої верхньої тумби:
11. Розмір нижньої стінки:-----1шт.----(382 800)
12. Розмір верхньої стінки:-----1шт.----(420 800)
13. Розмір перегородки:-----2шт.----(382 1764)
14. Розмір полиці над шухлядою:-----2шт.----(382 764)
15. Розмір шухляди:-----1шт.----(800 200)
16. Розмір накладки:-----1шт.----(800 200)
17. Розмір дверки:-----1шт.----(800 800)
Розміри заготовок підставки під телевизор:
18. Розмір верхньої стінки:-----1шт.----(582 1618)
19. Розмір задньої стінки:-----1шт.----(1618 232)
20. Розмір перегородки:-----3шт.----(564 232)
Розміри заготовок полиць:
21. Розмір полиці меншої:-----1шт.----(250 1600)
22. Розмір полиці більшої:-----1шт.----(250 2400)
Розміри заготовок задніх стінок з ДВП:
23. Розмір для нижньої тумби:-----1шт.----(4000 500)
24. Розмір для лівої верхньої тумби:-----1шт.----(800 1800)
25. Розмір для правої верхньої тумби:-----1шт.----(1600 1000)
Розмір скляних полиць:
26. Розмір для лівої верхньої тумби:-----1шт.----(382 764)
27. Розмір для правої верхньої тумби:-----2шт.----(382 773)

```

Рис. 4.3. Файл з результатами розрахунків побудованої вітальні

Частина програмного коду функції *mybox* створеної за допомогою мови програмування C# з використанням .NET API в середовищі Microsoft Visual Studio .

```
[CommandMethod("mybox")]
public static void mybox()
{
    Form1 f1 = new Form1();
    f1.Show();

    // Встановлюємо поточний документ і базу даних, починаємо
    транзакцію
    Document acDoc = Application.DocumentManager.MdiActiveDocument;
    Database acCurDb = acDoc.Database;

    using (Transaction acTrans =
acCurDb.TransactionManager.StartTransaction())
    {
        // Відкриваємо таблицю Блоку для читання
        BlockTable acBlkTbl;
        acBlkTbl = acTrans.GetObject(acCurDb.BlockTableId,
OpenMode.ForRead) as BlockTable;

        // Відкриваємо запис таблиці Блоку простір Моделі (model
space) для запису
        BlockTableRecord acBlkTblRec;
        acBlkTblRec =
acTrans.GetObject(acBlkTbl[BlockTableRecord.ModelSpace],
OpenMode.ForWrite) as
BlockTableRecord;

        double zaz = 0.75;
        double tdvp = 4;
        double shur = Convert.ToDouble(f1.w.Text);
        double glub = Convert.ToDouble(f1.g.Text);
        double vusnt = Convert.ToDouble(f1.h.Text);
        double vusvt = Convert.ToDouble(f1.h1.Text);
        double glubvt = Convert.ToDouble(f1.g1.Text);
        double vusvpt = Convert.ToDouble(f1.h2.Text);
        double vuslvt = Convert.ToDouble(f1.h3.Text);
        double vusshyh = Convert.ToDouble(f1.h4.Text);
        double vusppa = Convert.ToDouble(f1.h5.Text);
        double shurp = Convert.ToDouble(f1.g2.Text);
        double sklo = Convert.ToDouble(f1.comboBox2.Text);
        double tpl = Convert.ToDouble(f1.tovpl.Text);
        double per = shur / 5;

        // Створюємо 3D Вох_____низ тумби нижньої
        Solid3d nuzt = new Solid3d();
        nuzt.SetDatabaseDefaults();
        nuzt.CreateBox(glub-tpl, shur, tpl);
        // Встановлюємо центр 3D тіла_____низ тумби нижньої
        nuzt.TransformBy(Matrix3d.Displacement(new Point3d((glub-tpl)
/ 2, shur / 2, tpl / 2) -
Point3d.Origin));
```

```

//-----
// Створюємо 3D Вох____верх тумби нижньої
Solid3d verht = new Solid3d();
verht.SetDatabaseDefaults();
verht.CreateBox(glub+20, shur, tpl);
// Встановлюємо центр 3D тіла____верх тумби нижньої
verht.TransformBy(Matrix3d.Displacement(new Point3d(glub /
2+10, shur / 2, vusnt-tpl / 2) -
Point3d.Origin));
//-----
// Створюємо 3D Вох____ліва частина тумби
Solid3d livacht = new Solid3d();
livacht.SetDatabaseDefaults();
livacht.CreateBox(glub-tpl, tpl, vusnt - (tpl+tpl));
// Встановлюємо центр 3D тіла____ліва частина тумби
livacht.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, tpl / 2, (vusnt - tpl) / 2+(tpl/2)) -
Point3d.Origin));
//-----
// Створюємо 3D Вох____права частина тумби
Solid3d pravacht = new Solid3d();
pravacht.SetDatabaseDefaults();
pravacht.CreateBox(glub-tpl, tpl, vusnt - (tpl+tpl));
// Встановлюємо центр 3D тіла____права частина тумби
pravacht.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, shur-(tpl/2), (vusnt - tpl) / 2+(tpl/2)) -
Point3d.Origin));
//-----
// Створюємо 3D Вох____перегородка 1 тумби
Solid3d per1t = new Solid3d();
per1t.SetDatabaseDefaults();
per1t.CreateBox(glub-tpl, tpl, vusnt - (tpl + tpl));
// Встановлюємо центр 3D тіла____перегородка 1 тумби
per1t.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, per, (vusnt - tpl) / 2 + (tpl / 2)) -
Point3d.Origin));
//-----
// Створюємо 3D Вох____перегородка 2 тумби
Solid3d per2t = new Solid3d();
per2t.SetDatabaseDefaults();
per2t.CreateBox(glub-tpl, tpl, vusnt - (tpl + tpl));
// Встановлюємо центр 3D тіла____перегородка 2 тумби
per2t.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, per*2, (vusnt - tpl) / 2 + (tpl / 2)) -
Point3d.Origin));
//-----
// Створюємо 3D Вох____перегородка 3 тумби
Solid3d per3t = new Solid3d();
per3t.SetDatabaseDefaults();
per3t.CreateBox(glub-tpl, tpl, vusnt - (tpl + tpl));
// Встановлюємо центр 3D тіла____перегородка 3 тумби
per3t.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, per * 3, (vusnt - tpl) / 2 + (tpl / 2)) -
Point3d.Origin));
//-----
// Створюємо 3D Вох____перегородка 4 тумби
Solid3d per4t = new Solid3d();
per4t.SetDatabaseDefaults();
per4t.CreateBox(glub-tpl, tpl, vusnt - (tpl + tpl));
// Встановлюємо центр 3D тіла____перегородка 4 тумби
per4t.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, per * 4, (vusnt - tpl) / 2 + (tpl / 2)) -
Point3d.Origin));

```

```

//-----
// Створюємо 3D Вох___дверка 1
Solid3d dv1 = new Solid3d();
dv1.SetDatabaseDefaults();
dv1.CreateBox(tpl, per-zaz, vusnt - tpl );
// Встановлюємо центр 3D тіла___дверка 1
dv1.TransformBy(Matrix3d.Displacement(new Point3d(glub-tpl/2,
(per-zaz)/2, (vusnt-tpl)/2 ) -
Point3d.Origin));
//-----
// Створюємо 3D Вох___дверка 2
Solid3d dv2 = new Solid3d();
dv2.SetDatabaseDefaults();
dv2.CreateBox(tpl, per - zaz*2, vusnt - tpl);
// Встановлюємо центр 3D тіла___дверка 2
dv2.TransformBy(Matrix3d.Displacement(new Point3d(glub - tpl
/ 2, per + (per / 2), (vusnt - tpl) / 2) -
Point3d.Origin));
//-----
// Створюємо 3D Вох___дверка 3
Solid3d dv3 = new Solid3d();
dv3.SetDatabaseDefaults();
dv3.CreateBox(tpl, per - zaz*2, vusnt - tpl);
// Встановлюємо центр 3D тіла___дверка 3
dv3.TransformBy(Matrix3d.Displacement(new Point3d(glub - tpl
/ 2, per*2 + (per / 2), (vusnt - tpl) / 2) -
Point3d.Origin));
//-----
// Створюємо 3D Вох___дверка 4
Solid3d dv4 = new Solid3d();
dv4.SetDatabaseDefaults();
dv4.CreateBox(tpl, per - zaz*2, vusnt - tpl);
// Встановлюємо центр 3D тіла___дверка 4
dv4.TransformBy(Matrix3d.Displacement(new Point3d(glub - tpl
/ 2, per*3 + (per / 2), (vusnt - tpl) / 2) -
Point3d.Origin));
//-----
// Створюємо 3D Вох___дверка 5
Solid3d dv5 = new Solid3d();
dv5.SetDatabaseDefaults();
dv5.CreateBox(tpl, per - zaz, vusnt - tpl);
// Встановлюємо центр 3D тіла___дверка 5
dv5.TransformBy(Matrix3d.Displacement(new Point3d(glub - tpl
/ 2, per * 4 + (per / 2)+(zaz/2), (vusnt - tpl) / 2) -
Point3d.Origin));
//-----
// Створюємо 3D Вох___нижня полиця правої верхньої тумби
Solid3d nppvvt = new Solid3d();
nppvvt.SetDatabaseDefaults();
nppvvt.CreateBox(glubvt-tpl, per*2, tpl);
// Встановлюємо центр 3D тіла___нижня полиця правої
верхньої тумби
nppvvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2-tpl/2, per * 4, vusvt+vusnt+(tpl/2)) -
Point3d.Origin));
//-----
// Створюємо 3D Вох___правий бік правої верхньої тумби
Solid3d pppvvt = new Solid3d();
pppvvt.SetDatabaseDefaults();
pppvvt.CreateBox(glubvt-tpl, tpl, vuspvt-tpl*2);
// Встановлюємо центр 3D тіла___правий бік правої верхньої
тумби

```

```

        pbpvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2 - tpl / 2, shur - tpl / 2, vuspvt / 2 + vusnt + vusvt) -
                                                    Point3d.Origin));
//-----
// Створюємо 3D Вох____лівий бік правої верхньої тумби
Solid3d lbpvt = new Solid3d();
lbpvt.SetDatabaseDefaults();
lbpvt.CreateBox(glubvt-tpl, tpl, vuspvt - tpl * 2);
// Встановлюємо центр 3D тіла____лівий бік правої верхньої
тумби
        lbpvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2-tpl/2, per*3+tpl/2, vuspvt / 2 + vusnt + vusvt) -
                                                    Point3d.Origin));
//-----
// Створюємо 3D Вох____перегородка правої верхньої тумби
Solid3d perpvt = new Solid3d();
perpvt.SetDatabaseDefaults();
perpvt.CreateBox(glubvt-tpl, tpl, vuspvt - tpl * 2);
// Встановлюємо центр 3D тіла____перегородка правої верхньої
тумби
        perpvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2-tpl/2, per * 4 , vuspvt / 2 + vusnt + vusvt) -
                                                    Point3d.Origin));

```

Висновки до розділу

Під час написання цього розділу було проведено дослідження програмного забезпечення сімейства AutoCAD. Провівши різносторонній аналіз було вирішено використати мову програмування С# для написання додатку під програму AutoCAD. Описавши етапи встановлення бібліотеки AutoCAD. NET API було визначено що даний пакет дозволяє створювати програми в моделювальному середовищі напряму звертаючись до методів проектувального виробу та маніпулювати ним.

РОЗДІЛ 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

5.1 Опис ідеї проекту

Уявіть, що витрачаєте весь свій час на заняття, яке вам подобається.

Уявіть, що ви працюєте над власним проектом, ви більше не гвинтик у системі іншої компанії.

Уявіть, що сьогодні ваш останній робочий день як працівника чужої компанії.

Через деякий час відкрийте ноутбук у своєму домашньому офісі, відкрийте власний магазин чи аутсорсингову компанію.

Ви більше не виконуєте чийсь накази.

Стань сам собі босом і формуй своє нове майбутнє.

Ця бізнес-модель добре зарекомендувала себе на прикладі «випадкових» підприємців.

Для успішного стартап-проекту і можливості не “прогоріти” на початку шляху існують певні аспекти дій:

- одне з найважливіших це правильна ідея для майбутнього проекту, разом з ретельним дослідженням ринку даної області, у якій проект буде вести свою діяльність. Адже якщо новий продукт не задовільняє нагальну потребу ринку споживачів то потреба в такому продукті відпаде дуже швидко. Отже і вкладення в такий проект виявляться безперечно марними;
- пошук у команду досвідчених спеціалістів вибраної сфери діяльності;
- залучення інверторів які зацікавлені у вкладенні коштів в проблемну область продукту або знайти стартовий капітал з власних грошових джерел;
- створення бізнес-плану для правильного розпорядження грошовими коштами в рамках створення нового продукту;
- прийняття дедлайну кінцевої версії проекту.

5.2 Маркетингова програма стартап-проекту

Після успішного втілення ідеї в реальний продукт після проведених дій у підрозділі 5.1, перш ніж опублікувати його потрібно запускати деякий пробний період на узгоджений час для певного вузького кола споживачів з метою тестування та виявлення критичних недоліків.

Успішно пройшовши етап первинного тестування та виправлення усіх знайдених недоліків настав час задуматись над глобальним запуском продукту. Для того, щоб охопити якомога більшу аудиторію, потрібно неодмінно провести маркетингову реклам-компанію з найнеобхіднішою інформацією. З метою максимально продуктивно та якісно провести запуск проекту, буде хорошим рішенням найняти маркетингову компанію яка опублікує таргетовану рекламу серед потенційно зацікавлених споживачів. Для легкої впізнаності продукту найкращим кроком буде створення власного логотипу. Він повинен поєднувати в собі сучасні підходи до дизайну брендваної реклами, наприклад мінімалізм, а також нагадувати, хоч і віддалено про те що реалізовано брендваним продуктом.

Якщо підбити підсумки, брендова продукція - це:

- ознака ввічливості в діловому етикеті;
- ефективні способи збільшення продажів, залучення нових клієнтів і утримання старих клієнтів;
- ефективні засоби підвищення впізнаваності бренду;
- існуюча методика формування позитивного іміджу компанії;
- чудовий вибір для заохочення співробітників.

5.3 Аналіз технологічних можливостей реалізації ідей проекту

Проект буде реалізовано за допомогою технології AutoCAD .NET API. Цей технічний підхід вибраний не випадково. Адже, ця технологія є широко відомою, швидкою в освоєнні, використовується як досвідченими

спеціалістами так і початківцями. Тому проблем або труднощів під час реалізації власного стартап проекту у великій кількості виникнути немає, на відміну від маловідомих технологій через які буде витрачатись додатковий час команди для освоєння програмного та технічного забезпечення продукту на базі погано вибраної технології.

За допомогою технології AutoCAD .NET API можна виконувати технічні завдання будь-якої складності, адже вона підтримує широкий спектр можливих реалізацій у вибраній області.

Висновки до розділу

Підбиваючи підсумки цього розділу можна сказати що актуальність проблеми створення інтуїтивно легких додатків для моделювання меблевих виробів існує по сьогоднішній день. В рамках дослідження доцільності розробки стартап-проекту було визначено певні етапи, слідувавши яким можна охопити деяку частину ринку меблевої галузі та успішно розвинути свій продукт. Також під час дослідження було виявлено що рекламно-маркетингова компанія грає грандіозну роль в успішності молодого бізнесу, адже який би хороший і функціональний продукт не був, якщо ним користується надто мало споживачів, він дуже швидко стане збитковим.

ВИСНОВКИ

Розроблена САПР у вигляді застосунку до AutoCAD виконує проектування корпусних меблів із модульних конструкцій на прикладі вітальні.

Основний результат роботи зводиться до такого:

1. Проаналізовано наявне програмне забезпечення в сфері меблевого виробництва.

2. У розробленому програмному забезпеченні застосування мови програмування C# в середовищі Microsoft Visual Studio, технології AutoCAD .NET API, дозволило використовувати архітектуру AutoCAD.

3. Розроблений застосунок для AutoCAD забезпечує виведення усіх результатів геометричних розрахунків у файл для подальшого створення необхідної технічної документації.

Зрозумілий, а також наглядний UI / UX програми було створено з метою полегшення проектування меблевих виробів, для зручної побудови тривимірного каркасу вітальні, з можливістю подальшого вдосконалення дизайнерами використовуючи наявні функціональні можливості середовища AutoCAD.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кунву Ли. САПР технологических процесів / Ли. Кунву. – М. : Питер, 2017. – 560 с.
2. Кондаков А. И. САПР технологических процессов / А. И. Кондаков. – М. : Академия. – 2017. – 272 с.
3. Грабченко А. І. Теорія 3D моделювання / А. І. Грабченко, В. Л. Доброскок. – Х. : Видавництво НТУ "ХПІ", 2019. – 230 с.
4. Наумчук О. М. Основи систем автоматизованого проектування / О. М. Наумчук. – Рівне : Видавництво НУВГП, 2018. – 126 с.
5. Стариков А. В. САПР мебели. Автоматизированное конструирование изделий корпусной мебели в САПР "bCAD для Мебельщика" / А.В. Стариков. – Воронеж : ВГЛА, 2007. – 228 с.
6. Бунаков П. Ю. Новая парадигма проектирования САПР сложной корпусной мебели для показанного программного производства / П.Ю. Бунаков, А.В. Стариков, А.А. Старикова, В.Н. Харин. – М. : МГУ, 2007. – 321с.
7. Microsoft. Разработка Windows-приложений на Microsoft Visual Basic .NET и Microsoft Visual C# / Microsoft. – М. : Официальные издания Microsoft Press, 2018. – 514 с.
8. Агуров П. С#. Разработка компонентов в MS Visual Studio / П. Агуров. – СПб. : БХВ Петербург, 2018. – 467 с.
9. Нэш Т. С# Ускоренный курс для профессионалов / Т. Нэш. – М. : Вильямс, 2018. – 576 ср.
10. Нейгел К. С# и платформа .NET 4.5 для профессионалов / К. Нейгел, Б. Ивсен, Дж. Глинн. – М. : Диалектика-Вильямс, 2019. – 1392 с.
11. Троелсен Э. С# и платформа .NET. Библиотека программиста / Э Троелсен. – СПб. : БХВ Петербург, 2018. – 796 с.
12. Шилдт Г. Полный справочник по С# / Г. Шилдт. – М. : Вильямс, 2018. – 744 с.

13. Фролов А. Язык С# / А. Фролов, Г. Фролов. – М. : Диалог-МИФИ, 2018. – 560 с.
14. Вернер З. AutoCAD 2020. “Руководство чертежника, конструктора, архитектора / З. Вернер – М. : ООО “Бином – Пресс”, 2020. – 763 с.
15. Полещук Н. Н. САПР на базе AutoCAD - как это делается / Н. Н. Полещук. – СПб : БХВ-Петербург, 2004. – 1168 с.
16. AutoCAD .NET Developer's Guide – Режим доступа: <http://docs.autodesk.com/ACD/2010/ENU/AutoCAD%20.NET%20Developer%27s%20Guide/index.html> - доступный станом на 06.12.2021
17. AutoCAD DevBlog – Режим доступа: <https://adndevblog.typepad.com/autocad> - доступный станом на 06.12.2021
18. ADN Open CIS – Режим доступа: <https://adn-cis.org/articles> - доступный станом на 06.12.2021
19. Студопедія: Методи й алгоритми комп'ютерної графіки – Режим доступа: <https://studopedia.org/5-24292.html> - доступный станом на 06.12.2021
20. Інфопедія: Моделі і їхні параметри в САПР - Режим доступа: <https://infopedia.su/6x581d.html> - доступный станом на 06.12.2021
21. Основы сапр для студентів - Режим доступа: <https://uchebana5.ru/cont/1825332-p12.html> - доступный станом на 06.12.2021
22. Будуємо дім: Меблева промисловість - Режим доступа: <https://www.stroimdom.com.ua/topic/162591-mebleva-promislovist> - доступный станом на 06.12.2021
23. Системи проектування сінтаро: САПР - Режим доступа: www.cintaro.com.ua - доступный станом на 06.12.2021

ДОДАТОК А

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Autodesk.AutoCAD.Runtime;
using Autodesk.AutoCAD.ApplicationServices;
using Autodesk.AutoCAD.DatabaseServices;
using Autodesk.AutoCAD.Geometry;
using Autodesk.AutoCAD.EditorInput;

namespace my
{
    public class Class1
    {
        [CommandMethod("mybox")]
        public static void mybox()
        {
            Form1 f1 = new Form1();
            f1.Show();

            Document acDoc = Application.DocumentManager.MdiActiveDocument;
            Database acCurDb = acDoc.Database;

            using (Transaction acTrans =
acCurDb.TransactionManager.StartTransaction())
            {
                BlockTable acBlkTbl;
                acBlkTbl = acTrans.GetObject(acCurDb.BlockTableId,
                                                OpenMode.ForRead) as BlockTable;

                BlockTableRecord acBlkTblRec;
                acBlkTblRec =
acTrans.GetObject(acBlkTbl[BlockTableRecord.ModelSpace],
                    OpenMode.ForWrite) as BlockTableRecord;

                double zaz = 0.5;
                double tdvp = 3;
            }
        }
    }
}
```

```

double shur = Convert.ToDouble(f1.w.Text);
double glub = Convert.ToDouble(f1.g.Text);
double vusnt = Convert.ToDouble(f1.h.Text);
double vusvt = Convert.ToDouble(f1.h1.Text);
double glubvt = Convert.ToDouble(f1.g1.Text);
double vuspvt = Convert.ToDouble(f1.h2.Text);
double vuslvt = Convert.ToDouble(f1.h3.Text);
double vusshyh = Convert.ToDouble(f1.h4.Text);
double vusppa = Convert.ToDouble(f1.h5.Text);
double shurp = Convert.ToDouble(f1.g2.Text);
double sklo = Convert.ToDouble(f1.comboBox2.Text);
double tpl = Convert.ToDouble(f1.tovpl.Text);
double per = shur / 5;

Solid3d nuzt = new Solid3d();
nuzt.SetDatabaseDefaults();
nuzt.CreateBox(glub-tpl, shur, tpl);
nuzt.TransformBy(Matrix3d.Displacement(new Point3d((glub-tpl)
/ 2, shur / 2, tpl / 2) -
Point3d.Origin));
//-----
Solid3d verht = new Solid3d();
verht.SetDatabaseDefaults();
verht.CreateBox(glub+20, shur, tpl);
verht.TransformBy(Matrix3d.Displacement(new Point3d(glub /
2+10, shur / 2, vusnt-tpl / 2) -
Point3d.Origin));
//-----
Solid3d livacht = new Solid3d();
livacht.SetDatabaseDefaults();
livacht.CreateBox(glub-tpl, tpl, vusnt - (tpl+tpl));
livacht.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, tpl / 2, (vusnt - tpl) / 2+(tpl/2)) -
Point3d.Origin));
//-----
Solid3d pravacht = new Solid3d();
pravacht.SetDatabaseDefaults();
pravacht.CreateBox(glub-tpl, tpl, vusnt - (tpl+tpl));
pravacht.TransformBy(Matrix3d.Displacement(new
Point3d((glub-tpl) / 2, shur-(tpl/2), (vusnt - tpl) / 2+(tpl/2)) -
Point3d.Origin));
//-----

```

```

Solid3d per1t = new Solid3d();
per1t.SetDatabaseDefaults();
per1t.CreateBox(glub-tpl, tpl, vusnt - (tpl + tpl));
per1t.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, per, (vusnt - tpl) / 2 + (tpl / 2)) -
Point3d.Origin));
//-----
Solid3d per2t = new Solid3d();
per2t.SetDatabaseDefaults();
per2t.CreateBox(glub-tpl, tpl, vusnt - (tpl + tpl));
// Встановлюємо центр 3D тіла____перегородка 2 тумби
per2t.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, per*2, (vusnt - tpl) / 2 + (tpl / 2)) -
Point3d.Origin));
//-----
Solid3d per3t = new Solid3d();
per3t.SetDatabaseDefaults();
per3t.CreateBox(glub-tpl, tpl, vusnt - (tpl + tpl));
per3t.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, per * 3, (vusnt - tpl) / 2 + (tpl / 2)) -
Point3d.Origin));
//-----
Solid3d per4t = new Solid3d();
per4t.SetDatabaseDefaults();
per4t.CreateBox(glub-tpl, tpl, vusnt - (tpl + tpl));
per4t.TransformBy(Matrix3d.Displacement(new Point3d((glub-
tpl) / 2, per * 4, (vusnt - tpl) / 2 + (tpl / 2)) -
Point3d.Origin));
//-----
Solid3d dv1 = new Solid3d();
dv1.SetDatabaseDefaults();
dv1.CreateBox(tpl, per-zaz, vusnt - tpl );
dv1.TransformBy(Matrix3d.Displacement(new Point3d(glub-tpl/2,
(per-zaz)/2, (vusnt-tpl)/2 ) -
Point3d.Origin));
//-----
Solid3d dv2 = new Solid3d();
dv2.SetDatabaseDefaults();
dv2.CreateBox(tpl, per - zaz*2, vusnt - tpl);
dv2.TransformBy(Matrix3d.Displacement(new Point3d(glub - tpl
/ 2, per + (per / 2), (vusnt - tpl) / 2) -
Point3d.Origin));

```

```

//-----
Solid3d dv3 = new Solid3d();
dv3.SetDatabaseDefaults();
dv3.CreateBox(tpl, per - zaz*2, vusnt - tpl);
dv3.TransformBy(Matrix3d.Displacement(new Point3d(glub - tpl
/ 2, per*2 + (per / 2), (vusnt - tpl) / 2) -
Point3d.Origin));
//-----
Solid3d dv4 = new Solid3d();
dv4.SetDatabaseDefaults();
dv4.CreateBox(tpl, per - zaz*2, vusnt - tpl);
dv4.TransformBy(Matrix3d.Displacement(new Point3d(glub - tpl
/ 2, per*3 + (per / 2), (vusnt - tpl) / 2) -
Point3d.Origin));
//-----
Solid3d dv5 = new Solid3d();
dv5.SetDatabaseDefaults();
dv5.CreateBox(tpl, per - zaz, vusnt - tpl);
dv5.TransformBy(Matrix3d.Displacement(new Point3d(glub - tpl
/ 2, per * 4 + (per / 2)+(zaz/2), (vusnt - tpl) / 2) -
Point3d.Origin));
//-----
Solid3d nppvvt = new Solid3d();
nppvvt.SetDatabaseDefaults();
nppvvt.CreateBox(glubvt-tpl, per*2, tpl);
nppvvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2-tpl/2, per * 4, vusvt+vusnt+(tpl/2)) -
Point3d.Origin));
//-----
Solid3d pbpvt = new Solid3d();
pbpvt.SetDatabaseDefaults();
pbpvt.CreateBox(glubvt-tpl, tpl, vuspvt-tpl*2);
pbpvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2 - tpl / 2, shur - tpl / 2, vuspvt / 2 + vusnt + vusvt) -
Point3d.Origin));
//-----
Solid3d lbpvt = new Solid3d();
lbpvt.SetDatabaseDefaults();
lbpvt.CreateBox(glubvt-tpl, tpl, vuspvt - tpl * 2);
lbpvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2-tpl/2, per*3+tpl/2, vuspvt / 2 + vusnt + vusvt) -
Point3d.Origin));

```

```

//-----
Solid3d perpvt = new Solid3d();
perpvt.SetDatabaseDefaults();
perpvt.CreateBox(glubvt-tpl, tpl, vuspvt - tpl * 2);
perpvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2-tpl/2, per * 4 , vuspvt / 2 + vusnt + vusvt) -
Point3d.Origin));
//-----
Solid3d vpvt = new Solid3d();
vpvt.SetDatabaseDefaults();
vpvt.CreateBox(glubvt+20, per * 2, tpl);
vpvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2+10, per * 4, vuspvt+vusvt + vusnt - (tpl / 2)) -
Point3d.Origin));
//-----
Solid3d p1pvt = new Solid3d();
p1pvt.SetDatabaseDefaults();
p1pvt.CreateBox(glubvt-tpl, per-tpl-tpl/2 , tpl);
p1pvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt/2-
tpl/2, shur-(per/2)-tpl/4,tpl/2+vusnt+vusvt+vusshyh-tpl) -
Point3d.Origin));
//-----
Solid3d p2pvt = new Solid3d();
p2pvt.SetDatabaseDefaults();
p2pvt.CreateBox(glubvt - tpl, per - tpl - tpl / 2, tpl);
p2pvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2-tpl/2, per*3+per/2 + tpl / 4, tpl / 2 + vusnt + vusvt + vusshyh - tpl) -
Point3d.Origin));
//-----
Solid3d sh1pvt = new Solid3d();
sh1pvt.SetDatabaseDefaults();
sh1pvt.CreateBox(tpl, per-zaz, vusshyh);
sh1pvt.TransformBy(Matrix3d.Displacement(new Point3d(tpl / 2
+ glubvt-tpl, shur - (per / 2)+zaz/2,vusshyh/2+vusnt+vusvt) -
Point3d.Origin));
//-----
Solid3d sh2pvt = new Solid3d();
sh2pvt.SetDatabaseDefaults();
sh2pvt.CreateBox(tpl, per - zaz, vusshyh);
sh2pvt.TransformBy(Matrix3d.Displacement(new Point3d(tpl / 2
+ glubvt - tpl, per*3+per/2-zaz/2, vusshyh / 2 + vusnt + vusvt) -

```

```

Point3d.Origin));
//-----
Solid3d nak1pvt = new Solid3d();
nak1pvt.SetDatabaseDefaults();
nak1pvt.CreateBox(tpl, per - zaz, vusshyh);
nak1pvt.TransformBy(Matrix3d.Displacement(new Point3d(tpl / 2
+ glubvt - tpl, shur - (per / 2) + zaz / 2, vusshyh / 2 + vusnt +
vusvt+vuspvt-vusshyh-tpl) -
Point3d.Origin));
//-----
Solid3d nak2pvt = new Solid3d();
nak2pvt.SetDatabaseDefaults();
nak2pvt.CreateBox(tpl, per - zaz, vusshyh);
nak2pvt.TransformBy(Matrix3d.Displacement(new Point3d(tpl / 2
+ glubvt - tpl, per * 3 + per / 2 - zaz / 2, vusshyh / 2 + vusnt + vusvt +
vuspvt - vusshyh - tpl) -
Point3d.Origin));
//-----
Solid3d sk1pvt = new Solid3d();
sk1pvt.SetDatabaseDefaults();
sk1pvt.CreateBox(glubvt - tpl, per - tpl - tpl / 2, sklo);
sk1pvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2 - tpl / 2, shur - (per / 2) - tpl / 4, tpl / 2 + vusnt + vusvt + vuspvt/2-
sklo-sklo/2-tpl/2) -
Point3d.Origin));
//-----
Solid3d sk2pvt = new Solid3d();
sk2pvt.SetDatabaseDefaults();
sk2pvt.CreateBox(glubvt - tpl, per - tpl - tpl / 2, sklo);
sk2pvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2 - tpl / 2, per * 3 + per / 2 + tpl / 4, tpl / 2 + vusnt + vusvt + vuspvt /
2 - sklo - sklo / 2 - tpl / 2) -
Point3d.Origin));
//-----
Solid3d nplvt = new Solid3d();
nplvt.SetDatabaseDefaults();
nplvt.CreateBox(glubvt - tpl, per, tpl);
nplvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2 - tpl / 2, per/2, vusvt + vusnt + (tpl / 2)) -
Point3d.Origin));
//-----
Solid3d lblvt = new Solid3d();

```

```

lblvt.SetDatabaseDefaults();
lblvt.CreateBox(glubvt - tpl, tpl, vuslvt - tpl * 2);
lblvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2 - tpl / 2, tpl / 2, vuslvt / 2 + vusnt + vusvt) -
Point3d.Origin));
//-----
Solid3d pblvt = new Solid3d();
pblvt.SetDatabaseDefaults();
pblvt.CreateBox(glubvt - tpl, tpl, vuslvt - tpl * 2);
pblvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2 - tpl / 2, per-tpl/2, vuslvt / 2 + vusnt + vusvt) -
Point3d.Origin));
//-----
Solid3d vlvt = new Solid3d();
vlvt.SetDatabaseDefaults();
vlvt.CreateBox(glubvt + 20, per, tpl);
vlvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt / 2
+ 10, per/2, vuslvt + vusvt + vusnt - (tpl / 2)) -
Point3d.Origin));
//-----
Solid3d pllvt = new Solid3d();
pllvt.SetDatabaseDefaults();
pllvt.CreateBox(glubvt - tpl, per - tpl - tpl, tpl);
pllvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2 - tpl / 2, per / 2, tpl / 2 + vusnt + vusvt + vusshyh - tpl) -
Point3d.Origin));
//-----
Solid3d p2lvt = new Solid3d();
p2lvt.SetDatabaseDefaults();
p2lvt.CreateBox(glubvt - tpl, per - tpl - tpl, tpl);
p2lvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2 - tpl / 2, per / 2, vuspvt + vusvt + vusnt - (tpl / 2)-tpl) -
Point3d.Origin));
//-----
Solid3d shlvt = new Solid3d();
shlvt.SetDatabaseDefaults();
shlvt.CreateBox(tpl, per, vusshyh);
shlvt.TransformBy(Matrix3d.Displacement(new Point3d(tpl / 2 +
glubvt - tpl, per/2, vusshyh / 2 + vusnt + vusvt) -
Point3d.Origin));
//-----
Solid3d naklvt = new Solid3d();

```

```

naklvt.SetDatabaseDefaults();
naklvt.CreateBox(tpl, per, vusshyh);
naklvt.TransformBy(Matrix3d.Displacement(new Point3d(tpl / 2
+ glubvt - tpl, per/2, vusshyh / 2 + vusnt + vusvt + vuspvt - vusshyh - tpl)
-
Point3d.Origin));
//-----
Solid3d dvlvt = new Solid3d();
dvlvt.SetDatabaseDefaults();
dvlvt.CreateBox(tpl, per, vuslvt-vuspvt);
dvlvt.TransformBy(Matrix3d.Displacement(new Point3d(tpl / 2 +
glubvt - tpl, per / 2, (vuslvt - vuspvt) / 2 + vusnt + vusvt + vuspvt-tpl) -
Point3d.Origin));
//-----
Solid3d sklvt = new Solid3d();
sklvt.SetDatabaseDefaults();
sklvt.CreateBox(glubvt - tpl, per - tpl - tpl, sklo);
sklvt.TransformBy(Matrix3d.Displacement(new Point3d(glubvt /
2 - tpl / 2, per / 2, tpl / 2 + vusnt + vusvt + vuspvt / 2 - sklo - sklo / 2
- tpl / 2) -
Point3d.Origin));
//-----
Solid3d pnt1 = new Solid3d();
pnt1.SetDatabaseDefaults();
pnt1.CreateBox(shurp, per+per, tpl);
pnt1.TransformBy(Matrix3d.Displacement(new Point3d(shurp/2,
per / 2 + per + per/2, tpl / 2 + vusnt + vusvt + vuspvt / 2 - sklo - sklo / 2
- tpl / 2) -
Point3d.Origin));
//-----
Solid3d pnt2 = new Solid3d();
pnt2.SetDatabaseDefaults();
pnt2.CreateBox(shurp, per + per + per, tpl);
pnt2.TransformBy(Matrix3d.Displacement(new Point3d(shurp / 2,
per / 2 + per +per/2+ per / 2, tpl / 2 + vusnt + vusvt + vuspvt+(vuslvt-
vuspvt)/2 ) -
Point3d.Origin));
//-----
Solid3d ppa1 = new Solid3d();
ppa1.SetDatabaseDefaults();
ppa1.CreateBox(glub - tpl - tpl, tpl, vusppa - tpl);

```

```

        ppa1.TransformBy(Matrix3d.Displacement(new Point3d(glub / 2 ,
per, vusppa / 2 + vusnt-tpl/2) -
                                                    Point3d.Origin));
//-----
Solid3d ppa2 = new Solid3d();
ppa2.SetDatabaseDefaults();
ppa2.CreateBox(glub - tpl - tpl, tpl, vusppa - tpl);
ppa2.TransformBy(Matrix3d.Displacement(new Point3d(glub / 2,
per+per, vusppa / 2 + vusnt - tpl / 2) -
                                                    Point3d.Origin));
//-----
Solid3d ppa3 = new Solid3d();
ppa3.SetDatabaseDefaults();
ppa3.CreateBox(glub - tpl - tpl, tpl, vusppa - tpl);
ppa3.TransformBy(Matrix3d.Displacement(new Point3d(glub / 2,
per+per+per, vusppa / 2 + vusnt - tpl / 2) -
                                                    Point3d.Origin));
//-----
Solid3d zsppa = new Solid3d();
zsppa.SetDatabaseDefaults();
zsppa.CreateBox(tpl, per+per+tpl, vusppa - tpl);
zsppa.TransformBy(Matrix3d.Displacement(new Point3d(tpl/2,
per+per , vusppa / 2 + vusnt - tpl / 2) -
                                                    Point3d.Origin));
//-----
Solid3d vppa = new Solid3d();
vppa.SetDatabaseDefaults();
vppa.CreateBox(glub-tpl, per + per + tpl, tpl);
vppa.TransformBy(Matrix3d.Displacement(new Point3d((glub -
tpl)/ 2, per + per, tpl / 2 + vusnt+vusppa-tpl) -
                                                    Point3d.Origin));
//-----
Solid3d zsnt = new Solid3d();
zsnt.SetDatabaseDefaults();
zsnt.CreateBox(tdvp, shur, vusnt);
zsnt.TransformBy(Matrix3d.Displacement(new Point3d(tdvp / 2-
tdvp, shur / 2, vusnt / 2) -
                                                    Point3d.Origin));
//-----
Solid3d zslvt = new Solid3d();
zslvt.SetDatabaseDefaults();
zslvt.CreateBox(tdvp, per, vuslvt);

```

```

        zslvt.TransformBy(Matrix3d.Displacement(new Point3d(tdvp/2-
tdvp, per / 2, vuslvt/2+vusnt+vusvt) -
                                                    Point3d.Origin));
//-----
Solid3d zspvt = new Solid3d();
zspvt.SetDatabaseDefaults();
zspvt.CreateBox(tdvp, per+per, vuspvt);
zspvt.TransformBy(Matrix3d.Displacement(new Point3d(tdvp/2-
tdvp, per * 4, vuspvt/2+vusnt+vusvt) -
                                                    Point3d.Origin));

acBlkTblRec.AppendEntity(nuzt);
acTrans.AddNewlyCreatedDBObject(nuzt, true);
acBlkTblRec.AppendEntity(verht);
acTrans.AddNewlyCreatedDBObject(verht, true);
acBlkTblRec.AppendEntity(livacht);
acTrans.AddNewlyCreatedDBObject(livacht, true);
acBlkTblRec.AppendEntity(pravacht);
acTrans.AddNewlyCreatedDBObject(pravacht, true);
acBlkTblRec.AppendEntity(per1t);
acTrans.AddNewlyCreatedDBObject(per1t, true);
acBlkTblRec.AppendEntity(per2t);
acTrans.AddNewlyCreatedDBObject(per2t, true);
acBlkTblRec.AppendEntity(per3t);
acTrans.AddNewlyCreatedDBObject(per3t, true);
acBlkTblRec.AppendEntity(per4t);
acTrans.AddNewlyCreatedDBObject(per4t, true);
acBlkTblRec.AppendEntity(dv1);
acTrans.AddNewlyCreatedDBObject(dv1, true);
acBlkTblRec.AppendEntity(dv2);
acTrans.AddNewlyCreatedDBObject(dv2, true);
acBlkTblRec.AppendEntity(dv3);
acTrans.AddNewlyCreatedDBObject(dv3, true);
acBlkTblRec.AppendEntity(dv4);
acTrans.AddNewlyCreatedDBObject(dv4, true);
acBlkTblRec.AppendEntity(dv5);
acTrans.AddNewlyCreatedDBObject(dv5, true);
acBlkTblRec.AppendEntity(nppvt);
acTrans.AddNewlyCreatedDBObject(nppvt, true);
acBlkTblRec.AppendEntity(pbpvt);
acTrans.AddNewlyCreatedDBObject(pbpvt, true);
acBlkTblRec.AppendEntity(lbpvt);
acTrans.AddNewlyCreatedDBObject(lbpvt, true);

```

```
acBlkTblRec.AppendEntity(perpvt);
acTrans.AddNewlyCreatedDBObject(perpvt, true);
acBlkTblRec.AppendEntity(vpvt);
acTrans.AddNewlyCreatedDBObject(vpvt, true);
acBlkTblRec.AppendEntity(plpvt);
acTrans.AddNewlyCreatedDBObject(plpvt, true);
acBlkTblRec.AppendEntity(p2pvt);
acTrans.AddNewlyCreatedDBObject(p2pvt, true);
acBlkTblRec.AppendEntity(sh1pvt);
acTrans.AddNewlyCreatedDBObject(sh1pvt, true);
acBlkTblRec.AppendEntity(sh2pvt);
acTrans.AddNewlyCreatedDBObject(sh2pvt, true);
acBlkTblRec.AppendEntity(nak1pvt);
acTrans.AddNewlyCreatedDBObject(nak1pvt, true);
acBlkTblRec.AppendEntity(nak2pvt);
acTrans.AddNewlyCreatedDBObject(nak2pvt, true);
acBlkTblRec.AppendEntity(sk1pvt);
acTrans.AddNewlyCreatedDBObject(sk1pvt, true);
acBlkTblRec.AppendEntity(sk2pvt);
acTrans.AddNewlyCreatedDBObject(sk2pvt, true);
acBlkTblRec.AppendEntity(nplvt);
acTrans.AddNewlyCreatedDBObject(nplvt, true);
acBlkTblRec.AppendEntity(lblvt);
acTrans.AddNewlyCreatedDBObject(lblvt, true);
acBlkTblRec.AppendEntity(pblvt);
acTrans.AddNewlyCreatedDBObject(pblvt, true);
acBlkTblRec.AppendEntity(vlvt);
acTrans.AddNewlyCreatedDBObject(vlvt, true);
acBlkTblRec.AppendEntity(pllvt);
acTrans.AddNewlyCreatedDBObject(pllvt, true);
acBlkTblRec.AppendEntity(p2lvt);
acTrans.AddNewlyCreatedDBObject(p2lvt, true);
acBlkTblRec.AppendEntity(shlvt);
acTrans.AddNewlyCreatedDBObject(shlvt, true);
acBlkTblRec.AppendEntity(naklvt);
acTrans.AddNewlyCreatedDBObject(naklvt, true);
acBlkTblRec.AppendEntity(dvlvt);
acTrans.AddNewlyCreatedDBObject(dvlvt, true);
acBlkTblRec.AppendEntity(sklvt);
acTrans.AddNewlyCreatedDBObject(sklvt, true);
acBlkTblRec.AppendEntity(pnt1);
acTrans.AddNewlyCreatedDBObject(pnt1, true);
```

```

acBlkTblRec.AppendEntity(pnt2);
acTrans.AddNewlyCreatedDBObject(pnt2, true);
acBlkTblRec.AppendEntity(ppa1);
acTrans.AddNewlyCreatedDBObject(ppa1, true);
acBlkTblRec.AppendEntity(ppa2);
acTrans.AddNewlyCreatedDBObject(ppa2, true);
acBlkTblRec.AppendEntity(ppa3);
acTrans.AddNewlyCreatedDBObject(ppa3, true);
acBlkTblRec.AppendEntity(zsppa);
acTrans.AddNewlyCreatedDBObject(zsppa, true);
acBlkTblRec.AppendEntity(vppa);
acTrans.AddNewlyCreatedDBObject(vppa, true);
acBlkTblRec.AppendEntity(zsnt);
acTrans.AddNewlyCreatedDBObject(zsnt, true);
acBlkTblRec.AppendEntity(zslvt);
acTrans.AddNewlyCreatedDBObject(zslvt, true);
acBlkTblRec.AppendEntity(zspvt);
acTrans.AddNewlyCreatedDBObject(zspvt, true);

ViewportTableRecord acVportTblRec;
acVportTblRec =
acTrans.GetObject(acDoc.Editor.ActiveViewportId,
    OpenMode.ForWrite) as ViewportTableRecord;
acVportTblRec.ViewDirection = new Vector3d(1, -1, 1);
acDoc.Editor.UpdateTiledViewportsFromDatabase();
acTrans.Commit();
    }
}

}

//Виведення розмірів у файл
private void button2_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFile1 = new SaveFileDialog();
    saveFile1.DefaultExt = "*.txt";
    saveFile1.Filter = "Text files|*.txt";
    if (saveFile1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK &&
        saveFile1.FileName.Length > 0)
    {

```

```

using (StreamWriter sw = new StreamWriter(saveFile1.FileName,
true))
{
    double zaz = 0.75;
    double per = Convert.ToDouble(w.Text) / 5;
    sw.WriteLine("Розміри заготовок плит:");
    sw.WriteLine("");
    sw.WriteLine("Розміри заготовок нижньої тумби:");
    sw.WriteLine("");
    sw.WriteLine("1. Розмір нижньої стінки:-----
-----1шт.---- (" + (Convert.ToDouble(g.Text) -
Convert.ToDouble(tovpl.Text)) + " " + Convert.ToDouble(w.Text)+")");
    sw.WriteLine("2. Розмір верхньої стінки:-----
-----1шт.---- (" + (Convert.ToDouble(g.Text) + 20) + " " +
Convert.ToDouble(w.Text) + ")");
    sw.WriteLine("3. Розмір перегородки:-----
-----6шт.---- (" + (Convert.ToDouble(g.Text) -
Convert.ToDouble(tovpl.Text)) + " " + (Convert.ToDouble(h.Text) -
(Convert.ToDouble(tovpl.Text)*2)) + ")");
    sw.WriteLine("4. Розмір дверки:-----
-----5шт.---- (" + (per - zaz) + " " +
(Convert.ToDouble(h.Text)-Convert.ToDouble(tovpl.Text)) + ")");
    sw.WriteLine("");
    sw.WriteLine("Розміри заготовок правої верхньої тумби:");
    sw.WriteLine("");
    sw.WriteLine("5. Розмір нижньої стінки:-----
-----1шт.---- (" + (Convert.ToDouble(g1.Text) -
Convert.ToDouble(tovpl.Text)) + " "+ (per*2) + ")");
    sw.WriteLine("6. Розмір верхньої стінки:-----
-----1шт.---- (" + (Convert.ToDouble(g1.Text) + 20) + " " + (per
* 2) + ")");
    sw.WriteLine("7. Розмір перегородки:-----
-----3шт.---- (" + (Convert.ToDouble(g1.Text) -
Convert.ToDouble(tovpl.Text)) + " " + (Convert.ToDouble(h2.Text) -
(Convert.ToDouble(tovpl.Text)*2)) + ")");
    sw.WriteLine("8. Розмір полиці над шухлядою:-----
-----2шт.---- (" + (Convert.ToDouble(g1.Text) -
Convert.ToDouble(tovpl.Text)) + " " + (per - Convert.ToDouble(tovpl.Text) -
Convert.ToDouble(tovpl.Text) / 2) + ")");

```

```

        sw.WriteLine("9. Розмір шухляди:-----
-----2шт.----(" + (per - zaz) + " " + Convert.ToDouble(h4.Text)
+ "));

        sw.WriteLine("10. Розмір накладки:-----
-----2шт.----(" + (per - zaz) + " " + Convert.ToDouble(h4.Text)
+ "));

        sw.WriteLine("");
        sw.WriteLine("Розміри заготовок лівої верхньої тумби:");
        sw.WriteLine("");
        sw.WriteLine("11. Розмір нижньої стінки:-----
-----1шт.----(" + (Convert.ToDouble(g1.Text) -
Convert.ToDouble(tovpl.Text)) + " " + per + "));
        sw.WriteLine("12. Розмір верхньої стінки:-----
-----1шт.----(" + (Convert.ToDouble(g1.Text) + 20) + " " + per
+ "));

        sw.WriteLine("13. Розмір перегородки:-----
-----2шт.----(" + (Convert.ToDouble(g1.Text) -
Convert.ToDouble(tovpl.Text)) + " " + (Convert.ToDouble(h3.Text) -
(Convert.ToDouble(tovpl.Text) * 2)) + "));
        sw.WriteLine("14. Розмір полиці над шухлядою:-----
-----2шт.----(" + (Convert.ToDouble(g1.Text) -
Convert.ToDouble(tovpl.Text)) + " " + (per - Convert.ToDouble(tovpl.Text) -
Convert.ToDouble(tovpl.Text)) + "));
        sw.WriteLine("15. Розмір шухляди:-----
-----1шт.----(" + per + " " + Convert.ToDouble(h4.Text) +
""));
        sw.WriteLine("16. Розмір накладки:-----
-----1шт.----(" + per + " " + Convert.ToDouble(h4.Text) +
""));
        sw.WriteLine("17. Розмір дверки:-----
-----1шт.----(" + per + " " + (Convert.ToDouble(h3.Text) -
Convert.ToDouble(h2.Text)) + "));
        sw.WriteLine("");
        sw.WriteLine("Розміри заготовок підставки під
телевізор:");
        sw.WriteLine("");
        sw.WriteLine("18. Розмір верхньої стінки:-----
-----1шт.----(" + (Convert.ToDouble(g.Text) -
Convert.ToDouble(tovpl.Text)) + " " + (per+per+Convert.ToDouble(tovpl.Text))
+ "));

```

```

        sw.WriteLine("19. Розмір задньої стінки:-----
-----1шт.----(" + (per + per + Convert.ToDouble(tovpl.Text)) +
" " + (Convert.ToDouble(h5.Text) - Convert.ToDouble(tovpl.Text)) + ")");
        sw.WriteLine("20. Розмір перегородки:-----
-----3шт.----(" + (Convert.ToDouble(g.Text) -
Convert.ToDouble(tovpl.Text) - Convert.ToDouble(tovpl.Text)) + " " +
(Convert.ToDouble(h5.Text) - Convert.ToDouble(tovpl.Text)) + ")");
        sw.WriteLine("");
        sw.WriteLine("Розміри заготовок полиць:");
        sw.WriteLine("");
        sw.WriteLine("21. Розмір полиці меншої:-----
-----1шт.----(" + Convert.ToDouble(g2.Text) + " " + (per*2 ) +
")");
        sw.WriteLine("22. Розмір полиці більшої:-----
-----1шт.----(" + Convert.ToDouble(g2.Text) + " " + (per*3) +
")");
        sw.WriteLine("");
        sw.WriteLine("Розміри заготовок задніх стінок з ДВП:");
        sw.WriteLine("");
        sw.WriteLine("23. Розмір для нижньої тумби:-----
-----1шт.----(" + Convert.ToDouble(w.Text) + " " +
Convert.ToDouble(h.Text) + ")");
        sw.WriteLine("24. Розмір для лівої верхньої тумби:-----
-----1шт.----(" + per + " " + Convert.ToDouble(h3.Text) + ")");
        sw.WriteLine("25. Розмір для правої верхньої тумби:-----
-----1шт.----(" + per*2 + " " + Convert.ToDouble(h2.Text) +
")");
        sw.WriteLine("");
        sw.WriteLine("Розмір скляних полиць:");
        sw.WriteLine("");
        sw.WriteLine("26. Розмір для лівої верхньої тумби:-----
-----1шт.----(" + (Convert.ToDouble(g1.Text) -
Convert.ToDouble(tovpl.Text)) + " " + (per - Convert.ToDouble(tovpl.Text) -
Convert.ToDouble(tovpl.Text)) + ")");
        sw.WriteLine("27. Розмір для правої верхньої тумби:-----
-----2шт.----(" + (Convert.ToDouble(g1.Text) -
Convert.ToDouble(tovpl.Text)) + " " + (per - Convert.ToDouble(tovpl.Text) -
Convert.ToDouble(tovpl.Text)/2) + ")");
        sw.WriteLine("");

        sw.Close();
    } } }

```