Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерный наук и кибербезопасности
Высшая школа программной инженерии

ОТЧЁТ

«Объектно-Ориентированное Моделирование в AnyDynamics»

по дисциплине «Основы объектно-ориентированного моделирования»

Выполнила
студентка гр. 5140904/30202 Ли Ицзя

Руководитель Сениченков Ю. Б.

Санкт-Петербург
2024

1 Описание решаемой задачи

Задание 4: Построить библиотеку классов, с помощью которой можно одновременно рисовать фазовые портреты систем и этих же систем, записанных в векторно-матричной форме. Абстрактный класс должен содержать все нужные параметры. Родительский класс должен работать с уравнениями в скалярной форме. Класс-наследник – с уравнениями в матричной форме. Отдельный класс должен искать неподвижные точки.

$$
\begin{gathered}
x_{n+1} = \frac{x_n\left(a - x_n - y_n\right)}{3} \\
y_{n+1} = \frac{y_n\left(b y_n - x_n\right)}{3} \\
\mathrm{a} = \{5.0, 5.1\}, \\
\mathrm{b} = \{3.0, 3.1\};
\end{gathered}
$$

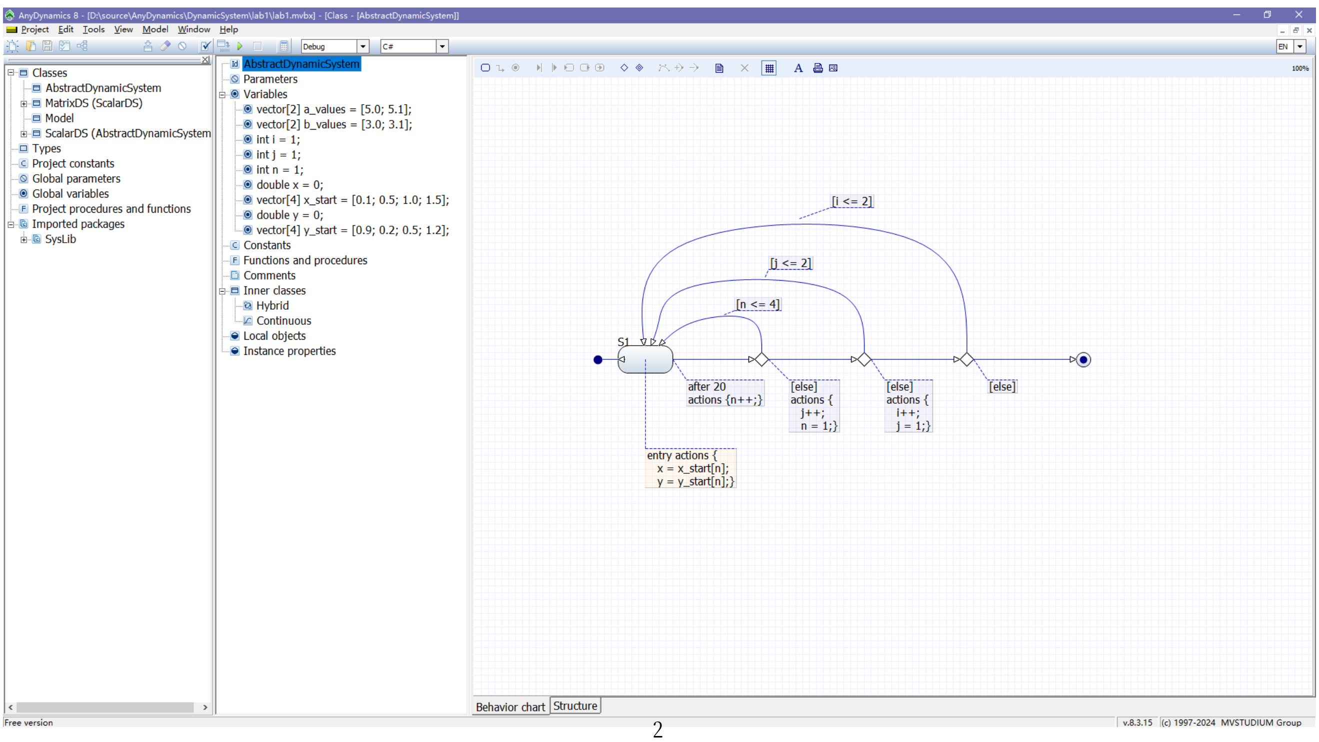
2 Реализация модели в среде AnyDynamics

Нам необходимо реализовать следующие части согласно требованиям темы:

1. Абстрактный класс (AbstractDynamicSystem)
2. Родительский класс (ScalarDS)
3. Класс-наследник (MatrixDS)

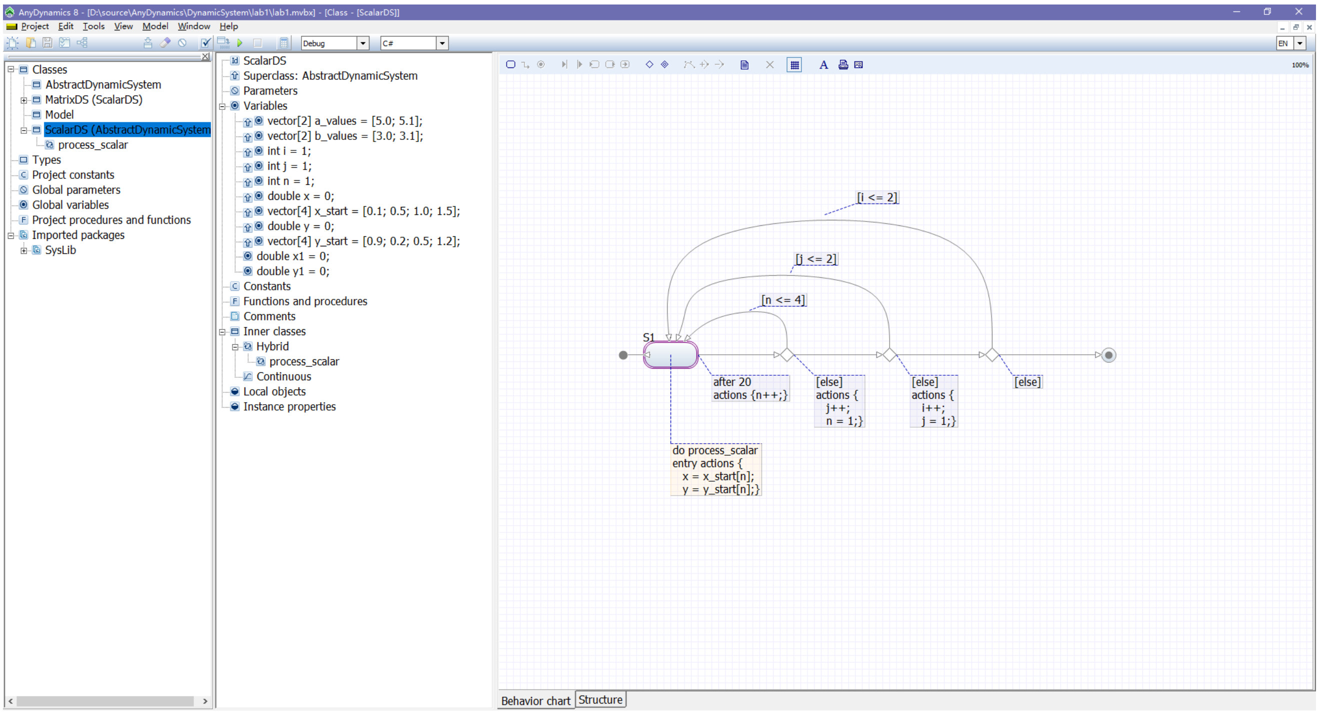
2.1 Абстрактный класс

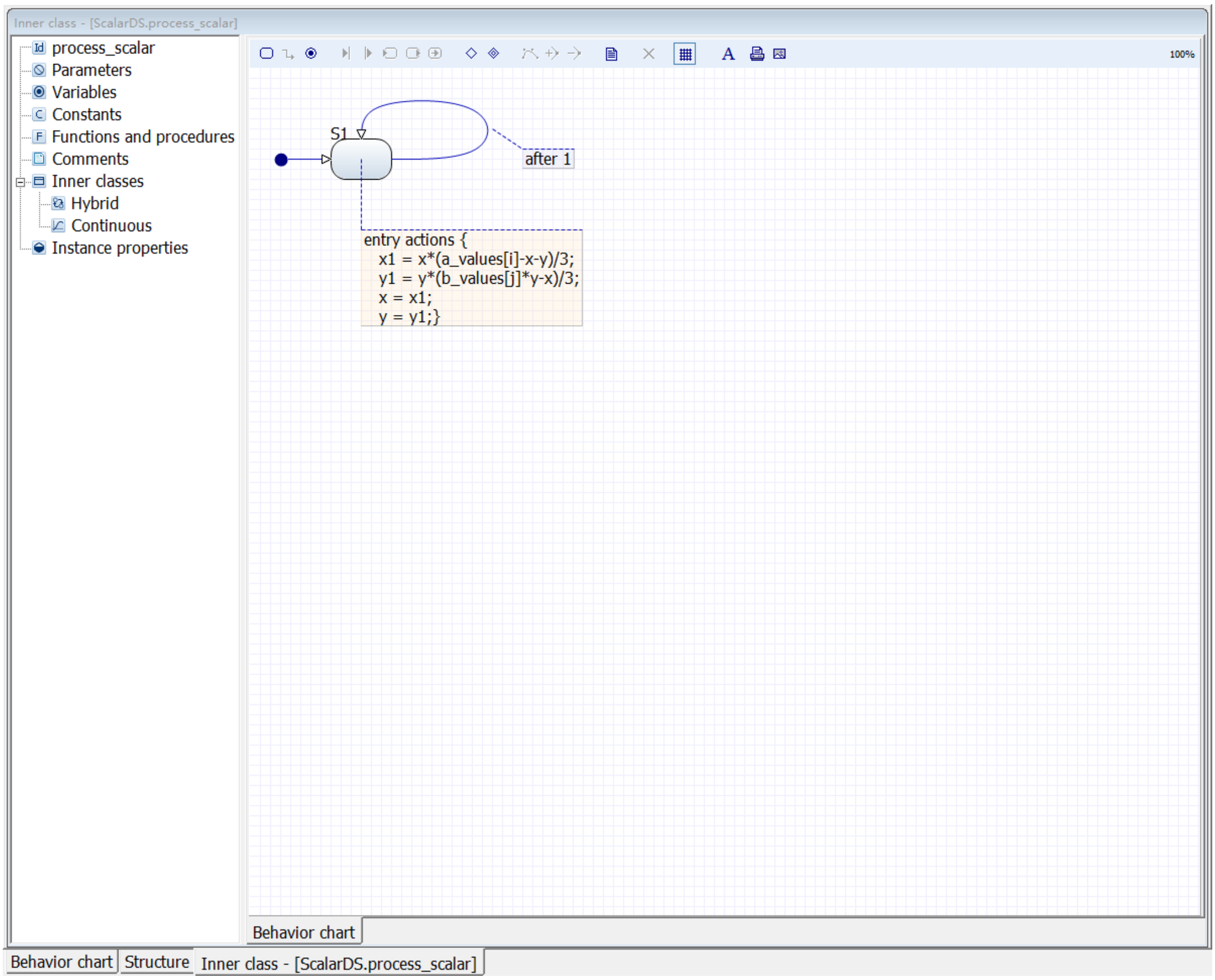
Этот абстрактный класс определяет базовый набор свойств и методов для построения фазовых графиков и системного анализа.



2.2 Родительский класс (ScalarDS)

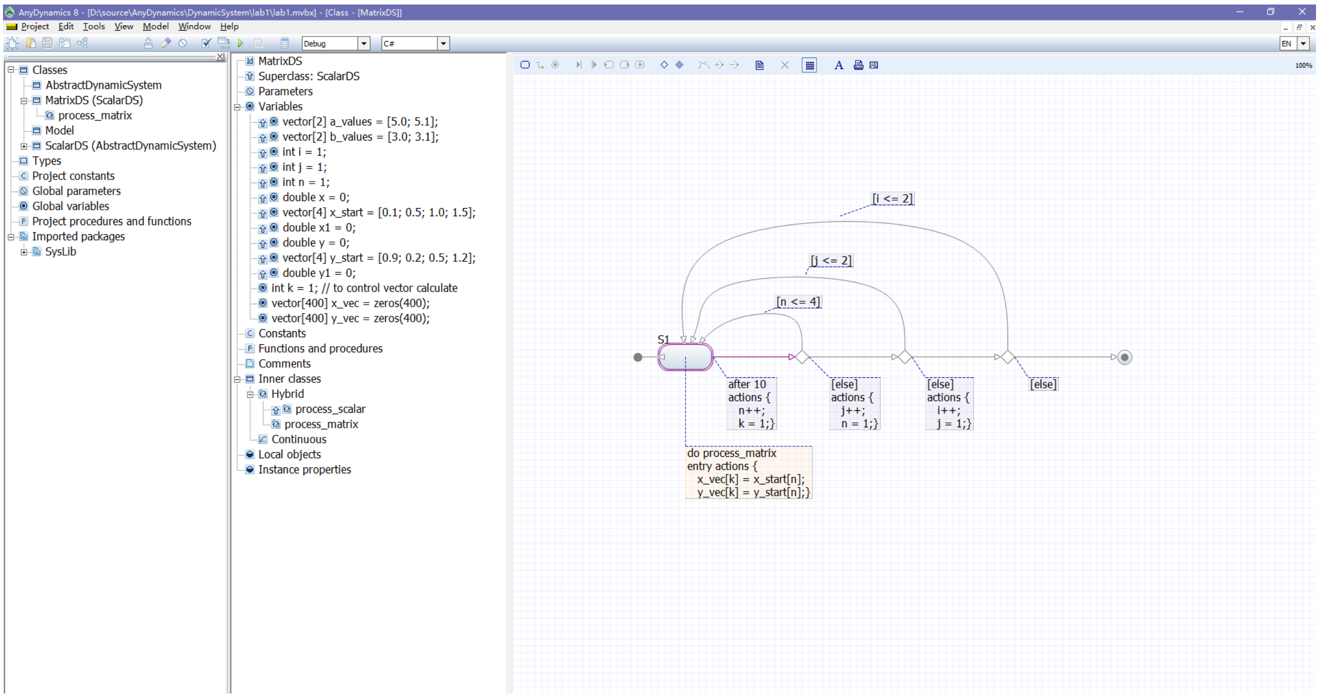
Родительский класс, реализующий уравнения в скалярной форме.

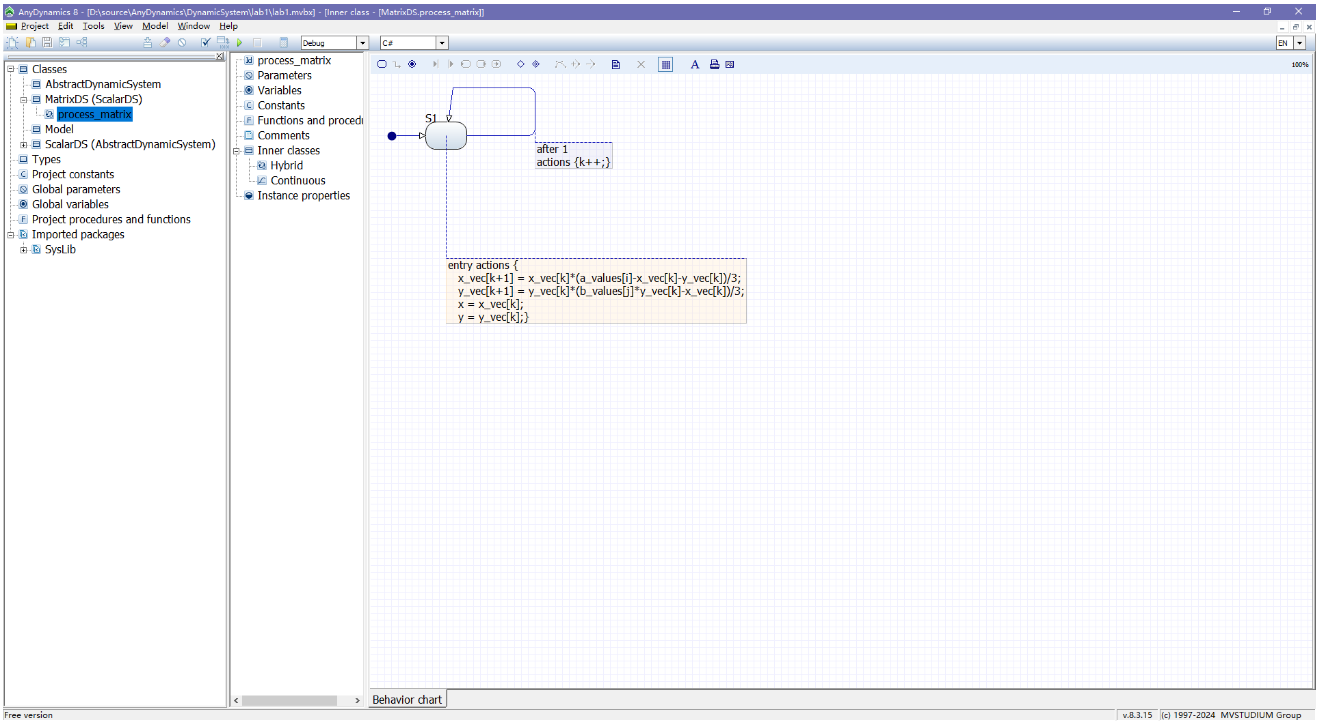




2.3 Класс-наследник (MatrixDS)

Реализуйте подкласс (MatrixDS), унаследованный от родительского класса ScalarDS: этот подкласс специализируется на обработке уравнений в матричной форме (векторно-матричной форме). Это означает, что по сравнению с родительским классом он может обрабатывать более сложные математические модели, особенно те системы, которые требуют матричных и векторных вычислений. Подклассы должны расширять функциональность родительского класса для поддержки анализа уравнений в матричной форме.

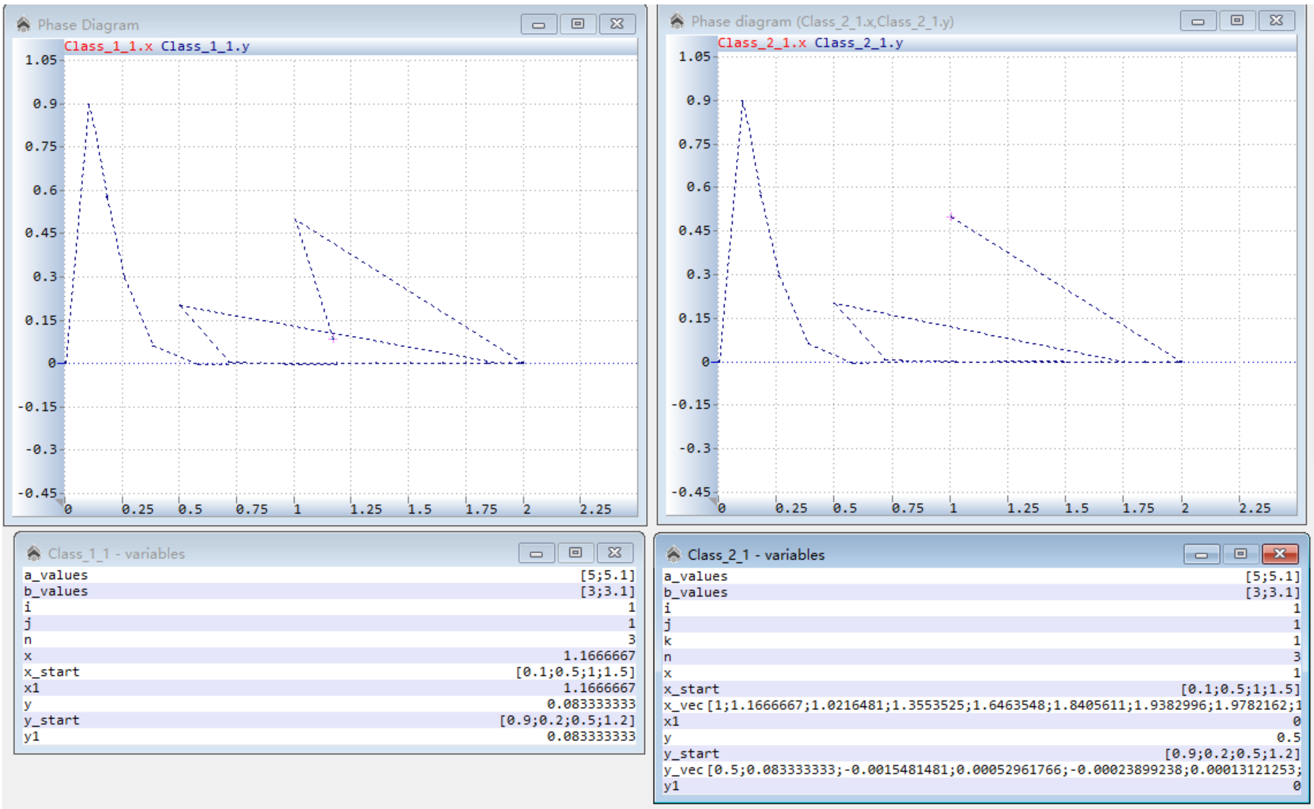




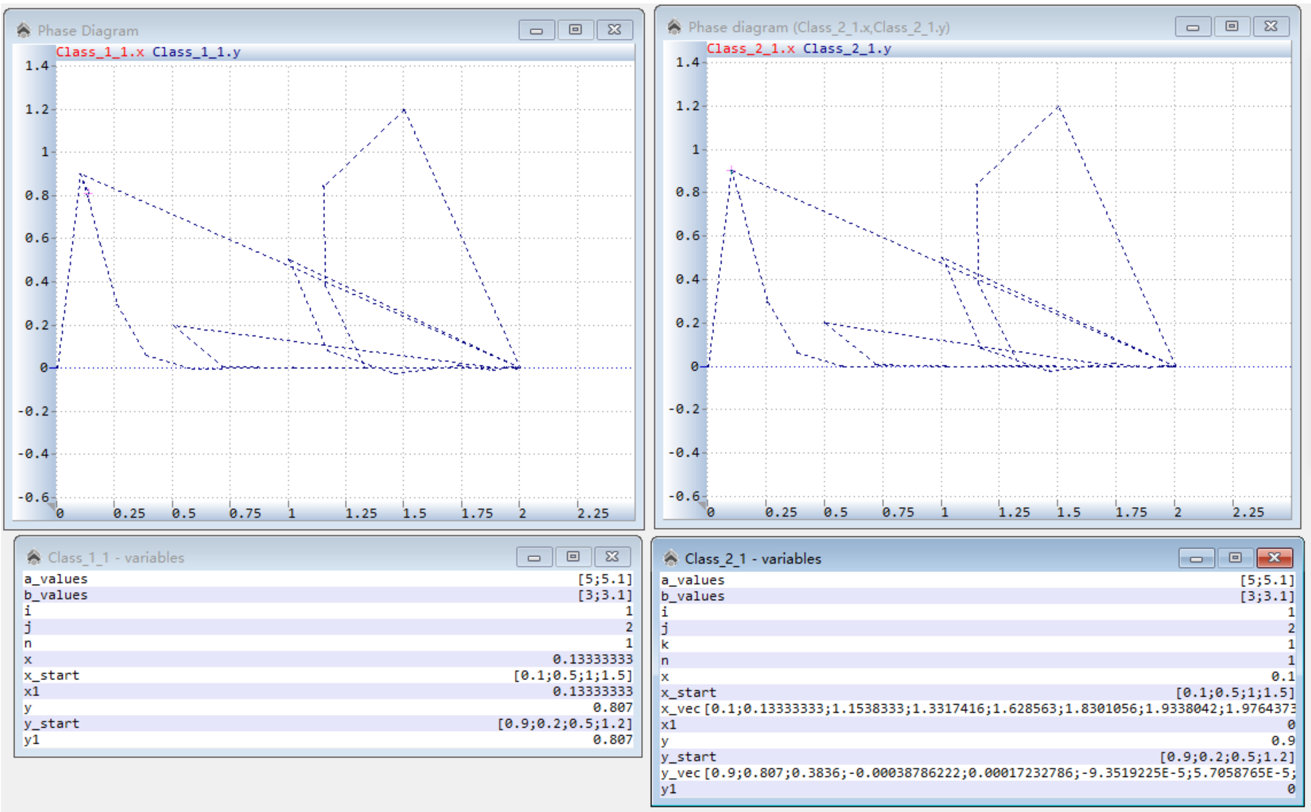
3 Анализ результата

3.1 Фазовые портреты

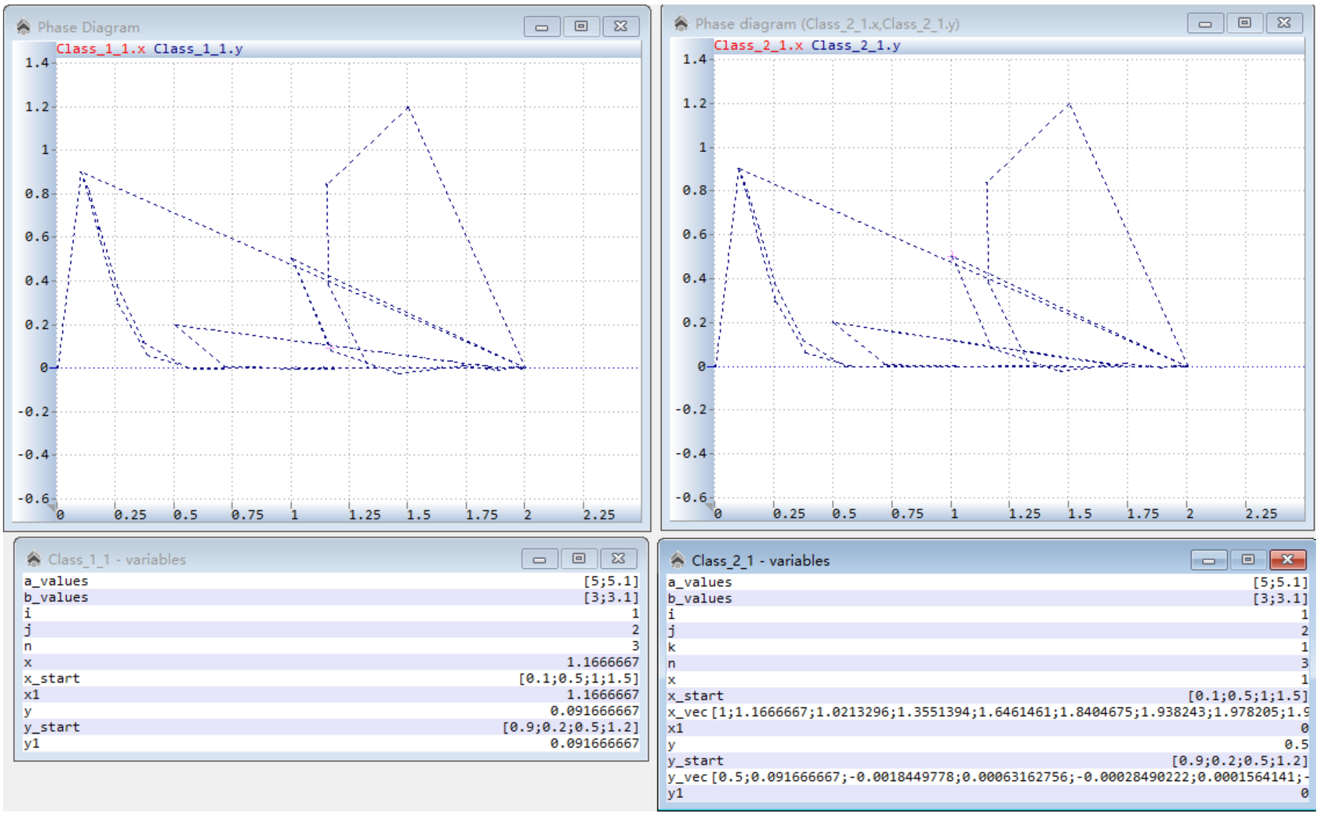
t=20:



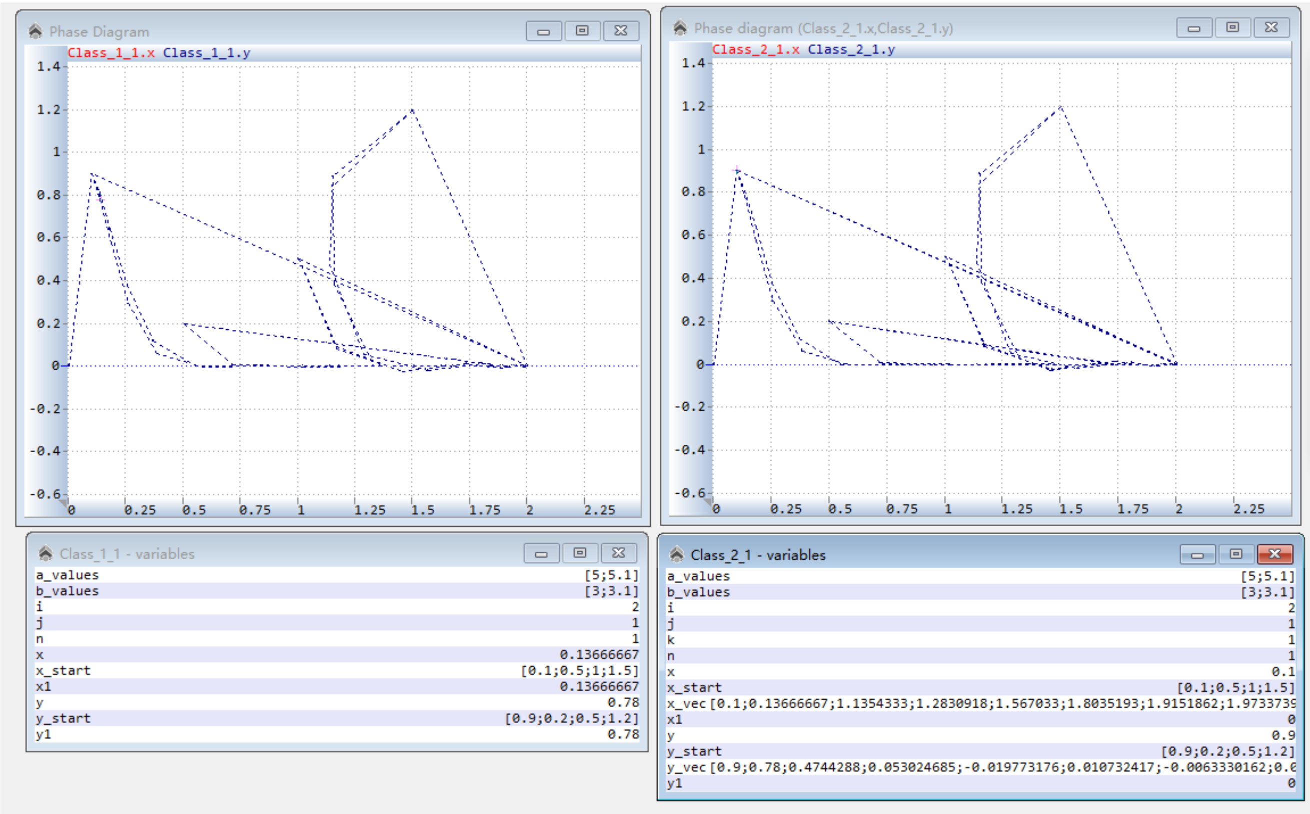
t=40:



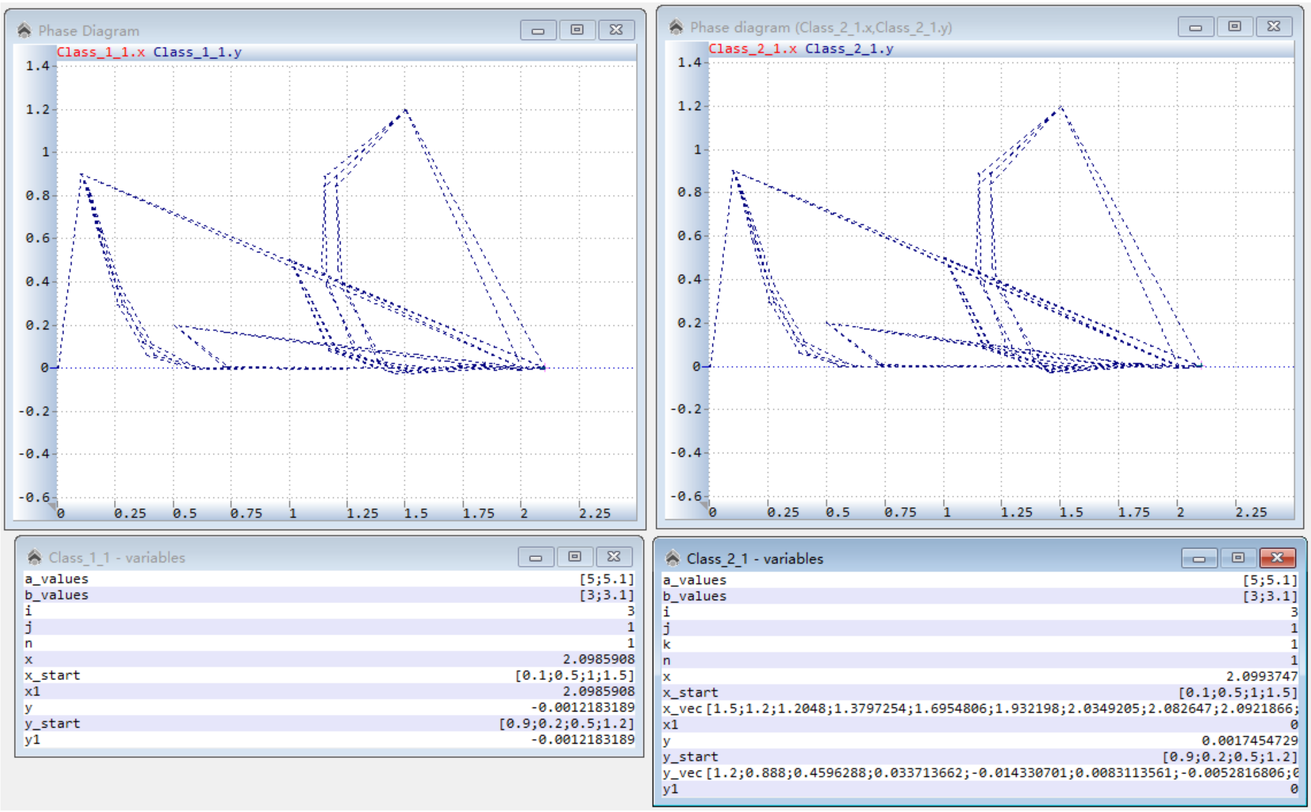
t=60:

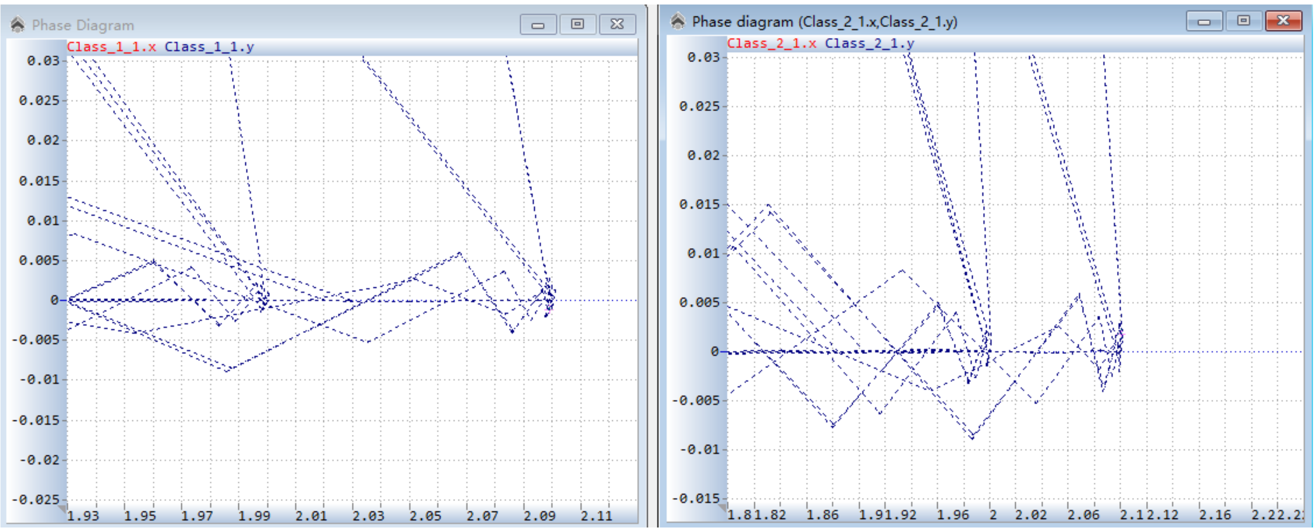

Phase Diagram
Class_1_1.x Class_1_1.y
Phase diagram (Class_2_1.x,Class_2_1.y)
Class_2_1.x Class_2_1.y
Class_1_1 - variables
a_values [5;5.1]
b_values [3;3.1]
i 1
j 2
n 3
x 1.1666667
x_start [0.1;0.5;1;1.5]
x1 1.1666667
y 0.091666667
y_start [0.9;0.2;0.5;1.2]
y1 0.091666667
Class_2_1 - variables
a_values [5;5.1]
b_values [3;3.1]
i 1
j 2
k 1
n 3
x 1
x_start [0.1;0.5;1;1.5]
x_vec [1;1.1666667;1.0213296;1.3551394;1.6461461;1.8404675;1.938243;1.978205;1.9
x1 0
y 0.5
y_start [0.9;0.2;0.5;1.2]
y_vec [0.5;0.091666667;-0.0018449778;0.00063162756;-0.00028490222;0.0001564141;-
y1 0

t=80:


Phase Diagram
Class_1_1.x Class_1_1.y
Phase diagram (Class_2_1.x,Class_2_1.y)
Class_2_1.x Class_2_1.y
Class_1_1 - variables
a_values [5;5.1]
b_values [3;3.1]
i 2
j 1
n 1
x 0.13666667
x_start [0.1;0.5;1;1.5]
x1 0.13666667
y 0.78
y_start [0.9;0.2;0.5;1.2]
y1 0.78
Class_2_1 - variables
a_values [5;5.1]
b_values [3;3.1]
i 2
j 1
k 1
n 1
x 0.1
x_start [0.1;0.5;1;1.5]
x_vec [0.1;0.13666667;1.1354333;1.2830918;1.567033;1.8035193;1.9151862;1.973373
x1 0
y 0.9
y_start [0.9;0.2;0.5;1.2]
y_vec [0.9;0.78;0.4744288;0.053024685;-0.019773176;0.010732417;-0.0063330162;0.0
y1 0

t=240:





Из анализа заданных точек траектории мы знаем, что при каждом наборе параметров и начальном состоянии траектория системы со временем приближается к определенному значению, и эти конкретные значения могут быть неподвижными точками системы. Фиксированная точка — это точка, в которой состояние больше не меняется во время последовательных итераций, а именно $x_{n+1} = x_n$ и $y_{n+1} = y_n$.

Для каждого заданного начального состояния мы наблюдаем, что траектория постепенно стабилизируется и стремится к определенной точке. В случае a= 5,0 траектория со временем приближается к точке (2, 0). В случае a= 5,1 траектория со временем приближается к точке (2.1, 0).

a	b	Приблизительная точка
5.0	3.0	(2.00000011, 1.85622657e-07)
5.0	3.1	(1.99998634, -2.04587005e-05)

a	b	Приблизительная точка
5.1	3.0	(2.10001208, 1.72693965e-05)
5.1	3.1	(2.09997589, -3.44286818e-05)

3.2 Анализ неподвижных точек

Неподвижные точки — это точки, удовлетворяющие условиям $x_{n+1} = x_n$ и $y_{n+1} = y_n$, что означает, что система больше не будет меняться после достижения этих точек.

Из данного уравнения мы можем проанализировать фиксированную точку, установив $x_{n+1} = x_n = x$ и $y_{n+1} = y_n = y$, чтобы получить:

$$x = \frac{x(a - x - y)}{3}$$
$$y = \frac{y(by - x)}{3}$$

Рассматривая случай неподвижных точки y, мы можем обнаружить, что если $y = 0$, то второе уравнение может быть удовлетворено (обратите внимание, что если $b \neq 0$, то $y = 0$ является решением). Затем, подставив $y = 0$ в первое уравнение, мы можем найти (x). Первое уравнение упрощается до:

$$x = \frac{x(a - x)}{3}$$

Чтобы это уравнение выполнялось, помимо $x = 0$, x также может быть таким значением, что $a - x = 3$, то есть $x = a - 3$.

для (a = 5.0) Когда a = 5.0, решением является $x = 5.0–3 = 2$, поэтому фиксированной точкой является (2, 0).

для (a = 5.1) Когда a = 5.1, решением является $x = 5.1 - 3 = 2.1$, поэтому фиксированной точкой является (2.1, 0).

Следовательно, разные значения a приводят к разным точкам устойчивого состояния системы. Такие тонкие изменения параметров приводят к изменениям в динамическом поведении системы, особенно в положении фиксированной точки динамической системы, подчеркивая чувствительную зависимость динамической системы от параметров.

4 Заключение

В ходе выполнения данного задания была разработана и успешно реализована библиотека классов для построения фазовых портретов динамических систем, включая обработку систем, описываемых как в скалярной, так и в векторно-матричной форме. Разработка библиотеки осуществлялась согласно заданию, включающему в себя создание абстрактного класса, родительского класса для работы с уравнениями в скалярной форме и класса-наследника для работы с уравнениями в векторно-матричной форме.

Развитие проекта в среде AnyDynamics демонстрирует значимость и преимущества объектно-ориентированного подхода в моделировании и анализе сложных динамических систем. Реализация представленных классов и методов позволяет глубже анализировать системы, предоставляя инструменты для построения фазовых портретов и нахождения неподвижных точек, что является ключевым аспектом при исследовании поведения динамических систем.