

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

ІНЖЕНЕРНО-ХІМІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА «АВТОМАТИЗАЦІЯ ХІМІЧНИХ ВИРОБНИЦТВ»

**МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ
МОДУЛЬНОЇ КОНТРОЛЬНОЇ РОБОТИ
З КРЕДИТНОГО МОДУЛЯ «ВІЗУАЛЬНЕ
ПРОГРАМУВАННЯ ПРИКЛАДНИХ ПРОГРАМ »
КУРСУ “ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ – 2”
для студентів напрямку підготовки “6.050202 – Автоматизація та
комп’ютерно-інтегровані технології ”**



Київ
НТУУ «КПІ»
2015

Методичні вказівки до виконання модульної контрольної роботи з кредитного модуля «Візуальне програмування прикладних програм» курсу “Прикладне програмне забезпечення – 2” для студентів напрямку підготовки “6.050202-Автоматизація та комп’ютерно-інтегровані технології” [Текст] / Укладач, В. М. Ковалевський, // – К.: НТУУ «КПІ», 2015. – 105 с.

*Гриф надано Методичною і Вченою радою ІХФ «КПІ»
(Протокол № 4 від 27 квітня 2015 р.)*

Навчальне видання

**МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ
МОДУЛЬНОЇ КОНТРОЛЬНОЇ РОБОТИ
З КРЕДИТНОГО МОДУЛЯ «ВІЗУАЛЬНЕ
ПРОГРАМУВАННЯ ПРИКЛАДНИХ ПРОГРАМ»
КУРСУ “ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ – 2”
для студентів напрямку підготовки “6.050202 – Автоматизація та
комп’ютерно-інтегровані технології”**

Укладач: *Ковалевський Валерій Михайлович*, канд. техн. наук, доцент.

Відповідальний

за випуск: *А. І. Жученко*, док. техн. наук, професор.

Рецензент: *В. І. Сівецький*, канд. техн. наук, професор.

Авторська редакція

© НТУУ «КПІ», 2015 рік

ЗМІСТ

	стор.
1. Загальні зведення до виконання модульної контрольної роботи курсу «Прикладне програмне забезпечення – 2»	5
2. Постановка завдання до виконання модульної контрольної роботи кредитного модуля «Візуальне програмування прикладних програм»	6
3. Приклад розробки завдань до модульної контрольної роботи № 2	11
3.1 Технологічна схема та опис процесів з виробництва хлорметанів	11
3.2 Приклад з розробки матеріалів до прикладної C++ програми модульної контрольної роботи № 2	19
3.2.1 Мнемосхема виробництва хлорметанів та текст опису технологічного процесу	19
3.2.2 Схема конструкції і текст опису процесу в технологічному апараті “Хлоратор “	24
3.2.3 Схема конструкції і текст опису процесу в технологічному апараті “Абсорбер “	27
4. Приклади з техніки створення і програмування графічного меню команд прикладної C++ програми	29
4.1 Приклад № 1 з розробки графічного меню команд до Прикладної C++ програми	29
4.1.1 Програмування обробки подій маніпулятора “мишка“ у графічному меню команд “Технологія “	48
4.1.2 Програмування обробки подій маніпулятора “мишка“ у графічному меню команд “Апарати “	53
4.1.3 Програмування обробки подій маніпулятора “мишка“ у графічному меню команд “Інформація “	63

4.1.4 Програмування обробки подій маніпулятора “мишка“ у графічному меню команд “Вихід “	67
4.2 Приклад № 2 з розробки графічного меню команд до прикладної С++ програми	67
4.2.1 Програмування змінювання рисунків кнопок у графічному меню команд С++ програми	84
4.2.2 Програмування функцій для компонент “Image “ з графічного меню команд С++ програми	88
5. Література	103
<u>Додатки:</u>	
Додаток Д1. Схема апарату “Розріджувач лугу “	104
Додаток Д2. Приклад опису технологічного процесу апарату “Розріджувач лугу “	105

1. ЗАГАЛЬНІ ЗВЕДЕННЯ ДО ВИКОНАННЯ МОДУЛЬНОЇ КОНТРОЛЬНОЇ РОБОТИ КУРСУ “ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ-2”

Навчання бакалаврів з напрямку підготовки 6.050202 “Автоматизація та комп’ютерно-інтегровані технології ” передбачає в навчальному курсі “Прикладне програмне забезпечення - 2 ” вивчення і отримання знань з кредитного модуля “Візуальне програмування прикладних програм ”. Даний кредитний модуль відносяться до циклу професійної та практичної підготовки бакалаврів, де передбачено продовження вивчення сучасної технології програмування прикладних інженерно-технічних задач. Навчальний матеріал кредитного модуля “Візуальне програмування прикладних програм ” відрізняється тим, що вивчається сучасна технологія і техніка застосування об’єктно-орієнтованого та візуального програмування прикладних програм на основі використання інтегрованого середовища C++ Builder. Модульна контрольна робота № 2 з візуального програмування передбачає виконання завдань з розробки прикладної C++ програми, у якій на формах вікон відображаються: мнемосхема технологічного процесу з хімічного виробництва; схеми до конструкцій основних апаратів та описи їх технологічних процесів; графіки зміни у часі параметрів процесу у технологічному апараті.

Знання студентів з кредитного модуля “Візуальне програмування прикладних програм ” на протязі навчального семестру контролюються по завданням, які виконуються до модульної контрольної роботи № 2 на тему «**Обробка подій маніпулятора “мишка” у графічному меню команд прикладної C++ програми “Мнемосхема технологічного процесу хімічного виробництва ”**». Модульна контрольна робота № 2 орієнтована на контроль наступних знань:

- методики підготовки графічних матеріалів до розробляємої прикладної C++ програми з візуальним графічним меню команд;

- правил з розробки і побудови алгоритму з використання графічних матеріалів у меню команд прикладної C++ програми;
- методики з візуального програмування і налагодження прикладної програми у середовищі C++ Builder згідно розробленого алгоритму до графічного меню команд;
- команд і функцій з обробки подій та визначення натиснутих клавіш і координат курсору маніпулятора “мишка” до дій користувача при роботі прикладної C++ програми.

Виконання модульної контрольної роботи № 2 з програмування прикладної програми в інтегрованому середовищі C++ Builder дає студентам відповідні практичні знання з самостійної розробки матеріалів до задачі програмування та знання по використанню інспектора об’єктів C++ Builder для налаштування віконних форм, параметрів до компонент та функцій для обробки команд у графічному меню програми.

2. ПОСТАНОВКА ЗАВДАННЯ ДО ВИКОНАННЯ МОДУЛЬНОЇ КОНТРОЛЬНОЇ РОБОТИ КРЕДИТНОГО МОДУЛЯ «ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ ПРИКЛАДНИХ ПРОГРАМ»

У інтегрованому середовищі C++ Builder необхідно розробити прикладну C++ програму з графічним меню команд, яка буде виводити в окремих вікнах на екран дисплею інформацію у вигляді мнемосхеми технологічного процесу та опису процесів завданого хімічного виробництва. Також C++ програма повинна у вікні виводити інформацію про властивості та призначення продукції хімічного виробництва. У графічному меню команд C++ програми повинна бути можливість переглянути:

- призначення технологічних апаратів на мнемосхемі процесів хімічного виробництва;

- описи технологічних процесів у визначених апаратах (2 апарата з мнемосхеми хімічного виробництва);

- графіки зміни технологічних параметрів у першого та другого технологічного апаратів, які використовуються у схемі хімічного виробництва. Також обов'язково потрібно розмістити інформацію про використану літературу з виконання контрольної роботи № 2 та створити меню “Про програму”.

Пункти меню команд прикладної С++ програми мають бути побудовані у вигляді графічного меню з обробкою подій до команд маніпулятора «мишка». Завдання з контрольної роботи № 2 (МКР-2) необхідно виконувати таким чином:

- підготовка графічних і текстових матеріалів до завданої схеми технологічного процесу хімічного виробництва;
- розробка графічних матеріалів для побудови графічного меню команд у основному вікні прикладної С++ програми;
- виконання візуального програмування у середовищі С++ Builder сформульованих задач та захист результатів з виконання завдань до модульної контрольної роботи № 2.

Перший етап виконання МКР-2. Робота студентів з задачі програмування (РСЗП) оцінюється відповідними балами по двох частинах виконаних завдань. Перша частина **РСЗП(Ч1)** з виконання завдань передбачає:

- розробку рисунка мнемосхеми до технологічного процесу завданого хімічного виробництва у програмі графічного редактора;
- підготовку в текстовому редакторі **Word** опису схеми технологічного процесу до завданого хімічного виробництва.

Друга частина **РСЗП(Ч2)** завдань передбачає аналіз роботи завданих технологічних апаратів хімічного виробництва та виконання:

- розробки схеми до конструкції технологічного апарату та відповідного опису технологічного процесу ([назва апарату № 1](#));

- розробки схеми до конструкції технологічного апарату та відповідного опису технологічного процесу (назва апарату № 2).

Виконані завдання з частин РСЗП(Ч1) та РСЗП(Ч2) оформлюються у вигляді для подальшого включення у звіт з МКР-2.

Розроблені матеріали завдань з частин РСЗП(Ч1) та РСЗП(Ч2) студенти використовують далі при виконанні програмування прикладної С++ програми до МКР-2.

Розроблена прикладна С++ програма з графічним меню команд подається до захисту з наступними матеріалами:

- виконавчого файлу (*.exe) прикладної С++ програми з графічним основним меню команд, побудованим на основі графічних компонент;
- файлів проекту **ProjectМКР2.bpr** до прикладної С++ програми та програмних модулів до віконних форм;
- записки з рисунками схем, описами процесів, блок-схемами алгоритмів та листингами С++ програми, яка розроблена до модульної контрольної роботи № 2. У записці також показуються зображення вікон з екрану дисплея по результатах роботи прикладної С++ програми .

Для виконання модульної контрольної роботи № 2 студентам видаються індивідуальні завдання у вигляді технологічної схеми до відповідного хімічного виробництва. Якщо користувач у графічному меню команд С++ програми обирає команду “Технологія ” (рис. 2-1) і потім виконує команду “Мнемосхема ”, то в цьому випадку заставка у прикладної С++ програми замінюється на рисунок мнемосхеми технологічного процесу хімічного виробництва, який стає далі фоном основного вікна і знизу з’являються підказки назв до технологічних апаратів і обладнання, на яких встановлюється показник (стрілка) маніпулятора “мишка”.

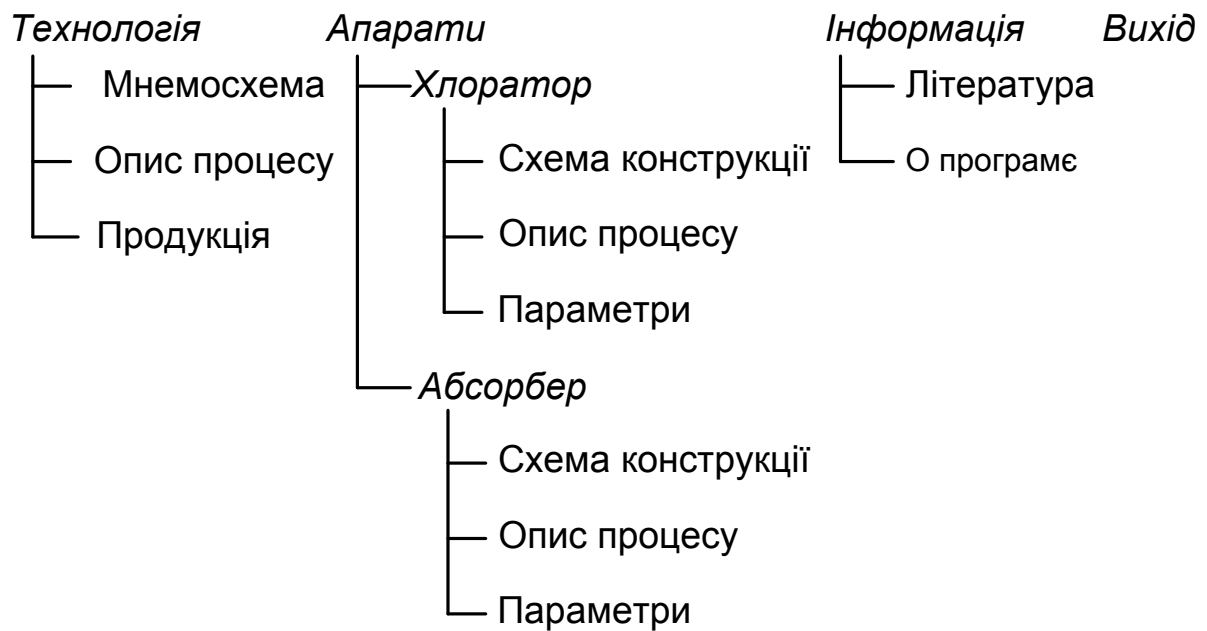


Рис. 2-1. Структура команд і підкоманд у графічному меню команд прикладної С++ програми у МКР-2.

Коли користувач у графічному меню С++ програми вибирає такі команди: «Апарати /Апарат (назва апарату)/ Схема конструкції», або «Апарати /Апарат (назва апарату)/ / Опис процесу», або «Апарати /Апарат (назва апарату)/ / Параметри», тоді інформація до цих команд повинна показуватись в окремому вікні. При виконанні у меню “Апарати ” команди “Апарат (назва апарату)/ / Параметри ” повинні показуватися в окремому вікні графіки зміни параметрів процесу до вибраного технологічного апарату. Для побудови даних графіків необхідно у С++ Builder використовувати компоненту **TChart**.

Після запуску виконавчого файлу *.exe з МКР-2 на екрані дисплея повинно з’явитися основне вікно С++ програми з такою заставкою:

МОДУЛЬНА КОНТРОЛЬНА РОБОТА КРЕДИТНОГО МОДУЛЯ
«ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ ПРИКЛАДНИХ ПРОГРАМ»

НА ТЕМУ: ОБРОБКА ПОДІЙ МАНІПУЛЯТОРА “МИШКА”
У ГРАФІЧНОМУ МЕНЮ КОМАНД ПРИКЛАДНОЇ
C++ ПРОГРАМИ “МНЕМОСХЕМА ТЕХНОЛОГІЧНОГО
ПРОЦЕСУ ХІМІЧНОГО ВИРОБНИЦТВА”

Розробник C++ програми студент(ка)

П. І. Б.

ІХФ гр. ЛА-№ 1 або ЛА-№ 2

Київ-КПІ 20XX р.

Матеріали з результатів програмування прикладної C++ програми до модульної контрольної роботи № 2 для захисту оформлюються у вигляді записки з наступним змістом:

Зміст	стор.
1. Завдання до контрольної роботи	2
2. Основне графічне меню команд прикладної програми “Мнемосхема технологічного процесу (назва хімічного виробництва)”	3
2.1 Інформація для показу по командах графічного меню прикладної C++ програми з модульної контрольної роботи № 2 (рисунок основного графічного меню команд C++ програми та відповідна інформація до підкоманд)	6
3. Алгоритм обробки подій маніпулятора “мишка” при перегляду структури апаратів на зображенні мнемосхеми технологічної схеми хімічного виробництва (блок-схема алгоритму на	

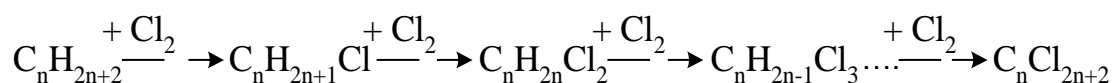
аркушу формату А3)	15
4. Структура віконних форм у C++ програми і алгоритми їх відкриття та закриття (блок-схема алгоритму на аркушу формату А3)	17
5. Листінги програмних модулів прикладної C++ програми	19
6. Результати роботи C++ програми з МКР № 2	25
7. Література	27
8. Додатки:	
7.1 Виконавчий файл прикладної C++ програми і файли проекту з МКР № 2	27
7.2 Рисунок заставки до прикладної C++ програми.....	28

3. ПРИКЛАД РОЗРОБКИ ЗАВДАНЬ ДО МОДУЛЬНОЇ КОНТРОЛЬНОЇ РОБОТИ № 2

Методика і приклад виконання завдань до модульної контрольної роботи № 2 далі буде розглядатися у вигляді необхідних та відповідних дій для підготовки схем апаратів та текстів з описами процесів для візуального програмування прикладної C++ програми з мнемосхемою технологічного процесу виробництва хлорметанів.

3.1 Технологічна схема та опис процесів з виробництва хлорметанів

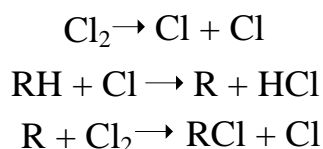
При хлоруванні парафінових вуглеводнів у реакції атоми хлору заміщають атоми водню, які відщеплюються, утворюючи хлористий водень. При цьому можна одержати моно-, ди-, три- і поліхлороподібні відповідно до таких реакцій:



Процес такого заміщення був відкритий французьким ученим Дюма і називається металепсією. При безпосередній взаємодії хлору з парафіновими вуглеводнями (або з боковими ланцюгами алкіл бензолів) молекула хлору насамперед дисоціює на атоми, які потім взаємодіють з вуглеводнем і при цьому утворюється вільний радикал, реагуючи з іншою молекулою хлору, утворює хлорпохідне вуглеводню, а один атом хлору звільняється.

Дисоціацію хлору на атоми можна спричинити тепловою або променистою енергією. Практично хлорування проводять у газовій та рідкій фазах з застосуванням каталізаторів або без них. Рідко фазне хлорування може здійснюватися також у розчині, в середовищі розплавлених солей.

Таким чином, хлорування парафінових вуглеводнів відбувається за ланцюговою реакцією:

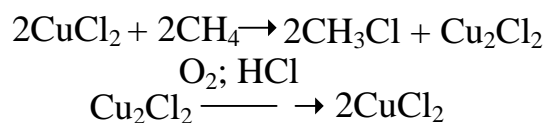


Фотохімічний метод хлорування парафінових вуглеводнів не набув значного поширення в промисловості через високу вартість потрібного устаткування. Основним способом є термічне хлорування в газовій фазі при температурі, достатній для помітної дисоціації хлору (понад 250-300 градусів). Реакція хлорування проходить з виділенням тепла, яке використовують для підігрівання реагентів. Реакція термічного хлорування відбувається з великою швидкістю. Температурний режим процесу підтримується за рахунок фізичного тепла надлишкового вуглеводню. Щоб уникнути місцевих перегрівів, що призводять до розщеплення молекул парафінів необхідне старання змішування хлору з вуглеводнем і для цього хлор вводять у реакційну зону через сопла, розміщені в напрямку потоку вуглеводню.

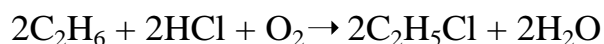
Каталітичне хлорування парафінових вуглеводнів відбувається звичайно при нижчих температурах, ніж термічне хлорування, тому що каталізатори

прискорюють утворення дихлорпохідних і продуктів за рахунок більш повного заміщення атомів водню хлором. В якості каталізаторів застосовують хлориди міді, сурми, олова, кремнію, йоду, сірки, які нанесені на високопористі матеріали (активне вугілля, пемза, силікагель та ін.). Досить активним каталізатором є, наприклад, хлорна мідь CuCl_2 , нанесена на високопористий матеріал.

Хлорпохідні вуглеводнів утворюються при взаємодії вуглеводнів не тільки з елементарним хлором, але і з деякими його сполуками, наприклад, з хлористим сульфурилом SO_2Cl_2 , фосгеном COCl_2 і т.д.. Так, газоподібні вуглеводні можна хлорувати при 400 градусах хлорною міддю, яка в тонкоподрібненому стані суспендується в газі. Хлористу мідь, яка при цьому утворюється, окислюють повітрям при 325 - 400 градусах і обробляють хлористим воднем:

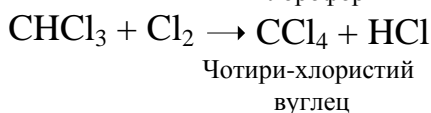
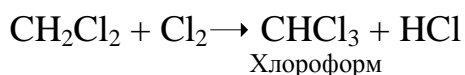
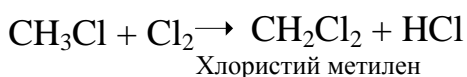
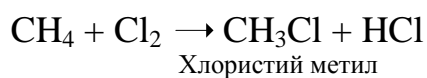


Регеновану хлорну мідь знову використовують для хлорування. Великий практичний інтерес становить метод окислювального хлорування вуглеводнів хлористим воднем, наприклад, одержання хлорпохідних пропусканням суміші вуглеводню, хлористого водню і кисню над каталізатором:



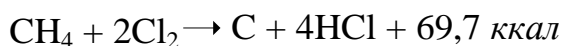
Використовуючи для цього процесу хлористий водень, який є побічним продуктом хлорування парафінових вуглеводнів газоподібним хлором, можна було б значно підвищити економічність виробництва аліфатичних хлорорганічних сполук. Такі процеси добре розроблені для ароматичного ряду сполук. При вивченні процесів хлорування парафінових вуглеводнів було відкрито ряд закономірностей, які справедливі для більшості реакцій. Встановлено, що коли температура хлорування нижча за температуру розщеплення хлорованого вуглеводню, хлорпохідне, яке при цьому утворюється, має стільки ж

вуглецевих атомів, скільки вихідний вуглеводень. При більш високих температурах хлорування і надлишку хлору молекула вуглеводню розщеплюється на радикали (CH_3^* , C_2H_5^*), які потім перетворюються в полі хлорпохідні CCl_4 , C_2Cl_6 . Хлорування з одночасним розщепленням молекули вуглеводню називається деструктивним хлоруванням. При помірних температурах швидкість заміщення на хлор атомів водню у третинних вуглецевих атомів більша, ніж у вторинних, а у вторинних – більша ніж у первинних. З підвищенням температури хлорування ця різниця зменшується і при 600 градусах заміщення відбувається з однаковою швидкістю. При підвищенні тиску в процесі хлорування вуглеводнів у газуватій фазі збільшується відносна швидкість заміщення водню при первинному вуглецевому атомі. При безпосередньому хлоруванні потоку метану не вдається одержати яке-небудь одне індивідуальне хлорпохідне, тому що практично відповідно до хімічних реакцій завжди одержується суміш всіх чотирьох хлорпохідних:



Створюючи певні умови процесу, можна спрямувати реакцію в бік утворення переважного потрібного хлорпохідного. Процес здійснюється при великому молярному надлишку метану і порівняно при високій температурі (400 °C і вище), якщо потрібно одержати головним чином хлористий метил CH_3Cl , а для переважного одержання чотири хлористого вуглецю потрібно надлишок хлору.

При хлоруванні метану, як і інших вуглеводнів, слід уникати значного підвищення температури. При температурі вище 500 - 550 градусів може статись вибух з виділенням вуглецю і хлористого водню:



Тому необхідно уникати місцевих перегрівів реагуючих газів. Виходи різних хлорпохідних метану залежно від умов процесу наведені у таблиці 3.1. Хлористий метил (монохлорметан) можна одержати хлоруванням метану при 400-450 градусах в присутності каталізатора (хлориди металів, осаджені на пемзі) при десятикратному молярному надлишку метану. В цих умовах приблизно 80-85 % хлору, який надходить на хлорування, витрачається на утворення хлористого метилу.

Хлористий метил являє собою безбарвний газ, з запахом ефіру, горить безбарвним полум'ям; температура кипіння - 23,7 градусів; температура замерзання - 97,6 градусів; питома вага хлористого метилу при температурі кипіння дорівнює 0,992 г/см³; прихована теплота випаровування 430,5 кдж/кг. В 100 г води розчиняється 0,74 г CH₃Cl. Хлористий метил застосовується як холодоагент у холодильних установках. В промисловості органічного синтезу він використовується, як метилуючий засіб. При пропусканні хлористого метилу з водяною парою над вапном при 300-350 градусах можна одержати з хорошим виходом метиловий спирт.

Таблиця 3.1

Виходи хлорпохідних метану					
Температура, °С	Молярне відношення Cl ₂ : CH ₄	Склад продуктів реакції, % мол.			
		CH ₃ Cl	CH ₂ Cl ₂	CHCl ₃	CCl ₄
440	1:2	62	30	7	1
440	1,1:1	37	41	19	3
440	1,68:1	19	43	33	4
440	1,98:1	11	35	45	9
440	2,28:1	5	29	52	14
440	3,02:1	3	15	53	29
440	3,31:1	—	6	43	51
460	3,88:1	—	—	4	96

Хлористий метилен (метиленхлорид, дихлорметан) CH₂Cl₂ одержують з хорошим виходом хлорування суміші метану та хлористого метилу. Хлори-

стий метил і частина метану хлорують до метиленхлориду, а інша частина метану – до хлористого метилу. З продуктів реакції добувають хлористий метилен, а газ, що лишився (метан + хлористий метил), додають до свіжого газу, який містить метан, нагрівають суміш до 130 градусів і подають на змішування з хлором. Одержана суміш містить близько 20 % об'ємного хлору, 60 % метану (відношення метану до хлору 3:1), 6 % хлористого метилу і 0,1-0,2 % хлористого метилу, решта – азот, окис і двоокис вуглецю. Ця суміш газів і парів при 100 градусах надходить у хлоратор, де здійснюється хлорування при температурі близько 500 градусів.

Хлоратор являє собою вертикальний хімічний реактор 4 на рис. 3.1 у вигляді циліндричного апарату з сталевим корпусом, який футерований зсередини двома шарами діабазових плиток і шаром шамотної цегли. Ззовні хлоратор ізольований шаром азбесту. Перед введенням у хлоратор реакційних газів в топці спалюють горючий газ і продукти згоряння нагрівають футеровку апарату і цегляну насадку, які акумулюють тепло. Після досягнення потрібної температури в апарат 4 через верхній штуцер вводять суміш вуглеводнів і хлору. Нагрівання газів потрібне тільки для ініціювання реакції, а далі екзотермічна реакція хлорування відбувається автотермічно. Продукти реакції відводяться з нижньої частини хлоратора. В процесі хлорування незначна частина метану зазнає глибоких перетворень, при цьому утворюються смолисті продукти і сажа. Щоб запобігти потраплянню цих речовин у відповідний газопровід, в кільцевому просторі хлоратора укладається шар керамічних кілець висотою близько 200 мм.

Періодично, в міру нагромадження в хлораторі сажі та смолистих речовин, їх випалюють. Для цього припиняють подачу реакційних газів у хлоратор, запалюють у топці горючий газ і подають продукти його згоряння разом з надлишковим повітрям в камеру хлорування.

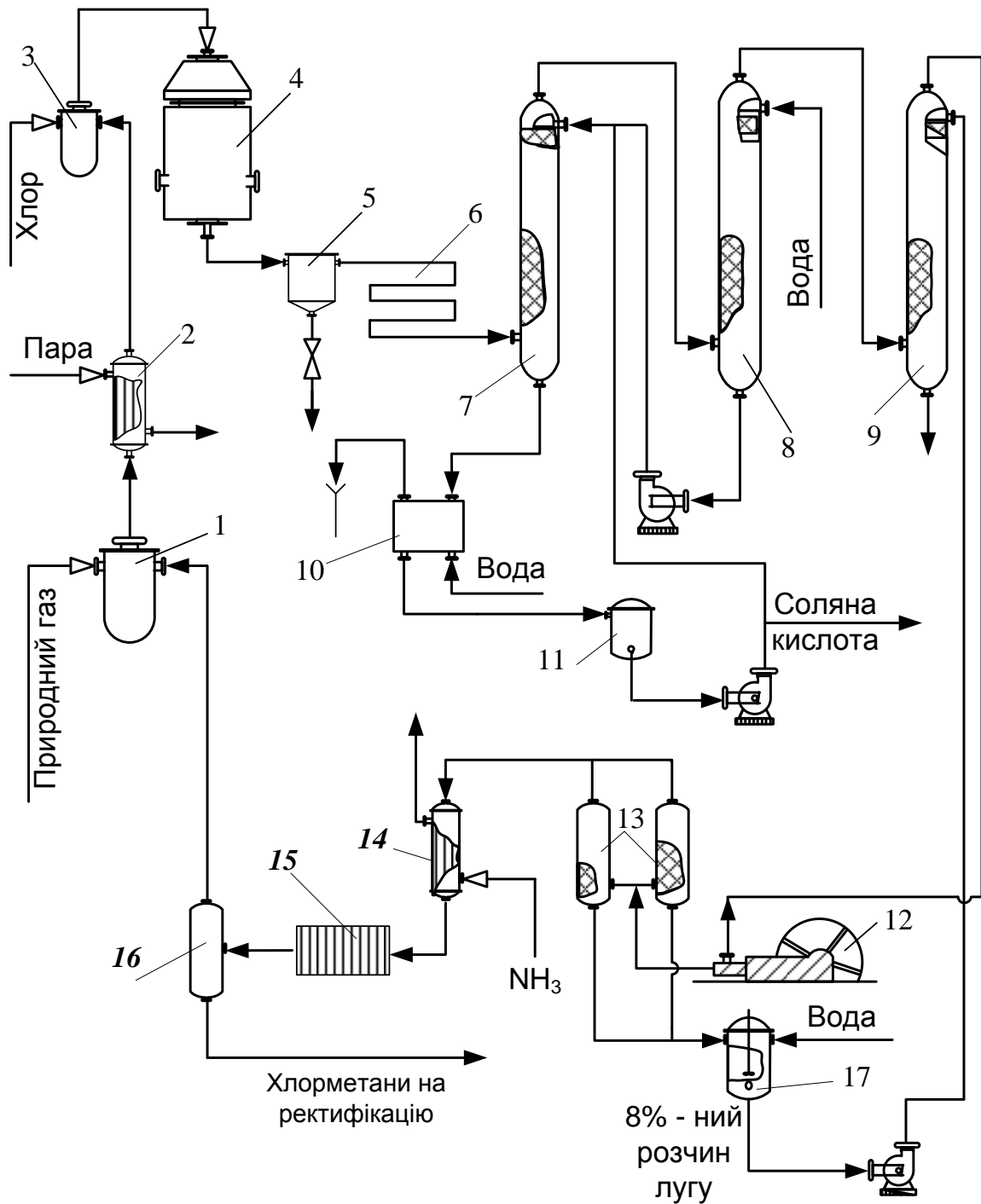


Рис. 3.1. Схема технологічного процесу “Виробництво хлорметанів” до прикладу з виконання модульної контрольної роботи № 2: 1, 3 – змішувачі; 2 – підігрівник; 4 – хлоратор; 5 – сажовловлювач; 6 – повітряний холодильник; 7, 8 – абсорбери; 9 – колона нейтралізаційна; 10 – графітовий холодильник; 11 – збірник концентрованої соляної кислоти; 12 – компресор; 13 – осушувач; 14 – конденсатор; 15 – фільтр; 16 – сепаратор; 17 – розріджувач лугу.

Після видалення речовин, що містять вуглець, вогонь у топці гасять і в хлоратор знову подають реакційні гази. Продукти хлорування відводять з нижньої частини реактора 4, пропускають через сажовловлювач 5 і охолоджують у повітряному холодильнику 6 до 100 градусів.

У газоподібних продуктах хлорування міститься приблизно 55-57 % метану, що не прореагував, 17-18 % хлористого водню, 9-10 % хлористого метилу, 5-6 % метиленхлориду і близько 1,5 % вищих хлоридів, а решта це азот, окис і двоокис вуглецю.

З потоку газоподібних продуктів реакції насамперед добувають хлористий водень. Для цього використовуються три колонних апарати, з яких два перших (за ходом газів) апарати 7 і 8 є абсорберами, а апарат 9 здійснює нейтралізацію слідів хлористого водню. Абсорбер 8 зрошують розбавленою соляною кислотою. Концентрована соляна кислота, що витікає з абсорбера, охолоджується до 20-30 градусів у графітовому холодильнику 10. Гази, які відходять з абсорбера 7, подаються у другий абсорбер 8 для добування з них залишків хлористого водню, а розведена соляна кислота, що утворюється, надходить на зрошення абсорбера 7.

Для нейтралізації слідів хлористого водню гази промивають в колоні 9 холодним 8 % водним розчином лугу. Одночасно у процесі нейтралізації гази охолоджуються.

Для видалення хлорметанів гази, відділені від хлористого водню, стискають в одноступінчастому компресорі 12 до 2 – 8 атм. і пропускають в осушувачах 13 через твердий луг, який відбирає з газів вологу. Потім гази охолоджуються в кожухотрубному конденсаторі 14 аміаком, що випаровується при $-53\text{ }^{\circ}\text{C}$ і при цьому відбуваються виморожування вологи та конденсація хлорметанів. Після відокремлення льоду на фільтрі 15 і рідких хлорметанів у сепараторі 16 гази повертають на хлорування. Рідка суміш хлорметанів має приблизно такий склад (в % ваг.):

Хлористий метил 28 – 32; Хлороформ 12 – 14;
Метиленхлорид 50 – 53; Чотирихлористий вуглець ... 3 – 5.

З цієї суміші ректифікацією одержують індивідуальні сполуки. Хлористий метил повертають на хлорування (повністю або частково).

Хлористий метилен являє собою безбарвну рідину з запахом хлороформу; він кипить при 40,1 °С, замерзає при – 96,6 °С, пит. вага 1,335 г/см³(при 15 °С); прихована теплота випаровування 78,7 ккал/кг.

Метиленхлорид добре розчиняє ефіри целюлози, жири, масла і каучук, важко гідролізується, негорючий, його пари не утворюють вибухових сумішей з повітрям. Тому він є цінним промисловим розчинником і може використовуватись також для очистки мастил від парафіну.

3.2 Приклад з розробки матеріалів до прикладної С++ програми модульної контрольної роботи № 2

За допомогою матеріалів опису технологічної схеми процесів з виробництва хлорметанів, наведених у пункті 3.1, виконаємо підготовку інформації, яка далі буде запрограмована у прикладній С++ програмі для модульної контрольної роботи № 2.

3.2.1 Мнемосхема виробництва хлорметанів та текст опису технологічного процесу

Схема технологічного процесу (рис. 3.1) використовується далі для розробки навчального прикладу у вигляді С++ програми *“Мнемосхема технологічного процесу з виробництва хлорметанів”*. Відповідно до 2 пункту даних методичних вказівок зміст матеріалів РСЗП(Ч1) передбачає розробку наступних завдань:

- Завдання № 1. Створення файлу (.doc) з текстом опису технологічного процесу виробництва хлорметанів відповідно до схеми рис. 3.1;

- Завдання № 2. Виготовлення у графічному редакторі рисунка процесу виробництва хлорметанів у вигляді мнемосхеми відповідно до завданої схеми рис. 3.1.
- Завдання № 3. Створення файлу (.doc) з текстом опису властивостей хімічної продукції – суміші хлорметанів.

Виконання завдання № 1. На основі тексту з опису процесів переробки углеводів у процесі виробництва хлорметанів методом хлорування метану, наведених у пункті 3.1 до схеми рисунка 3.1, формуємо у текстовому редакторі **Word** файл відповідного тексту для показу у вікні **Form2** при виборі у меню “Технологія ” команди “Опис схеми ”. Приклад опису схеми з виробництва хлорметанів до МКР № 2 може мати такий текст:

Опис технологічного процесу з виробництва хлорметанів

Перед введенням у хлоратор реакційних газів в топці спалюють горючий газ і продукти згоряння нагрівають футеровку апарату і цегляну насадку, які акумулюють тепло. Суміш газів після змішувачів 1 і 3 містить близько 20 % об’ємного хлору, 60 % метану (відношення метану до хлору 3:1), 6 % хлористого метилу і 0,1-0,2 % хлористого метилену, решта – азот, окис і двоокис вуглецю. Ця суміш газів і парів при 100 градусах надходить у реактор (хлоратор), де проводять хлорування при температурі близько 500 градусів. Після досягнення потрібної температури в хлоратор через верхній штуцер вводять суміш вуглеводнів і хлору. Нагрівання потрібне тільки для ініціювання реакції. Далі екзотермічна реакція хлорування відбувається автотермічно. Продукти реакції відводяться з нижньої частини хлоратора.

В процесі хлорування незначна частина метану зазнає глибоких перетворень, при цьому утворюються смолисті продукти і сажа. Щоб запобігти потраплянню цих речовин у відвідний газопровід, в кільцевому просторі хлоратора укладається шар керамічних кілець висотою у 200 мм.

Періодично, в міру нагромадження в хлораторі сажі та смолистих речовин, їх випалюють. Для цього припиняють подачу в хлоратор реакційних газів, запалюють у топці горючий газ і подають продукти його згоряння разом з надлишковим повітрям в камеру хлорування.

Після видалення речовин, що містять вуглець, вогонь у топці гасять і в хлоратор знову подають реакційні гази. Продукти хлорування відводять з нижньої частини реактора 4, пропускають через сажовловлювач 5 і охоло-

джують у повітряному холодильнику 6 до 100 градусів. У газоподібних продуктах хлорування міститься приблизно 55-57 % метану, що не прореагував, 17-18 % хлористого водню, 9-10 % хлористого метилу, 5-6 % метиленхлориду і близько 1,5 % вищих хлоридів, решта-азот, окис і двоокис вуглецю.

З газоподібних продуктів реакції насамперед добувають хлористий водень. Для цього встановлюють три колонних апарати, з яких два перших (за ходом газів) апарати 7 і 8 є абсорберами, а в апараті 9 здійснюється нейтралізація слідів хлористого водню. Абсорбер 8 зрошують розбавленою соляною кислотою. Концентрована соляна кислота, що витікає з абсорбера, охолоджується до 20-30 градусів у графітовому холодильнику 10. Гази, які відходять з абсорбера 7, подають у другий абсорбер 8 для добування з них залишків хлористого водню, а розведена соляна кислота, що утворюється, надходить на зрошення абсорбера 7. Для нейтралізації слідів хлористого водню газу промивають в колоні 9 холодним 8 % водним розчином лугу. Одночасно з нейтралізацією потік газів охолоджується.

Для видалення хлорметанів газу, відмиті від хлористого водню, стискають в одноступінчастому компресорі 12 до 2 – 8 атм. і пропускають в осушувачах 13 через твердий луг, який відбирає з газів вологу. Потім газу охолоджують в кожухотрубному конденсаторі 14 аміаком, що випаровується при - 53°; при цьому відбуваються виморожування вологи і конденсація хлорметанів. Після відокремлення льоду на фільтрі 15 і рідких хлорметанів у сепараторі 16 газу повертають на хлорування.

З цієї суміші ректифікацією одержують індивідуальні сполуки. Хлористий метил повертають на хлорування (повністю або частково). Рідка суміш хлорметанів має приблизно такий склад (в % ваг.):

Хлористий метил	28 – 32;	Хлороформ ///.....	12 – 14;
Метиленхлорид	50 – 53;	Чотирихлористий вуглець	3 – 5.

Виделений вище в рамці текст буде описом процесів у виробництві хлорметанів і зберігаємо у файл з назвою **Scheme.rtf** та розміщуємо з доступом у каталогах МКР2MainMenu \ Хлорування метану \ Data\Main \ для розглядаемого навчального прикладу з модульної контрольної роботи № 2.

Виконання завдання № 2. За допомогою технологічної схеми (рис. 3.1) з виробництва хлористого метилену у графічному редакторі робимо рисунок у вигляді мнемосхеми, який показано на рис. 3.2.

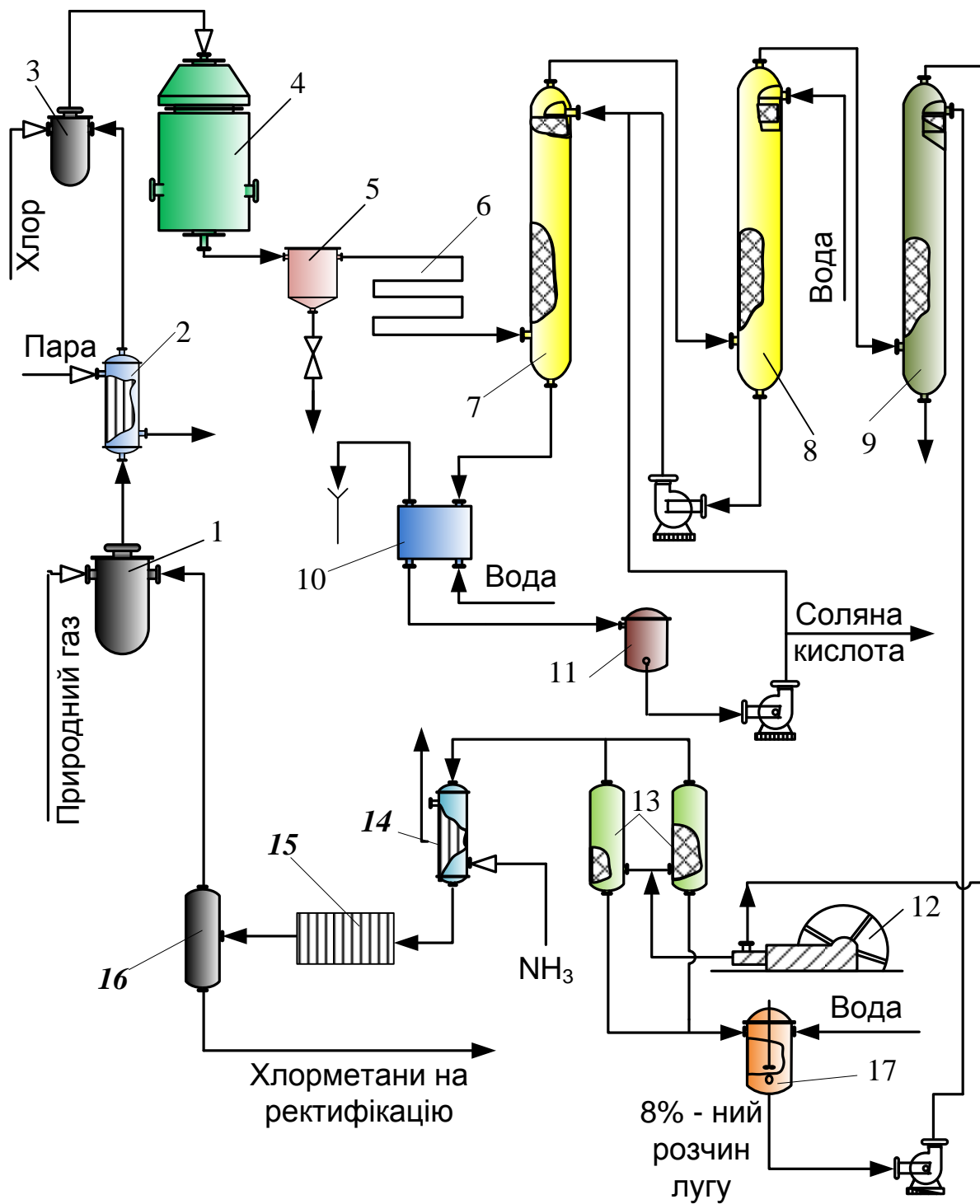


Рис. 3.2. Мнемосхема технологічного процесу з виробництва хлорметанів.

При виконанні у програмі C++ команди “Схема” з меню “Технологія” буде з’являтися рис. 3.2 у вікні основної форми **Form1** і далі цей рисунок буде залишатися, як загальний фон основного вікна. При установці курсора миш-

ки на якійсь апарат, зображений на мнемосхемі виробництва хлорметанів, у вікні відразу ж буде з'являтися підказка у вигляді назви технологічного апарату.

Мнемосхему виробництва хлорметанів (рис. 3.2) зберігаємо у файл з назвою **Scheme.bmp** і розміщуємо з доступом у папках МКР2MainMenu \ Хлорування метану \ Data \ Main \ для розглядаємого навчального прикладу з модульної контрольної роботи № 2.

Виконання завдання № 3. На основі тексту з опису процесів хлорування метану з отриманням суміші хлорметанів, наведених у пункті 3.1, формуємо файл у текстовому редакторі **Word**, наприклад, з таким текстом опису продукції для показу у вікні **Form2** при виборі команди “Продукція” у меню “Технологія”:

Хлорметани, як хімічна продукція хлорування метану

Практично не розчинний у воді (2 г/л) дихлорметан змішується з більшістю органічних розчинників. Легколетучий (40 °С), утворює азеотропну суміш із водою (т.стосів. 38,1 °С, 98,5 % М.). Метиленхлорид (дихлорметан) реагує із хлором з утворенням хлороформу і чотири хлористого вуглецю. З йодом при 200 °С дає CH_2Cl_2 , із бромом при 25-30 °С у присутності алюмінію дає бромхлорметан. При нагріванні з водою гідролізується до CH_2 і HCl . При нагріванні зі спиртовим розчином NH_3 до 100-125 °С утворюється гексаметилентетрамин. Реакція з водяним розчином NH_3 при 200 °С приводить до метиламіну, мурашиної кислоти і HCl . З ароматичними з'єднаннями в присутності AlCl_3 метиленхлорид вступає в реакцію Фріделя - Крафтса, наприклад, з бензолом утворює дифенілметан. Дешевина, висока здатність розчиняти багато органічних речовин, легкість видалення, відносно мала токсичність привела до широкого застосування його як розчинник для проведення реакцій, екстракцій у тому числі і у лабораторіях. Використовують у сумішах для зняття лаку, знежирення поверхонь. У харчовій промисловості використовують для готування екстракту хмелю і інших харчових препаратів. Висока летючість дозволяє використовувати для зпінення поліуретанів і в хроматографії.

Він же використовується для склеювання пластиків: полістирол, полікарбонат, поліетилентерефталат, Абс-Пластик, але не для поліетилену і поліпропілену.

У промисловості в суміші з полімерами використовують для створення форми, яка швидко застигає за рахунок випару дихлорметану.

Виделений вище текст в рамці буде описом властивостей хлорметанів, як продукції хімічного виробництва. Зберігаємо цей текст у файл з назвою **Products.rtf** з доступом у каталогах МКР2MainMenu \ Хлорування метану \ Data \ Main\ для розглядаємого навчального прикладу з модульної контрольної роботи № 2.

3.2.2 Схема конструкції і текст опису процесу в технологічному апараті “Хлоратор”

Відповідно до пункту 2 даних методичних вказівок зміст матеріалів з другої частини завдань РСЗП(Ч2) передбачає розробку наступних завдань:

- Завдання № 1. Створення файлу (.doc) з текстом опису технологічного процесу в апараті “Хлоратор”;
- Завдання № 2. Виготовлення у графічному редакторі рисунка схеми з конструкції технологічного апарату “Хлоратор”;
- Завдання № 3. Створення файлу (.doc) з текстом опису технологічного процесу в апараті “Абсорбер”;
- Завдання № 4. Виготовлення у графічному редакторі рисунка схеми з конструкції технологічного апарату “Абсорбер”.

Виконання завдання № 1 до РСЗП(Ч2). За допомогою текстового редактора **Word** виготовляємо, наприклад, наступний текст з опису технологічного процесу в апараті “Хлоратор” і зберігаємо у файл для вивіду у відповідному вікні форми при виборі команди “Опис процесу” у підменю “Хлоратор” з меню “Апарати”:

Опис технологічного процесу в хлораторі

Потік суміши у хлоратор містить близько 20 % об’ємного хлору, 60 % метану (відношення метану до хлору 3:1), 6 % хлористого метилу і 0,1-0,2 % хлористого метилену, решта – азот, окис і двоокис вуглецю. Ця суміш газів і парів при 100 градусах надходить у реактор (хлоратор), де здійснюють хло-

рування при температурі близько до 500 градусів.

Перед введенням реакційних газів у хлоратор в топці спалюють горючий газ і продукти згоряння нагрівають футеровку апарату і цегляну насадку, які акумулюють тепло. Після досягнення потрібної температури в апарат через верхній штуцер подається суміш вуглеводнів і хлору. Нагрівання потрібне тільки для ініціювання реакції, а далі екзотермічна реакція хлорування відбувається автотермічно. Продукти реакції відводяться з нижньої частини апарату. В процесі хлорування незначна частина метану зазнає глибоких перетворень, при цьому утворюються смолисті продукти і сажа. Щоб запобігти потраплянню цих речовин у відповідний газопровід, в кільцевому просторі хлоратора укладаються шар керамічних кілець висотою близько до 200 мм. Періодично, в міру нагромадження в хлораторі сажі та смолистих речовин, їх випалюють. Для цього припиняють подачу в хлоратор реакційних газів, запалюють у топці горючий газ і подають продукти його згоряння разом з надлишковим повітрям в камеру хлорування.

Після видалення речовин, що містять вуглець, у топці вогонь гасять і в хлоратор знову подають реакційні гази. Продукти хлорування відводять з нижньої частини реактора, пропускають через сажовловлювач і охолоджують у повітряному холодильнику до 100 градусів.

У газоподібних продуктах хлорування міститься приблизно 55-57 % об'ємно метану, що не прореагував, 17-18 % хлористого водню, 9-10 % хлористого метилу, 5-6 % метиленхлориду і близько 1,5 % вищих хлоридів, решта це азот, окис і двоокис вуглецю.

Виделений вище в рамці текст буде описом технологічного процесу в апараті “Хлоратор”. Зберегаємо цей текст у файл з назвою **Process.rtf** і розміщуємо з доступом у каталогах МКР2MainMenu \Хлорування метану \ Data \ Device1\ для розглядаємого навчального прикладу з модульної контрольної роботи № 2.

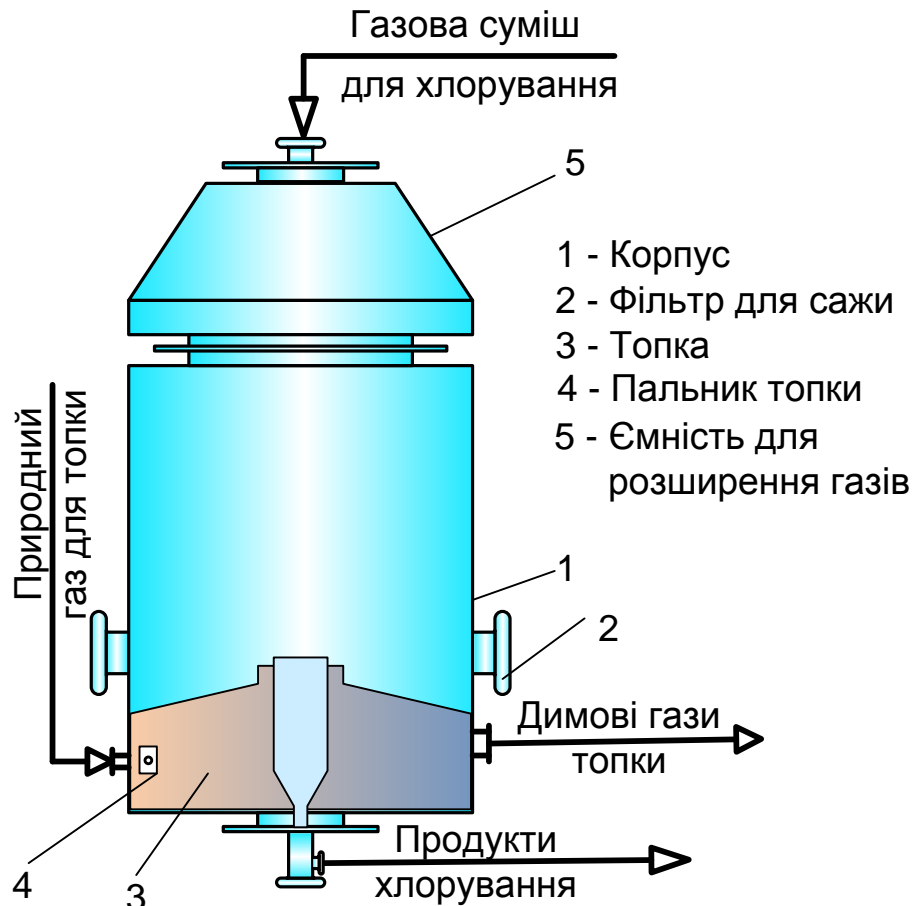


Рис. 3.2-2. Схема конструкції технологічного апарату “Хлоратор”.

Виконання завдання № 2 до РСЗП(Ч2). За допомогою графічного редактора робимо рисунок схеми з конструкції технологічного апарату “Хлоратор”, який показано на рис. 3.2-2 і при виконанні команди “Конструкція” у підменю “Хлоратор” з меню “Апарати” буде з’являтися у вікні відповідної форми C++ програми.

Схему конструкції технологічного апарату “Хлоратор” зберігаємо у файл з назвою **Device.bmp** і розміщуємо з доступом у каталогах МКРЗMainMenu \ Хлорування метану \ Data \ Device1\ для розглядаемого навчального прикладу з модульної контрольної роботи № 2.

3.2.3 Схеми конструкції і текст опису процесу в технологічному апараті “Абсорбер”

Виконання завдання № 3 до РСЗП(Ч2). За допомогою текстового редактора **Word** створюємо, наприклад, наступний текст з описом технологічного процесу в апараті “Абсорбер” і зберігаємо у файл для показу у вікні відповідної форми при виборі команди “Опис процесу” у підменю відповідної форми при виборі команди “Опис процесу” у підменю “Абсорбер” в меню “Апарати”:

Опис технологічного процесу в абсорбері

Абсорбцією називається процес поглинання індивідуального газу, а також виборчого поглинання одного чи декількох компонентів газової суміші рідким поглиначем - абсорбентом. Поглинання газу може відбуватися або за рахунок його розчинення в абсорбенті, або в результаті його хімічної взаємодії з абсорбентом. У першому випадку процес називають фізичною абсорбцією, а в другому випадку - хемосорбцією. Можливо вести також сполучення обоїх цих процесів. Абсорбуємі компоненти газової суміші називають абсорбтивом, а неабсорбуємі - інертом. Абсорбентами служать індивідуальні розчини рідини активного компонента в рідкому розчиннику. В усіх випадках до абсорбентів пред'являються ряд вимог. Серед яких істотними є: висока абсорбційна здатність, селективність, низький тиск парів, хімічна інертність стосовно матеріалів з котрих виготовлений корпус абсорбера. Абсорбція у хімічних виробництвах використовується для здійснення ряду процесів, до яких відносяться:

- одержання цільових продуктів, наприклад, абсорбція HCl водою для одержання соляної кислоти;
- виділення коштовних компонентів з газових сумішей;
- видалення шкідливих домішок CO і CO_2 з технологічного потоку газів.

Необхідно відзначити, що один і той же тип апарату залежно від умов роботи може працювати в різних режимах. Так, наприклад, абсорбер з насадкою може працювати як в плівковому режимі, так і в барботажному.

Поверхневі абсорбери. Поверхня контакту фаз в поверхневих абсорберах створюється за рахунок фіксованої поверхні, або дзеркала рідини (власне поверхні процесу з абсорбції), або поточної плівки рідини (плівкові абсорбери), тобто поверхня контакту фаз в апараті визначається площею елемента у апараті, наприклад, насадки. Ці апарати можна розділити на наступні типи: поверхневі абсорбери з горизонтальним дзеркалом рідини; абсорбери з нерухомою насадкою; плівкові абсорбери; механічні плівкові

абсорбери. Апарати з рухомою насадкою займають проміжне положення між насадками і барботажними абсорберами.

Розпилюючі абсорбери. У розпилюючих абсорберах поверхні контакту фаз утворюється шляхом розпилюванням рідини в об'ємі на краплі.

Барботажні абсорбери. У барботажних абсорберах поверхня міжфазного контакту розривається потоками газових цівок або бульбашок, що розподіляються по рідині.

Виделений вище в рамці текст буде описом технологічного процесу в апараті “Абсорбер”. Зберегаємо цей текст у файл з назвою **Process.rtf** з доступом у каталогах МКРЗMainMenu \ Хлорування метану \ Data \ Device2 \ для розглядаємого навчального прикладу з модульної контрольної роботи № 2.

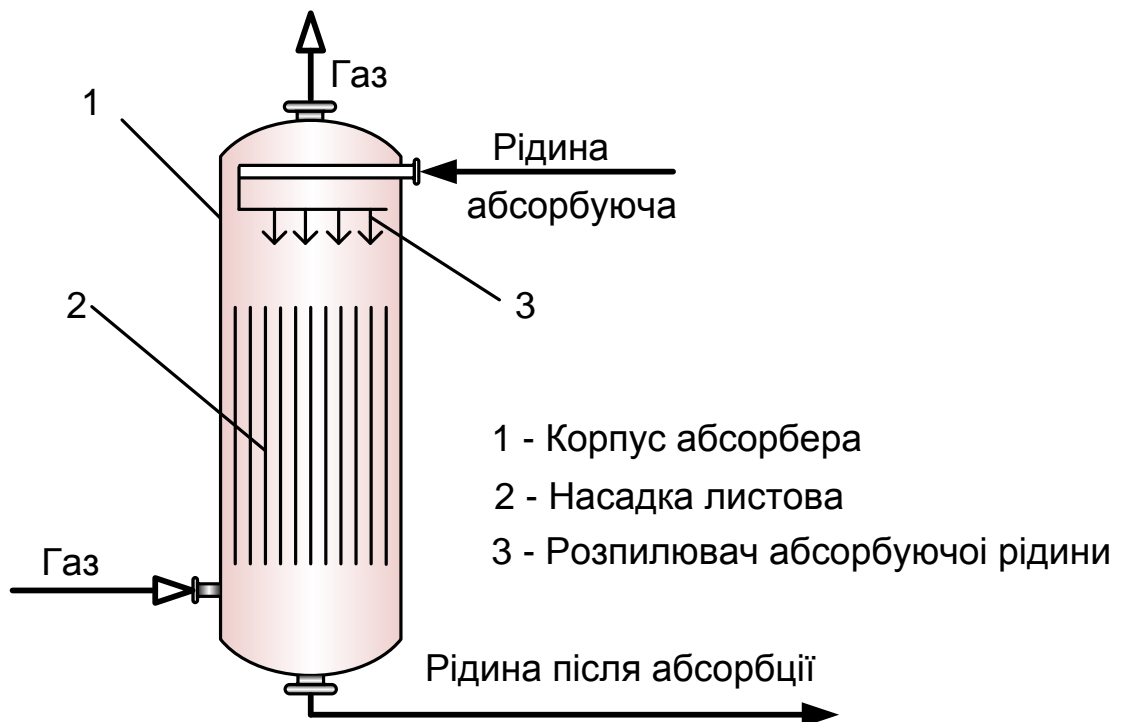


Рис. 3.2-3. Схема конструкції технологічного апарату “Абсорбер”.

Виконання завдання № 4 до РСЗП(Ч2). За допомогою графічного редактора робимо рисунок схеми з конструкції технологічного апарату “Абсорбер”, який показано на рис. 3.2-3. При виконанні команди “Конструкція”

у підменю “Абсорбер” в меню “Апарати” буде з’являтися рис. 3.2-3 у вікні відповідної форми C++ програми.

Схему (рис. 3.2-3) конструкції технологічного апарату “Абсорбер” зберігаємо у файл з назвою **Device.bmp** і розміщуємо з доступом у каталогах МКРЗMainMenu \ Хлорування метану \ Data \ Device2\ для розглядаємого навчального прикладу з модульної контрольної роботи № 2.

4. ПРИКЛАДИ З ТЕХНІКИ СТВОРЕННЯ І ПРОГРАМУВАННЯ ГРАФІЧНОГО МЕНЮ КОМАНД ПРИКЛАДНОЇ C++ ПРОГРАМИ

Візуальне програмування прикладних C++ програм з графічними меню команд мають багато можливостей, що показує робота системи Windows, яка функціонує у графічному режимі. Методику розробки і програмування C++ програми з графічним меню команд розглянемо у вигляді прикладів з двох варіантів по створюванню у C++ Builder прикладної C++ програми з мнемосхемою технологічного процесу виробництва хлорметанів.

4.1 Приклад № 1 з розробки графічного меню команд до прикладної C++ програми

В даному навчальному прикладі C++ програма з графічним меню команд буде функціонувати таким чином:

- у лівій частині основної форми **Form1** будуть розташовані 4 малюнки у вигляді кнопок з командами графічного меню: “Технологія”, “Апарати”, “Інформація” і “Вихід”;

- при наведенні курсору маніпулятора “мишка” на малюнки кнопок з назвами “Технологія”, “Апарати” або “Інформація” ці перші малюнки змінюються на інші, одночасно з цим активізуються відповідні малюнки кнопок з підменю графічного меню команд;

- для меню “Технологія” з’являються малюнки кнопок, які відповідають за виконання команд у підменю “Схема”, “Опис схеми” і “Продукція”;
- для меню “Апарати” з’являються малюнки кнопок, за допомогою яких буде виконання команд з підменю “Хлоратор” та “Абсорбер” ;
- при наведенні мишки на малюнок кнопки кожного підменю “Хлоратор” або “Абсорбер” буде виконуватися поява малюнків кнопок для виконання підкоманд “Конструкція”, “Опис процесу” і “Параметри”;
- для меню “Інформація” з’являються малюнки кнопок, які відповідають за виконання команд підменю “Про програму” і “Література”;
- при наведенні курсору маніпулятора “мишка” на меню “Вихід” форма курсору змінюється, а сам малюнок основного вікна прикладної С++ програми залишається таким самим;
- при наведенні курсору маніпулятора “мишка” на рисунки кнопок з назвами “Схема”, “Опис схеми”, “Продукція”, “Конструкція”, “Опис процесу”, “Параметри”, “Про програму” та “Література” форма курсора мишки буде відповідно змінюватися.

Усі необхідні рисунки з кнопками і відповідними написами для графічного меню команд прикладної С++ програми створюються у окремій програмі графічного редактора. Для створення графічного меню команд до прикладної С++ програми потрібно в середовищі С++ Builder виконувати такі наступні кроки і дії.

Крок 1. Створюємо новий проект файлів для прикладної С++ програми з модульної контрольної роботи № 2. Далі відповідно виконуємо:

- команду **File/New/Application** для створення нової чистої форми **Form1**;
- переходимо у вікно інспектора об’єктів та у полі **Caption** задаємо назву “Мнемосхема технологічного процесу виробництва хлорметанів” для основного вікна прикладної С++ програми;

- команду **File/Save** та задаємо назву проекту і файлу модуля форми (*.cpp) файлу програми.

Крок 2. Встановлюємо на **Form1** компоненту **Image** з сторінки **Additional** у бібліотеці компонентів та маркерною рамкою визначаємо бажаний розмір поля для розміщення схеми технологічного процесу виробництва хлорметанів в якості основного фону прикладної C++ програми. Якщо потрібно розмір форми можна також змінити.

Крок 3. Встановлюємо на **Form1** компоненту **Label** із сторінки **Additional** у бібліотеці компонентів та розміщуємо у лівий нижній край поля форми.

Крок 4. Натискаємо лівою кнопкою мишки на формі у полі **Image1** та в інспекторі об'єктів у полі **Picture** натискаємо на значок з трьома крапками. Кнопкою **Load** вказуємо шлях розташування рисунка схеми технологічного процесу виробництва хлорметанів (формат *.bmp) та натискаємо назву “**Открыть**” і у вікні **Picture Editor** натискаємо “**OK**”. Правою кнопкою мишки натискаємо на полі **Image1** та виконуємо команду **Control/Send to back**.

Крок 5. На мнемосхемі технологічного процесу, яка завантажилась у компоненту **Image1** створюємо окремі маркерні поля (**Image2...Image18**) навколо кожного з апаратів, які розташовані на рисунку мнемосхеми.

Крок 6. Далі необхідно розмістити на **Form1** компоненти **Image19, Image20..Image39** із сторінки **Additional** у бібліотеці компонентів так, як показано на рис. 4-1 для подальшого завантаження у них рисунків кнопок, які будуть складати графічне меню команд прикладної C++ програми. Також компонентам **Image19, Image20, Image21, Image22** задаємо при допомозі інспектора об'єктів такі розміри: у полях **Width** «97», а у полях **Height** «73».

Компонентам **Image23, Image24, Image25** у вікні інспектора об'єктів задаємо розміри у полях **Width** «145», а у полях **Height** «73». Також для компонент **Image26... Image39** у вікні інспектора об'єктів задаємо розміри у полях **Width** «25», а у полях **Height** «105».

Крок 7. Задаємо відображення назви апарату в **Label1** при переміщенні курсора маніпулятора «мишка» по мнемосхемі, завантаженої у **Image1**.

Натискаємо на полі **Image1** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо у вікно редактора коду, де у шаблон функції **Image1MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image1MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption="";  
Mig_T=0;  
if (Mig==1)  
{  
Image23->Visible = false;  
Image24->Visible = false;  
Image25->Visible = false;  
Image26->Visible = false;  
Image27->Visible = false;  
Image28->Visible = false;  
Image29->Visible = false;  
Image30->Visible = false;  
Image31->Visible = false;  
Image32->Visible = false;  
Image33->Visible = false;  
Image34->Visible = false;  
Image35->Visible = false;  
Image36->Visible = false;  
Image37->Visible = false;  
Image38->Visible = false;  
Image39->Visible = false;  
Image19->Picture->LoadFromFile("Data/Icons/Технологія_1.bmp");  
Image20->Picture->LoadFromFile("Data/Icons/Апарати_1.bmp");  
Image21->Picture->LoadFromFile("Data/Icons/Інформація_1.bmp");  
Mig=0;  
}  
}  
//-----
```


Натискаємо на полі **Image2** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо у вікно редактора коду, де у функцію **Image2MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image2MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("3 - Змішувач");  
}  
//-----
```

Натискаємо на **Image3** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо у вікно редактора коду, де у функцію **Image3MouseMove** додаємо текст:

```
//-----  
void __fastcall TForm1::Image3MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("4 - Хлоратор");  
}  
//-----
```

Натискаємо на **Image4** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки миші переходимо до вікна редактора коду, де у функцію **Image4MouseMove** додаємо текст:

```
//-----  
void __fastcall TForm1::Image4MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("1 - Змішувач");  
}  
//-----
```

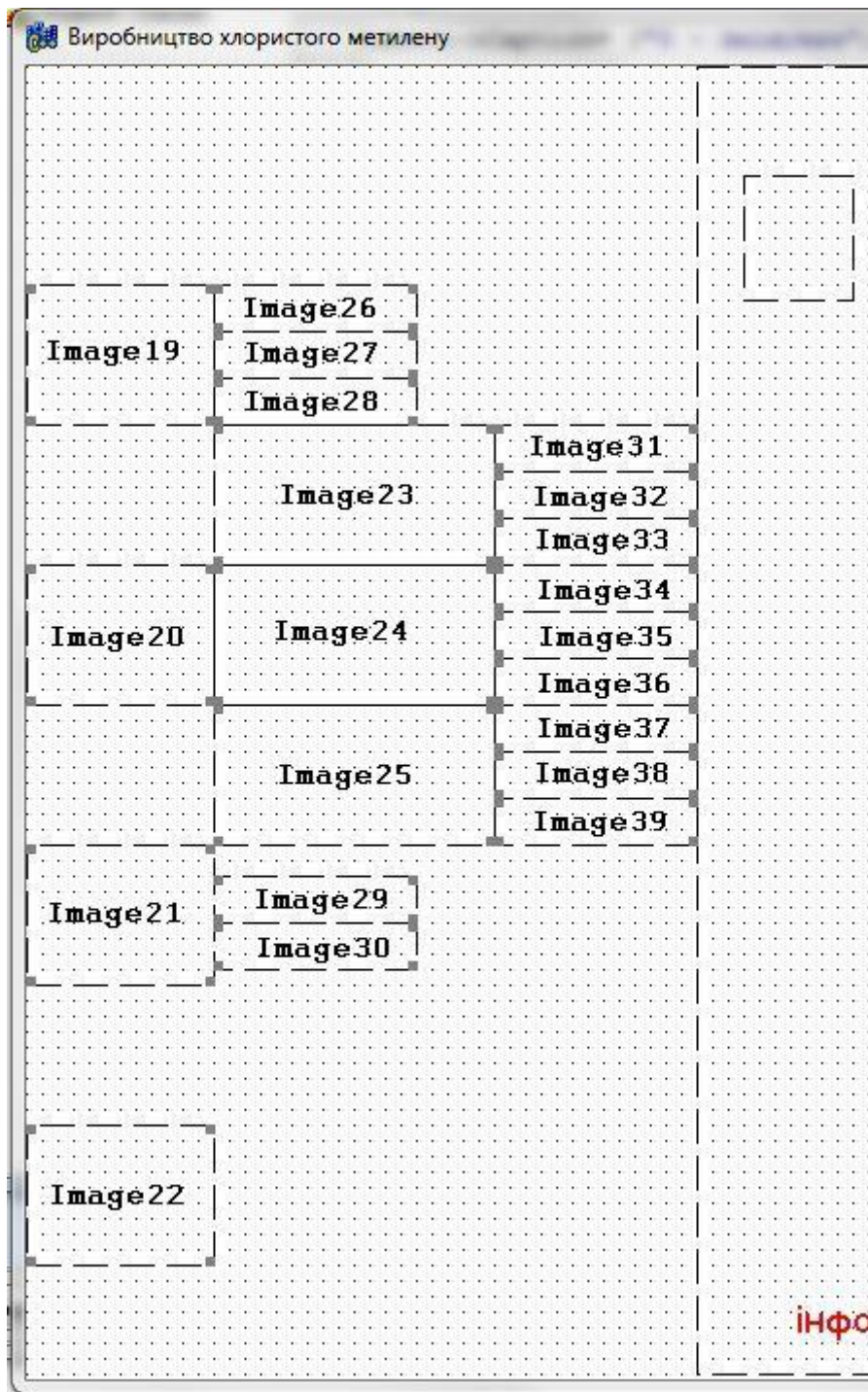


Рис. 4-1. Схема розміщення на Form1 компонент від Image19 до Image39 для рисунків кнопок з назвами команд графічного меню C++ програми.

Натискаємо на **Image5** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image5MouseMove** додаємо текст:

```
//-----  
void __fastcall TForm1::Image5MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)
```

```
{  
Label1->Caption= ("2 - Підігрівник");  
}
```

```
//-----
```

Натискаємо на **Image6** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image6MouseMove** додаємо текст:

```
//-----  
void __fastcall TForm1::Image6MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)
```

```
{  
Label1->Caption= ("1 - Сажовловлювач");  
}
```

```
//-----
```

Натискаємо на **Image7** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image7MouseMove** додаємо текст:

```
//-----  
void __fastcall TForm1::Image7MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)
```

```
{  
Label1->Caption= ("16 - Сепаратор");  
}
```

```
//-----
```

Натискаємо на **Image8** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image8MouseMove** додаємо текст:

```
//-----  
void __fastcall TForm1::Image8MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)
```

```
{  
Label1->Caption= ("10 – Графітовий холодильник");  
}  
//-----
```

Натискаємо на **Image9** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image9MouseMove** додаємо текст:

```
//-----  
void __fastcall TForm1::Image9MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("7 – Абсорбер");  
}  
//-----
```

Натискаємо на **Image10** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image10MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image10MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("8 – Абсорбер");  
}  
//-----
```

Натискаємо на **Image11** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image11MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image11MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("9 – Розріджувач луку");  
}  
//-----
```

```
//-----
```

Натискаємо на **Image12** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image12MouseMove** додаємо наступний текст:

```
//-----
```

```
void __fastcall TForm1::Image12MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
    Label1->Caption= ("11 – Збірник концентрованої соляної кислоти");
}
//-----
```

Натискаємо на **Image13** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image13MouseMove** додаємо наступний текст:

```
//-----
```

```
void __fastcall TForm1::Image13MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
    Label1->Caption= ("12 – Компресор");
}
//-----
```

Натискаємо на **Image14** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image14MouseMove** додаємо наступний текст:

```
//-----
```

```
void __fastcall TForm1::Image14MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
    Label1->Caption= ("15 – Фільтр");
}
//-----
```

Натискаємо на **Image15** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки пере-

ходимо до вікна редактора коду де у функцію **Image15MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image15MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("14 – Конденсатор");  
}  
//-----
```

Натискаємо на **Image16** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image16MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image16MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("15 –осушувачі");  
}  
//-----
```

Натискаємо на **Image17** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image17MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image17MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("17 – Розріджувач луку");  
}  
//-----
```

Натискаємо на **Image18** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image18MouseMove** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image18MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
Label1->Caption= ("6 – Повітряний холодильник");
}
//-----
```

Крок 8. Натискаємо лівою кнопкою мишки на формі по полю **Image1** та у вікні інспектора об'єктів у полі **Picture** натискаємо на значок з трьома крапками. Кнопкою **Clear** видаляємо рисунок фону та натискаємо кнопку «**OK**».

Крок 9. Задаємо завантаження фону в **Image1** та малюнків кнопок, які складають графічне меню команд С++ програми при допомозі полей **Image19....Image39** при старті прикладної програми. На вільному просторі форми натискаємо лівою кнопкою миші та в інспекторі об'єктів обираємо **Events** і подвійним натисканням миші у полі **OnCreate** переходимо до редактора коду прикладної С++ програми.

У функцію **FormCreate** необхідно додати наступний текст, який забезпечить завантаження фону в **Image1** та рисунков кнопок, що складають графічне меню у **Image19....Image39**; зробить поля **Image2....Image18** та **Image23....Image39** невидимими і також активізує інший візуальний курсор на полях **Image22....Image39**:

```
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
Image23->Picture->LoadFromFile("Data/Icons/Хлоратор_1.bmp");
Image24->Picture->LoadFromFile("Data/Icons/Абсорбер_1.bmp");
Image1->Picture->LoadFromFile("Data/Main/Background.bmp");
Image2->Visible = false;
Image3->Visible = false;
Image4->Visible = false;
Image5->Visible = false;
Image6->Visible = false;
Image7->Visible = false;
Image8->Visible = false;
Image9->Visible = false;
}
```

```
Image10->Visible = false;
Image11->Visible = false;
Image12->Visible = false;
Image13->Visible = false;
Image14->Visible = false;
Image15->Visible = false;
Image16->Visible = false;
Image17->Visible = false;
Image18->Visible = false;
Image23->Visible = false;
Image24->Visible = false;
Image26->Visible = false;
Image27->Visible = false;
Image28->Visible = false;
Image29->Visible = false;
Image30->Visible = false;
Image31->Visible = false;
Image32->Visible = false;
Image33->Visible = false;
Image34->Visible = false;
Image35->Visible = false;
Image36->Visible = false;
Image37->Visible = false;
Image38->Visible = false;
Image39->Visible = false;
Image19->Picture->LoadFromFile("Data/Icons/Технологія_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Апарати_1.bmp");
Image21->Picture->LoadFromFile("Data/Icons/Інформація_1.bmp");
Image22->Picture->LoadFromFile("Data/Icons/Вихід_1.bmp");
Image23->Picture->LoadFromFile("Data/Icons/Хлоратор_1.bmp");
Image24->Picture->LoadFromFile("Data/Icons/Абсорбер_1.bmp");
Image26->Picture->LoadFromFile("Data/Icons/Схема_1.bmp");
Image27->Picture->LoadFromFile("Data/Icons/Опис схеми_1.bmp");
Image28->Picture->LoadFromFile("Data/Icons/Продукція_1.bmp");
Image29->Picture->LoadFromFile("Data/Icons/Про програму_1.bmp");
Image30->Picture->LoadFromFile("Data/Icons/Література_1.bmp");
Image31->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");
Image32->Picture->LoadFromFile("Data/Icons/Опис процесу_1.bmp");
Image33->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");
Image34->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");
Image35->Picture->LoadFromFile("Data/Icons/Опис процесу_1.bmp");
Image36->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");
```



```

Image37->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");
Image38->Picture->LoadFromFile("Data/Icons/Опис процесу_1.bmp");
Image39->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");
Image22->Cursor = crHandPoint;
Image26->Cursor = crHandPoint;
Image27->Cursor = crHandPoint;
Image28->Cursor = crHandPoint;
Image29->Cursor = crHandPoint;
Image30->Cursor = crHandPoint;
Image31->Cursor = crHandPoint;
Image32->Cursor = crHandPoint;
Image33->Cursor = crHandPoint;
Image34->Cursor = crHandPoint;
Image35->Cursor = crHandPoint;
Image36->Cursor = crHandPoint;
Image37->Cursor = crHandPoint;
Image38->Cursor = crHandPoint;
Image39->Cursor = crHandPoint;
}
//-----

```

Крок 10. На вільному просторі форми **Form1** натискаємо лівою кнопкою мишки. У вікні інспектора об'єктів обираємо **Events** і у полі **OnMove** подвійним натисканням мишки переходимо до вікна редактора коду C++ програми та у функцію **FormMouseMove** додаємо наступий текст:

```

//-----
void __fastcall TForm1::FormMouseMove(TObject *Sender, TShiftState Shift,
int X, int Y)
{
if (Mig==1)
{
Image23->Visible = false;
Image24->Visible = false;
Image25->Visible = false;
Image26->Visible = false;
Image27->Visible = false;
Image28->Visible = false;
Image29->Visible = false;
Image30->Visible = false;
Image31->Visible = false;
Image32->Visible = false;

```

```

Image33->Visible = false;
Image34->Visible = false;
Image35->Visible = false;
Image36->Visible = false;
Image37->Visible = false;
Image38->Visible = false;
Image39->Visible = false;
Image19->Picture->LoadFromFile("Data/Icons/Технологія_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Апарати_1.bmp");
Image21->Picture->LoadFromFile("Data/Icons/Інформація_1.bmp");
Mig=0;
}
Mig_T=0;
Mig_A=0;
Mig_I=0;
Mig_V=0;
}
//-----

```

Крок 11. Переходимо на форму **Form1** і натискаємо на **Image3**. У вікні інспектора об'єктів у полі **Cursor** обираємо **crHandPoint** і потім вкладку **Events** та у полі **OnClick** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image3Click** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image3Click(TObject *Sender)
{
Form3->Caption="Конструкція хлоратора";
Form3->Image1->Picture->LoadFromFile("Data/Device1/Device.bmp");
Form3->ShowModal();
}
//-----

```

Крок 12. Переходимо на форму **Form1** і натискаємо на **Image9**. У вікні інспектора об'єктів у полі **Cursor** обираємо **crHandPoint** і потім вкладку **Events** та у полі **OnClick** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image9Click** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image9Click(TObject *Sender)
{
Form3->Caption="Конструкція абсорбера";
}

```

```
Form3->Image1->Picture->LoadFromFile("Data/Device2/Device.bmp");
Form3->ShowModal();
}
//-----
```

Крок 13. Переходимо на форму **Form1** і натискаємо на **Image10**. У вікні інспектора об'єктів у полі **Cursor** обираємо **crHandPoint** і потім вкладку **Events** та у полі **OnClick** подвійним натисканням кнопки мишки переходимо до редактора коду, де у функцію **Image10Click** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image10Click(TObject *Sender)
{
Form3->Caption="Конструкція абсорбера";
Form3->Image1->Picture->LoadFromFile("Data/Device2/Device.bmp");
Form3->ShowModal();
}
//-----
```

Тепер при натисканні на компонентах **Image**, яки відповідають назвам “Хлоратор” або “Абсорбер” буде відкриватися форма **Form3** з окремим рисунком схеми відповідного апарату.

Папка **Data** знаходиться у папці з файлами проекту прикладної програми. Файл “**Background.bmp**” є основним фоновим малюнком прикладної програми. Файли з папки **Data/Icons** є кнопками графічного меню програми. Файли з індексом **1** відповідають пасивному стану кнопок, а файли з індексом **2** відповідають активізованому стану кнопок. Тепер створимо додаткові форми, які нам знадобляться для відображення інформації відповідно до пунктів графічного меню команд.

Створення форми Form2. Виконуємо команду **File/New/Form** для створення нової чистої форми **Form2**. Встановлюємо на **Form2** компоненту **RichEdit** з сторінки **Win32** у бібліотеці компонентів. Натискаємо на новостворену компоненту **RichEdit1** та у полі **Align** інспектора об'єктів обираємо **alClient**. Таким чином **RichEdit1** буде налаштовуватись до розмірів форми **Form2**.

Створення форми Form3. Виконуємо команду **File/New/Form** для створення нової чистої форми **Form3**. У дереві об'єктів обираємо **Form3** та в полі **AutoSize** інспектора об'єктів обираємо значення **true**. Встановлюємо на **Form3** компоненту **Image** із сторінки **Additional** бібліотеки компонентів. Далі натискаємо на новостворену компоненту **Image1** та у полі **Width** інспектора об'єктів вводимо значення **385**, а у полі **Height** інспектора об'єктів вводимо значення **497**.

Створення форми Form4. Виконується далі такими кроками і діями:

Крок 1. Виконуємо команду **File/New/Form** для створення нової чистої форми **Form4**. Встановлюємо на **Form4** компоненту **Chart** із сторінки **Additional** у бібліотеці компонентів;

Крок 2. Натискаємо правою кнопкою мишки на нову компоненту **Chart1** та обираємо **Edit Chart**. Переходимо у вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо "OK". Переходимо у вкладку **Chart/Titles** та у полі вводу параметра набираємо "Температура". Переходимо у вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** та виходимо з меню **Edit Chart**;

Крок 3. Встановлюємо на **Form4** компоненту **Chart** із сторінки **Additional** у бібліотеці компонентів;

Крок 4. Натискаємо правою кнопкою мишки на нову компоненту **Chart2** та обираємо **Edit Chart**. Переходимо у вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо "OK". Переходимо у вкладку **Chart/Titles** та у полі вводу параметра набираємо "Витрата сировини". Переходимо у вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** для виходу з меню **Edit Chart**;

Крок 5. У дереві об'єктів обираємо **Form4** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnCreate** подвійним натисканням кнопки

мишки переходимо до вікна редактора коду, де у функцію **FormCreate** додаємо наступний текст:

```
//-----  
void __fastcall TForm4::FormCreate(TObject *Sender)  
{  
for (int i=0; i<=10; i++)  
{  
Series1->AddXY(i+cos(i),13*i*i,"",clRed);  
Series2->AddXY(12*i*i+25, i*i+15*i,"",clRed);  
}  
}  
//-----
```

Створення форми Form5. Виконуємо для цього такі кроки і дії:

Крок 1. Виконуємо команду **File/New/Form** для створення нової чистої форми **Form5**. Встановлюємо на **Form5** компоненту **Chart** з сторінки **Additional** у бібліотеці компонентів;

Крок 2. Натискаємо правою кнопкою мишки на нову компоненту **Chart1** та обираємо **Edit Chart**. Переходимо у вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо “**OK**”. Переходимо у вкладку **Chart/Titles** та у полі вводу параметра набираємо “**Температура**”. Переходимо у вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** та виходимо з меню **Edit Chart**;

Крок 3. Встановлюємо на **Form5** компоненту **Chart** із сторінки **Additional** у бібліотеці компонентів;

Крок 4. Натискаємо правою кнопкою миші на нову компоненту **Chart2** та обираємо **Edit Chart**. Переходимо у вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо “**OK**”. Переходимо у вкладку **Chart/Titles** та у полі вводу параметра набираємо “**Витрата сировини**”. Переходимо у вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** для виходу з меню **Edit Chart**;

Крок 5. У дереві об'єктів обираємо **Form5**. У вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnCreate** подвійним натисканням мишки переходимо до вікна редактора коду і у функцію **FormCreate** додаємо текст:

```
//-----  
void __fastcall TForm5::FormCreate(TObject *Sender)  
{  
for (int i=0; i<=10; i++)  
{  
Series1->AddXY(i+sin(i),13*i*i,"",clRed);  
Series2->AddXY(14*i+25*i*sin(i), cos(i)*i*i*i+15*i,"",clRed);  
}  
}  
//-----
```

Створення форми Form6. Виконуємо для цього такі кроки і дії:

Крок 1. Виконуємо команду **File/New/Form** для створення нової чистої форми **Form6**. Встановлюємо на **Form6** компоненту **Chart** із сторінки **Additional** у бібліотеці компонентів;

Крок 2. Натискаємо правою кнопкою мишки на нову компоненту **Chart1** та обираємо **Edit Chart**. Переходимо на вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо “**OK**”. Переходимо на вкладку **Chart/Titles** та у полі вводу параметра набираємо “**Температура**”. Переходимо на вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** для виходу з меню **Edit Chart**;

Крок 3. Встановлюємо на **Form6** компоненту **Chart** із сторінки **Additional** у бібліотеці компонентів;

Крок 4. Натискаємо правою кнопкою мишки на нову компоненту **Chart2** та обираємо **Edit Chart**. Переходимо на вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо “**OK**”. Переходимо на вкладку **Chart/Titles** та у полі вводу параметра набираємо “**Витрата сировини**”. Пе-

переходимо на вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** для виходу з меню **Edit Chart**;

Крок 4. У дереві об'єктів обираємо **Form6** і в вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnCreate** подвійним натисканням мишки переходимо у вікно редактора коду, де у функцію **FormCreate** додаємо текст:

```
//-----  
void __fastcall TForm6::FormCreate(TObject *Sender)  
{  
for (int i=0; i<=10; i++)  
{  
Series1->AddXY(i+tan(i),13*i*i,"",clRed);  
Series2->AddXY(13*i+25*tan(i), exp(i)*i*i+15*i,"",clRed);  
}  
}  
//-----
```

Створення форми AboutBox. Виконуємо для цього такі кроки і дії:

Крок 1. Виконуємо команду **File/New/Other** та обираємо **About box** для створення нової форми. За допомогою дерева об'єктів обираємо **AboutBox** та у полі **Caption** інспектора об'єктів вводимо “**Про програму**”;

Крок 2. У списку дерева об'єктів обираємо **Copyright** та у полі **Caption** інспектора об'єктів вводимо “**Розробив _ПІБ_**”;

Крок 3. У дереві об'єктів обираємо **Label1** та у полі **Caption** інспектора об'єктів вводимо “**Група _№ групи**”;

Крок 4. У дереві об'єктів обираємо **ProductName** та у полі **Caption** інспектора об'єктів вводимо “**Модульна контрольна робота**”;

Крок 5. Через дерево об'єктів обираємо **Version** і у полі **Caption** вікна інспектора об'єктів вводимо “**Версія 1.0**”. У вікні редактора коду зверху до тексту додаємо наступний текст:

```
//-----  
#include <vcl.h>  
#pragma hdrstop
```

```

#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit4.h"
#include "Unit5.h"
#include "Unit6.h"
#include "Unit7.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int Mig=0;
int Mig_T=0;
int Mig_A=0;
int Mig_I=0;
int Mig_V=0;
int Mig_AA=0;
int Mig_AX=0;
int Mig_AN=0;

```

Це дасть змогу викликати на виконання форми **Form2**, **Form 3**, **Form 4**, **Form 5**, **Form 6** і **Form 7**, знаходячись на формі **Form1**. Також вводимо ряд змінних, які допоможуть реалізувати роботу графічного меню команд та уникнення певних проблем (запобігання мерехтіння зображень і т.д.).

4.1.1 Програмування обробки подій маніпулятора “мишка” у графічному меню команд “Технологія”

При наведенні курсору маніпулятора “мишка” на меню “Технологія” необхідно, щоб у графічному меню команд рисунок кнопки змінювався на інший та одночасно з цим активізувалися відповідні підменю цього меню.

Для меню “Технологія” потрібно активізувати рисунки кнопок, які відповідають підменю “Схема”, “Опис схеми” і “Продукція”. Для цього далі виконуємо такі кроки і дії.

Крок 1. Натискаємо мишкою на **Image19** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кноп-

ки мишки переходимо до вікна редактора коду, де у шаблон функції **Image19MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image19MouseMove(TObject *Sender,  
    TShiftState Shift, int X, int Y)  
{  
if (Mig_T==0)  
{  
Image19->Picture->LoadFromFile("Data/Icons/Технологія_2.bmp");  
Mig_T=1;  
}  
Image23->Visible = false;  
Image24->Visible = false;  
Image25->Visible = false;  
Image26->Visible = true;  
Image27->Visible = true;  
Image28->Visible = true;  
Image29->Visible = false;  
Image30->Visible = false;  
Image31->Visible = false;  
Image32->Visible = false;  
Image33->Visible = false;  
Image34->Visible = false;  
Image35->Visible = false;  
Image36->Visible = false;  
Image37->Visible = false;  
Image38->Visible = false;  
Image39->Visible = false;  
Mig=1;  
}  
//-----
```

Таким чином при наведенні курсору маніпулятора “мишка” на компоненту **Image19**, яка відповідає графічному меню “Технологія”, відразу на екрані дисплея стануть видимими рисунки кнопок, яки завантажуються у поля **Image26**, **Image27** і **Image28** для графічних підменю “Схема”, “Опис схеми” і “Продукція”. Крім того у саму компоненту **Image19** завантажується інший рисунок, на якому зображення показує активізований стан команд цього графічного меню команд.

Далі необхідно записати оператори, щоб при натисканні на кожному з трьох підменю відображалася бажана інформація, а рисунок кнопки, який відповідає підкоманді при натисканні та відпусканні мишки змінювався.

Програмування підменю “Схема”.

Крок 1. Натискаємо на **Image26** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnClick** подвійним щелчком мишки переходимо до вікна редактора коду, де у шаблон функції **Image26Click** додаємо текст:

```
//-----  
void __fastcall TForm1::Image26Click(TObject *Sender)  
{  
Image1->Picture->LoadFromFile("Data/Main/Scheme.bmp");  
Image2->Visible = true;  
Image3->Visible = true;  
Image4->Visible = true;  
Image5->Visible = true;  
Image6->Visible = true;  
Image7->Visible = true;  
Image8->Visible = true;  
Image9->Visible = true;  
Image10->Visible = true;  
Image11->Visible = true;  
Image12->Visible = true;  
Image13->Visible = true;  
Image14->Visible = true;  
Image15->Visible = true;  
Image16->Visible = true;  
Image17->Visible = true;  
Image18->Visible = true;  
Label1->Visible = true;  
}  
//-----
```

Крок 2. Натискаємо на **Image26** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image26MouseDown** додаємо наступний текст:

```
//-----
```

```

void __fastcall TForm1::Image26MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
Image26->Picture->LoadFromFile("Data/Icons/Схема_2.bmp");
}
//-----

```

Крок 3. Натискаємо на **Image26** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image26MouseUp** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image26MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
Image26->Picture->LoadFromFile("Data/Icons/Схема_1.bmp");
}
//-----

```

Програмування підменю "Опис схеми".

Крок 1. Натискаємо на **Image27** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnClick** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image27Click** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image27Click(TObject *Sender)
{
Form2->RichEdit1->Lines->LoadFromFile("Data/Main/Scheme.rtf");
Form2->Caption="Опис процесу отримання хлористого метилену";
Form2->ShowModal();
}
//-----

```

Крок 2. Натискаємо на **Image27** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image27MouseDown** додаємо наступний текст:

```

//-----

```

```

void __fastcall TForm1::Image27MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
Image27->Picture->LoadFromFile("Data/Icons/Опис схеми_2.bmp");
}
//-----

```

Крок 3. Натискаємо на **Image27** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image27MouseUp** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image27MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
Image27->Picture->LoadFromFile("Data/Icons/Опис схеми_1.bmp");
}
//-----

```

Програмування підменю "Продукція".

Крок 1. Натискаємо на **Image28** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnClick** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image28Click** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image28Click(TObject *Sender)
{
Form2->RichEdit1->Lines->LoadFromFile("Data/Main/Products.rtf");
Form2->Caption="Продукція";
Form2->ShowModal();
}
//-----

```

Крок 2. Натискаємо на **Image28** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки пе-

переходимо до вікна редактора коду, де у функцію **Image28MouseDown** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image28MouseDown(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image28->Picture->LoadFromFile("Data/Icons/Продукція_2.bmp");  
}  
//-----
```

Крок 3. Натискаємо на **Image28** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image28MouseUp** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image28MouseUp(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image28->Picture->LoadFromFile("Data/Icons/Продукція_1.bmp");  
}  
//-----
```

4.1.2 Програмування обробки подій маніпулятора “мишка” у графічному меню команд “Апарати”

При наведенні курсору маніпулятора “мишка” на меню “Апарати” необхідно, щоб у графічному меню команд рисунок кнопки змінювався та одночасно з цим активізувалися відповідні підменю до цього меню. Для меню “Апарати” активізуються рисунки кнопок, які відповідають підменю “Хлоратор” та “Абсорбер”. Також для кожного підменю активізуються підкоманди “Конструкція”, “Опис процесу” та “Параметри”. Для програмування відповідних команд виконуємо такі кроки і дії.

Крок 1. Натискаємо на **Image20** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки пе-

переходимо до вікна редактора коду, де у функцію **Image20MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image20MouseMove(TObject *Sender,  
    TShiftState Shift, int X, int Y)  
{  
if(Mig_A==0)  
{  
Image20->Picture->LoadFromFile("Data/Icons/Апарати_2.bmp");  
Mig_A=1;  
}  
if(Mig_AA==1)  
{  
Image24->Picture->LoadFromFile("Data/Icons/Абсорбер_1.bmp");  
Mig_AA=0;  
}  
if(Mig_AX==1)  
{  
Image23->Picture->LoadFromFile("Data/Icons/Хлоратор_1.bmp");  
Mig_AX=0;  
}  
Image23->Visible =true;  
Image24->Visible = true;  
Image26->Visible = false;  
Image27->Visible = false;  
Image28->Visible = false;  
Image29->Visible = false;  
Image30->Visible = false;  
Image31->Visible = false;  
Image32->Visible = false;  
Image33->Visible = false;  
Image34->Visible = false;  
Image35->Visible = false;  
Image36->Visible = false;  
Image37->Visible = false;  
Image38->Visible = false;  
Image39->Visible = false;  
Mig=1;  
}  
//-----
```

Таким чином при наведенні курсору маніпулятора “мишка” на компоненту **Image20**, яка відповідає графічному меню “Апарати”, відразу на екрані дисплея стануть видимими рисунки кнопок, які завантажуються у поля **Image23** і **Image24** для підменю “Хлоратор” та “Абсорбер”. Крім того у саму компоненту **Image20** завантажуються інший рисунок, на якому зображення показує активізований стан команди цього пункту графічного меню.

Далі необхідно записати такі оператори, щоб при виборі кожного з цих підменю відображалася бажана інформація, а рисунок кнопки, який відповідає підменю при натисканні та відпусканні мишки змінювався.

Програмування підкоманди “Хлоратор”.

Зробимо так, щоб при наведенні курсору маніпулятора “мишка” на **Image23**, яка відповідає підменю “Хлоратор”, стають видимими **Image31**, **Image32**, **Image33**, які відповідають підкомандам “Конструкція”, “Опис процесу” та “Параметри”. Крім того у саму компоненту **Image23** необхідно завантажити інший рисунок, на якому зображення показує активізований стан команди цього графічного підменю. Для програмування відповідних команд виконуємо такі кроки і дії.

Крок 1. Натискаємо на **Image23** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image23MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image23MouseMove(TObject *Sender,  
    TShiftState Shift, int X, int Y)  
{  
    if(Mig_AX==0)  
    {  
        Image23->Picture->LoadFromFile("Data/Icons/Хлоратор_2.bmp");  
        Mig_AX=1;  
    }  
    if(Mig_AA==1)  
    {
```

```

Image24->Picture->LoadFromFile("Data/Icons/Абсорбер_1.bmp");
Mig_AA=0;
}
Image26->Visible = false;
Image27->Visible = false;
Image28->Visible = false;
Image29->Visible = false;
Image30->Visible = false;
Image31->Visible = true;
Image32->Visible = true;
Image33->Visible = true;
Image34->Visible = false;
Image35->Visible = false;
Image36->Visible = false;
Image37->Visible = false;
Image38->Visible = false;
Image39->Visible = false;
}
//-----

```

Далі необхідно записати такі оператори, щоб при натисканні на кожній з цих підкоманд відображалась бажана інформація, а рисунок кнопки, який відповідає підкоманді при натисканні та відпусканні мишки змінювався.

Програмування підкоманди “Конструкція”.

Крок 1. Натискаємо на **Image31** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnClick** подвійним щелчком мишки переходимо до вікна редактора коду, де у функцію **Image31Click** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image31Click(TObject *Sender)
{
Form4->Caption="Конструкція хлоратора";
Form4->Image1->Picture->LoadFromFile("Data/Device1/Device.bmp");
Form4->ShowModal();
}
//-----

```

Крок 2. Натискаємо на **Image31** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки пе-

переходимо до вікна редактора коду, де у функцію **Image31MouseDown** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image31MouseDown(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image31->Picture->LoadFromFile("Data/Icons/Конструкція_2.bmp");  
}  
//-----
```

Крок 3. Натискаємо на **Image31** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image31MouseUp** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image31MouseUp(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image31->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");  
}  
//-----
```

Програмування підкоманди "Опис процесу".

Крок 1. Натискаємо на **Image32** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnClick** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image32Click** додаємо текст:

```
//-----  
void __fastcall TForm1::Image32Click(TObject *Sender)  
{  
Form2->RichEdit1->Lines->LoadFromFile("Data/Device1/Process.rtf");  
Form2->Caption="Опис процесу в хлораторі";  
Form2->ShowModal();  
}  
//-----
```

Крок 2. Натискаємо на **Image32** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки переходимо до вікна редактора коду. де у функцію **Image32MouseDown** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image32MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
Image32->Picture->LoadFromFile("Data/Icons/Опис процесу_2.bmp");
}
//-----
```

Крок 3. Натискаємо на **Image32** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image32MouseUp** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image32MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
Image32->Picture->LoadFromFile("Data/Icons/Опис процесу_1.bmp");
}
//-----
```

Програмування підкоманди "Параметри".

Крок 1. Натискаємо на **Image33** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnClick** подвійним щелчком мишки переходимо до вікна редактора коду, де у функцію **Image33Click** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image33Click(TObject *Sender)
{
Form4->ShowModal();
}
//-----
```

Крок 2. Натискаємо на **Image33** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image33MouseDown** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image33MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
```

```
Image33->Picture->LoadFromFile("Data/Icons/Параметри_2.bmp");  
}  
//-----
```

Крок 3. Натискаємо на **Image33** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image33MouseUp** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image33MouseUp(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image33->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");  
}  
//-----
```

Програмування підменю “Абсорбер”.

Зробимо так, щоб при наведенні курсору маніпулятора “мишка” на **Image24**, яка відповідає підменю “Абсорбер”, стають видимими **Image34**, **Image35**, **Image36**, яки відповідають підкомандам “Конструкція”, “Опис процесу” та “Параметри”. Крім того у саму компоненту **Image24** необхідно завантажити інший рисунок кнопки, на який зображення показує активізований стан команд у графічному меню. Для програмування команд далі виконуємо такі наступні кроки і дії.

Крок 1. Натискаємо на **Image24** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image24MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image24MouseMove(TObject *Sender,  
    TShiftState Shift, int X, int Y)  
{  
if(Mig_AA==0)  
{  
Image24->Picture->LoadFromFile("Data/Icons/Абсорбер_2.bmp");
```

```

Mig_AA=1;
}
if(Mig_AX==1)
{
Image23->Picture->LoadFromFile("Data/Icons/Хлоратор_1.bmp");
Mig_AX=0;
}
Image26->Visible = false;
Image27->Visible = false;
Image28->Visible = false;
Image29->Visible = false;
Image30->Visible = false;
Image31->Visible = false;
Image32->Visible = false;
Image33->Visible = false;
Image34->Visible = true;
Image35->Visible = true;
Image36->Visible = true;
Image37->Visible = false;
Image38->Visible = false;
Image39->Visible = false;
}
//-----

```

Далі необхідно записати такі оператори, щоб при натисканні на кожну з трьох підкоманд відображалась бажана інформація, а рисунок кнопки, який відповідає підкоманді при натисканні та відпусканні мишки змінювався.

Програмування підкоманди “Конструкція”.

Крок 1. Натискаємо на **Image34** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnClick** подвійним натисканням клавиши мишки переходимо до вікна редактора коду, де у шаблон функції **Image34Click** додаємо текст:

```

//-----
void __fastcall TForm1::Image34Click(TObject *Sender)
{
Form4->Caption="Конструкція абсорбера";
Form4->Image1->Picture->LoadFromFile("Data/Device1/Device.bmp");
Form4->ShowModal();
}

```

//-----

Крок 2. Натискаємо на **Image34** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image34MouseDown** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image34MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
Image34->Picture->LoadFromFile("Data/Icons/Конструкція_2.bmp");
}
//-----
```

Крок 3. Натискаємо на **Image34** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image34MouseUp** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image34MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
Image34->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");
}
//-----
```

Програмування підкоманди "Опис процесу".

Крок 1. Натискаємо на **Image35** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnClick** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image35Click** додаємо текст:

```
//-----
void __fastcall TForm1::Image35Click(TObject *Sender)
{
Form2->RichEdit1->Lines->LoadFromFile("Data/Device1/Process.rtf");
Form2->Caption="Опис процесу в абсорбері";
Form2->ShowModal();
}
```

```
}
```

```
//-----
```

Крок 2. Натискаємо на **Image35** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image35MouseDown** додаємо наступний текст:

```
//-----
```

```
void __fastcall TForm1::Image35MouseDown(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image35->Picture->LoadFromFile("Data/Icons/Опис_процесу_2.bmp");  
}
```

```
//-----
```

Крок 3. Натискаємо на **Image35** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image35MouseUp** додаємо наступний текст:

```
//-----
```

```
void __fastcall TForm1::Image35MouseUp(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image35->Picture->LoadFromFile("Data/Icons/Опис_процесу_1.bmp");  
}
```

```
//-----
```

Програмування підкоманди "Параметри".

Крок 1. Натискаємо на **Image36** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnClick** подвійним щелчком мишки переходимо до вікна редактора коду, де у функцію **Image36Click** додаємо наступний текст:

```
//-----
```

```
void __fastcall TForm1::Image36Click(TObject *Sender)  
{  
Form5->ShowModal();  
}
```

```
//-----
```

Крок 2. Натискаємо на **Image36** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки пе-

переходимо до вікна редактора коду, де у функцію **Image36MouseDown** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image36MouseDown(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image36->Picture->LoadFromFile("Data/Icons/Параметри_2.bmp");  
}  
//-----
```

Крок 3. Натискаємо на **Image36** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image36MouseUp** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image36MouseUp(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image36->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");  
}  
//-----
```

4.1.3 Програмування обробки подій маніпулятора “мишка” у графічному меню команд “Інформація”

При наведенні курсору маніпулятора “мишка” на меню “Інформація” необхідно, щоб у графічному меню команд рисунок кнопки змінювався на інший та одночасно з цим активізувалися відповідні підкоманди цього меню.

Для меню “Іформаця” активізуються рисунки кнопок, які відповідають підкомандам “Про програму” та “Література”.

Крок 1. Натискаємо на **Image21** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image21MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image21MouseMove(TObject *Sender,
```

```

    TShiftState Shift, int X, int Y)
{
if(Mig_I==0)
{
Image21->Picture->LoadFromFile("Data/Icons/Інформація_2.bmp");
Mig_I=1;
}
Image23->Visible = false;
Image24->Visible = false;
Image26->Visible = false;
Image27->Visible = false;
Image28->Visible = false;
Image29->Visible = true;
Image30->Visible = true;
Image31->Visible = false;
Image32->Visible = false;
Image33->Visible = false;
Image34->Visible = false;
Image35->Visible = false;
Image36->Visible = false;
Image37->Visible = false;
Image38->Visible = false;
Image39->Visible = false;
Mig=1;
}
//-----

```

Таким чином при наведенні курсору маніпулятора “мишка” на компоненту **Image21**, яка відповідає графічному меню “Інформація”, відразу на екрані дисплея стануть видимими рисунки кнопок, які завантажуються у поля **Image29** та **Image30** для підменю “Про програму” і “Література”. Крім того у саму компоненту **Image21** завантажувється інший рисунок, на якому зображення показує активізований стан команд цього графічного меню. Далі необхідно записати такі оператори, щоб при натисканні на кожній з двох підкоманд відображалась бажана інформація, а рисунок кнопки, який відповідає підменю при натисканні та відпусканні мишки змінювався.

Програмування підкоманди “Про програму”.

Крок 1. Натискаємо на **Image29** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnClick** подвійним щелчком мишки переходимо до вікна редактора коду, де у функцію **Image29Click** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image29Click(TObject *Sender)  
{  
AboutBox->ShowModal();  
}  
//-----
```

Крок 2. Натискаємо на **Image29** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image29MouseDown** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image29MouseDown(TObject *Sender,  
TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image29->Picture->LoadFromFile("Data/Icons/Про програму_2.bmp");  
}  
//-----
```

Крок 3. Натискаємо на **Image29** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image29MouseUp** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image29MouseUp(TObject *Sender,  
TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image29->Picture->LoadFromFile("Data/Icons/Про програму_1.bmp");  
}  
//-----
```

Програмування підкоманди “Література”.

Крок 1. Натискаємо на **Image30** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnClick** подвійним натисканням кнопки мишки пе-

переходимо до вікна редактора коду, де у функцію **Image30Click** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image30Click(TObject *Sender)  
{  
Form2->Caption= ("Література");  
Form2->RichEdit1->Lines->LoadFromFile("Data/Main/Literature.rtf");  
Form2->ShowModal();  
}  
//-----
```

Крок 2. Натискаємо на **Image30** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseDown** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image30MouseDown** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image30MouseDown(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image30->Picture->LoadFromFile("Data/Icons/Література_2.bmp");  
}  
//-----
```

Крок 3. Натискаємо на **Image30** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image30MouseUp** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image30MouseUp(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
Image30->Picture->LoadFromFile("Data/Icons/Література_1.bmp");  
}  
//-----
```

4.1.4 Програмування обробки подій маніпулятора “мишка” у графічному меню команд “Вихід”

При розташуванні курсору маніпулятора “мишка” у графічному меню команд на рисунку кнопки “Вихід необхідно, щоб змінювався зовнішній вигляд курсору та виконувався вихід з прикладної С++ програми модульної контрольної роботи № 2.

Крок 1. Натискаємо на **Image22** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnClick** подвійним щелчком мишки переходимо до вікна редактора коду, де у функцію **Image22Click** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image22Click(TObject *Sender)  
{  
Close();  
}  
//-----
```

4.2 Приклад № 2 з розробки графічного меню команд до прикладної С++ програми

Розглянемо методику розробки і програмування другого варіанту графічного меню команд С++ програми до модульної контрольної роботи № 2, щоб показати, як можна змінювати вигляд різноманітних графічних меню команд та які дії користувача і розроблені матеріали до прикладної програми залишаються стандартними, коли структура графічного меню команд буде відповідати рис. 4-2. Також додаємо у меню «Апарати» ще один апарат «Розріджувач луку» з мнемосхеми процесу виробництва хлорметанів ([дивись додаток Д1](#)).

Прикладна С++ програма повинна мати графічне меню команд у іншому вигляді і працювати за таким варіантом обробки рисунків кнопок з назвами команд. У лівій частині **Form1** будуть розташовані 4 компоненти Image — **Image19**, **Image20**, **Image21** та **Image22**. В них С++ програма буде заванта-

жувати рисунки кнопок до графічного меню. В залежності від того, яке підменю основного графічного меню визначено користувачем, тоді у ці компоненти відповідно розміщуватимуться інші рисунки кнопок. Залежність рисунків кнопок до команд графічного меню програми докладно показана у таблицях № 4-1 та № 4-2. Переміщення по підкомандах меню програми здійснюється за допомогою натискання лівої кнопки маніпулятора “мишка”. Компонента **Image22** використовується для виходу з самої прикладної програми, або для повернення у попередній підпункт меню команд. В основі дії другого варіанту C++ програми лежить використання двох змінних — *ur* та *sh*. Змінна *sh* використовується для запобігання мерехтінню зображення на екрані при багаторазовому звертанні до підменю “Схема” з меню команд “Технологія”. Змінна *ur* допомагає зробити можливим використання лише 4 компонент **Image** за рахунок присвоєння цій змінній різних значень в залежності від поточного підпункту меню команд C++ програми та перевірки цих значень при допомозі умовних операторів. Відповідні значення змінної *ur* показані під таблицями № 4-1 та № 4-2. Усі необхідні рисунки з кнопками і відповідними написами для графічного меню команд прикладної C++ програми створюються у програмі графічного редактора. Для створення графічного меню команд у прикладній C++ програмі потрібно в середовищі C++ Builder виконувати такі наступні кроки і дії.

Крок 1. Створюємо новий проект для C++ програми до модульної контрольної роботи № 2. Виконуємо команду **File/New/Application** для створення чистої форми **Form1**. Переходимо у вікно інспектора об’єктів та у полі **Caption** задаємо назву прикладної програми: “Мнемосхема технологічного процесу виробництва хлорметанів”. Виконуємо команду **File/Save** та задаємо назву проекту і файлам (*.cpp) з розробляємої програми.

Таблиця № 4-1. Залежності до основного графічного меню команд.

Компонента	Зображення у компоненті			
Panel1	Меню	Технологія	Апарати	Інформація
Image19	Технологія	Схема	Хлоратор	Про програму
Image20	Апарати	Опис схеми	Абсорбер	Література
Image21	Інформація	Продукція	Розріджувач лугу	<Пусто>
Image22	Вихід	Назад	Назад	Назад
Значення ur	0	1	2	6

Таблиця № 4-2. Залежності до графічного підменю “Апарати”.

Компонента	Зображення у компоненті		
Panel1	Хлоратор	Абсорбер	Розріджувач лугу
Image19	Конструкція	Конструкція	Конструкція
Image20	Опис процесу	Опис процесу	Опис процесу
Image21	Параметри	Параметри	Параметри
Image22	Назад	Назад	Назад
Значення ur	3	4	5

Крок 2. Встановлюємо на **Form1** компоненту **Image** з сторінки **Additional** у бібліотеці компонентів та маркерною рамкою визначаємо розмір бажаного поля для розміщення мнемосхеми технологічного процесу, як основного фону прикладної C++ програми. Якщо потрібно розмір форми можна також змінити.

Крок 3. Встановлюємо на **Form1** компоненту **Label** з сторінки **Additional** у бібліотеці компонентів та розміщуємо його у лівий нижній край форми.

Крок 4. Натискаємо лівою кнопкою миші на формі по полю **Image1** та у вікні інспектора об'єктів у полі **Picture** натискаємо на значок з трьома крапками. Кнопкою **Load** вказуємо шлях розташування рисунка мнемосхеми хімічного виробництва (хлорметанів) у форматі (*.bmp) та натискаємо “Отк-

рыть” у вікні **Picture Editor** і далі натискаємо “**OK**”. Правою кнопкою мишки натискаємо на полі **Image1** та виконуємо команду **Control/Send to back**.

Крок 5. На рисунку мнемосхеми, що завантажився у поле **Image1**, за допомогою компоненти **Image** із сторінки **Additional** створюємо окремі поля навколо кожного з апаратів на мнемосхемі (**Image2.....Image18**), аналогічно як показано на рис. 4-2.

Крок 6. На **Form1** відповідно до рис. 4-2 розміщуємо компоненти **Image19, Image20, Image21** і **Image22** із сторінки **Additional** бібліотеки і **Panel1** з сторінки **Standard** для подальшого завантаження у них рисунків кнопок, які будуть складати графічне меню прикладної C++ програми.

Компонентам **Image19, Image20, Image21** і **Image22** у вікні інспектора об’єктів задаємо значення до розмірів: у полі **Width** «185» і у полі **Height** «105». Компоненті **Panel1** у вікні інспектора об’єктів задаємо розміри: у полі **Width** «185» і у полі **Height** «49».

Крок 7. Задаємо у **Label1** відображення підказки з назви апарату при переміщенні маніпулятора “мишка” по мнемосхемі, завантаженої у **Image1**.

Натискаємо мишкою на **Image1** і у вікні інспектора об’єктів обираємо **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image1MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image1MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
    Label1->Caption=("");  
}  
//-----
```

Далі натискаємо на **Image2** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням лівої кнопки мишки переходимо до вікна редактора коду і у функцію **Image2MouseMove** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image2MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
    Label1->Caption= ("3 - Змішувач");
}
//-----
```

Також натискаємо на **Image3** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image3MouseMove** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image3MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
    Label1->Caption= ("4 - Хлоратор");
}
//-----
```

Далі натискаємо на **Image4** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **Image4MouseMove** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image4MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
    Label1->Caption= ("1 - Змішувач");
}
//-----
```

Натискаємо на **Image5** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image5MouseMove** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image5MouseMove(TObject *Sender, TShiftState Shift,
```

```

    int X, int Y)
{
Label1->Caption= ("2 - Підігрівник");
}
//-----

```

Натискаємо на **Image6** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image6MouseMove** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image6MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
Label1->Caption= ("1 - Сажовловлювач");
}
//-----

```

Натискаємо на **Image7** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image7MouseMove** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image7MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
Label1->Caption= ("16 - Сепаратор");
}
//-----

```

Натискаємо на **Image8** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image8MouseMove** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image8MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)

```



```
{  
Label1->Caption= ("10 – Графітовий холодильник");  
}  
//-----
```

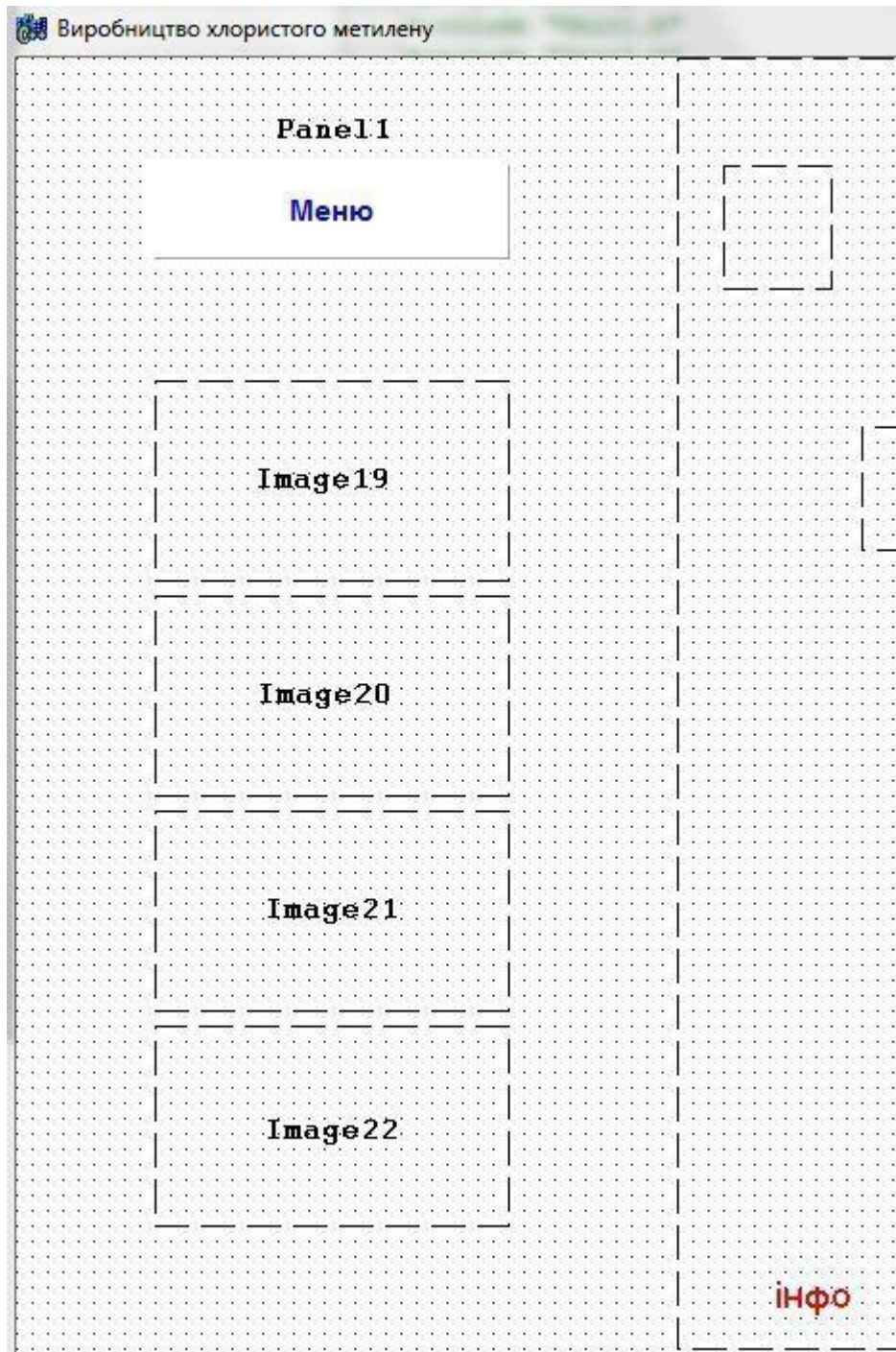


Рис. 4-2. Схема розміщення на Form1 компонент Image19, Image20, Image21, Image22 та Panel1.

Натискаємо на **Image9** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image9MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image9MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("7 – Абсорбер");  
}  
//-----
```

Натискаємо на **Image10** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image10MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image10MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("8 – Абсорбер");  
}  
//-----
```

Натискаємо на **Image11** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image11MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image11MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("9 – Розріджувач луку");  
}  
//-----
```

Натискаємо на **Image12** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо

до вікна редактора коду, де у шаблон функції **Image12MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image12MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("11 – Збірник концентрованої соляної кислоти");  
}  
//-----
```

Натискаємо на **Image13** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду де у шаблон функції **Image13MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image13MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("12 – Компресор");  
}  
//-----
```

Натискаємо на **Image14** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де шаблон функції **Image14MouseMove** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image14MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
Label1->Caption= ("15 – Фільтр");  
}  
//-----
```

Натискаємо на **Image15** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image15MouseMove** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image15MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
    Label1->Caption= ("14 – Конденсатор");
}
//-----
```

Натискаємо на **Image16** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image16MouseMove** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image16MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
    Label1->Caption= ("15 –осушувачі");
}
//-----
```

Натискаємо на **Image17** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image17MouseMove** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image17MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
    Label1->Caption= ("17 – Розріджувач лугу");
}
//-----
```

Натискаємо на **Image18** і у вікні інспектора об'єктів обираємо вкладку **Events** та у полі **OnMouseMove** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **Image18MouseMove** додаємо наступний текст:

```
//-----
void __fastcall TForm1::Image18MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
```

```

{
Label1->Caption= ("6 – Повітряний холодильник");
}
//-----

```

Крок 7. Натискаємо лівою кнопкою мишки на формі по полю **Image1** та у вікні інспектора об'єктів у полі **Picture** натискаємо на значок з трьома крапками. Кнопкою **Clear** видаляємо малюнок фону та натискаємо "OK".

Крок 8. Задаємо для **Image1** завантаження рисунка мнемосхеми та рисунків кнопок до команд графічного меню у **Image19**, **Image20**, **Image21** і **Image22**, які з'являються на екрані при старті прикладної C++ програми.

На вільному просторі форми **Form1** натискаємо лівою кнопкою мишки. У вікні інспектора об'єктів обираємо **Events** та подвійним натисканням мишки у полі **OnCreate** переходимо до вікна редактора коду прикладної програми. Далі у функцію **FormCreate** необхідно додати оператори для завантаження фону в **Image1** та рисунків кнопок з командами до графічного меню у **Image19**..... **Image39**. Також необхідно зробити невидимими поля **Image2**.....**Image18** та активізувати інший візуальний курсор мишки на полях **Image19**, **Image20**, **Image21** і **Image22**:

```

//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
Image19->Picture->LoadFromFile("Data/Icons/Технологія_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Апарати_1.bmp");
Image21->Picture->LoadFromFile("Data/Icons/Інформація_1.bmp");
Image22->Picture->LoadFromFile("Data/Icons/Вихід_1.bmp");
Image1->Picture->LoadFromFile("Data/Main/Background.bmp");
Image2->Visible = false;
Image3->Visible = false;
Image4->Visible = false;
Image5->Visible = false;
Image6->Visible = false;
Image7->Visible = false;
Image8->Visible = false;
Image9->Visible = false;
Image10->Visible = false;

```

```

Image11->Visible = false;
Image12->Visible = false;
Image13->Visible = false;
Image14->Visible = false;
Image15->Visible = false;
Image16->Visible = false;
Image17->Visible = false;
Image18->Visible = false;
Image19->Cursor = crHandPoint;
Image20->Cursor = crHandPoint;
Image21->Cursor = crHandPoint;
Image22->Cursor = crHandPoint;
}
//-----

```

Крок 9. Переходимо на форму **Form1** і натискаємо на **Image3**. У вікні інспектора об'єктів у полі **Cursor** обираємо **crHandPoint** і вкладку **Events** та у полі **OnClick** подвійним натисканням мишки переходимо до редактора коду, де у шаблон функції **Image3Click** додаємо текст:

```

//-----
void __fastcall TForm1::Image3Click(TObject *Sender)
{
Form3->Caption="Конструкція хлоратора";
Form3->Image1->Picture->LoadFromFile("Data/Device1/Device.bmp");
Form3->ShowModal();
}
//-----

```

Крок 10. Переходимо на форму **Form1** і натискаємо на **Image9**. У вікні інспектора об'єктів у полі **Cursor** обираємо **crHandPoint** і вкладку **Events** та у полі **OnClick** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у функцію **Image9Click** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image9Click(TObject *Sender)
{
Form3->Caption="Конструкція абсорбера";
Form3->Image1->Picture->LoadFromFile("Data/Device2/Device.bmp");
Form3->ShowModal();
}
//-----

```

Крок 11. Переходимо на форму **Form1** і натискаємо на **Image10**. У вікні інспектора об'єктів у полі **Cursor** обираємо **crHandPoint** і вкладку **Events** та у полі **OnClick** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image10Click** додаємо текст:

```
//-----  
void __fastcall TForm1::Image10Click(TObject *Sender)  
{  
Form3->Caption="Конструкція абсорбера";  
Form3->Image1->Picture->LoadFromFile("Data/Device2/Device.bmp");  
Form3->ShowModal();  
}  
//-----
```

Крок 12. Переходимо на форму **Form1** та натискаємо на **Image11**. У вікні інспектора об'єктів у полі **Cursor** обираємо **crHandPoint** і вкладку **Events** та у полі **OnClick** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image11Click** додаємо текст:

```
//-----  
void __fastcall TForm1::Image11Click(TObject *Sender)  
{  
Form3->Caption="Конструкція розріджувача луку";  
Form3->Image1->Picture->LoadFromFile("Data/Device3/Device.bmp");  
Form3->ShowModal();  
}  
//-----
```

Тепер при натисканні на компонентах **Image**, яки розташовані у вікні на місцях назв апаратів: “Хлоратор”, “Абсорбер” і “Розріджувач луку” буде відкриватися форма **Form3** з окремим рисунком схеми відповідного апарату.

Папка “Data” знаходиться у папці з файлами проекту прикладної програми. Файл “Background.bmp” є основним фоновим рисунком у вікні прикладної C++ програми з контрольної роботи № 2 (варіант графічного меню № 2). Файли з папки “Data/Icons” є рисунками кнопок до команд графічного меню програми. Файли з індексом 1 відповідають пасивному стану кнопок у меню команд, а файли з індексом 2 відповідають активному стану кнопки. Далі

створюємо додаткові форми, які знадобляться для відображення інформації при виконанні підкоманд графічного меню команд.

Створення і налаштування форми Form2.

Крок 1. Виконуємо команду **File/New/Form** для створення нової чистої форми **Form2**. Встановлюємо на **Form2** компоненту **RichEdit** з сторінки **Win32** у бібліотеці компонентів. Натискаємо на новостворену компоненту **RichEdit1** та у полі **Align** інспектора об'єктів обираємо **alClient**. Таким чином **RichEdit1** буде розтягуватись на екрані до розмірів форми **Form2**.

Створення і налаштування форми Form3.

Крок 1. Виконуємо команду **File/New/Form** для створення нової чистої форми **Form3**. Через дерево об'єктів обираємо **Form3** та у полі **AutoSize** інспектора об'єктів обираємо **true**. Встановлюємо на **Form3** компоненту **Image** із сторінки **Additional** у бібліотеці компонентів. Натискаємо на полі новоствореної компоненти **Image1** і у полі **Width** вікна інспектора об'єктів вводимо розмір «385», а у полі **Height** вводимо розмір «497».

Створення і налаштування форми Form4.

Крок 1. Виконуємо команду **File/New/Form** для створення нової чистої форми **Form4**. Встановлюємо на полі **Form4** компоненту **Chart** із сторінки **Additional** у бібліотеці компонентів.

Крок 2. Натискаємо правою кнопкою мишки на полі новоствореної компоненти **Chart1** та обираємо **Edit Chart**. Далі переходимо на вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо “OK”. Переходимо у вкладку **Chart/Titles** та у полі вводу набираємо для параметру назву “Температура”. Переходимо у вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** та виходимо з меню **Edit Chart**.

Крок 3. Встановлюємо на нолі **Form4** компоненту **Chart** із сторінки **Additional** у бібліотеці компонентів.

Крок 4. Натискаємо правою кнопкою мишки на полі новоствореної компоненти **Chart2** та обираємо **Edit Chart**. Далі переходимо на вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо “**OK**”. Переходимо у вкладку **Chart/Titles** та у полі вводу набираємо для параметру назву “**Витрата сировини**”. Переходимо у вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** та виходимо з меню **Edit Chart**.

Крок 5. У дереві об’єктів обираємо **Form4**. У вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnCreate** подвійним натисканням кнопки мишки переходимо до вікна редактора коду і у функцію **FormCreate** додаємо наступний текст:

```
//-----  
void __fastcall TForm4::FormCreate(TObject *Sender)  
{  
for (int i=0; i<=10; i++)  
{  
Series1->AddXY(i+cos(i),13*i*i,"",clRed);  
Series2->AddXY(12*i*i+25, i*i+15*i,"",clRed);  
}  
}  
//-----
```

Створення і налаштування форми Form5.

Крок 1. Виконуємо команду **File/New/Form** для створення нової чистої форми **Form5**. Встановлюємо на полі **Form5** компоненту **Chart** із сторінки **Additional** у бібліотеці компонентів.

Крок 2. Натискаємо правою кнопкою мишки на полі новоствореної компоненти **Chart1** та обираємо **Edit Chart**. Далі переходимо на вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо “**OK**”. Переходимо у вкладку **Chart/Titles** та у полі вводу набираємо для параметру назву “**Температура**”. Переходимо у вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** та виходимо з меню **Edit Chart**.

Крок 3. Встановлюємо на полі **Form5** компоненту **Chart** із сторінки **Additional** у бібліотеці компонентів.

Крок 4. Натискаємо правою кнопкою мишки на полі новоствореної компоненти **Chart2** та обираємо **Edit Chart**. Далі переходимо на вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо “**OK**”. Переходимо у вкладку **Chart/Titles** та у полі вводу набираємо для параметру “**Витрата сировини**”. Переходимо у вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** та виходимо з меню **Edit Chart**.

Крок 5. У дереві об’єктів обираємо **Form5**. У вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnCreate** подвійним натисканням мишки переходимо до вікна редактора коду, де у шаблон функції **FormCreate** додаємо наступний текст:

```
//-----  
void __fastcall TForm5::FormCreate(TObject *Sender)  
{  
for (int i=0; i<=10; i++)  
{  
Series1->AddXY(i+sin(i),13*i*i,"",clRed);  
Series2->AddXY(14*i+25*i*sin(i), cos(i)*i*i*i+15*i,"",clRed);  
}  
}  
//-----
```

Створення і налаштування форми Form6.

Крок 1. Виконуємо команду **File/New/Form** для створення нової чистої форми **Form6**. Встановлюємо на **Form6** компоненту **Chart** з сторінки **Additional** у бібліотеці компонентів.

Крок 2. Натискаємо правою кнопкою мишки на полі новоствореної компоненти **Chart1** та обираємо **Edit Chart**. Далі переходимо на вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо “**OK**”. Переходимо у вкладку **Chart/Titles** та у полі вводу набираємо для параметру назву “**Температура**”. Переходимо у вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** та виходимо з меню **Edit Chart**.

Крок 3. Встановлюємо на полі **Form6** компоненту **Chart** із сторінки **Additional** у бібліотеці компонентів.

Крок 4. Натискаємо правою кнопкою мишки на полі новоствореної компоненти **Chart2** та обираємо **Edit Chart**. Далі переходимо на вкладку **Chart/Series**, натискаємо **Add**, обираємо **Fast Line** та натискаємо “**OK**”. Переходимо у вкладку **Chart/Titles** та у полі вводу набираємо для параметру “**Витрата сировини**”. Переходимо на вкладку **Chart/Legend** та знімаємо галочку з пункту **Visible**. Натискаємо **Close** та виходимо з меню **Edit Chart**.

Крок 5. У дереві об’єктів обираємо **Form6** і у вікні інспектора об’єктів обираємо вкладку **Events** та у полі **OnCreate** подвійним натисканням мишки переходимо до вікна редактора коду, де у функцію **FormCreate** додаємо наступний текст:

```
//-----  
void __fastcall TForm6::FormCreate(TObject *Sender)  
{  
for (int i=0; i<=10; i++)  
{  
Series1->AddXY(i+tan(i),13*i*i,"",clRed);  
Series2->AddXY(13*i+25*tan(i), exp(i)*i*i+15*i,"",clRed);  
}  
}  
//-----
```

Створення і налаштування форми AboutBox.

Крок 1. Виконуємо команду **File/New/Other** та обираємо там **Aboutbox** для створення нової форми **AboutBox**. Через дерево об’єктів обираємо форму **AboutBox** та у полі **Caption** інспектора об’єктів вводимо “Про програму”.

Крок 2. Через дерево об’єктів обираємо **Copyright** та у полі **Caption** інспектора об’єктів вводимо “Розробив **_ПІБ**”.

Крок 3. Через дерево об’єктів обираємо **Label1** та у полі **Caption** інспектора об’єктів вводимо “Група **_№ групи**”.

Крок 4. Через дерево об’єктів обираємо **ProductName** та у інспекторі об’єктів в полі **Caption** вводимо “Модульна контрольна робота”.

Крок 5. Через дерево об'єктів обираємо **Version** і у вікні інспектора об'єктів в полі **Caption** вводимо “**Версія 2.0**” та у вікні редактора коду зверху додамо наступний текст:

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include "Unit1.h"  
#include "Unit2.h"  
#include "Unit3.h"  
#include "Unit4.h"  
#include "Unit5.h"  
#include "Unit6.h"  
#include "Unit7.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//----- Об'ява допоміжних змінних -----  
int ur=0;  
int sh=0;
```

Це дає змогу викликати на виконання **Form2**, **Form3**, **Form4**, **Form5**, **Form6** і **Form7** знаходячись на формі **Form1**. Також ми введемо ряд змінних, які допоможуть нам реалізувати роботу графічного меню та уникнення певних проблем (запобігання мерехтіння зображень і т.п.).

4.2.1 Програмування змінювання рисунків кнопок у графічному меню команд C++ програми

Дії для оновлення графічного меню “Технологія”.

Коли користувач знаходиться у загальному графічному меню команд необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на полі компоненти **Image19** одночасно у компонентах **Image19**, **Image20**, **Image21** і **Image22** з'являлися нові рисунки кнопок, які відповідають командам з меню “Технологія”. Тобто у **Image19** необхідно розміщувати рисунок кнопки до

підменю “Схема”, у **Image20** – рисунок кнопки до підменю “Опис схеми”, у **Image21** – рисунок кнопки до підменю “Продукція”, а у компоненту **Image22** розміщувати рисунок кнопки з написом “Назад”.

Також необхідно реалізувати і повернення у загальне графічне меню. Відповідно коли користувач знаходиться у меню “Технологія” необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на полі компоненти **Image22** одночасно у компоненти **Image19**, **Image20**, **Image21** та **Image22** розміщувались рисунки кнопок, які б відповідали пунктам загального меню. Тобто у **Image19** відновлювався рисунок кнопки “Технологія”, в **Image20** – рисунок кнопки “Апарати”, а у **Image21** – рисунок кнопки “Інформація” і відповідно у **Image22** – рисунок кнопки “Вихід”.

Дії для оновлення графічного меню “Апарати”.

При натисканні лівої кнопки маніпулятора “мишка” на компоненті **Image20** одночасно у компоненти **Image19**, **Image20**, **Image21** та **Image22** повинні розміщуватися нові рисунки кнопок, які відповідають за виконання команд у підменю “Апарати”. Тобто на полі **Image19** необхідно розміщувати рисунок кнопки підменю “Хлоратор”, у **Image20** – рисунок кнопки підменю “Абсорбер”, у **Image21** – рисунок кнопки підменю “Розріджувач луку”, а у **Image22** – рисунок кнопки з назвою “Назад”.

Необхідно виконувати і повернення у загальне графічне меню, коли користувач знаходиться у меню “Апарати”. Для цього необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на компоненті **Image22** у компоненти **Image19**, **Image20**, **Image21**, **Image22** розміщувались рисунки кнопок, які відповідають командам загального меню. Тобто на полі **Image19** розміщувався рисунок кнопки “Технологія”, на полі **Image20** – відповідно рисунок кнопки “Апарати”, на полі **Image21** – рисунок кнопки “Інформація”, а на полі **Image22** розміщувався рисунок кнопки “Вихід”.

Дії для оновлення графічного підменю “Хлоратор”.

Для випадку, коли користувач вибрав у графічному меню “Апарати” необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на полі компоненти **Image19** одночасно у компоненти **Image19**, **Image20**, **Image21**, **Image22** розміщувались відповідні рисунки кнопок до підменю “Хлоратор». Тобто на **Image19** розміщувався відповідний рисунок кнопки підкоманди “Конструкція”, на **Image20** – підкоманди “Опис процесу”, на полі **Image21** – підкоманди “Параметри”, а на полі **Image22** розміщувався рисунок кнопки з командою “Назад”.

Коли користувач знаходиться у підменю “Хлоратор”, також потрібно передбачити і повернення у попередній стан графічного меню. Для цього необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на полі компоненті **Image22** одночасно на полях компонент **Image19**, **Image20**, **Image21** та на **Image22** розміщувались відповідні рисунки кнопок до підменю з меню “Апарати”. Тобто на полі **Image19** розміщувався рисунок кнопки до підменю “Хлоратор”, на **Image20** – до підменю “Абсорбер”, на **Image21** – до підменю “Розріджувач лугу”, а на полі **Image22** розміщувався рисунок кнопки з командою “Назад”.

Дії для оновлення графічного підменю “Абсорбер”.

Для випадку, коли користувач у графічному меню вибрав “Апарати” необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на компоненті **Image20** одночасно на полях компонент **Image19**, **Image20**, **Image21** та **Image22** розміщувались відповідні рисунки кнопок до підменю “Абсорбер”. Тобто на **Image19** розміщувався відповідний рисунок кнопки підкоманди “Конструкція”, на **Image20** – підкоманди “Опис процесу”, на **Image21** – під-

команди “Параметри”, а на **Image22** розміщувався рисунок кнопки з командою “Назад”.

Коли користувач знаходиться у підменю “Абсорбер”, також потрібно передбачити і повернення у попередній стан графічного меню. Для цього необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на полі компоненті **Image22** одночасно на полях компонент **Image19**, **Image20**, **Image21** і на **Image22** розміщувались відповідні рисунки кнопок до підменю з меню “Апарати”. Тобто на полі **Image19** розміщувався рисунок кнопки до підменю “Хлоратор”, на **Image20** – до підменю “Абсорбер”, на **Image21** – до підменю “Розріджувач лугу”, а на полі **Image22** розміщувався рисунок кнопки з командою “Назад”.

Дії для оновлення графічного підменю “Розріджувач лугу”.

Для випадку, коли користувач у графічному меню вибрав “Апарати” необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на компоненті **Image21** одночасно на полях компонент **Image19**, **Image20**, **Image21** та **Image22** розміщувались відповідні рисунки кнопок до підменю “Розріджувач лугу”. Тобто на **Image19** розміщувався відповідний рисунок кнопки підкоманди “Конструкція”, на **Image20** – підкоманди “Опис процесу», на **Image21** – підкоманди “Параметри”, а на полі компоненти **Image22** розміщувався рисунок кнопки з командою “Назад”.

Коли користувач знаходиться у підменю “Розріджувач лугу”, також потрібно передбачити і повернення у попередній стан графічного меню. Для цього необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на полі компоненті **Image22** одночасно на полях компонент **Image19**, **Image20**, **Image21** і на **Image22** розміщувались відповідні рисунки кнопок до підменю з меню “Апарати”. Тобто на полі **Image19** розміщувався рисунок кнопки до підменю “Хлоратор”, на **Image20** – до підменю “Абсорбер”, на **Image21** – до

підменю “Розріджувач лугу”, а на полі **Image22** розміщувався рисунок кнопки з командою “Назад”.

Дії для оновлення графічного меню “Інформація”.

Коли користувач знаходиться у загальному графічному меню команд необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на полі компоненти **Image21** одночасно у компонентах **Image19**, **Image20**, **Image21** і **Image22** з’являлися нові рисунки кнопок, які відповідають командам з меню “Інформація”. Тобто у **Image19** необхідно розміщувати рисунок кнопки до підменю “Про програму”, у **Image20** необхідно розміщувати рисунок кнопки до підменю “Література”, а у **Image22** розміщувати рисунок кнопки з командою “Назад”, а компонента **Image21** ставала невидимою.

Також необхідно реалізувати і повернення у загальне графічне меню. Відповідно коли користувач знаходиться у меню “Інформація” необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на полі компоненти **Image21** одночасно у компоненти **Image19**, **Image20**, **Image21** та **Image22** розміщувались рисунки кнопок, які б відповідали пунктам загального меню. Тобто у **Image19** відновлювався рисунок кнопки “Технологія”, у **Image20** – рисунок кнопки “Апарати”, у **Image21** – рисунок кнопки “Інформація”, а у **Image22** – рисунок кнопки “Вихід”. Коли користувач набуває на екрані загальне графічне меню команд, тоді необхідно, щоб при натисканні лівої кнопки маніпулятора “мишка” на компоненті **Image22** виконувався вихід з прикладної C++ програми.

4.2.2 Програмування функцій для компонент “Image” з графічного меню команд C++ програми

Крок 1. Натискаємо кнопкою мишки на полі **Image19**. Далі у вікні інспектора об’єктів обираємо вкладку **Events** і у полі **OnClick** подвійним натискан-

ням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції

Image19Click додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image19Click(TObject *Sender)  
{  
if(ur==0)  
{  
Image19->Picture->LoadFromFile("Data/Icons/Схема_1.bmp");  
Image20->Picture->LoadFromFile("Data/Icons/Опис_схеми_1.bmp");  
Image21->Picture->LoadFromFile("Data/Icons/Продукція_1.bmp");  
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");  
ur=1;  
Panel1->Caption = "Технологія";  
}  
else  
{  
if(ur==1)  
{  
if(sh==0)  
{  
Image1->Picture->LoadFromFile("Data/Main/Scheme.bmp");  
sh=1;  
}  
Image2->Visible = true;  
Image3->Visible = true;  
Image4->Visible = true;  
Image5->Visible = true;  
Image6->Visible = true;  
Image7->Visible = true;  
Image8->Visible = true;  
Image9->Visible = true;  
Image10->Visible = true;  
Image11->Visible = true;  
Image12->Visible = true;  
Image13->Visible = true;  
Image14->Visible = true;  
Image15->Visible = true;  
Image16->Visible = true;  
Image17->Visible = true;  
Image18->Visible = true;  
Label1->Visible = true;
```

```

}
if(ur==2)
{
Image19->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Опис Процесу_1.bmp");
Image21->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
ur=3;
Panel1->Caption = "Хлоратор";
}
else
{
if(ur==3)
{
Form3->Caption="Конструкція хлоратора";
Form3->Image1->Picture->LoadFromFile("Data/Device1/Device.bmp");
Form3->ShowModal();
}
if(ur==4)
{
Form3->Caption="Конструкція абсорбера";
Form3->Image1->Picture->LoadFromFile("Data/Device2/Device.bmp");
Form3->ShowModal();
}
if(ur==5)
{
Form3->Caption="Конструкція розріджувача луѓу";
Form3->Image1->Picture->LoadFromFile("Data/Device3/Device.bmp");
Form3->ShowModal();
}
if(ur==6)
{
AboutBox->ShowModal();
}
}
}
}
}
//-----

```

Крок 2. Натискаємо кнопкою мишки на полі **Image20**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnClick** подвійним натискан-

ням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image20Click** додаємо наступний текст:

```
//-----  
void __fastcall TForm1::Image20Click(TObject *Sender)  
{  
if(ur==0)  
{  
Image19->Picture->LoadFromFile("Data/Icons/Хлоратор_1.bmp");  
Image20->Picture->LoadFromFile("Data/Icons/Абсорбер_1.bmp");  
Image21->Picture->LoadFromFile("Data/Icons/Розріджувач луку_1.bmp");  
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");  
Panel1->Caption = "Апарати";  
ur=2;  
}  
else  
{  
if(ur==1)  
{  
Form2->RichEdit1->Lines->LoadFromFile("Data/Main/Scheme.rtf");  
Form2->Caption="Опис процесу виробництва хлорметанів";  
Form2->ShowModal();  
}  
if(ur==2)  
{  
Image19->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");  
Image20->Picture->LoadFromFile("Data/Icons/Опис Процесу_1.bmp");  
Image21->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");  
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");  
Panel1->Caption = "Абсорбер";  
ur=4;  
}  
else  
{  
if(ur==3)  
{  
Form2->RichEdit1->Lines->LoadFromFile("Data/Device1/Process.rtf");  
Form2->Caption="Опис процесу в хлораторі";  
Form2->ShowModal();  
}  
if(ur==4)  
{
```

```

Form2->RichEdit1->Lines->LoadFromFile("Data/Device2/Process.rtf");
Form2->Caption="Опис процесу в абсорбері";
Form2->ShowModal();
}
if(ur==5)
{
Form2->RichEdit1->Lines->LoadFromFile("Data/Device3/Process.rtf");
Form2->Caption="Опис процесу в розріджувачі луку";
Form2->ShowModal();
}
if(ur==6)
{
Form2->Caption= ("Література");
Form2->RichEdit1->Lines->LoadFromFile("Data/Main/Literature.rtf");
Form2->ShowModal();
}
}
}
}
}
//-----

```

Крок 3. Натискаємо кнопкою мишки на полі **Image21**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnClick** подвійним натисканням кнопки мишки переходимо до редактора коду, де у шаблон функції **Image21Click** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image21Click(TObject *Sender)
{
if(ur==0)
{
Image19->Picture->LoadFromFile("Data/Icons/Про програму_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Література_1.bmp");
Image21->Visible = false;
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
Panel1->Caption = "Інформація";
ur=6;
}
else
{
if(ur==1)
{

```

```

Form2->RichEdit1->Lines->LoadFromFile("Data/Main/Products.rtf");
Form2->Caption="Продукція";
Form2->ShowModal();
}
if(ur==2)
{
Image19->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Опис_процесу_1.bmp");
Image21->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
Panel1->Caption = "Розріджувач лугу";
ur=5;
}
else
{
if(ur==3)
{
Form4->ShowModal();
}
if(ur==4)
{
Form5->ShowModal();
}
if(ur==5)
{
Form6->ShowModal();
}
}
}
}
}
}
//-----

```

Крок 4. Натискаємо кнопкою мишки на полі **Image22**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnClick** подвійним натисканням кнопки мишки переходимо до редактора коду, де у шаблон функції **Image22Click** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image22Click(TObject *Sender)
{
if(ur==0)

```

```

{
Close();
}
else
{
if(ur==1)
{
Image19->Picture->LoadFromFile("Data/Icons/Технологія_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Апарати_1.bmp");
Image21->Picture->LoadFromFile("Data/Icons/Інформація_1.bmp");
Image22->Picture->LoadFromFile("Data/Icons/Вихід_1.bmp");
ur=0;
Panel1->Caption = "Меню";
}
if(ur==2)
{
Image19->Picture->LoadFromFile("Data/Icons/Технологія_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Апарати_1.bmp");
Image21->Picture->LoadFromFile("Data/Icons/Інформація_1.bmp");
Image22->Picture->LoadFromFile("Data/Icons/Вихід_1.bmp");
ur=0;
Panel1->Caption = "Меню";
}
if(ur==3)
{
Image19->Picture->LoadFromFile("Data/Icons/Хлоратор_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Абсорбер_1.bmp");
Image21->Picture->LoadFromFile("Data/Icons/Розріджувач лугу_1.bmp");
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
ur=2;
Panel1->Caption = "Апарати";
}
if(ur==4)
{
Image19->Picture->LoadFromFile("Data/Icons/Хлоратор_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Абсорбер_1.bmp");
Image21->Picture->LoadFromFile("Data/Icons/Розріджувач лугу_1.bmp");
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
ur=2;
Panel1->Caption = "Апарати";
}
if(ur==5)

```

```

{
Image19->Picture->LoadFromFile("Data/Icons/Хлоратор_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Абсорбер_1.bmp");
Image21->Picture->LoadFromFile("Data/Icons/Розріджувач лугу_1.bmp");
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
ur=2;
Panel1->Caption = "Апарати";
}
if(ur==6)
{
Image19->Picture->LoadFromFile("Data/Icons/Технологія_1.bmp");
Image20->Picture->LoadFromFile("Data/Icons/Апарати_1.bmp");
Image21->Visible = true;
Image21->Picture->LoadFromFile("Data/Icons/Інформація_1.bmp");
Image22->Picture->LoadFromFile("Data/Icons/Вихід_1.bmp");
Panel1->Caption = "Меню";
ur=0;
}
}
}
}
//-----

```

Крок 5. Натискаємо кнопкою мишки на полі **Image19**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnMouseDown** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image19MouseDown** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image19MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
if(ur==0)
{
Image19->Picture->LoadFromFile("Data/Icons/Технологія_2.bmp");
}
if(ur==1)
{
Image19->Picture->LoadFromFile("Data/Icons/Схема_2.bmp");
}
if(ur==2)
{

```

```

Image19->Picture->LoadFromFile("Data/Icons/Хлоратор_2.bmp");
}
if(ur==3)
{
Image19->Picture->LoadFromFile("Data/Icons/Конструкція_2.bmp");
}
if(ur==4)
{
Image19->Picture->LoadFromFile("Data/Icons/Конструкція_2.bmp");
}
if(ur==5)
{
Image19->Picture->LoadFromFile("Data/Icons/Конструкція_2.bmp");
}
if(ur==6)
{
Image19->Picture->LoadFromFile("Data/Icons/Про програму_2.bmp");
}
}
}
//-----

```

Крок 6. Натискаємо кнопкою мишки на полі **Image19**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image19MouseUp** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image19MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
if(ur==0)
{
Image19->Picture->LoadFromFile("Data/Icons/Технологія_1.bmp");
}
if(ur==1)
{
Image19->Picture->LoadFromFile("Data/Icons/Схема_1.bmp");
}
if(ur==2)
{
Image19->Picture->LoadFromFile("Data/Icons/Хлоратор_1.bmp");
}
}
}

```



```

}
if(ur==3)
{
Image19->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");
}
if(ur==4)
{
Image19->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");
}
if(ur==5)
{
Image19->Picture->LoadFromFile("Data/Icons/Конструкція_1.bmp");
}
if(ur==6)
{
Image19->Picture->LoadFromFile("Data/Icons/Про програму_1.bmp");
}
}
}
//-----

```

Крок 7. Натискаємо кнопкою мишки на полі **Image20**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnMouseDown** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image20MouseDown** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image20MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
if(ur==0)
{
Image20->Picture->LoadFromFile("Data/Icons/Апарати_2.bmp");
}
if(ur==1)
{
Image20->Picture->LoadFromFile("Data/Icons/Опис схеми_2.bmp");
}
if(ur==2)
{
Image20->Picture->LoadFromFile("Data/Icons/Абсорбер_2.bmp");
}
}
}

```

```

if(ur==3)
{
Image20->Picture->LoadFromFile("Data/Icons/Опис_процесу_2.bmp");
}
if(ur==4)
{
Image20->Picture->LoadFromFile("Data/Icons/Опис_процесу_2.bmp");
}
if(ur==5)
{
Image20->Picture->LoadFromFile("Data/Icons/Опис_процесу_2.bmp");
}
if(ur==6)
{
Image20->Picture->LoadFromFile("Data/Icons/Література_2.bmp");
}
}
}
//-----

```

Крок 8. Натискаємо кнопкою мишки на полі **Image20**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image20MouseUp** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image20MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
if(ur==0)
{
Image20->Picture->LoadFromFile("Data/Icons/Апарати_1.bmp");
}
if(ur==1)
{
Image20->Picture->LoadFromFile("Data/Icons/Опис_схеми_1.bmp");
}
if(ur==2)
{
Image20->Picture->LoadFromFile("Data/Icons/Абсорбер_1.bmp");
}
if(ur==3)

```

```

{
Image20->Picture->LoadFromFile("Data/Icons/Опис процесу_1.bmp");
}
if(ur==4)
{
Image20->Picture->LoadFromFile("Data/Icons/Опис процесу_1.bmp");
}
if(ur==5)
{
Image20->Picture->LoadFromFile("Data/Icons/Опис процесу_1.bmp");
}
if(ur==6)
{
Image20->Picture->LoadFromFile("Data/Icons/Література_1.bmp");
}
}
}
//-----

```

Крок 9. Натискаємо кнопкою мишки на полі **Image21**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnMouseDown** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image21MouseDown** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image21MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
if(ur==0)
{
Image21->Picture->LoadFromFile("Data/Icons/Інформація_2.bmp");
}
if(ur==1)
{
Image21->Picture->LoadFromFile("Data/Icons/Продукція_2.bmp");
}
if(ur==2)
{
Image21->Picture->LoadFromFile("Data/Icons/Розріджувач лугу_2.bmp");
}
if(ur==3)
{

```

```

Image21->Picture->LoadFromFile("Data/Icons/Параметри_2.bmp");
}
if(ur==4)
{
Image21->Picture->LoadFromFile("Data/Icons/Параметри_2.bmp");
}
if(ur==5)
{
Image21->Picture->LoadFromFile("Data/Icons/Параметри_2.bmp");
}
}
}
//-----

```

Крок 10. Натискаємо кнопкою мишки на полі **Image21**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image21MouseUp** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image21MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
if(ur==0)
{
Image21->Picture->LoadFromFile("Data/Icons/Інформація_1.bmp");
}
if(ur==1)
{
Image21->Picture->LoadFromFile("Data/Icons/Продукція_1.bmp");
}
if(ur==2)
{
Image21->Picture->LoadFromFile("Data/Icons/Розріджувач лугу_1.bmp");
}
if(ur==3)
{
Image21->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");
}
if(ur==4)
{
Image21->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");
}
}
}

```

```

}
if(ur==5)
{
Image21->Picture->LoadFromFile("Data/Icons/Параметри_1.bmp");
}
}
//-----

```

Крок 11. Натискаємо кнопкою мишки на полі **Image22**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnMouseDown** подвійним натисканням кнопки мишки переходимо до вікна редактора коду, де у шаблон функції **Image22MouseDown** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image22MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
if(ur==0)
{
Image22->Picture->LoadFromFile("Data/Icons/Вихід_2.bmp");
}
if(ur==1)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_2.bmp");
}
if(ur==2)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_2.bmp");
}
if(ur==3)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_2.bmp");
}
if(ur==4)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_2.bmp");
}
if(ur==5)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_2.bmp");
}
}

```

```

if(ur==6)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_2.bmp");
}
}
//-----

```

Крок 12. Натискаємо кнопкою мишки на полі **Image22**. Далі у вікні інспектора об'єктів обираємо вкладку **Events** і у полі **OnMouseUp** подвійним натисканням кнопки мишки переходимо до редактора коду, де у шаблон функції **Image22MouseUp** додаємо наступний текст:

```

//-----
void __fastcall TForm1::Image22MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
if(ur==0)
{
Image22->Picture->LoadFromFile("Data/Icons/Вихід_1.bmp");
}
if(ur==1)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
}
if(ur==2)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
}
if(ur==3)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
}
if(ur==4)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
}
if(ur==5)
{
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");
}
if(ur==6)

```

```
{  
Image22->Picture->LoadFromFile("Data/Icons/Назад_1.bmp");  
}  
}  
//-----
```

5. ЛІТЕРАТУРА

1. Архангельский, А. Я. Программирование в С++ Builder 6 [Текст] / А. Я. Архангельский // – М.: ЗАО «Издательство БИНОМ», 2002. – 1152 с. Библиогр.: с. 1150–1151. 4000 экз. ISBN 5-7989-0239-0.
2. Архангельский, А. Я. С++ Builder 6. Справочное пособие. Книга 1. Язык С++ [Текст] / А. Я. Архангельский // – М.: ЗАО «Издательство БИНОМ», 2002. – 554 с. Библиогр.: с. 541–543. 4000 экз. ISBN 5-9518-0007-2.
3. Архангельский, А. Я. С++ Builder 6. Справочное пособие. Книга 2. Классы и компоненты [Текст] / А. Я. Архангельский // – М.: ЗАО «Издательство БИНОМ», 2002. – 528 с. Библиогр.: с. 525–526. 4000 экз. ISBN 5-9518-0009-9.
4. Культин, Н. Б. Самоучитель С++ Builder [Текст] / Н. Б. Культин // – СПб.: БХВ-Петербург, 2004. – 320 с. Библиогр.: с. 317. 4000 экз. ISBN 5-94157-378-2.
5. Шилд, Г. Полный справочник по С++ [Текст] / Г. Шилд // 4-е издание. : Пер. с англ. – М.: Издательский дом “Вильямс”, 2006. – 800 с. 3000 экз. ISBN 5-8459-0489-7.

Додатки:

Додаток Д1. Схема апарату «Розріджувач лугу»

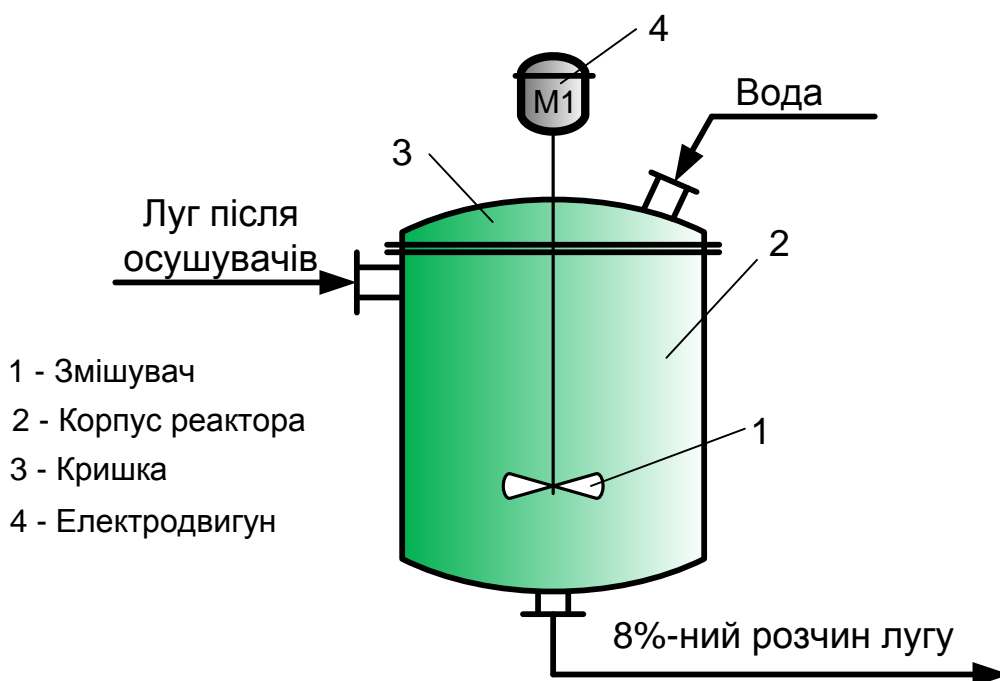


Рис. Д1. Схема конструкції апарату «Розріджувач лугу».

Додаток Д2. Приклад опису технологічного процесу апарату «Розріджувач лугу»

Опис технологічного процесу з розріджування лугу

Процес розріджування лугу виконується в хімічному реакторі методом змішування потоку після осушувачів з потоком води. За допомогою мешалки змішуються речовини і таким чином робиться 8 % розчин лугу для подачі на нейтралізаційну колону. Процес у реакторі це перемішування речовин однакового агрегатного стану з одержанням гомогенного 8 % розчину лугу. Методи перемішування, конструкції пристроїв, що перемішують, і їхні робочі режими залежать від агрегатного стану і фізичних властивостей речовин, що перемішуються, а також від вимог, пред'являємим до одержання суміші. Суміш може бути однофазної (розчин) чи двофазної (іноді багатофазної). Дуже часто зустрічаються двофазні суміші, у яких

суцільною фазою являється рідина, дисперсною масою - дрібні краплі іншої нерозчинної рідини чи газіві пухирці, тверді частки. В усіх випадках пристрій, що переміщує робочу масу в апараті (хімічному реакторі), повинен забезпечувати одержання однорідної суміші при максимальній продуктивності і мінімальній витраті енергії. У хімічних виробництвах широко застосовуються чотири методи перемішування (змішування) в рідких середовищах: 1) за допомогою механічних мішалок з обертальним чи коливальним рухом; 2) барбатажний - шляхом подачі в рідке середовище газу чи пари; 3) розмішування в потоці нерухомих турболізуючих пристроїв; 4) з використанням струминних і відцентрових насосів. Для процесу розріджування луку в реакторі використовується механічна мішалка, тому розглянемо особливості роботи механічних мішалок у рідкому середовищі. Перемішування рідини будь-яким методом зводиться до багаторазового відносного переміщення елементів в обсязі маси. Складний рух рідини, що виникає в апараті (реакторі) при обертанні мішалки, можна розкласти на складові: радіальну (обертання уздовж радіуса); тангенціальну (по касательній до окружності, описуваної кінцем мішалки); осьову (уздовж осі вала). У мішалках різних конструкцій ці складові знаходяться в різних співвідношеннях. Існують, однак, мішалки, у яких дві чи навіть усі три складові порівнянні, тому класифікують механічні мішалки по їх конструктивних ознаках. Найбільш простою конструкцією є плоско-лопастна мішалка, що складається з ряду вертикальних плоских лопатей прямокутної форми, закріплених до валу, що обертається електродвигуном. При роботі такої мішалки утворюється лійка і може всмоктуватися повітря з поверхні змішуваної маси. Величина лійки залежить від розмірів апарату (реактора), а також розмірів лопатей та їх кількості на валу. Листові мішалки являють собою плоско-лопастеву мішалку з однією великою і високою лопатю на валу. Ці мішалки створюють обертальний рух всього обсягу маси в апараті і тому на внутрішній частині корпусу апарату встановлюються відбивні ребра. Для інтенсивного процесу перемішування рідин застосовуються швидкохідні пропелерні мішалки. Робочим елементом такої мішалки є лопаті (від 2 до 6 штук) з безупинним нахилом, що змінюється, і вони прикріплені до втулки вала та схожі на гребний гвинт. При роботі пропелерної мішалки виникають циркуляційні потоки з насосним ефектом, який залежить від в'язкості змішуваних рідин. Пропелерні мішалки застосовуються для перемішування маловязких рідин для процесів розчинення і хімічного перетворення. Технологічний процес розріджування луку виконується в хімічному реакторі з пропелерною мішалкою.

Рис. Д2. Приклад опису технологічного процесу.