

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

*Спеціальність: 121 «Інженерія програмного забезпечення»*

*Освітня програма: «Розробка програмного забезпечення»*

*Група: 4РП-05*

# **Дипломний проект**

**здобувача освіти денної форми навчання  
РП.05.20.000.ДП**

***СИНЯЄВА  
МИКИТИ ВОЛОДИМИРОВИЧА***

**м. Одеса  
2022 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **121 «Інженерія програмного забезпечення»**

Освітня програма: **«Розробка програмного забезпечення»**

Група: **4РП-05**

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

### **Розробка алгоритмічного та програмного забезпечення для моделювання та візуалізації динамічної системи**

Проектний матеріал складається з пояснювальної записки на 69 сторінках та графічного (презентаційного) матеріалу на 18 аркушах (слайдах).

Дипломник \_\_\_\_\_ (Синяєв М.В.)

Керівник \_\_\_\_\_ (Кривченко Ю.В.)

#### **Консультанти:**

з економічної частини \_\_\_\_\_ (Копайгородська Т.Г.)

з охорони праці \_\_\_\_\_ (Чорновол Н.І.)

з дотримання вимог ЄСКД \_\_\_\_\_ (Петрашова В.І.)

старший консультант \_\_\_\_\_ (Скорнякова О.В.)

#### **До захисту допущений**

Голова циклової комісії \_\_\_\_\_ (Скорнякова О.В.)

Завідувач відділення \_\_\_\_\_ (Суліма Ю.Ю.)

Захист «    » \_\_\_\_\_ 2022 р.

Протокол ДКК № \_\_\_\_\_

Оцінка ДКК \_\_\_\_\_

Секретар ДКК \_\_\_\_\_

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Відділення комп'ютерних систем Комісія КТ та Ш  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР \_\_\_\_\_  
Беркань І.В.  
« \_\_\_\_\_ » \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ**

**на дипломний проект (роботу)**

Здобувачеві (здобувачці) освіти Синяєву Микиті Володимировичу  
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка алгоритмічного та програмного забезпечення для моделювання та візуалізації динамічної системи

затверджена наказом по коледжу від « \_\_\_\_\_ » \_\_\_\_\_ 2022 р. № \_\_\_\_\_

2. Термін здачі закінченого проекту (роботи) \_\_\_\_\_

3. Вихідні данні до проекту (роботи) \_\_\_\_\_

1. Система рекурентних рівнянь для отримання множини точок динамічної системи

2. Параметри відомих динамічних систем

3. Ознаки класифікації динамічних систем та їх основні характеристики

4. Візуалізувати динамічну систему у 3-вимірному просторі та їх зрізи з 4-вимірного простору

5. Моделювання та візуалізацію виконувати засобами ООП та графічного інтерфейсу OpenGL

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Математична модель динамічної системи; Вибір технологій та інструментів для реалізації моделі; Розробка схеми обчислювального процесу; Розробка структур даних та їх взаємодії у програмі; Розробка блок-схем алгоритмів програми; Розробка інтерфейсу програмного застосунку; Результати моделювання динамічної системи; Економічна частина; Охорона праці

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Послідовність етапів побудови математичної моделі; Система ітераційних рівнянь для опису динамічної системи; Інтерфейс програми для моделювання динамічної системи; Схема обчислювального процесу в програмі; Діаграма взаємодії об'єктів АТД у програмі; БСА функцій у класах; Інтерфейс програми у покроковому режимі і етапи побудови динамічної системи; Інтерфейс програми у процесі формування перетину з 4-вимірної множини; Візуалізація динамічної системи у хаотичному, квазіперіодичному та дивному режимах

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Технологічний	Кривченко Ю.В.		
Економічна частина	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Скорнякова О.В.		

7. Дата видачі завдання \_\_\_\_\_

Керівник Кривченко Ю.В. \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

#### КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Вступ. Постановка мети та задач проектування	5.05.2022	
2	Вивчення системи рекурентних рівнянь	7.05.2022	
3	Аналіз відомих динамічних систем та їх параметрів	9.05.2022	
4	Загальний опис реалізації моделі	11.05.2022	
5	Вибір і опис інструментарію для реалізації моделі	13.05.2022	
6	Розробка інтерфейсу та опис вимог до ПЗ	16.05.2022	
7	Розробка схеми обчислювального процесу	18.05.2022	
8	Розробка алгоритмів візуалізації	20.05.2022	
9	Розробка алгоритмів реалізації перетину Пуанкаре	23.05.2022	
10	Розробка алгоритмів реалізації розрахунків	25.05.2022	
11	Розробка діаграми взаємодії АТД у програмі	27.05.2022	
12	Аналіз результатів, підготовка слайдів презентації	30.05.2022	
13	Економічні розрахунки та питання з охорони праці	3.06.2022	
14	Підготовка графічної частини проекту	6.06.2022	
15	Підготовка проекту до захисту та тестування ПП	8.06.2022	

Дипломник \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)



# ЗМІСТ

Вступ.....	6
1 Аналітична частина .....	7
1.1 Математична модель динамічної системи.....	7
1.2 Вибір технологій та інструментів для реалізації моделі.....	10
1.2.1 Застосований інструментарій фреймворку Qt.....	11
1.2.2 Використання STL та OpenGL.....	13
1.3 Розробка схеми обчислювального процесу.....	14
1.4 Розробка структур даних та їх взаємодії у програмі.....	16
1.5 Розробка блоків схеми алгоритмічної програми.....	24
1.6 Розробка інтерфейсу програмного застосування.....	43
1.7 Результати моделювання динамічної системи.....	50
1.7.1 Хаотичний режим динамічної системи.....	50
1.7.2 Квaziперіодичний режим динамічної системи.....	53
1.7.3 Дієний режим у динамічній системі.....	55
2 Економічна частина.....	57
3 Охорона праці.....	62
Висновки .....	67
Перелік використаних джерел.....	68
ДОДАТОК А. Фрагмент коду функції Dimension класу Fractal.....	69





зв'язів між характеристиками об'єкта;

– аналітичний огляд літературних джерел, аналіз та порівняння побудованих раніше моделей подібних об'єктів;

– аналіз та узагальнення всього накопиченого матеріалу, розроблення загального плану створення математичної моделі.

В деяких випадках дослідження внутрішньої будови та порядку функціонування об'єкта моделювання неможливе або економічно недоцільне. Тому можна розглядати об'єкт як «чорний ящик», стосовно якого нам відомі лише його входи та виходи.



Рисунок 1.1. Послідовність етапів побудови математичної моделі

Зл	Ар	№ доку	Прова	Дата

РП 05.20.003.00 ДП ПЗ

Ар

8



одні вихідний потік з першого рівня. Наявність таких відмінностей в системах призводить до деяко різних властивостей.

Комп'ютерна модель динамічної системи у багатовимірному фазовому просторі дозволить ефективно досліджувати розривок та параметри динамічної системи.

У даній роботі для моделювання динамічної системи у тривимірному просторі необхідно побудувати безліч точок, що генеруються системою рекурентних рівнянь. Необхідно забезпечити візуалізацію безлічі, можливість її обертання навколо осей координат і перетин площинами. У програмній моделі мають розраховуватися фрактальна і інформаційна розмірності. Аналогічну задачу передбачається вирішувати і у чотиривимірному просторі. Особливістю завдання – побудова тривимірного перетину Пуанкаре (прямокутного паралелепіпеда) чотиривимірної безлічі точок, а також візуалізація частини безлічі, що потрапила в перетин. Візуалізація динамічної системи та розрахунок її параметрів дозволить визначати межі існування режимів: точка, граничний цикл, хаотичний режим, періодичний режим.

## 1.2 Вибір технологій та інструментів для реалізації моделі

Програмний додаток для моделювання та візуалізації динамічної системи створено з використанням принципів об'єктно-орієнтованого програмування (ООП). Такій підхід дозволить описати всі структури, реалізовані в моделі, за допомогою класів, процедур і функцій. Програмний продукт реалізований на мові C++ (стандарту C++ 11) за допомогою фреймворку Qt (версія 5.11.2 для x64). У жості компілятора для середовища Qt використано компілятор версії 14.0 Microsoft Visual Studio C++ 2019. Для реалізації послідовних і асоціативних контейнерів, а також алгоритмів роботи з ними і ітераторів застосовується вбудована в C++ бібліотека класів-контейнерів – стандартна бібліотека шаблонів STL. Візуалізація графічних об'єктів буде виконана за допомогою стандартного графічного інтерфейсу OpenGL на основі модуля QtOpenGL, розрахованого на застосування системної бібліотеки OpenGL.

										Апр
										10
Зл	Апр	№ докум	Провак	Дата	РП 05.20.003.00 ДП ПЗ					





QMainWindow або QDialog у якості вікна програми, проте в Qt будь-який віджет може бути вікном.

Для розміщення віджетів у вікні використовуються менеджери компоновки (layout managers). Менеджери компоновки можуть містити як віджети, так і інші менеджери компоновки, зокрема менеджер к горизонтального QHBoxLayout і вертикального QVBoxLayout компоновки. Зовнішній менеджер компоновки (типу QHBoxLayout) є головним. Він встановлюється за допомогою команди setLayout(mainLayout) і відповідальний за всю область, займану діалоговим вікном. Інші три менеджера компоновки (leftLayout, topLeftLayout і rightLayout) є внутрішніми. Особливістю класів менеджерів компоновки є те, що вони не є віджетами. Вони походять від класу QLayout, який в свою чергу походить від класу QObject. При роботі програми менеджери компоновки невидимі. При додаванні внутрішніх менеджерів компоновки до батьківського менеджера компоновки (команди leftLayout -> addLayout(topLeftLayout), mainLayout -> addLayout(leftLayout) і mainLayout -> addLayout(rightLayout)) для них автоматично встановлюється батьківський зв'язок. Далі, коли головний менеджер компоновки встановлюється для діалогу, він стає дочірнім елементом діалогу і всі віджети в менеджерах компоновки стають дочірніми елементами діалогу.

### 1.2.2 Використання STL та OpenGL

У стандарт C++ 11 входить власна збудована бібліотека класів-контейнерів – стандартна бібліотека шаблонів STL (standard template library). З STL пов'язано кілька основних понять: контейнери, алгоритми і ітератори. Контейнер є способом організації зберігання даних. У STL включені найбільш корисні з них. Контейнери STL підключаються до програми за допомогою шаблонних класів, отже, можна легко змінити тип збережених в них даних. Під алгоритмами в STL мають на увазі процедури, застосовувані до контейнерів для обробки їх даних різними способами: алгоритми сортування, копіювання, пошуку і об'єднання. Алгоритми представлені в STL у вигляді шаблонних функцій, однак вони не є методами класів-контейнерів – це абсолютно незалежні функції. Ітератори є

										Апр
										13
Зл	Апр	№ докум	Провак	Дата	РП 05.20.003.00 ДП ПЗ					

узагальненням концепції вказівників. Вони посилаються на елементи контейнера, їх можна інкрементувати, як звичайні вказівники і вони будуть посылатися послідовно на всі елементи контейнера. Ітератори – ключова частина всього STL, оскільки вони зв'язують алгоритми з контейнерами. Контейнери STL поділяються на дві категорії: послідовні і асоціативні. До послідовних відносяться вектори (типу `vector`), списки `list` і черги з двостороннім доступом `deque`. Серед асоціативних виділяють безлічі `set`, мультимножини `multiset`, відображення `map` і мультивідображення `multimap`.

OpenGL є стандартним програмним інтерфейсом, призначеним для відтворення тривимірної графіки. Додатки Qt можуть відображати графіку 3D, використовувачи модуль `QtOpenGL`, який розрахований на застосування системної бібліотеки OpenGL. Цей модуль являє собою клас `QGLWidget`, для якого можна створювати підкласи для розробки власних віджетів, що малюються з використанням команд OpenGL.

### 1.3 Розробка схеми обчислювального процесу

Розглянута вище модель динамічної системи у багатовимірному фазовому просторі описана системою трьох або чотирьох ітераційних рівнянь для 3-вимірного фазового простору (1.4) та для 4-вимірного (1.5).

$$\Phi(x, y, z) = \begin{cases} x_{n+1} = x_n - k_{xy} p x_n^2 + k_{yz} q y_n^2 + x_{in} \\ y_{n+1} = y_n + k_{xy} p x_n^2 - (k_{xy} + k_{yz}) q y_n^2 + k_{yz} r z_n^2 \\ z_{n+1} = z_n + k_{yz} q y_n^2 - (k_{xy} + k_{yz}) r z_n^2 \end{cases} \quad (1.4)$$

$$\Psi(x, y, z, w) = \begin{cases} x_{n+1} = x_n - k_{xy} p x_n^2 + k_{yz} q y_n^2 - x_{in} \\ y_{n+1} = y_n + k_{xy} p x_n^2 - (k_{xy} + k_{yz}) q y_n^2 + k_{yz} r z_n^2 \\ z_{n+1} = z_n + k_{yz} q y_n^2 - (k_{xy} + k_{yz}) r z_n^2 + k_{zw} s w_n^2 \\ w_{n+1} = w_n + k_{zw} s z_n^2 - (k_{xy} + k_{yz}) w_n^2 \end{cases} \quad (1.5)$$

де  $x, y, z, w$  – динамічні змінні, що описують, відповідно, стан кожного з рівнів динамічної системи;  $k_{ij}$  – перехідні коефіцієнти, що характеризують статичну і динамічну взаємодію рівнів відповідно;  $p, q, r, s$  – розподільні



кореляційну, інформаційну розмірності.

У програмному комплексі реалізовано візуалізацію множини точок, що генеруються системою рекурентних рівнянь. У багатовимірному фазовому просторі візуалізація цієї безлічі за допомогою перетину Пуанкаре дозволяє визначити особливості еволюції динамічної системи.

Реалізація ітеративної процедури (1.4, 1.5) передбачає задання початкових значень динамічних змінних  $x, y, z, w$  і їх фіксування після кожної ітерації. Зміна значень цих змінних передбачається лише при отриманні розрахованих на їх основі нових значень. При реалізації у програмі заданих систем рівнянь (1.4, 1.5) використано принципово інший алгоритм розрахунку змінних, який реалізує послідовний облік змін значень вже розрахованих змінних при розрахунку інших змінних на поточній ітерації. Ця якісна зміна алгоритму розрахунку призводить, відповідно, до зміни шляхів еволюції даної системи. При реалізації моделі передбачена візуалізація траєкторії генезису досліджуваної системи в багатовимірному фазовому просторі станів рівня і можливість оцінки її характеристик. Узагальнена схема обчислювального процесу, реалізованого в програмному комплексі, показана на рис. 1.2.

#### 1.4 Розробка структур даних та їх взаємодії у програмі

Програмний комплекс містить набір класів і інкапсульованих в них функцій для реалізації заданої функціональності. Асоціативні контейнерні класи типу відображення (map) представляють собою динамічні структури даних, в яких кожен елемент має один ключовий об'єкт або поле і один цільовий (асоційований) об'єкт або поле (поле значень або даних).

Ключове поле (ключ) однозначно відповідає полю даних. У контейнерах типу множини і відображення не може бути двох елементів, що володіють одним і тим же ключем. Тип об'єкта або поля ключа може бути будь-яким (int, double, char, string, QString), як і тип об'єкту або поля даних (int, double, char, float і ін.). У програмній реалізації зазначені типи об'єктів представляють собою окремі поля певного типу (в основному, типу string, int і double).

										Апр
										14
Зл	Апр	№ докум	Проект	Дата	РП 05.20.003.00 ДП ПЗ					

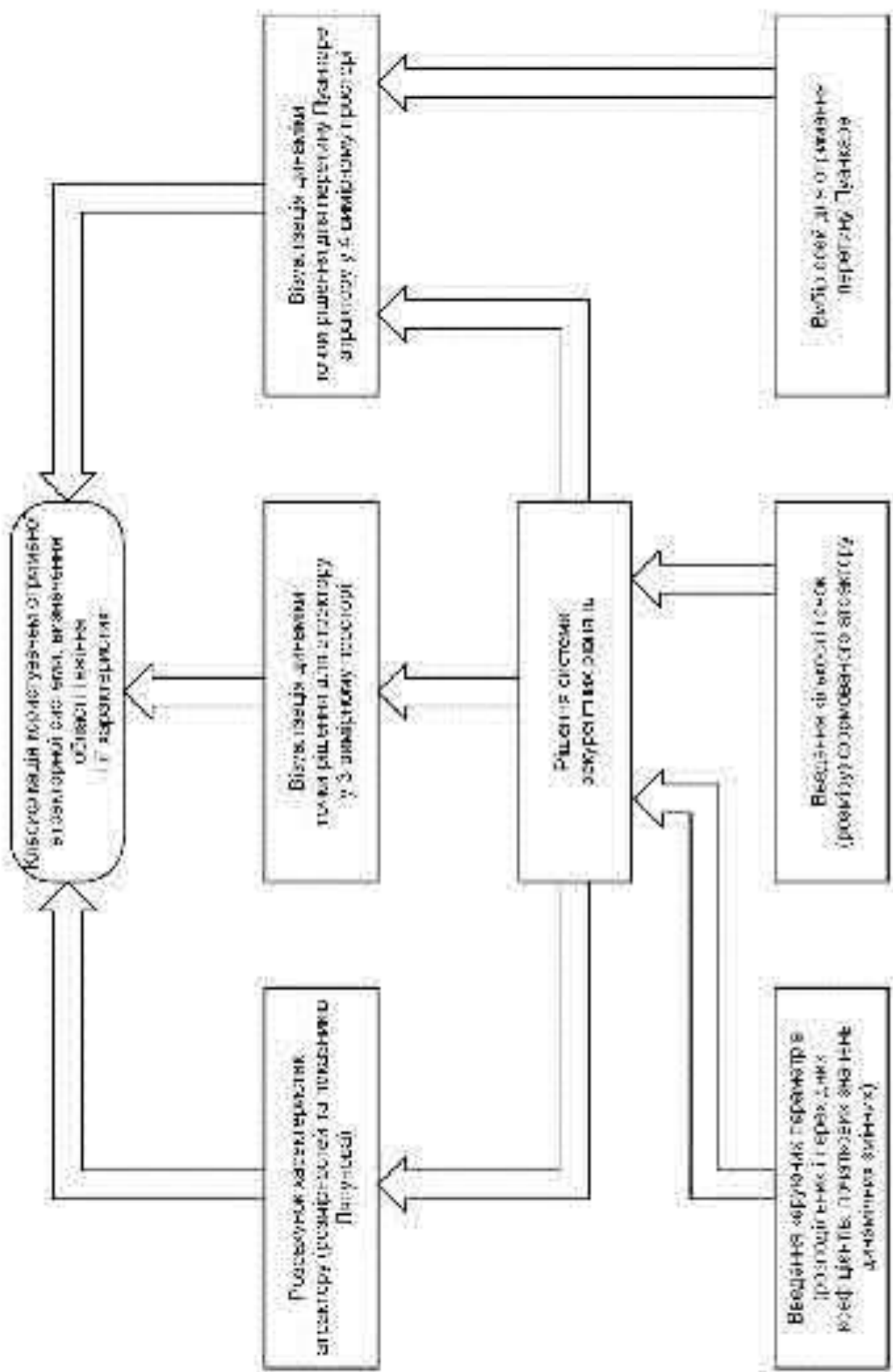


Рисунок 1.2. Схема обчислювального процесу у програмному комплексі

Зм	Арх	№ розд	Примк	Дата

РП 05.20.003.00 ДП ПЗ

Взаємозв'язки і взаємодія класів програмних модулів системи визначені функціональністю програмного комплексу і спирається на загальні принципи об'єктно-орієнтованої організації даних. Породжувані об'єкти АТД (абстрактних типів даних) пов'язані функціонально (рис. 1.3). На схемі показані наступні види зв'язків:

- ← виклик функцій і передача параметрів;
- ◁ генералізація (успадкування);
- ◁ створення екземпляра класу;
- ◇ агрегація.

Екземпляри класу *AtDialog* отримують доступ до об'єктів класів *Fractal*, *Tetrahedron*, *Series*, *Poincare\_s*, *Figure\_s*, *Fourth\_dim*. Посилання на екземпляр класу *Tetrahedron &t* і на екземпляр класу *Fractal &f* передаються першому параметру класу *AtDialog* при створенні останнього. Їх значення (відповідні адреси) присвоюються вказівниками даного екземпляра класу *AtDialog*, що мають тип, відповідно, класу *Tetrahedron* і класу *Fractal*.

У програмному комплексі визначені наступні структури даних:

- *AtDialog* – клас діалогового вікна, успадкований від класу *QDialog*:  
*class AtDialog: public QDialog;*
- *Fractal* – клас, який відповідає за розрахунок точок безлімі, сортування точок і обчислення величин розмірностей;
- *Tetrahedron* – клас, успадкований від класу *QGLWidget*, який відповідає за відображення точок безлімі в основному вікні *OpenGL* і візуальне сортування точок: *class Tetrahedron: public QGLWidget;*
- *Series* – клас, успадкований від класу *QGLWidget*, який відповідає за відображення точок множини в основному вікні *OpenGL* у вигляді декількох серій (фреймів) і візуальне сортування точок;
- *Poincare\_s* – клас, успадкований від класу *QGLWidget*, який відповідає за відображення точок перетину Пуанкаре у вікні *Poincare OpenGL*;

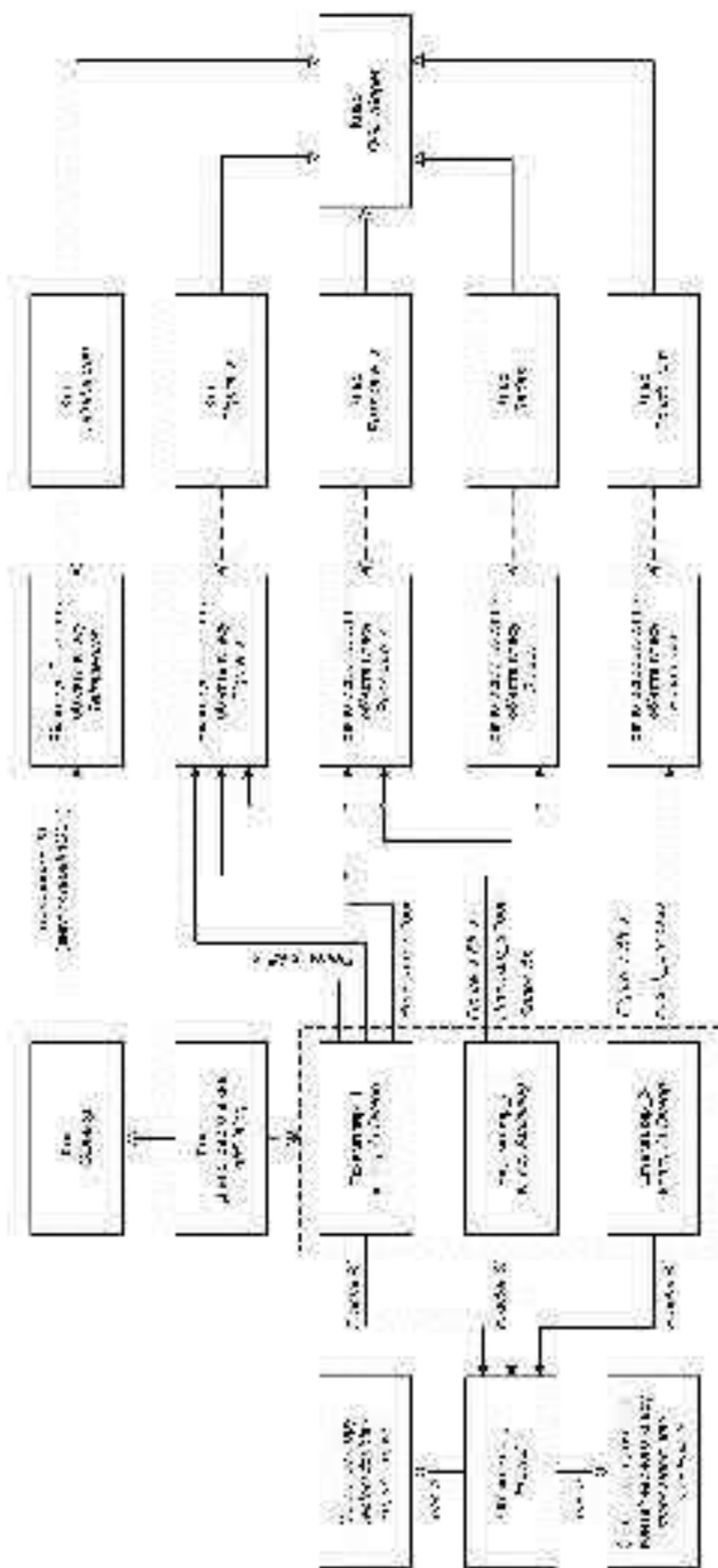


Рисунок 1.3. Диаграмма взаимодействия объектов АТД программного комплексу

Зл	Арх	№ докум	Проект	Дата

РП 05.20.003.00 ДП ПЗ

Арх

19

- *Figure\_3* – клас, успадкований від класу *OpenGLWidget*, який відповідає за відображення точок множини в певному секторі простору, межі якого визначаються в діалоговому вікні;
- *Fourth\_dim* – клас, успадкований від класу *OpenGLWidget*, який відповідає за відображення точок перетину Пуанкаре чотиривимірної моделі динамічної системи в основному вікні *OpenGL*;
- *x1 x2 z1 w1* – масиви координат точок безлічі;
- *x\_rnd y\_rnd z\_rnd w\_rnd* – масиви координат точок перетину Пуанкаре;
- *xp\_n yp\_n zp\_n wp\_n* – масиви координат точок, які є вершинами кубів тривимірної палетки;
- *rx1* – поточне значення позитивних координат *x, y, z* граничних точок сітки палетки;
- *rx2* – поточне значення негативних координат *x, y, z* граничних точок сітки палетки;
- *pf, pf* – вказівники на об'єкти класу *Fractal*;
- *N\_fr* – поточна кількість осередків палетки в процесі обчислення, відповідає одній визначеній точці графіку залежності  $\ln(N)$  від  $\varepsilon$ ;
- *pf > N\_fr* – поточне значення осередків палетки, які належать даній динамічній системі (доступ до цього значення здійснюється через вказівник на екземпляр класу *Fractal, pf*);
- *map <string int >pointsinPallets* – тип об'єкту асоціативного контейнерного класу, що містить в якості ключа інформацію про індекс осередку (перші три цілих числа в разі обчислення розмірностей тривимірного режиму або перші чотири числа в разі обчислення розмірностей чотиривимірного режиму), а також, якщо це необхідно, номер точки (крім першої або єдиної), а в якості значення – індекс точки в початкових масивах (*x1, y1, z1, w1*) множини; останній є однаковим для всіх координат;
- *map <string int >pointsinPallets\_in* – індекс існування точки з даними параметрами, тобто з даними ключем (0 або 1);
- *map <string int >pointsinPallets\_ct* – кількість точок в певній клітинці





вказівника на екземпляр класу `Tetrahedron`. Ця функція в свою чергу викликає функцію `draw()` класу `Tetrahedron`, що має доступ до вказівника на екземпляр класу `Fractal`. Функція `draw()` безпосередньо відповідає за відображення точок безлічі у вікні `OpenGL`. У той же час екземпляр класу `Tetrahedron` створюється при завантаженні робочих об'єктів в момент запуску програми з початковими стандартними коефіцієнтами рівнянь (крива в тривимірному просторі). Останні відрізняються від стандартних коефіцієнтів рівнянь діалогового вікна першого примірника класу `AttDialog`. Крім того, та ж функція `CalcClicked()`, виконавши розрахунок точок перетину Пуанкаре даної динамічної системи, викликає функцію відновлення відображення робочого об'єкту `Puancare repaintGL()` для показника на екземпляр класу `Puancare_s`, який створюється при завантаженні вікна `Puancare` в момент запуску програми. Якщо отримана в результаті розрахунків множина розходиться (з точки зору теорії множин), то видається попередження про те, що величини координат точок перевищили певні порогові значення. У цьому випадку, як правило, величини координат точок дуже швидко досягають наступних порогових значень, при яких подальші обчислення точок безлічі припиняються. При виборі користувачем після запуску додатка режиму чотиривимірної моделі запускається третій екземпляр діалогового вікна `AttDialog` (рис. 1.3). Він має доступ до посилань (& – ідентифікатор посилання) на екземпляри відповідних класів. Після виборі користувачем режиму тривимірної моделі в автоматичному режимі відображення динамічної системи запускається перший екземпляр діалогового вікна `AttDialog`, який також має доступ до посилань на екземпляри відповідних класів. При виборі покрокового режиму відображення динамічної системи запускається другий екземпляр діалогового вікна `AttDialog`.

Класи `Tetrahedron`, `Series`, `Puancare_s`, `Figure_s` і `Fourth_dim` успадковуються від класу `QGLWidget`. Функції класу `QGLWidget` `initializeGL()`, `resizeGL()`, `paintGL()` і `repaintGL()` пере визначаються в разі необхідності. Функції `repaintGL()` пере визначається для поновлення демонстрації зображень (сегмента) вікна `Segment`, який є віджетом класу `Figure_s`.

## 1.5 Розробка блок-схеми алгоритмів програми

Основна частина обчислень у динамічній системі і її розмірностей здійснюється за допомогою функції CalcClicked() (рис. 1.4) класу AtDialog, виклик якої відбувається при натисканні кнопки «Calculate» діалогового вікна програмного комплексу. Ця функція виконує розрахунок точок динамічної системи у тривимірному і чотиривимірному просторі, їх розмірностей і точок перетину Пуанкаре.

Для динамічної системи в тривимірному просторі обчислення точок безлічі відбувається за допомогою функції *AtDialog::var\_f\_N(pf, this)*, де *this* – вказівник на об'єкт діалогового вікна, яке здійснює даний виклик функції *var\_f\_N* (рис. 1.5). Ця функція відповідає за виклик функції *web\_d2* і наступний виклик функції перемальовування зображень в віджеті *Math* і "Step by step math draw": *AtDialog::var\_f\_N(Fractal point pf, AtDialog point\_d)*.

Для динамічної системи у чотиривимірному просторі обчислення точок безлічі відбувається з допомогою функції *AtDialog::var\_f\_N\_w(pf, this)*, де *this* – вказівник на об'єкт діалогового вікна, який здійснює даний виклик функції *var\_f\_N\_w* (рис. 1.6). Ця функція відповідає за виклик функції *web\_d\_w* і наступний виклик функції перемальовування зображень у віджеті "Fourth dimensional math": *AtDialog::var\_f\_N\_w(Fractal point pf, AtDialog point\_d)*.

Функція *web\_d2* (рис. 1.7) виконує формування векторів, що містять координати точок у тривимірному просторі на підставі розрахунків з допомогою функцій *func1*, *func2* і *func3* (обчислює, відповідно, координати точок *x*, *y*, *z* множини по формулі (1.4)):

```
void Fractal::web_d2(QLineEdit *d1, QLineEdit *d2, QLineEdit *d3,  
QLineEdit *d4, QLineEdit *d5, QLineEdit *d6, QLineEdit *d7,  
QLineEdit *d8, QLineEdit *d9, int fr_s)
```

Функція *web\_d\_w* (рис. 1.8) виконує формування векторів, що містять координати точок у чотиривимірному просторі на підставі розрахунків за допомогою функцій *func1*, *func2*, *func3* і *func\_w* (обчислює, відповідно, координати точок *x*, *y*, *z*, *w* множини за формулою (1.5)):

										Апр
										24
Зал	Апр	№ семестру	Прізвище	Дата	РП 05.20.003.00 ДП ПЗ					

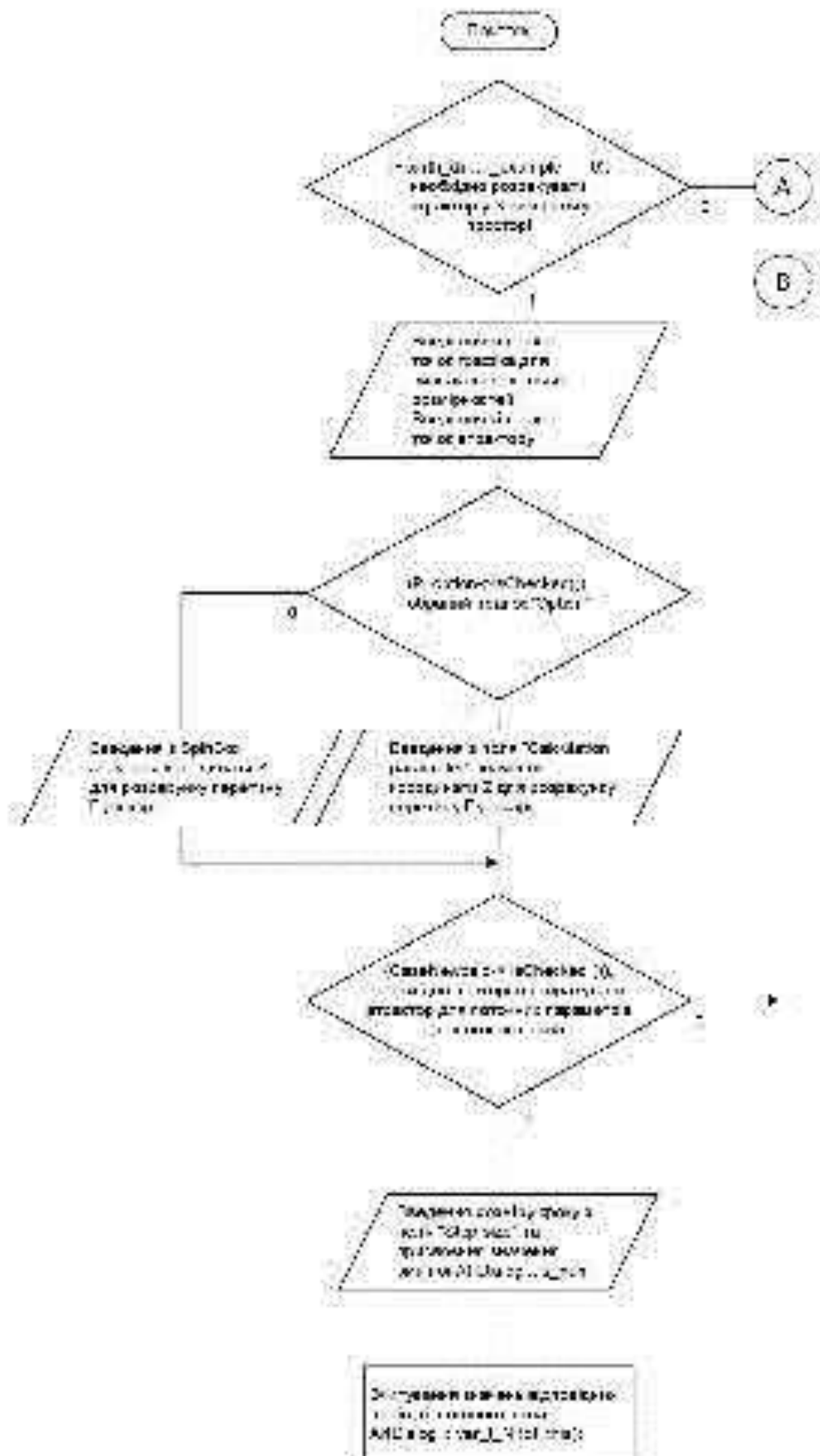


Рисунок 1.4а. БСА функції Calc Clicked() класу AtrDialog

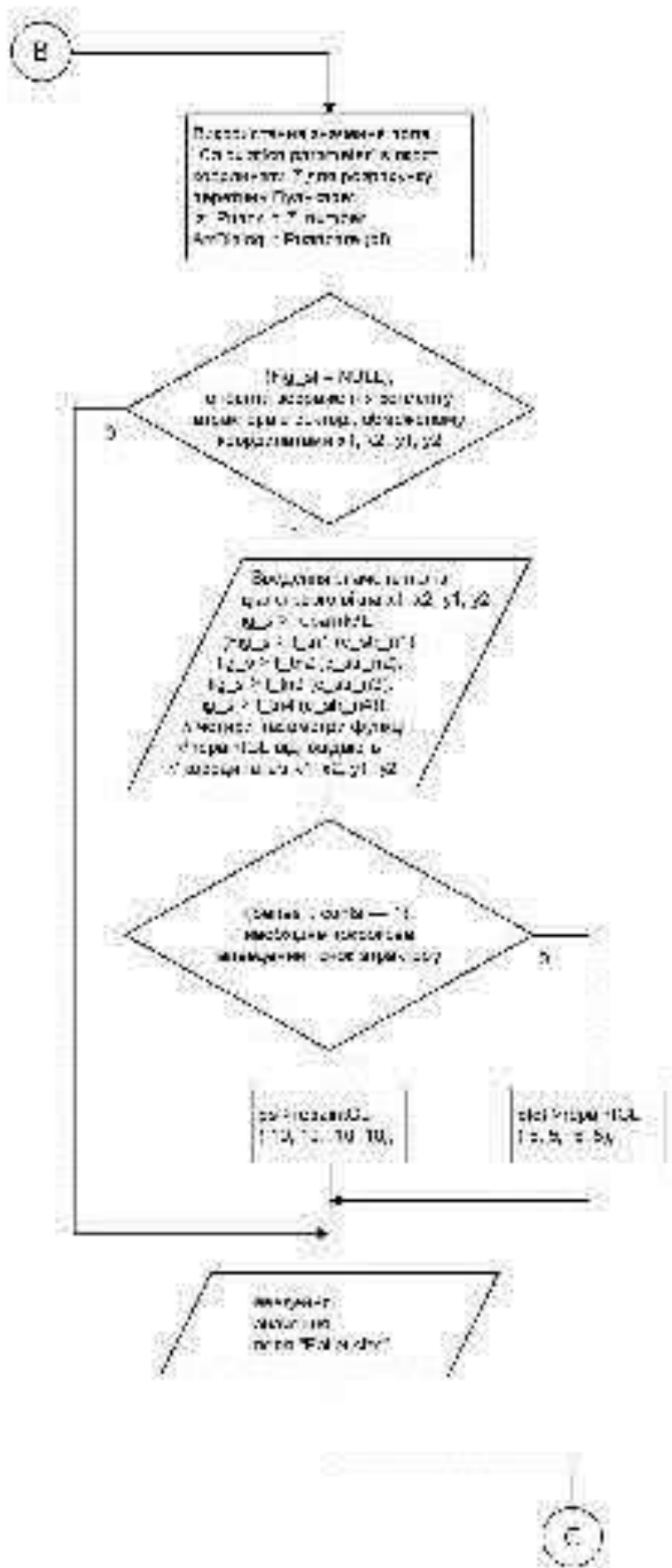


Рисунок 1.46. БСА функции CalcClicked() класса AtrDialog



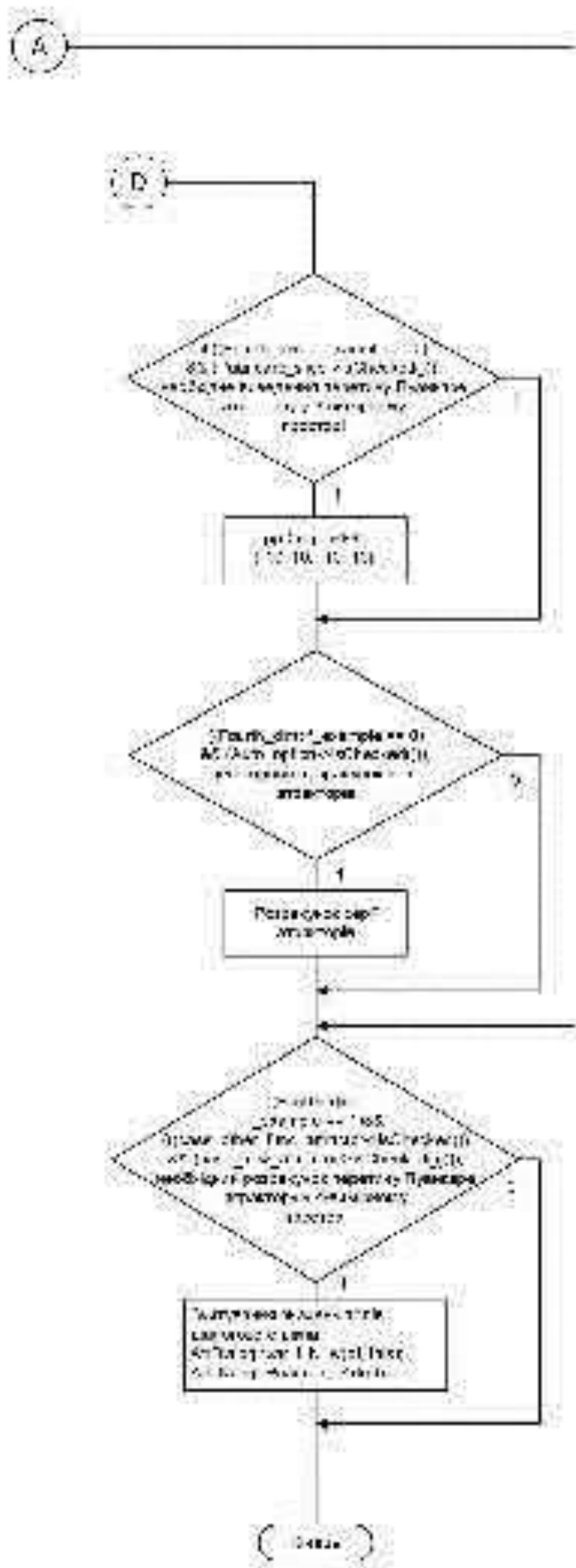


Рисунок 1.4г. БСА функції Calc Clicked() класу AtDiag

З.п.	А.р.	№ докум.	Проект	Дата

РП 05.20.003.00 ДП ПЗ



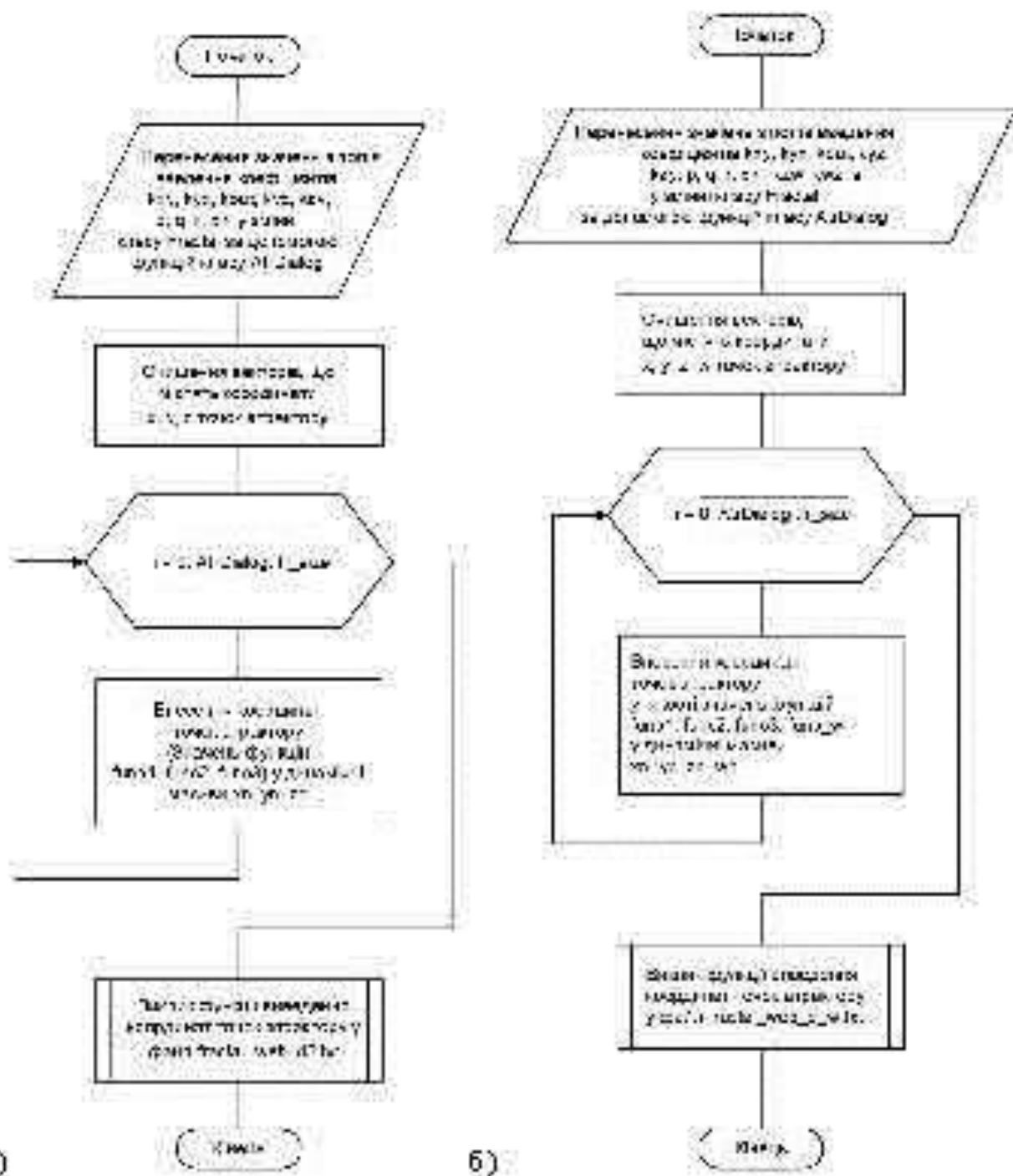


Рисунок 1.7. БСА функцій web\_d2 (а) і web\_d\_w (б) класу Fractal

У разі необхідності розрахунку розмірностей (при виборі відповідного параметру – Several N) функція CalcClicked() викликає виконання функції  $pf \rightarrow sev\_num\_exec(psl, pf, this)$ , яка в свою чергу викликає функцію  $pf \rightarrow sev\_num(psl, pf, pd)$ , де psl – ребро палетки, вказівник на об'єкт діалогового вікна. Функція sev\_num\_exec (рис. 1.8) реалізує очищення динамічних масивів і розрахунок розмірностей методом найменших квадратів:

*Fractal::sev\_num\_exec(double &psl, Fractal \*pf, AtrDialog \*pd).*

Зл	Арх	№ докум	Проект	Дата

РП 05.20.003.00 ДП ПЗ

Арх

20



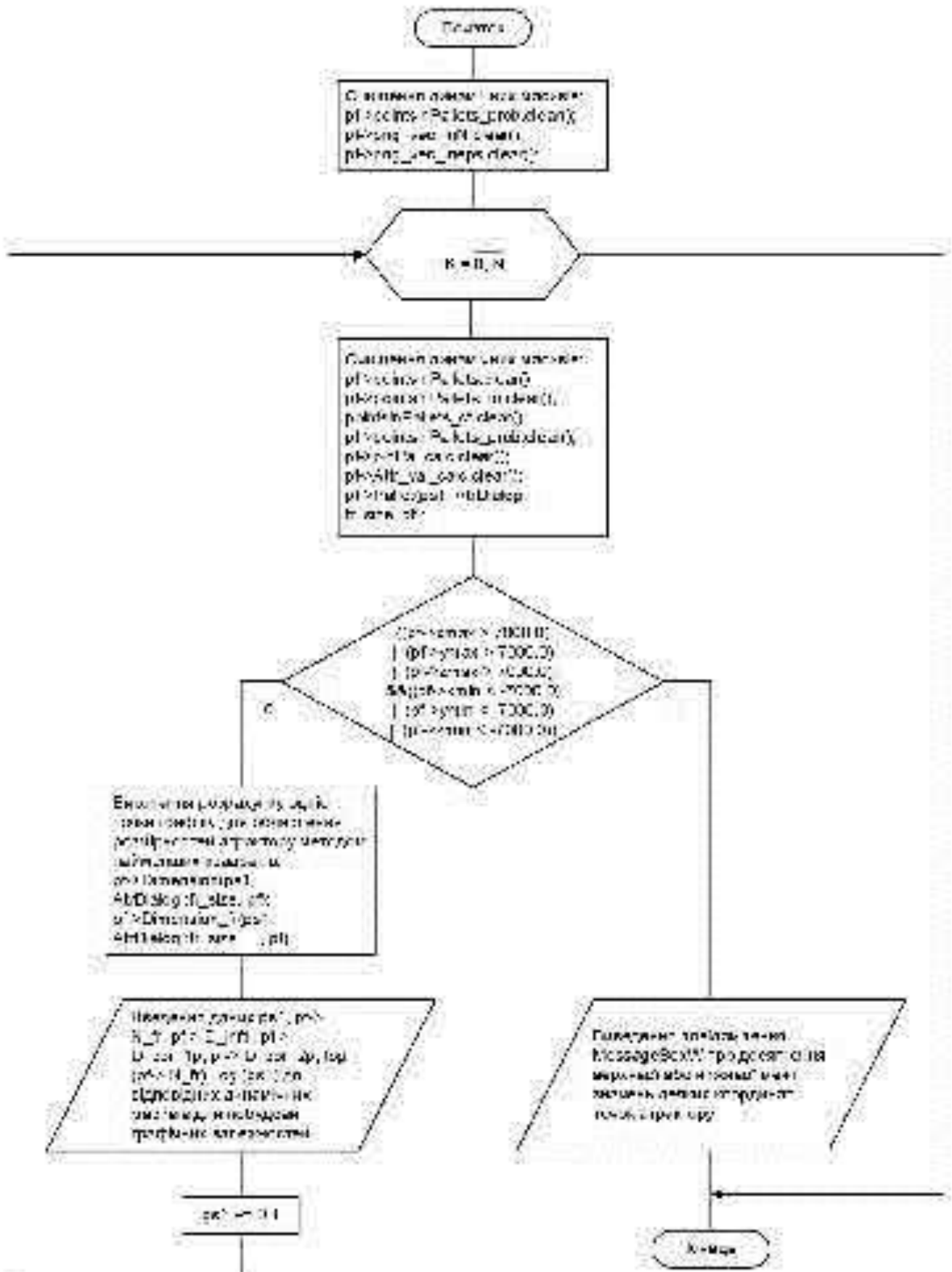


Рисунок 1.9. ВСА функції sev\_rum класу Fractal

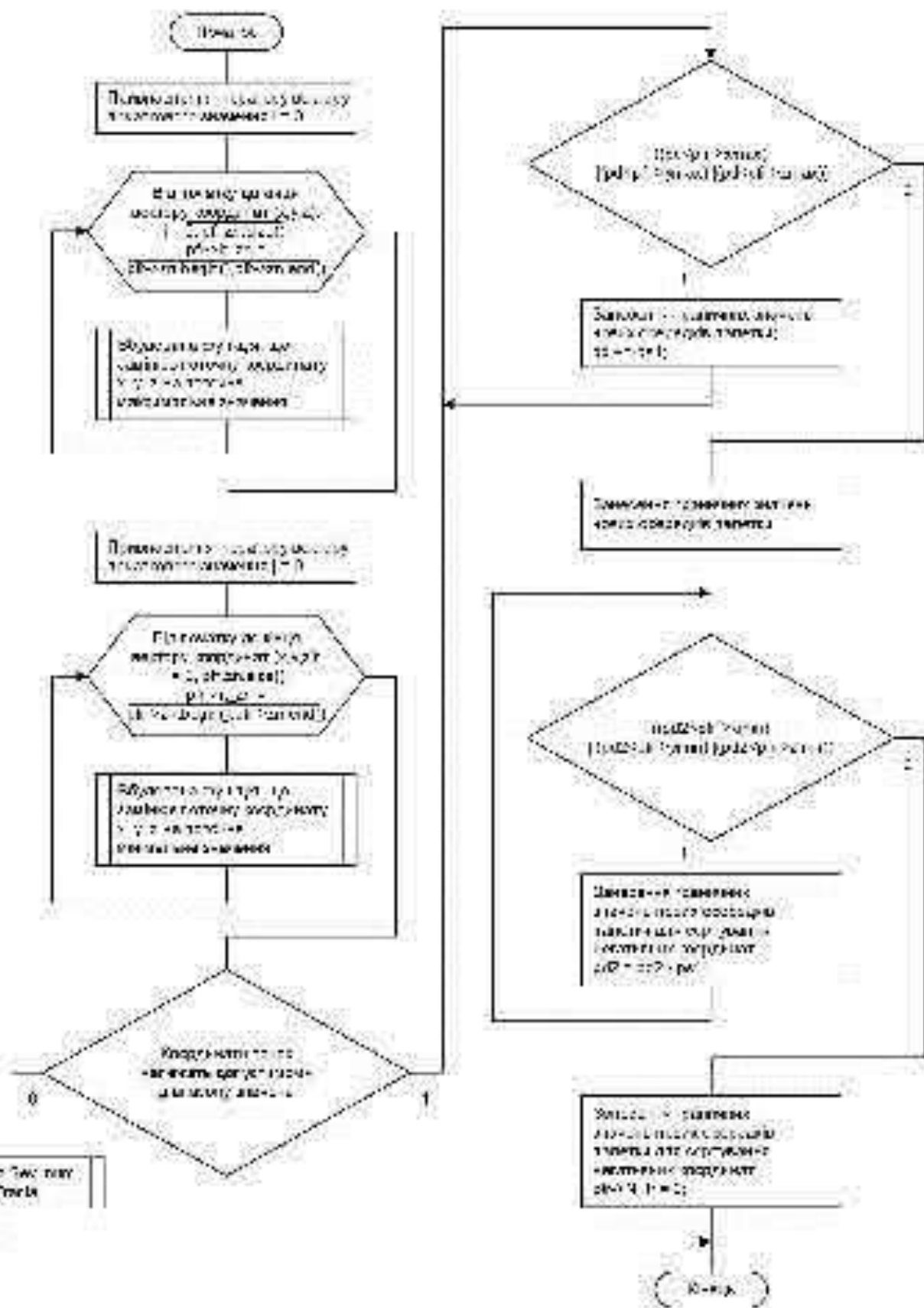


Рисунок 1.10. БСА функції Pallet класу Fractal



номер другої точки або точки далі по порядку, яка потрапляє в дану комірку. У відповідних полях значень об'єкта *prPointsInPallet* вноситься величина індексу *i*-го елемента об'єкта типу вектор *zn* [i], *ym* [i] і *zm* [i], в яких містяться координати *x*, *y*, *z* даної точки. У відповідних полях даних об'єкта *prPointsInPallets\_in* заноситься кількість точок, відповідна даному ключу (індексу комірки палетки), що потрапляють в цю комірку, тобто для об'єкта *prPointsInPallets\_in* ключем може бути тільки комбінація індексів *ix* за координатами *x*, *y*, *z* без будь-якого обліку від окремих точок у комірці.

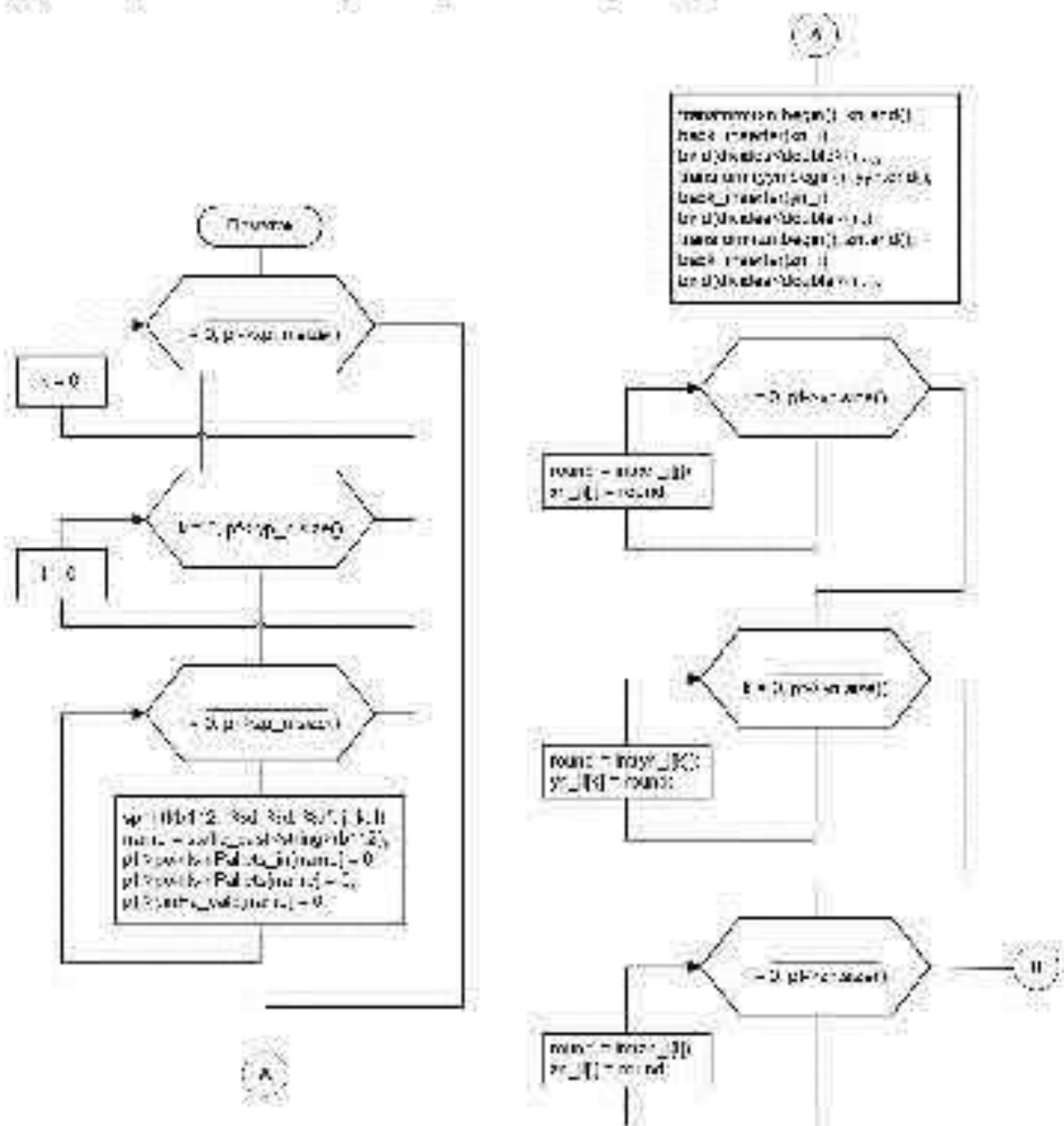


Рисунок 1.11 а. БСА функції Dimension класу Fractal

Зл	Арх	№ докум	Підпис	Дата

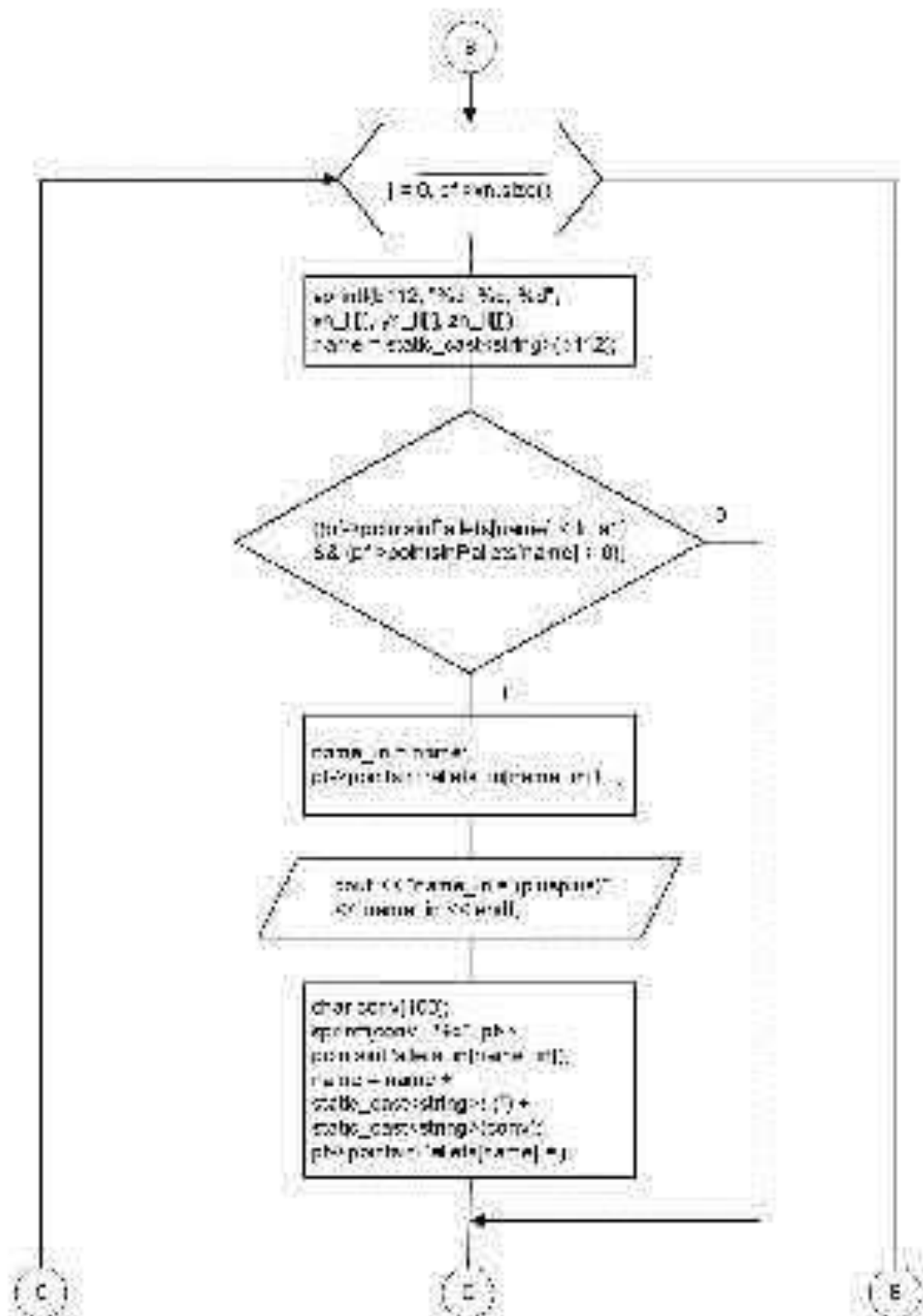


Рисунок 1.116. БСА функції Dimension класу Fractal

Спочатку об'єкти асоціативних контейнерних класів ініціалізуються початковими, нульовими значеннями цілових полів. Потім, шляхом перебору всіх комбінацій індексів комірок палетки, які відповідають певним діапазоном значень координат, з одночасним перебором первинних елементів динамічної системи (координат його точок) і перевірки, чи потрапляє дана точка в дану

Зл	Ар	№ доку	Прова	Дата

РП 05.20.003.00 ДП ПЗ

Ар

Зл

комп'юру, власне і відбувається привласнення значення цільового поля `points` у `palets`, відповідного ключовому полю.

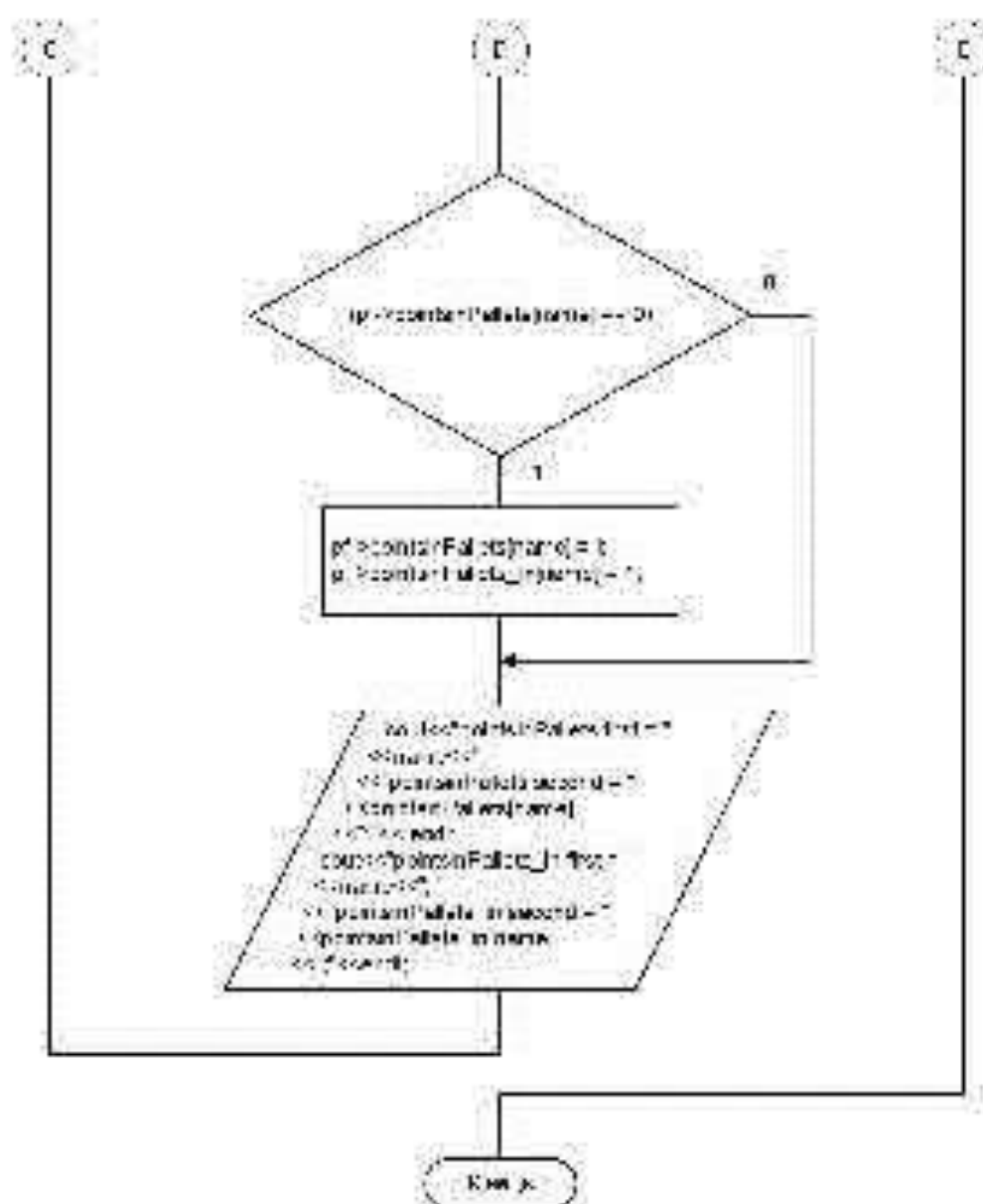


Рисунок 1.11в. БСА функції Dimension класу Fractal

У БСА на рис.1.11 блоки 1,3,5 реалізують привласнення для ініціалізації нульових значень всіх об'єктів, які використовуються для індексації точок і комірок палетки (заповнюють нулями асоціативні контейнери типу map бібліотеки STL). Блок 6 виконує формування комбінацій індексів по кожній осі від нульового до максимального значення отриманої сітки комірок палетки, де  $j, k, l$  – індекси комірок і точок, які знаходяться у відповідних комірках. У блоці 7 реалізується виклик функції transform, бінк для формування векторів, які зберігають нецілочисельні індекси координат відповідних точок ( $x_n, y_n, z_n$  – вектору

Зл	Арх	№ докум	Провак	Дата

РП 05.20.003.00 ДП ПЗ

Арх

37



приналежність точки комірки палетки і кількість таких точок в даній комірці. Одночасно здійснюється контроль над тим, щоб точка не враховувалася двічі. По закінченню роботи функції відбувається розрахунок однієї точки графіка для обчислення надалі першої і другої кореляційних розмірностей.

У програмі умовою перерахунку перетину Пуанкаре і подальшого оновлення робочого об'єкту OpenGL Пуанкаре є зміна ресурсу (поля перебору) діалогового вікна `rq_dsb_ru` типу `QDoubleSpinBox`. Взаємозв'язок функцій `AttrDialog`, що відповідають за перерахунок (перетину Пуанкаре), з даним ресурсом здійснюється за допомогою технології сигналів і слотів Qt 5 за допомогою двох наступних операторів:

```
QObject::connect(rq_dsb_ru, static_cast<void (QDoubleSpinBox::*)>
(double)>(&QDoubleSpinBox::valueChanged), this, &AttrDialog::enter_pdv);
QObject::connect(rq_dsb_ru, static_cast<void (QDoubleSpinBox::*)>
(double)>(&QDoubleSpinBox::valueChanged), this, &AttrDialog::refresh_pua).
```

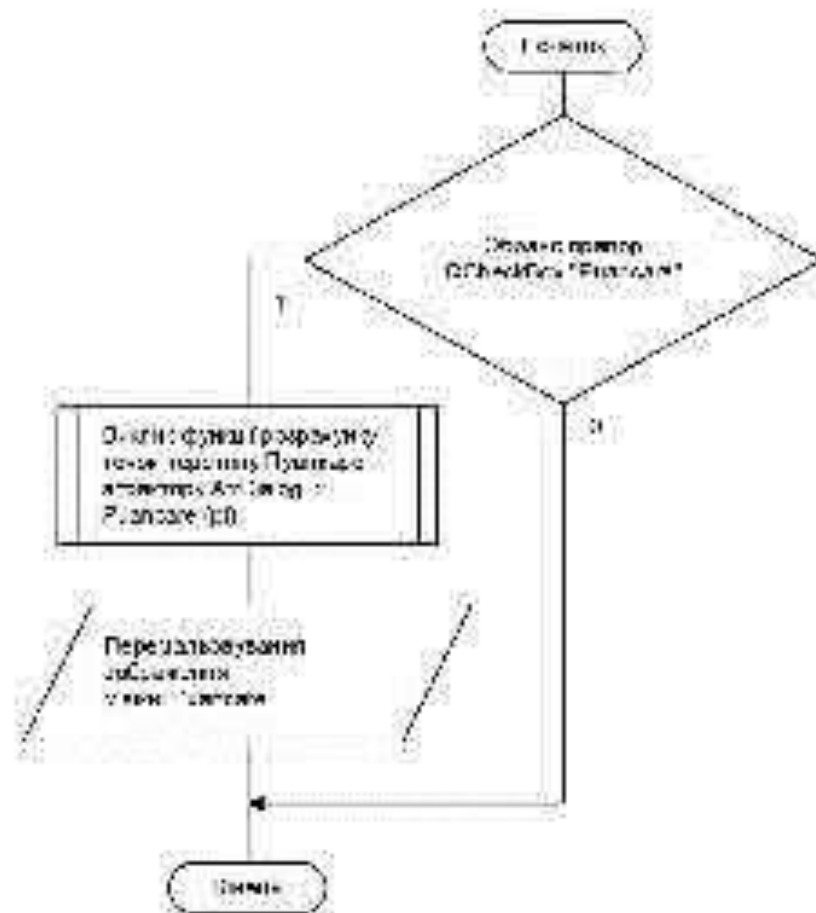


Рисунок 1.12. БСА функції `refresh_pua` класу `AttrDialog`



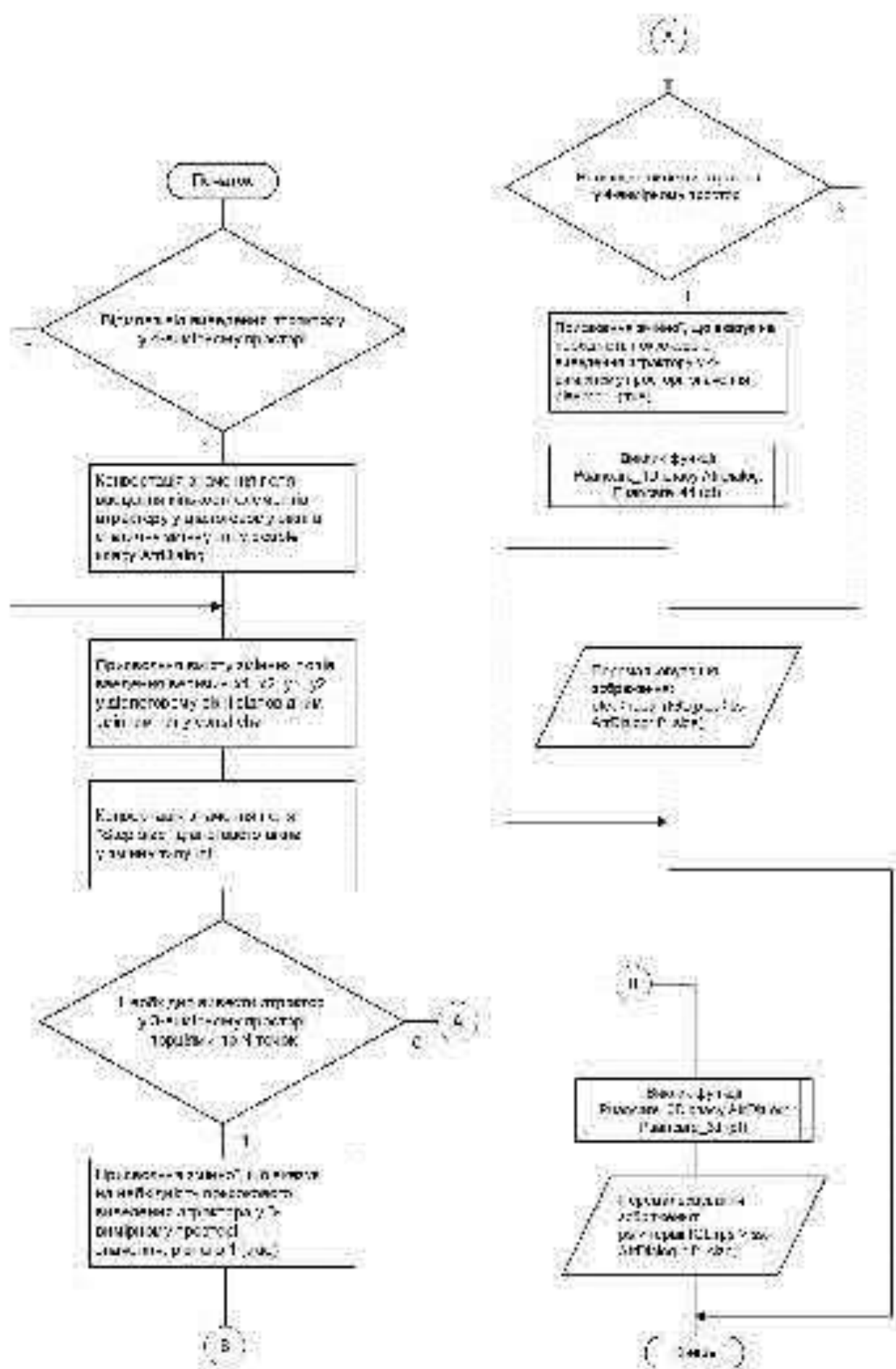


Рисунок 1.13. БСА функції Reproduce класу AtrDialog

Зл	Ар	№ доку	Прова	Дата

РП 05.20.003.00 ДП ПЗ

Ар

41











indices перед натисканням кнопки Calculate реалізується процедура розрахунку параметрів Лапунова з подальшим виведенням проміжних і кінцевих результатів у діалогове вікно текстового виведення.

Кнопки Open webd, Save webd дозволяють відкривати і зберігати сформовані безлічі множини в папки або додатку. Кнопка Compute from file ініціює розрахунок розмірностей із збереженого у файлі цілком. Кнопка Close призначена для виходу з програми.

Інтерфейс додатку при виборі покровкового режиму показаний на рис. 1.16. При цьому відображені 3 з 30 етапів еволюції динамічної системи з параметрами:  $k_{xy} = 0,9$ ;  $k_{yz} = 1,2$ ;  $k_{zx} = 0,95$ ;  $k_{xy} = 0,15$ ;  $k_{xyz} = 0,8$ ;  $p = 0,6$ ;  $q = 0,4$ ;  $r = 0,25$ ;  $x = 0,025$ ;  $y = 0,025$ ;  $z = 0,025$ ;  $x_n = 0,35$ ;  $N_{\text{max}} = 3000$ .

У покровковому режимі в робочий об'єм діалогового вікна Math на екран виводяться порції по  $N$  точок, де  $N$  – задається параметром Step Size у відповідному полі введення вікна Calculate Fractal. При натиснутій кнопці Reproduce series of frames відбувається послідовне виведення порцій точок динамічної системи. До закінчення поточного виведення точок множини вони виділені синім кольором, після закінчення повного виведення всіх точок динамічної системи їх колір стає жовтим.

На рис. 1.17 показаний інтерфейс для чотиризмірної моделі в момент формування перетину Пуанкаре по осі  $W$  для динамічної системи з параметрами:  $k_{xy} = 0,1$ ;  $k_{yz} = 0,1$ ;  $k_{zx} = 0,1$ ;  $k_{xy} = 0,1$ ;  $k_{xyz} = 0,2$ ;  $p = 1$ ;  $q = 1$ ;  $r = 1$ ;  $x = 346$ ;  $y = 6,99$ ;  $z = 0,76$ ;  $x_n = 1,556$ ;  $N_{\text{max}} = 10000$ . При цьому розраховані характеристики динамічної системи в чотиризмірному просторі склали:  $D_f = 3,81$ ;  $D_{\text{cut}} = 3,69$ . Розраховані на наступному етапі роботи програми характеристики перетину Пуанкаре по осі  $W$  для даного режиму склали:  $D_f = 2,73$ ;  $D_{\text{inf}} = 2,66$ ;  $D_{\text{cut}} = 2,56$ .

В інтерфейсі для чотиризмірної моделі передбачено діалогове вікно Calculate Fractal введення вихідних даних моделювання, діалогове вікно графічного виведення даних Fourth dimensional math для відображення перетину і діалогове вікно текстового виведення результатів розрахунку характеристик динамічної системи або перетину Пуанкаре.

У діалоговому вікні *Fourth dimensional math* виводиться зображення перетину Пуанкаре динамічної системи, утвореного точками, відстань між якими масштабується за допомогою скролінгу маніпулятора. Розрахунок точок в чотиризмірному просторі проводиться за рівнянням (1.5) з коефіцієнтами і початковими значеннями динамічних змінних, які вводяться у відповідні поля вводу (рис. 1.17). Розрахунок нового режиму динамічної системи і візуалізація перетину Пуанкаре у вікні *Fourth dimensional math* проводиться за допомогою клацання на кнопці *Calculate* в разі, якщо активовані прапор і *New calculation, 4 dim attractor* і *Poincare*.

При активації прапора *Several N* перед натисканням кнопки *Calculate* реалізується процедура розрахунку розмірностей динамічної системи в чотиризмірному просторі з подальшим виведенням проміжних і кінцевих результатів в діалогове вікно текстового введення.

При активації додатково прапора *Dim calc for Poincare slice* реалізується процедура розрахунку розмірностей перетину Пуанкаре в тризмірному просторі за обраною в полі *Axis* віссю (від 1, 2, 3, 4 для осей координат *X, Y, Z, W* відповідно) з подальшим виведенням проміжних і кінцевих результатів в діалогове вікно текстового введення (рис. 1.17).

Сканування всіх перетинів Пуанкаре для даної динамічної системи (з даними коефіцієнтами рівнянь і даною кількістю точок) можливо за допомогою натискання на кнопку *Reproduce series of frames* з включенням прапором *Retrieve 3d attractor* із зазначенням кількості кроків в полі введення *Row number*, кожен з яких, крім першого, виконується як і для зміни параметра Пуанкаре (кількість точок в перетині) на 0,04 для даної осі, починаючи з мінімального значення для даної динамічної системи.

Параметр *Row number* може набувати значень від 50 до 5000. Якщо в поле введення *Row number* введене число, менше 50 або більше 5000, то застосовується стандартне значення 100.

						РП 05.20.003.00 ДП ПЗ	Апр
Зл	Апр	№ докум	Провак	Дата			42

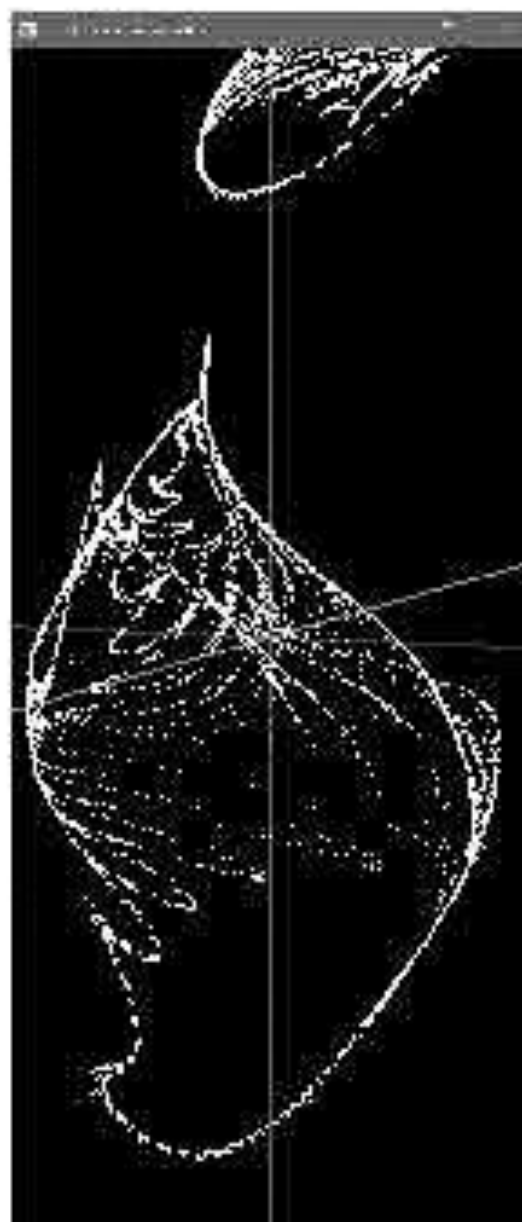
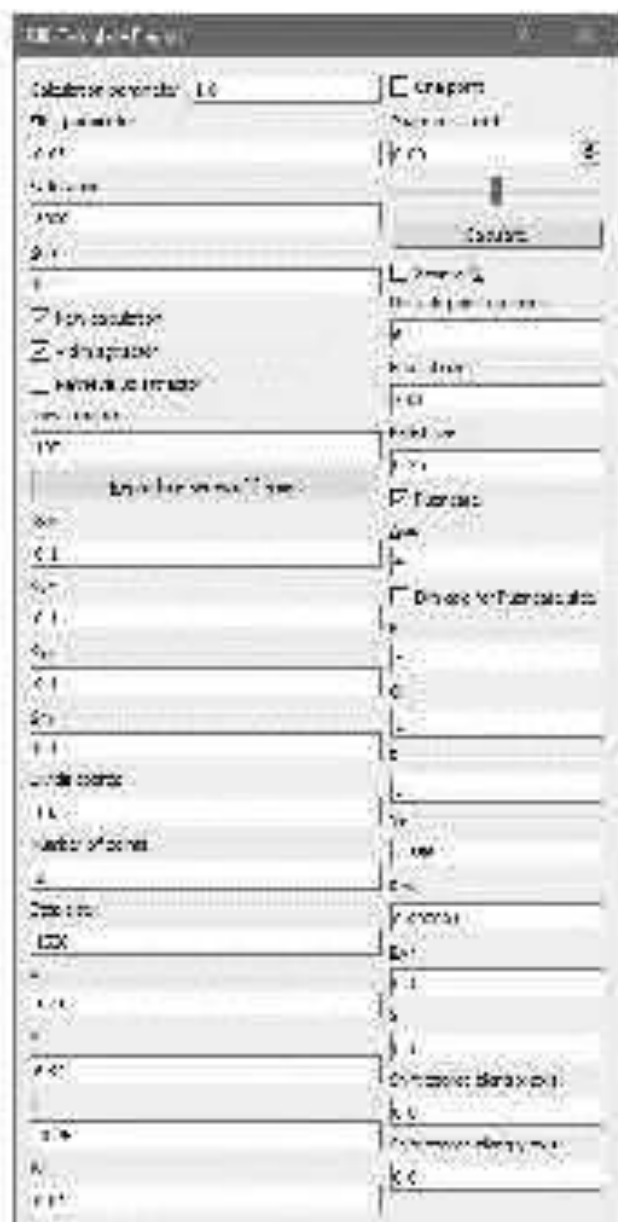


Рисунок 1.17. Інтерфейс для чотиривимірної моделі в процесі формування перетину по осі  $W$ .

Значення параметра Пуанкаре заноситься в файл `ria_gow.txt`, який знаходиться в папці `дво` додатку. Аналізуючи зміст цього файлу за координатою для даної осі і числом точок у відповідному перетині Пуанкаре можна відібрати найбільш характерні і об'ємні зрізи. У файл `ria_1.txt`, який знаходиться в папці `дво` додатку, записується координата точок перетину Пуанкаре кожен раз, коли змінюється значення координати  $Z$  у відповідному полі введення при включеному праторі `Puancare`. В полі введення `Slice parameter` задається число від 0.01 до 0.05, відповідне точності перетину Пуанкаре, тобто товщині зрізу.

Для контролю за процесом моделювання інформація про координати точок

Зал	Апр	№ докум	Проект	Дата

РП 05.20.003.00 ДП ПЗ

Апр

49

динамічної системи, а також проміжні дані розрахунку характеристик динамічної системи (розмірностей) виводяться у файли send.txt, s\_pua.txt, fractal\_web\_d1.txt і fractal\_web\_d2.txt, які знаходяться у папці db о додатку.

## 1.7 Результати моделювання динамічної системи

У даному підрозділі наведені результати моделювання (реконструкції) динамічних систем у чотиривимірному просторі з візуалізацією їх тривимірних перетинів Пуанкаре по осі  $w$ . Для кожного експерименту наведено початкові параметри моделювання – значення динамічних змінних і розподільних коефіцієнтів рекурентного рівняння (1.5) та кількість точок у множині ( $N_{\text{проб}}$ ), а також розраховані та виведені програмним комплексом результати – значення фрактальної розмірності ( $D_f$ ).

### 1.7.1 Хаотичний режим динамічної системи

1. Хаотичний режим, що виникає при параметрах:

$$\begin{array}{lll}
 k_{xy} = 0,5; & p = 0,001; & x = 0; \\
 k_{yx} = 0,4; & q = 0,09; & y = 0; \\
 k_{zx} = 0,3; & r = 0,04; & z = 0; \\
 k_{xz} = 0,3; & s = 0,1; & w = 0; \\
 k_{zw} = 0,000001; & x_0 = 6; & N_{\text{проб}} = 5000; \\
 k_{wz} = 0,1; & k_{wx} = 0,4; & 
 \end{array}$$

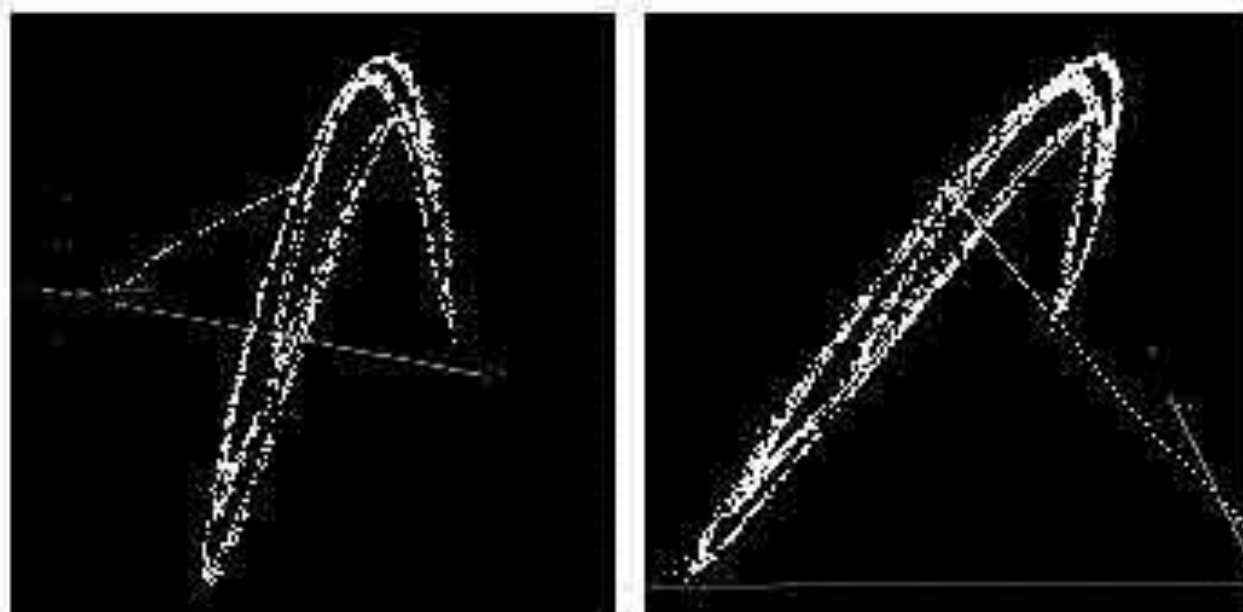


Рисунок 1.18. Динамічна система у хаотичному режимі 1

Зл	Ар	№ докум	Проває	Дата

РП 05.20.003.00 ДП ПЗ

Ар

30

Візуалізація тривимірного перетину Пуанкаре наведена на рис.1.18.  
 Фрактальна розмірність  $D_f = 3.29$ , фрактальна розмірність перетину Пуанкаре по осі  $w$   $D_{f_w} = 2.18$ .

2. Хаотичний режим, що виникає при параметрах:

$k_{xy} = 0,1;$	$p = 1;$	$x = 3;$
$k_{yx} = 0,1;$	$q = 1;$	$y = 7,9;$
$k_{yz} = 0,1;$	$r = 1;$	$z = -2,13;$
$k_{zy} = 0,1;$	$s = 0,1;$	$w = 0,01;$
$k_{xw} = 0,00001;$	$x_0 = 1.536;$	$N_{\text{генер}} = 5000.$
$k_{wx} = 0,1;$	$k_{\text{max}} = 0,2;$	

Візуалізація тривимірного перетину Пуанкаре наведена на рис.1.19.  
 Фрактальна розмірність  $D_f = 3.2$ , фрактальна розмірність перетину Пуанкаре по осі  $w$   $D_{f_w} = 2,08$ .



Рисунок 1.19. Динамічна система у хаотичному режимі 2

3. Хаотичний режим, що виникає при параметрах:

$k_{xy} = 0,65;$	$p = 1;$	$x = 0,927;$
$k_{yx} = 0,25;$	$q = 1;$	$y = 0,49;$
$k_{yz} = 0,65;$	$r = 1;$	$z = -0,13;$
$k_{zy} = 0,25;$	$s = 0,1;$	$w = 0,01;$
$k_{xw} = 0,000001;$	$x_0 = 0,435;$	$N_{\text{генер}} = 5000.$
$k_{wx} = 0,1;$	$k_{\text{max}} = 0,4;$	

Візуалізація тривимірного перетину Пуанкаре наведена на рис.1.20.  
 Фрактальна розмірність  $D_f = 3.58$ , фрактальна розмірність перетину Пуанкаре по осі  $w$   $D_{f_w} = 2,56$ .

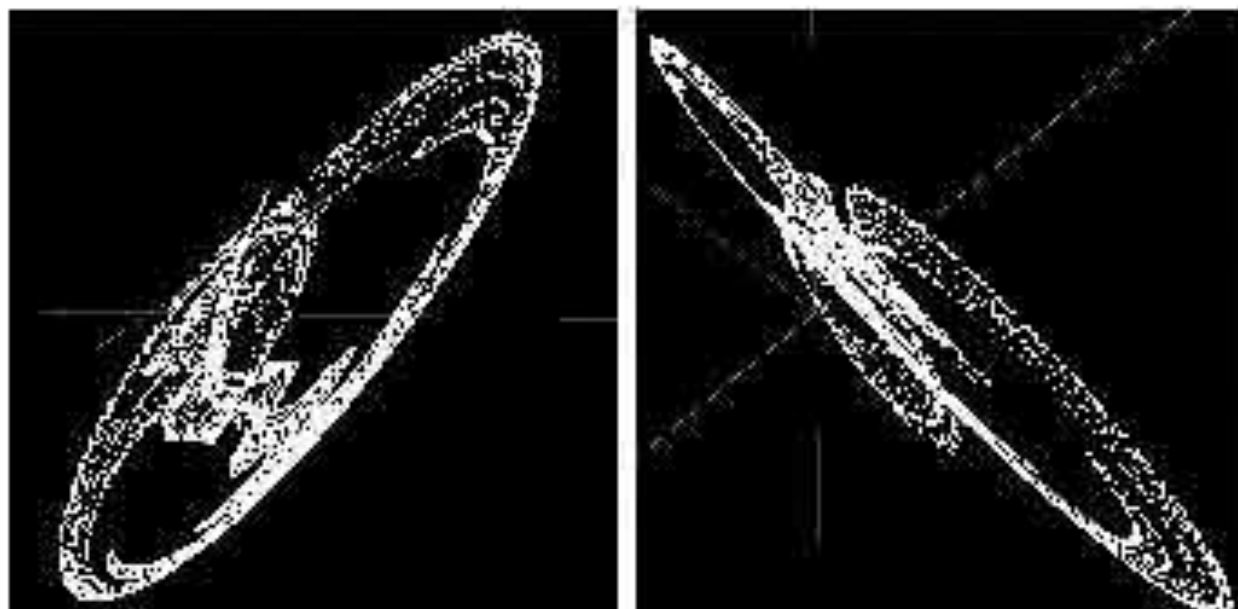


Рисунок 1.20. Динамічна система у хаотичному режимі 3

4. Хаотичний режим, що виникає при параметрах:

$k_{xy} = 0,5;$	$p = 0,01;$	$x = 17,93;$
$k_{yx} = 0,4;$	$q = 1;$	$y = 1,46;$
$k_{yz} = 0,3;$	$r = 1;$	$z = -0,74;$
$k_{zy} = 0,3;$	$s = 1;$	$w = 0,1;$
$k_{zw} = 0,001;$	$x_{max} = 0,475;$	$N_{max} = 5000.$
$k_{wx} = 1;$	$k_{wx} = 0,4;$	

Візуалізація тривимірного перетину Пуанкаре наведена на рис.1.21. Фрактальна розмірність  $D_f = 3,66$ , фрактальна розмірність перетину Пуанкаре по осі  $w$   $D_f = 2,64$ .

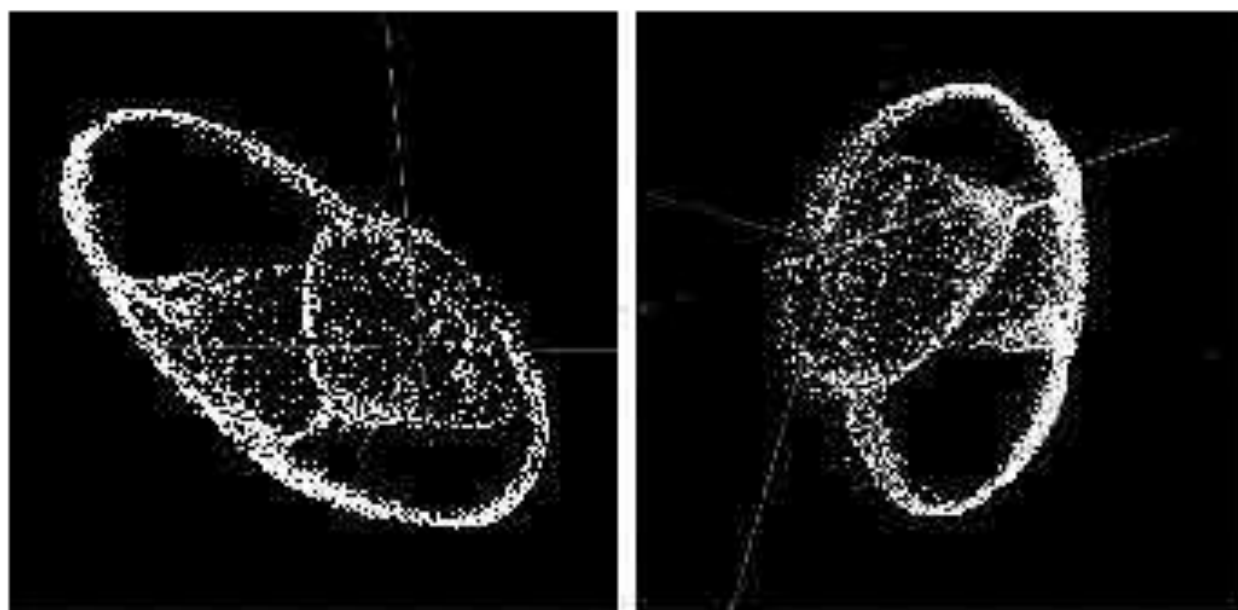


Рисунок 1.21. Динамічна система у хаотичному режимі 4

Зл	Ар	№ доку	Прова	Дата

РП 05.20.003.00 ДП ПЗ

Ар

32

## 1.7.2 Квазіперіодичний режим динамічної системи

1. Квазіперіодичний режим, що виникає при параметрах:

$$\begin{array}{lll} k_{xy} = 0,5; & p = 0,065; & x = 21,67; \\ k_{yx} = 0,4; & q = 0,03; & y = 15,53; \\ k_{yz} = 0,3; & r = 0,03; & z = 15,64; \\ k_{zy} = 0,3; & s = 0,1; & w = 0,01; \\ k_{xw} = 0,00001; & x_0 = 6,05; & N_{\text{генер}} = 5000; \\ k_{wx} = 0,1; & k_{ww} = 0,4; & \end{array}$$

Візуалізація тривимірного перетину Пуанкаре наведена на рис.1.22.

Фрактальна розмірність  $D_f = 3,14$ , фрактальна розмірність перетину Пуанкаре по осі  $w$   $D_f = 2,08$ .

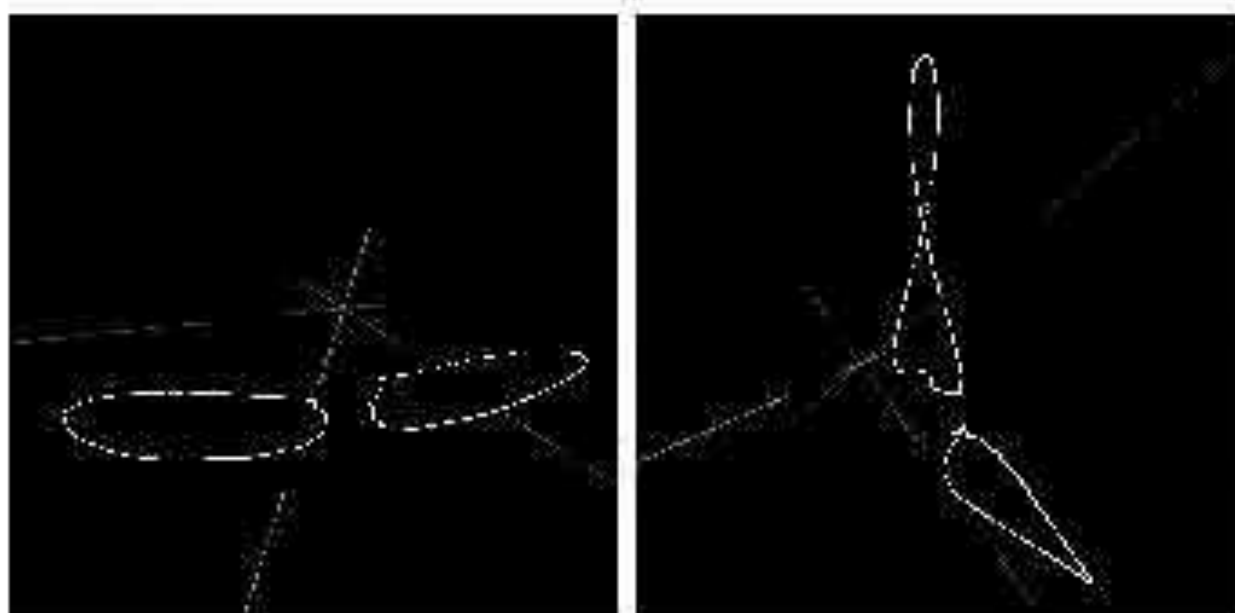


Рисунок 1.22. Динамічна система у квазіперіодичному режимі 1

2. Квазіперіодичний режим, що виникає при параметрах:

$$\begin{array}{lll} k_{xy} = 0,5; & p = 0,1; & x = 22; \\ k_{yx} = 0,4; & q = 0,1; & y = 10; \\ k_{yz} = 0,3; & r = 0,001; & z = -91; \\ k_{zy} = 0,3; & s = 0,1; & w = 0,01; \\ k_{xw} = 0,00001; & x_0 = 2,6; & N_{\text{генер}} = 5000; \\ k_{wx} = 0,1; & k_{ww} = 0,4; & \end{array}$$

Візуалізація тривимірного перетину Пуанкаре наведена на рис.1.23.

Фрактальна розмірність  $D_f = 3,21$ , фрактальна розмірність перетину Пуанкаре по осі  $w$   $D_f = 2,07$ .

Зл	Апр	№ семіна	Проває	Дата

РП 05.20.003.00 ДП ПЗ

Апр

53

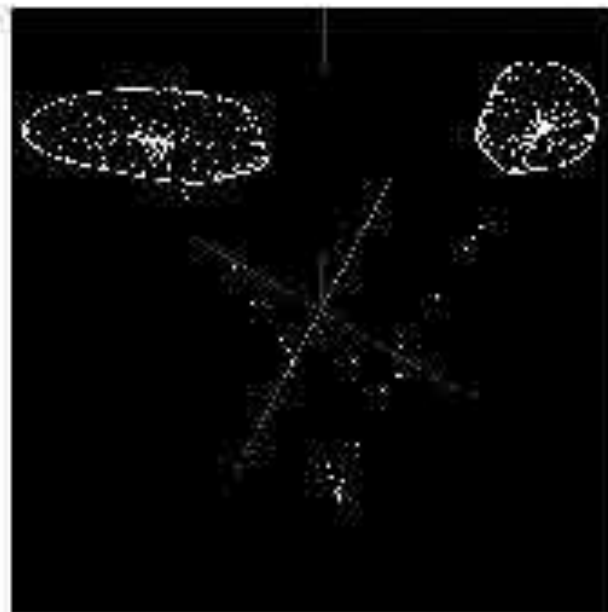


Рисунок 1.23. Динамічна система у квазіперіодичному режимі 2

3. Квазіперіодичний режим, що виникає при параметрах:

$k_{x\gamma} = 0,5;$	$p = 0,05;$	$x = 30;$
$k_{\gamma x} = 0,4;$	$q = 0,065;$	$y = 10;$
$k_{\gamma z} = 0,3;$	$r = 0,0075;$	$z = 10;$
$k_{xz} = 0,3;$	$\epsilon = 0,1;$	$w = 0,1;$
$k_{zw} = 0,000001;$	$x_m = 5,9;$	$N_{\text{генер}} = 5000.$
$k_{wx} = 0,1;$	$k_{zz} = 0,4;$	

Візуалізація тривимірного перетину Пуанкаре наведена на рис.1.24. Фрактальна розмірність  $D_f = 3,18$ , фрактальна розмірність перетину Пуанкаре по осі w  $D_f = 2,16$ .

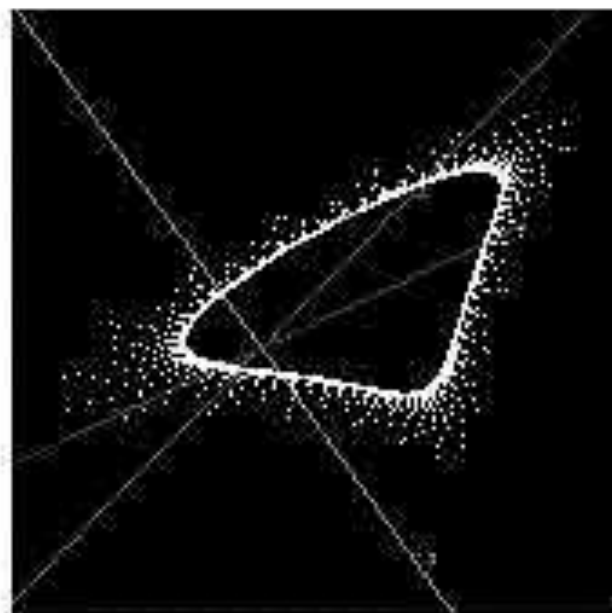


Рисунок 1.24. Динамічна система у квазіперіодичному режимі 3

### 1.7.3 Дивний режим у динамічній системі

1. Дивний режим, що виникає при параметрах:

$$\begin{array}{lll} k_{xy} = 0,8; & p = 0,95; & x = 0,025; \\ k_{yx} = 1,1; & q = 0,25; & y = 0,025; \\ k_{xz} = 1,0; & r = 0,2; & z = 0,025; \\ k_{xy} = 0,2; & s = 0,1; & w = 0,025; \\ k_{zw} = 0,0001; & x_0 = 0,33; & N_{\text{генер}} = 5000; \\ k_{wx} = 0,1; & k_{zw} = 0,48; & \end{array}$$

Візуалізація тривимірного перетину Пуанкаре наведена на рис.1.25.

Фрактальна розмірність  $D_f = 3.24$ , фрактальна розмірність перетину Пуанкаре по осі  $w$   $D_f = 2.13$ .

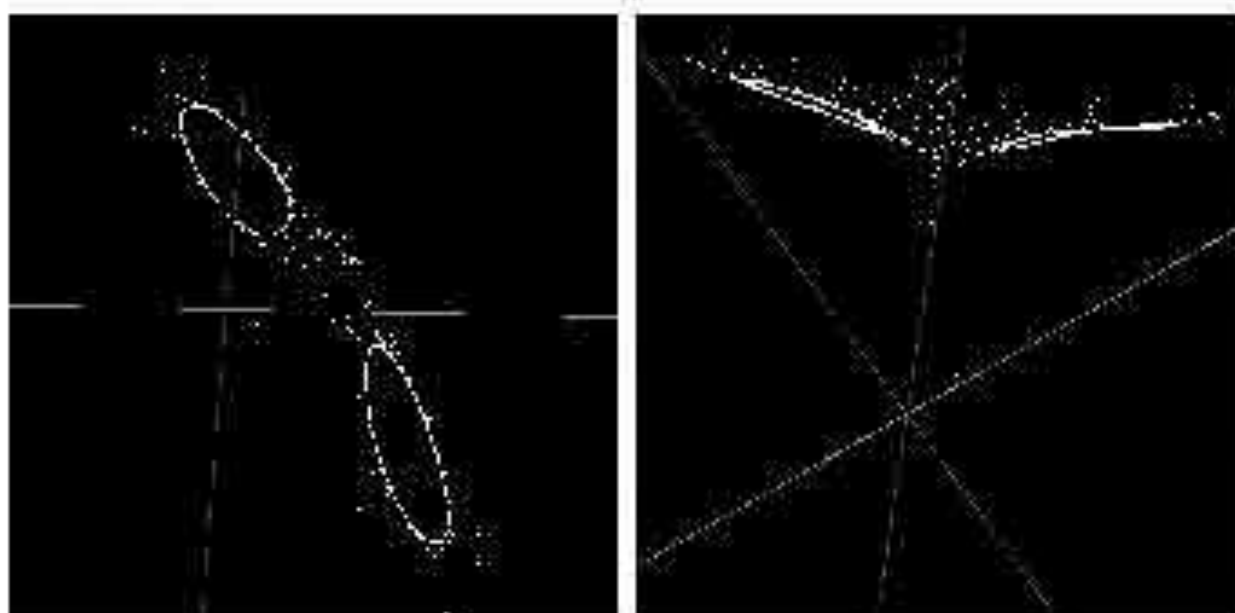


Рисунок 1.25. Дивний режим 1 динамічної системи

2. Дивний режим, що виникає при параметрах:

$$\begin{array}{lll} k_{xy} = 0,45; & p = 0,4; & x = 0,01; \\ k_{yx} = 0,5; & q = 0,2; & y = 0,01; \\ k_{xz} = 0,3; & r = 0,55; & z = 0,01; \\ k_{xy} = 0,4; & s = 0,1; & w = 0,01; \\ k_{zw} = 0,0001; & x_0 = 0,55; & N_{\text{генер}} = 5000; \\ k_{wx} = 0,1; & k_{zw} = 0,45; & \end{array}$$

Візуалізація тривимірного перетину Пуанкаре наведена на рис.1.26.

Фрактальна розмірність  $D_f = 3.51$ , фрактальна розмірність перетину Пуанкаре по осі  $w$   $D_f = 2.41$ .

Зл	Апр	№ сем	Проект	Дата

РП 05.20.003.00 ДП ПЗ

Апр

35

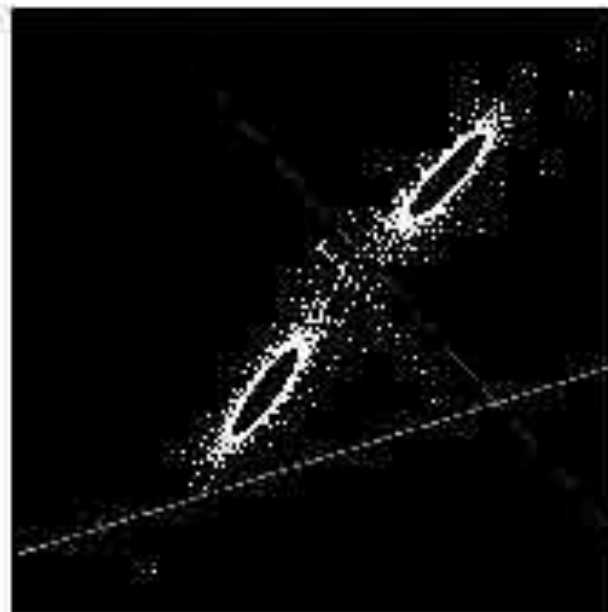


Рисунок 1.26. Дзеркальний режим 2 динамічної системи

Залежно від величин перехідних змінних і розподільних коефіцієнтів змінюється сценарій еволюції динамічної системи, описаної рівняннями (1.5).

Візуалізація відповідного типу динамічної системи свідчить про наявність детермінованого хаосу у динамічній системі (рис. 1.18-1.21). Динамічні системи на рис. 1.22-1.24 дають змогу говорити про відсутність хаосу та про періодичність (квазіперіодичність). Режим динамічної системи, показаний на рис. 1.25-1.26, веде до утворення дзеркального режиму, що представляє собою притягуючу безліч нестійких траєкторій у фазовому просторі – вони описують стохастичні автоколивання, підтримувані в динамічній системі за рахунок зовнішнього джерела. У системах також зустрічаються і інші сценарії розвитку хаосу, наприклад перехід до хаосу через тор, при якому після першої біфуркації з точки народжується двочастотний тор, в загальному випадку з ірраціональним співвідношенням частот. Але при варіюванні керуючого параметра виникають синхронізації частот різного порядку, цьому відповідають динамічні системи у вигляді послідовності декількох точок, кількість яких дорівнює ступеню синхронізації. Зі збільшенням величини вхідного потоку тор руйнується і народжується хаотичний режим динамічної системи.



У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сім кольором.

Вибравши аналог ПП, що містить  $V_0$  в умовних машинних командах, трудомісткості визначити на основі табл. 2.2

Таблиця 2.2

Обсяг ПП, тис. умов машинних команд	Норма часу, люд/год
100	229
200	244
300	262

На підставі отриманого значення, по довіднику, визначається зукрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючи умови розробки ПП, тобто в умовах комп'ютера,  $K_k=0,7-0,8$ ):  $T^* = 244 \times 0,8 = 195,2$  (люд/год).

Трудомісткість програмного продукту визначається як по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{i1} = T^* \times L_i \times K_n \quad (2.1)$$

$$T_{i2} = T^* \times L_i \times K_n \quad (2.2)$$

$$T_{i3} = T^* \times L_i \times K_n \times K_t \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

$L_i$  – питома вага  $i$ -го етапу розробки (див. табл. 2.2.);

$K_n$  – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.3.);

$K_t$  – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.4.)

Таблиця 2.3. Значення гистомікс коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП.

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L <sub>1</sub> )	0,15	0,12	0,12
ТП (L <sub>2</sub> )	0,16	0,15	0,11
РП (L <sub>3</sub> )	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K <sub>0</sub>
А	Принципово нові ПП	1,25 - 1,2
Б	ПП – розробок визначеного параметричного ряду	1,0 - 0,8
В	ПП маючи аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.5. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, %	Значення K <sub>1</sub>
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{\text{тз}} = T^{\text{тз}} * L_1 * K_0 = 195,2 * 0,12 * 0,7 = 16,4 \text{ (люд/година)} \quad (2.1)$$

Трудомісткість розробки технічного проекту

$$T_{\text{тп}} = T^{\text{тп}} * L_2 * K_0 = 195,2 * 0,11 * 0,7 = 15,0 \text{ (люд/година)} \quad (2.2)$$

Трудомісткість розробки робочого проекту

$$T_{рр} = T^* * L_3 * K_n * K_r = 195,2 * 0,61 * 0,7 * 0,7 = 58,3 \text{ (люд.днів)} \quad (2.3)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання  $N_{тз} = 2$  (стр), розробка ТП  $N_{тп} = 20$  (стр), розробка робочого проекту  $N_{рр} = 15$  (стр), повсювальна записка відповідно  $N_{пз} = 23$  (стр). Розрахунок зведений у таблицю 2.6.

Таблиця 2.6 Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, год.ч.		
	2	3	4
1. ТЗ	$T_{рр} = 16,4$	$T_{кк} = 0,7 * N_{тз} = 0,7 * 2 = 1,4$	$T_{кк} = 0,15 * N_{тз} = 0,15 * 2 = 0,3$
2. Розробка ТП	$T_{рр} = 15,0$	$T_{кк} = 0,7 * N_{тп} = 0,7 * 20 = 14$	$T_{кк} = 0,15 * N_{тп} = 0,15 * 20 = 3$
3. Розробка РП	$T_{рр} = 58,3$	$T_{кк} = 0,7 * N_{рр} = 0,7 * 15 = 10,5$	$T_{кк} = 0,15 * N_{рр} = 0,15 * 15 = 2,25$
4. Розробка ПЗ	$T_{пз} = 1,5, N_{пз} = 1,5, 23 = 34,5$	$T_{кк} = 0,7 * N_{пз} = 0,7 * 23 = 16,1$	$T_{кк} = 0,15 * N_{пз} = 0,15 * 23 = 3,45$
Усього, в т.ч.:	$\Sigma T = 175,2$		
- на розробку	$\Sigma T_p = 124,2$		
- контроль керівника		$\Sigma T_{кк} = 42$	
- нормоконтроль			$\Sigma T_{кк} = 9$

### 2.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машинно-години і витрати на розробку ПО. Розрахунок основної заробітної плати виконавців приведений у таблиці 2.7. Відповідно до статті 8 «Закону про Державний бюджет України на 2022»

встановлено мінімальну заробітну плату умисльному розмірі з 1 січня 2022 року - 6500 гривень; мінімальну погодинну тарифну ставку - 39,26 грн.

Таблиця 2.7 Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка грн.	Розрахунок, грн.
1. Розробка ПП	124,2	39,26	4 876,1
2. Контроль керівника	42	50,00	2 100
3. Нормоконтроль	9	50,00	450
Усього	-	-	$\Sigma 30 = 7 426,1$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 2.7

Таблиця 2.7 Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	70	2,0	140,0
				$B_{м1} = 140,0$
Транспортно-заготівельні витрати (10%)				$B_{м2} = 0,1 \times B_{м1} = 0,1 \times 140 = 14,00$
Усього				$B_{м} = B_{м1} + B_{м2} = 154,00$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.8.

						РП 05.20.003.00 ДП ПЗ	А гр.
Зл	А гр.	№ докум.	Провак	Дата			бі

Таблиця 2.8 Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	154,00	$B_m$ (див. табл. 4.7)
2. Основна заробітна плата	7 426,1	$Z_o$ (див. табл. 4.6)
3. Додаткова заробітна плата	1 113,9	$Z_d = 0,15 \times Z_o = 0,15 * 7 426,1$
4. Відрахування до єдиного фонду соціального внеску	1 878,8	$B_{c.c.v.} = 0,22 \times (Z_o + Z_d) = 0,22 * (7 426,1 + 1 113,9)$
5. Накладні витрати	2 227,83	$B_{нак.} = 0,3 \times Z_o = 0,3 * 7 426,1$
6. Повна собівартість	12 800,63	$C_{пов.} = B_m + Z_o + Z_d + B_{c.c.v.} + B_{нак.} = 154,00 + 7 426,1 + 1 113,9 + 1 878,8 + 2 227,83$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$П = (C_{пов.} * P) / 100 = (12 800,63 * 15) / 100 = 1 920,09 \text{ (2.4)}$$

Де  $P$  – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_o = C_{пов.} + П = 12 800,63 + 1 920,09 = 14 720,72 \text{ (2.5)}$$

Податок на додану вартість визначаємо по наступній формулі:

$$ПДВ = 0,2 * Ц_o = 0,2 * 14 720,72 = 2 944,14 \text{ (2.6)}$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_p = Ц_o + ПДВ = 14 720,72 + 2 944,14 = 17 664,86 \text{ (2.7)}$$

Зл	Агр	№ розр	Проект	Дата

РП 05.20.003.00 ДП ПЗ

Агр

02

## 3 ОХОРОНА ПРАЦІ

Сучасного розвитку суспільства має характерну особливість. Це зростання сфер діяльності людини, в яких використовуються інформаційні технології.

Широке розповсюдження отримали персональні комп'ютери. Однак їх використання загострило проблеми збереження власного та суспільного здоров'я, вимагає удосконалення існуючих та розробки нових підходів до організації робочих місць, проведення профілактичних заходів для запобігання розвитку негативних наслідків впливу ПК на здоров'я користувачів.

Забезпечення безпечних і здорових умов праці в значній мірі залежить від правильної оцінки небезпечних та шкідливих виробничих факторів. Однакові по складності зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища, надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин.

В дипломному розділі дипломного проекту розглядається питання охорони праці програміста на стадії вирішення ним питань розробки алгоритмічного та програмного забезпечення для моделювання та візуалізації динамічної системи.

### 3.1 Аналіз небезпечних та шкідливих чинників, що впливають на працівника

Оператори ПК і програмісти знітожуються із впливом таких фізично небезпечних і шкідливих виробничих факторів, як підвищений рівень шуму, підвищена температура зовнішнього середовища, недостатня освітленість робочої зони, електричний струм та інші. Тому на робочому місці програміста повинні бути створені умови для високопродуктивної праці.

До засобів захисту відносять: вентиляцію, штучне освітлення, звукоізоляцію. У системі мір, що забезпечують сприятливі умови праці, велике місце приділяється естетичним факторам: оформлення виробничого інтер'єра, устаткування, застосування функціональної музики та інше, які впливають на організм людини.

										А р.
										62
Зл	А р.	№ доку	Прова	Дата	РП 05.20.003.00 ДП ПЗ					

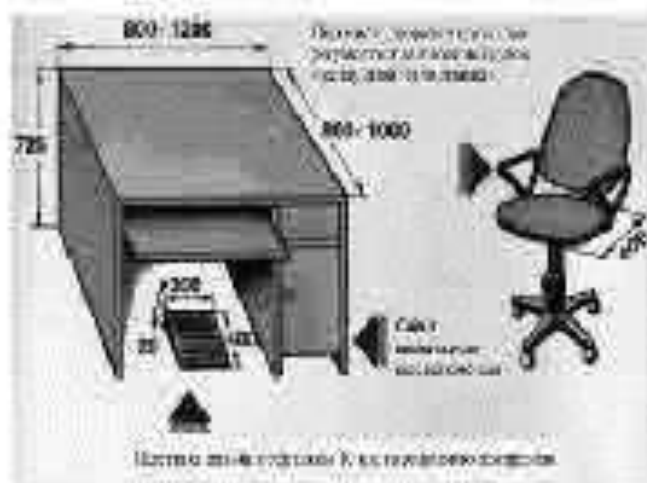




роботи при малій напрузі й не підключенням до захисного заземлення, а також підводити електроживлення до ПЕОМ від розетки за допомогою спеціальної вилки із заземлювчим контактом.

### 3.2.5 Організація робочого місця користувача ПК

Конструкція робочого місця й взаємне розташування всіх його елементів (сидіння, органи керування, засобу відображення інформації) відповідають антропометричним, фізіологічним і психологічним вимогам, а також характеру роботи. Конструкція робочих меблів дає можливість забезпечувати можливість індивідуального регулювання їх відповідно до потреб працівника для підтримки зручної пози. Робочий стіл повинен бути пофарбований матовою фарбою.



розташований так, що його верхній край перебуває на рівні очей, на відстані близько 70 см, що укладається в присутності рамки від 60 до 90 см. Частота мерехтіння екрана дорівнює 100 Гц, що відповідає умові більше 70 Гц.

Обладнання і організація робочого місця з ВДТ мають забезпечувати відповідність конструкцій всіх елементів робочого місця та їх взаємного розташування, ергономічним вимогам, з урахуванням характеру і особливостей трудової діяльності ( ДСанПіН 3.3.2.-007-98).

### 3.3 Пожежна безпека

Під пожежною безпекою розуміють систему державних і суспільних заходів, спрямованих на охорону від вогню людей і матеріальних цінностей.

Заходи щодо пожежної безпеки підрозділяють на дві основні групи: попередження пожеж і ліквідація вже виниклих пожеж. Пожежна профілактика –

					РП 05.20.003.00 ДП ПЗ	А пр.
Зл	А пр.	№ докум.	Проває	Дата		06

це комплекс заходів, спрямованих на попередження пожежі, створення умов, сприяючих швидкій ліквідації пожежі.

Противопожежний захист приміщення забезпечується застосуванням автоматичної установи пожежної сигналізації, наявністю засобів пожежогашіння, застосуванням основних будівельних конструкцій будинку з регламентованими межами вогнестійкості, організацією своєчасної евакуації людей.

Для ліквідації пожежі використовують перенесні засоби пожежогашіння, які призначені для гашіння пожежі початкової стадії їх розвитку. Вони є у всіх виробничих приміщеннях, це жак.

До перенесних засобів пожежогашіння відносяться : вогнегасники, пожежний інвентар (покривала з негорючого теплоізоляційного полотна грубововняної тканини або повсті, шланги з піском, бочки з водою, пожежні відра, совкові лопати) та пожежний інструмент (гаки, лом, совки тощо).

Всі приміщення повинні бути забезпечені перенесними засобами пожежогашіння: пожежним водопостачанням (пожежні врані ПК), пожежні щити з набором пожежного інструменту, вуглекислотними або порошковими вогнегасниками.

У випадку виникнення пожежі необхідно відключити електроживлення, викликати по телефону 101 пожежну команду, евакуювати людей із приміщення відповідно до плану евакуації і приступити до ліквідації пожежі.

					РП 05.20.003.00 ДП ПЗ	Ар.
Зл	Ар.	№ розр.	Провак	Дата		07

## ВИСНОВКИ

Розроблене алгоритмічне та програмне забезпечення для моделювання та візуалізації динамічної системи дозволяє досліджувати еволюцію абстрактних систем, які складаються з компонент, що взаємодіють за довідними законами. За допомогою перетинів з'являється можливість знижувати розмірність і візуалізувати динамічні системи, що спрощує їх класифікацію.

Розроблена для моделювання та візуалізації динамічної системи комп'ютерна модель дозволяє вивчати характер її поведінки, структуру і властивості її відбитків у трьох- і чотирьох-вимірному просторі. Основні параметри, що дозволяють класифікувати динамічні системи по типу – їх розрахована фрактальна розмірність, а також розташування по осях в фазовому просторі, вигляд відбитку перетину і щільність точок в різних областях їх візуалізації. Визначено три типи динамічних систем: хаотична, квазіперіодична і дивна. Залежно від величин перехідних змінних і розподільних коефіцієнтів змінюються сценарії еволюції динамічної системи, описаної рекурентними рівняннями запропонованої моделі. Тип сценарію знаходиться в однозначній відповідності з типом динамічної системи та її перетином.

Виконана візуалізація динамічної системи та аналіз можливих типів її розвитку в запропонованій математичній моделі показали, що є можливість керування сценаріями її розвитку, що може використовуватися в різних областях науки і техніки. Такими динамічними системами можуть бути атмосфера, турбулентні потоки, деякі види ритміч серця, біологічні популяції, суспільство як система комунікацій і його підсистеми: економічні, політичні, психологічні (культурно-історичні та інтер-культуральні) та інші соціальні системи.

Реалізоване програмне забезпечення створене на мові програмування C++, має зручний візуальний інтерфейс та оптимізовану внутрішню архітектуру, що дозволяє використовувати його на будь-якому ПК з системними вимогами, що пред'являються до ОС Windows 10.

										Апр
										02
Зл	Апр	№ докум	Провак	Дата	РП 05.20.003.00 ДП ПЗ					

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Данилов В.Я. Синергетичні методи аналізу: навч. посіб. / В.Я. Данилов, А.Ю. Зінченко. — К.: НТУУ «КПІ» ВП ВПК «Політехніка», 2011. — 340 с.
2. Комп'ютерне моделювання систем та процесів. Методи обчислень: навчальний посібник / Кветний Р. Н., Богач І. В., Бойко О. Р., Софіна О. Ю., Шушуря О. М.; за заг. ред. Р.Н. Кветного. — Вінниця: ВНТУ, 2012. — 193 с.
3. Самарский, А. А. Математическое моделирование. Идеи. Методы. Примеры / А. А. Самарский, А. П. Михайлов. — М., 2001.
4. Диференціальні моделі. Стійкість / А. М. Самойленко, С. Д. Борисенко, Дж. Матарazzo та ін. — К., 2000.
5. Хусаїнов Д.Я., Харченко ІІ., Шатирко А.В. Введення в моделювання динамічних систем: Навч. посібник — КНУ ім. Т. Шевченка, 2010.
6. Шильников Л. П., Шильников А. Л., Тураев Д. В., Чуа Л. Методы качественной теории в нелинейной динамике //Москва-Ижевск: Ин-т компьютерных исследований. — 2009.
7. Спротт Д. К. Элегантный хаос: алгебраически простые хаотические потоки //М. Ижевск: РХД, ИКИ. — 2012.
8. Дж. Гужензаймер, Жюльє П. Нелинейные колебания, динамические системы и бифуркации векторных полей //Москва-Ижевск: Ин-т комп., исп. — 2002.
9. Каток А. Б., Хасселблат Б. Введение в теорию динамических систем. — изд-во МЦНМО, 2005.
10. Кузнецов С.П. Динамический хаос и гиперболические аттракторы: от математики к физике //Москва – Ижевск: ИКИ, 2013.
11. Крутлов В.П., Кузнецов А.С., Кузнецов С.П. Гиперболический хаос в системах с параметрическим возбуждением паттернов стоячих волн //Нелинейная динамика. 2014. — Т. 10. — №3. — С. 265-277.
12. Кузнецов С.П. Динамический хаос. // М.: Физматлит, 2001. — 296 с.

					РП 05.20.003.00 ДП ПЗ	Агр.
Зл	Агр	№ доку	Проває	Дата		69



```

msgBox8.set WindowTitle("pf->xp_nsize()");
msgBox8.setText(0150);
if (fl_print == 1) {
if (msgBox8.exec() == QDialog::Yes) {;}
char d125[150];
for (j = 0; j < pf->xp_nsize(); j++)
for (k = 0; k < pf->yp_nsize(); k++)
for (l = 0; l < pf->zp_nsize(); l++) {
sprintf(d112 "%d %d %d", j, k, l);
name2 = static_cast<string>(d125);
pf->pointsInPallets_in[name] = 0;
pf->pointsInPallets[name] = 0; }
int m;
if ((fr_sl > 190) && (fr_sl <= 1500)) {
if (fopen_s(&file_dim, "c:\\ado\\adm.txt", "r") == NULL) {
else {MessageBox W(NULL, L"The error while attempting to open the file adm.txt",
L"Info", MB_OK);}
if (fl_print == 1) MessageBox W
(NULL, L"End of preliminary calculation of the fractal dimension.", L"Info", MB_OK);
if (file_dim) fclose(file_dim); }
sprintf(d150 "pf->xmax = %d", pf->xmax);
msgBox8.set WindowTitle("pf->xmax");
msgBox8.setText(0150);
sprintf(d150 "pf->xn_size() = %d", pf->xn_size());
msgBox8.set WindowTitle("pf->xn_size()");
msgBox8.setText(0150);
if (fl_print == 1) {
if (msgBox8.exec() == QDialog::Yes) {;}
if (fl_print == 1)
MessageBox W(NULL, L"Dimension! The end of the first phase!", L"Info", MB_OK);
int inf_dim_i;
inf_dim_i = 1;
npxmax = 0;
if (inf_dim_i == 0) {;}
for (j = 0; j < pf->xp_n2_size(); j++)
for (k = 0; k < pf->yp_n2_size(); k++)
for (l = 0; l < pf->zp_n2_size(); l++) {
sprintf(d112 "%d %d %d", j, k, l);
name = static_cast<string>(d112);
pf->pointsInPallets_in[name] = 0;
pf->pointsInPallets[name] = 0;
pf->pinPa_calc[name] = 0;
sprintf(d125 "%d", j);
name2 = static_cast<string>(d125);
pf->Attr_val_calc[name2] = 0;
sprintf(d112 "%d %d %d", j, k, l); // "%d", j, k, l);
name = static_cast<string>(d112);
pf->pointsInPallets_in[name] = 0;
pf->pointsInPallets[name] = 0;
pf->pinPa_calc[name] = 0;

```





```

else continue; }
sprintf(0150 "pf->pointinPallets_in[%s] = %d ", 0112, pf->pointinPallets_in[name_in]);
msgBox8.setWindowTitle("pf->pointinPallets_in[name_in]");
msgBox8.setText(0150);
if (fl_print == 1) {
if (msgBox8.exec() == QMessageBox::Yes) {
cout << "pointinPallets first = " << name << " "
<< "pointinPallets second = "
<< pointinPallets[name] << " " << endl;
cout << "pointinPallets_in first = " << name << " " << "pointinPallets_in second = "
<< pointinPallets_in[name] << " " << endl; } } }
for (i = 0; i < x_n = pf->x_n.begin(); i_y_n = pf->y_n.begin(); i_z_n = pf->z_n.begin();
i < pf->x_n.size(); i_x_n != pf->x_n.end(); i_y_n != y_n.end(); i_z_n != z_n.end(); i++, i_x_n++,
i_y_n++, i_z_n++)
for (j = 0; i_xp_n = pf->xp_n.begin(); j < pf->xp_n.size(); i_xp_n != pf->xp_n.end(); j++,
i_xp_n++)
for (k = 0; i_yp_n = yp_n.begin(); k < pf->yp_n.size(); i_yp_n != pf->yp_n.end(); k++,
i_yp_n++)
for (l = 0; i_zp_n = zp_n.begin(); l < pf->zp_n.size(); i_zp_n != pf->zp_n.end(); l++,
i_zp_n++) {
sprintf(0125 "%d", i);
name2 = static_cast<string>(0125);
if (pf->Attr_val_calc[name2] == 1) continue;
if (((i == 0) &&& ((i_x_n > 0) &&& (i_x_n < i_xp_n)))) {
sprintf(0112 "%d %d %d", j, k, l); // "%d", j, k, l);
name = static_cast<string>(0112);
sprintf(0150 "pf->pointinPallets_in[%s] = %d ", 0112, pf->pointinPallets_in[name_in]);
msgBox8.setWindowTitle("pf->pointinPallets_in[name_in]");
msgBox8.setText(0150);
if ((pf->pointinPallets_in[name] < fr_sl) &&& (pf->pointinPallets_in[name] > 0)) {
name_in = name;
pf->pointinPallets_in[name_in]++;
if ((i_x_n == 1) &&& (pf->pointinPallets_in[name_in] > Npmax)) {
Npmax = pf->pointinPallets_in[name_in];
pf->Attr_val_calc[name2]++;
sprintf(0150 "pf->pointinPallets_in[%s] = %d ", 0112, pf->pointinPallets_in[name_in]);
msgBox8.setWindowTitle("pf->pointinPallets_in[name_in]");
msgBox8.setText(0150);
cout << "name_in = plus plus" << name_in << endl;
char conv[100]; // string conv;
sprintf(conv, "%d", pf->pointinPallets_in[name_in]);
name = name + static_cast<string>(",") + static_cast<string>(conv);
pf->pointinPallets_in[name] = i;
pf->pointinPallets_in[name] = 1;
if (pf->pointinPallets_in[name] == 0) {
name_in = name;
pf->pointinPallets_in[name_in] = 1;
pf->Attr_val_calc[name2]++;
sprintf(0150, "pf->pointinPallets_in[%s] = %d ", 0112, pf->pointinPallets_in[name_in]);
msgBox8.setWindowTitle("pf->pointinPallets_in[name_in]"); // ...

```