

Міністерство освіти і науки України  
Український державний університет науки і технологій

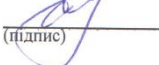
Факультет «Комп'ютерні технології і системи»  
Кафедра «Комп'ютерні інформаційні технології»

**Пояснювальна записка**  
до кваліфікаційної роботи бакалавра

на тему: «Розробка програми імітаційного моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі» за освітньою програмою: «Інженерія програмного забезпечення» зі спеціальності: «121 Інженерія програмного забезпечення»  
Виконав: студент групи «ПЗ1812»

Керівник:   
(підпис) студента

Нормоконтролер:   
(підпис)

  
(підпис)

Нікіта ЮХНО  
(Ім'я ПРІЗВИЩЕ)

зав. кафедрою Вадим ГОРЯЧКІН  
(посада, Ім'я ПРІЗВИЩЕ)

доц. Олена КУРОП'ЯТНИК  
(посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент

  
(підпис)

Дніпро – 2022 рік

Ministry of Education and Science of Ukraine  
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»  
Department «Computer information technology»

**Explanatory Note**  
to Bachelor's Thesis

on the topic: « Development of simulation program modeling of the mechanism of functioning macroeconomic system of the state on based on the classical market model» according to educational curriculum «Software engineering» in the Speciality: «121 Software engineering»

Done by the student of the group PZ1812:

Nikita YUHNO

Scientific Supervisor:

Vadym HORIACHKIN


Normative controller:

Olena KUROIATNYK

Dnipro – 2022

Міністерство освіти і науки України  
Український державний університет науки і технологій

Факультет: «Комп'ютерні технології і системи»  
Кафедра: «Комп'ютерні інформаційні технології»  
Рівень вищої освіти: бакалавр  
Освітня програма: «Інженерія програмного забезпечення»  
Спеціальність: «121 Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ  
Завідувач кафедри КІТ  
  
(підпис) Вадим ГОРЯЧКІН

Дата \_\_\_\_\_

### З А В Д А Н Н Я

на кваліфікаційну роботу бакалавра  
студенту Юхно Нікіті Андрійовичу

1. Тема роботи: «Розробка програми імітаційного моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі»

Керівник роботи: Горячкін Вадим Миколайович, завідувач кафедрою  
затверджені наказом № 77 ст від 08.12.2021

2. Строк подання студентом роботи: \_\_\_\_\_.\_\_\_\_.202\_ р.

3. Вихідні дані до дипломної роботи \_\_\_\_\_

4. Зміст пояснювальної записки (перелік питань до розробки):  
вступ, збір та аналіз вимог, зовнішнє проєктування, внутрішнє проєктування,  
тестування та налагодження програми, висновки.

5. Перелік демонстраційного матеріалу: приклади роботи програми-аналога,  
діаграма класів, результати налагодження програми.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання теми роботи	08.12.21	
2	Перегляд та аналіз аналогів	13.01.22	
3	Зовнішнє проектування	17.03.22	
4	Розробка програми	10.05.22	
5	Внутрішнє проектування	17.05.22	
6	Тестування та відлагодження	27.05.22	
7	Написання пояснювальної записки до програми	31.05.22	
8	Подання кваліфікаційної роботи до кафедри	13.06.22	
9	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	22.06.22	

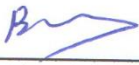
Студент

  
\_\_\_\_\_ (підпис)

Нікіта ЮХНО

(Ім'я ПРІЗВИЩЕ)

Керівник роботи:

  
\_\_\_\_\_ (підпис)

зав. кафедрою Вадим ГОРЯЧКІН

(посада, Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка складається з 8 розділів:

– вступ – в даному розділі описується сутність розробки, її актуальність.

Складається з 1 сторінки;

– збір вимог до програмного забезпечення – у цьому розділі описуються аналоги програми та література по даній предметній області, а також методика та результати опитування зацікавлених сторін для формування найбільш повних вимог до програмного забезпечення. Складається з 5 сторінок;

– зовнішнє і внутрішнє проектування – у цьому розділі проведений огляд вхідних і вихідних даних, формалізація задачі, розробка фізичного проекту, приводиться опис об'єктно-орієнтованого проектування, проектування інтерфейсу користувача, аналіз проекту, вибір мови програмування. Складається з 28 сторінок;

– тестування та налагодження – розділ включає в себе вибір стратегії тестування, опис тестів методами «чорного» та «білого» ящика. Також розділ включає в себе аналіз помилок, їх вплив на систему та вирішення проблеми. Складається з 17 сторінок;

– висновки. Складається з 1 сторінки;

– список літератури – включає в себе бібліографічний список використаної літератури. Складається з 2 сторінок;

– додатки – містить текст програми та інструкцію для користувача програми. Складається з 29 сторінок.

Кількість таблиць: 14.

Кількість рисунків: 18.

Ключові слова: економіка, макроекономіка, ефективна економічна модель, моделювання, форма, кнопка, клас, змінна.

## ЗМІСТ

Вступ.....	4
1 Збір та аналіз вимог.....	5
1.1 Опис аналогів .....	6
Висновок до розділу 1.....	9
2 Зовнішнє проектування.....	10
2.1 Функціональне призначення.....	10
2.2 Експлуатаційне призначення.....	10
2.3 Функціональні вимоги .....	10
2.4 Загальні вхідні дані .....	11
2.5 Окремі вхідні дані суб'єктів економічної діяльності.....	11
2.5.1 Виробники сировини (ВС).....	11
2.5.2 Виробники засобів виробництва (ВЗВ).....	12
2.5.3 Виробники споживчих товарів (ВСТ).....	12
2.5.4 Торгівля та Послуги (ТП) .....	13
2.5.5 Покупці споживчих товарів (ПСТ).....	13
2.5.6 Бюджет (Держава) (Б) .....	13
2.5.7 Комерційні банки (КБ).....	13
2.5.8 Нацбанк (НБ) .....	13
2.6 Вихідні дані .....	13
Висновки до розділу 2 .....	14
3 Внутрішнє проектування .....	15
3.1 Об'єктно-орієнтоване програмування.....	15
3.2 Опис відповідальності класів .....	16
3.3 Опис зв'язків класів .....	17
3.4 Діаграма класів програми .....	17
3.5 Опис специфікацій класів.....	18
3.5.1 Опис специфікацій класу Program.....	18
3.5.2 Опис специфікацій класу Form1 .....	18
3.5.3 Опис специфікацій класу Form2.....	19
3.5.4 Опис специфікацій класу Form3.....	19
3.5.5 Опис специфікацій класу Form4.....	19

3.5.6	Опис специфікацій класу Form5	19
3.5.7	Опис специфікацій класу Form6	20
3.5.8	Опис специфікацій класу Form7	20
3.5.9	Опис специфікацій класу Form8	20
3.5.10	Опис специфікацій класу Form9	20
3.5.11	Опис специфікацій класу InputData	21
3.5.12	Опис специфікацій класу Characteristics	23
3.5.13	Опис специфікацій класу Economy_events	24
3.5.14	Опис специфікацій класу Form10	24
3.5.15	Опис специфікацій класу Output_data	24
3.6	Опис інтерфейсної частини класів	24
3.6.1	Клас Program	24
3.6.2	Клас Form1	25
3.6.3	Клас Form2	29
3.6.4	Клас Form3	30
3.6.5	Клас Form4	30
3.6.6	Клас Form5	30
3.6.7	Клас Form6	31
3.6.8	Клас Form7	32
3.6.9	Клас Form8	33
3.6.10	Клас Form9	34
3.6.10	Клас Form10	34
3.7	Проектування інтерфейсу користувача	35
3.8	Аналіз проекту	36
3.9	Вибір мови програмування	36
	Висновки до розділу 3	37
4	Тестування та налагодження програми	38
4.1	Стратегія тестування	38
4.2	Специфікація та текст модулів	41
4.2.1	Опис специфікацій та текст функції Grafic_Update	41
4.3	Тестування методом «білої скриньки»	44
4.4	Тестування методом «чорної скриньки»	48

4.5 Вибір стратегії налагодження .....	53
4.6 Результати налагодження .....	54
Висновки до розділу 4 .....	54
Висновки.....	55
Література.....	56
Додатки.....	58

## ВСТУП

Дипломна робота спрямована на програмне моделювання високоефективної макроекономічної системи, яка повністю співпадає з тим, що пропонує сучасна економічна теорія і функціонує на основі таких самих економічних принципів.

Мета роботи полягає у створенні додатку, який виконує моделювання макроекономічної системи на основі дослідження фінансових потоків у макроекономіці між різними групами виробників і споживачів для кращого розуміння кінцевого результату їхньої діяльності.

Людина є ключовою фігурою економіки та її кінцевим споживачем. Завдяки людині здійснюється науково-технічний прогрес та розвивається економіка. Людина – це не «трудовий ресурс», не «робоча сила» і не «гвинтик», а основна дійова особа економічного процесу, для задоволення потреб якого все, власне, і робиться. Тому економіка має бути підпорядкована інтересам людини, а не людина має бути підпорядкована економіці. Вся економіка заснована лише на людській праці. Завдяки праці створюються всі блага, необхідні для людського життя: будівлі, машини, обладнання, предмети споживання та багато іншого. Людина, як біологічний вид, для свого повноцінного існування, обов'язково має харчуватися, носити одяг, мати житло, а також користуватися іншими різноманітними матеріальними та культурними благами. Кожна людина, незалежно від того, в якій сфері працює і що виробляє врешті-решт отримує за свою працю заробітну плату, яку потім має обміняти на необхідні йому споживчі блага.

Практичне призначення розроблюваної програми полягає в тому, щоб змоделювати макроекономічну систему на базі ринкової моделі.

## 1 ЗБІР ТА АНАЛІЗ ВИМОГ

Мета встановлення вимог полягає в тому, щоб дати розгорнуте визначення функціональних, а також не функціональних вимог, які учасники проекту очікують затвердити в системі, що реалізується та розгортається.

Методи збору вимог до програмного забезпечення. При вивченні предметної області та збору вимог про майбутній об'єкт інформатизації використовується методологія системного аналізу.

Метод бесіди – психологічний вербально-комунікативний метод, що полягає у веденні тематично спрямованого діалогу між дослідником і респондентом з метою отримання відомостей від останнього. Метод бесіди і консультацій найчастіше проводиться з керівниками підприємств і підрозділів або у формі ділової консультації з фахівцями з питань, що 40 мають глобальний характер і належать до визначення проблем і стратегій розвитку та управління підприємством.

Прототипування (prototyping) – це найбільш часто використовуваний сучасний метод виявлення вимог. Програмні прототипи конструюються для візуалізації системи або її частини для замовників з метою отримання їх відгуків. Прототип – це дуже ефективний спосіб виявлення вимог, які важко отримати від замовника за допомогою інших засобів.

Еволюційний прототип (evolutionary), який зберігається після виявлення вимог і використовується для створення кінцевого програмного продукту. Еволюційний прототип націлений на прискорення поставок товарів. Як правило, він концентрується на ясно викладених вимогах, тому першу версію продукту можна надати замовнику досить швидко (хоча її функціональні можливості, як правило, неповні)

При виконанні роботи були використані такі методи збору та аналізу вимог як метод діалогу та метод еволюційного прототипування. Комбінування методу діалогу та еволюційного прототипування дозволило швидко та зручно визначити вимоги та розробляти програму крок за кроком, поки усі вимоги не були покриті та погоджені із замовником.

Після збору та аналізу вимог, зібрана інформація була проаналізована і на її основі було розроблено бізнес-процес системи.

Бізнес процес для моделювання макроекономічної системи виконується наступним чином:

1. користувач задає вхідні дані;
2. відбувається моделювання системи із виводом результатів на екран у вигляді діаграм;
3. запис усіх розрахунків до файлу.

Бізнес процес є завершеним, так як користувач отримав усю необхідну інформацію про змодельовану систему у вигляді діаграм та у виглядів чистих розрахунків.

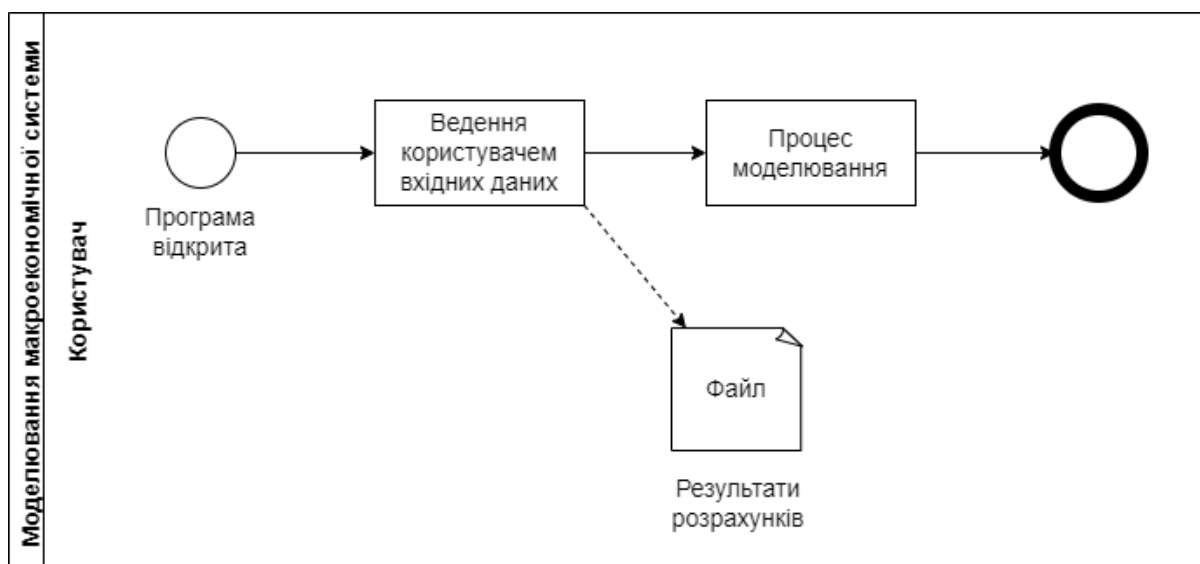


Рисунок 1.1 – Бізнес-процес моделювання макроекономічної системи

### 1.1 Опис аналогів

Існує тільки один аналог для моделювання макроекономічної моделі на базі безкризової економічної моделі та на базі класичної ринкової моделі - **SimpleModel**.

SimpleModel – перша версія додатку, яка була розроблена з метою моделювання макроекономічної системи.

Головною перевагою цієї програми є те, що вона виконує свою головну функцію – виконує моделювання макроекономічної системи на базі ринкової

моделі та на базі безкризової економічної моделі. Також вона має простий та зручний для користувача дизайн.

Недоліками програми є те, що всі вхідні дані програми вбудовані у програмний код, що не дає можливості змодельовати макроекономічну систему при різних даних. Також програма була розроблена з прив'язкою до старих систем Windows XP, Windows 7, що не дає гарантій на безвідмовну та надійну роботу програми.

Приклади роботи програми:

На рис. 1.2 продемонстрований результат роботи програми після її запуску. Користувач потрапляє на початкову форму програми, де він може розпочати моделювання макроекономічної системи на основі базової ринкової моделі.

На рис. 1.3 зображений результат роботи програми після натискання кнопки «Моделювання». На головній формі програми з'явилися графіки, які демонструють результати моделювання макроекономічної системи на основі базової ринкової моделі.

На рис. 1.4 зображений результат роботи програми на стовпчикових діаграмах після процесу моделювання макроекономічної системи на основі базової ринкової моделі протягом 42 днів.

На рис. 1.5 зображений файл з результатами моделювання макроекономічної системи на основі базової ринкової моделі. Цей файл створюється після натискання користувачем кнопки «Записати у файл».

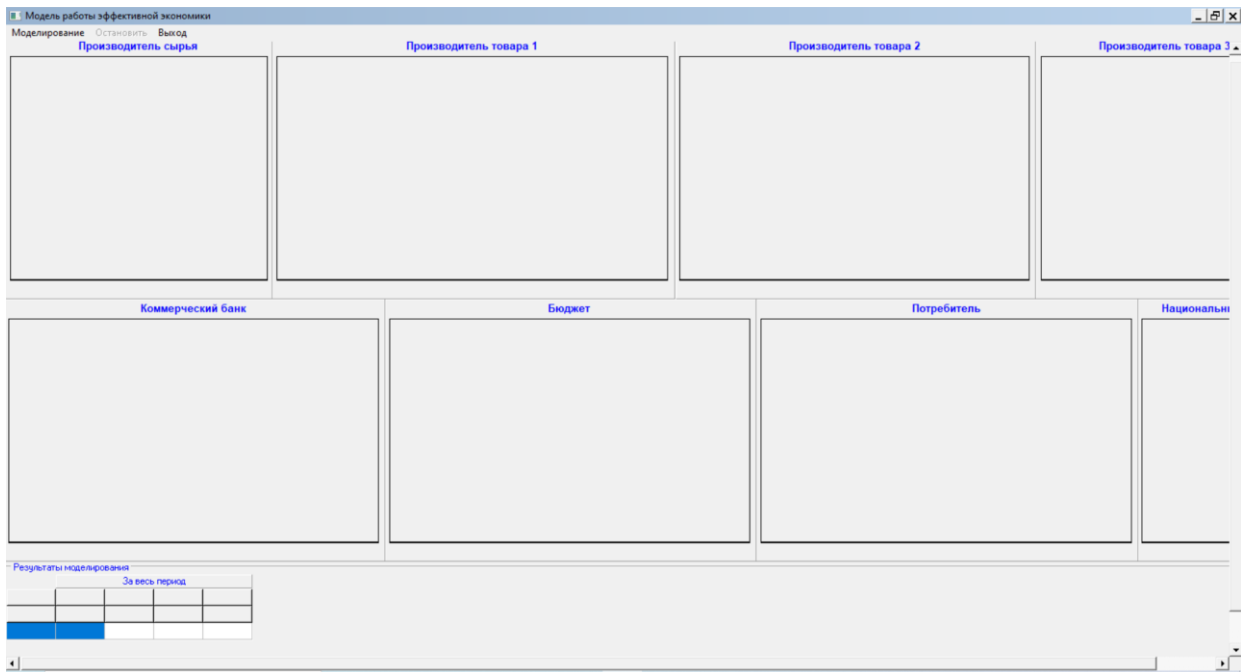


Рисунок 1.2 – Початкова форма програми

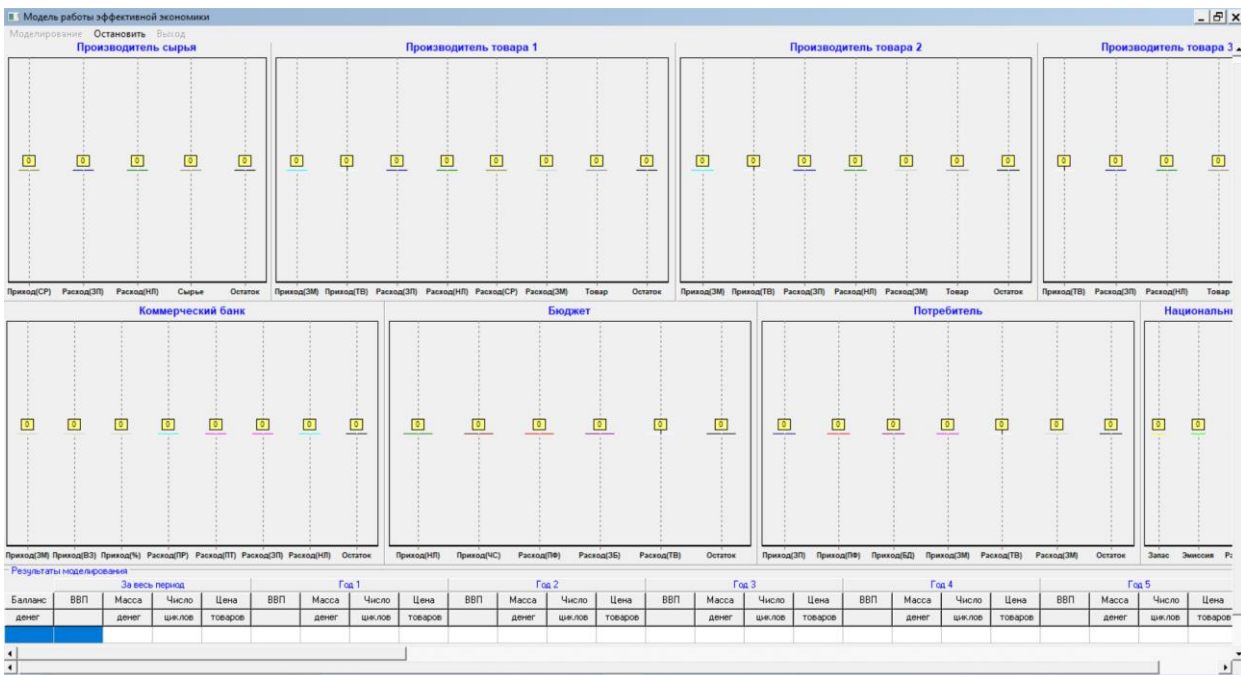


Рисунок 1.3 – Програма після натискання кнопки «Моделювання»

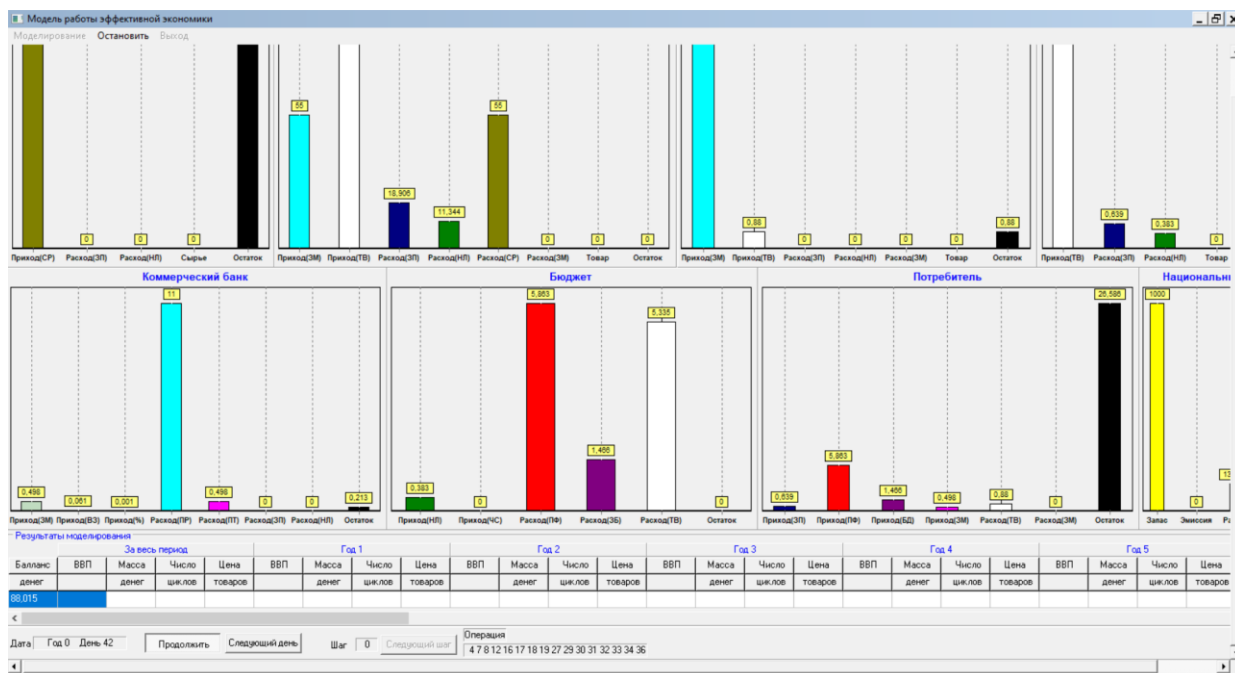


Рисунок 1.4 – Програма після моделювання 42 дня

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W		
13		17 000;10	625;6	375;43	000;18	000;32	564;0	000;0	000;17	000;23	715;9	510;0	100;0	000;0	000;0	000;0	000;2	319;0	417;0	250;0	000;7	107;0	071;9	51	
14		17 000;0	000;0	000;43	000;18	000;23	715;0	000;0	000;17	000;0	000;9	510;0	100;0	053;0	038;0	009;0	250;0	317;1	449;0	870;0	000;0	009;0	000;9	51	
15		17 000;0	000;0	000;43	000;15	000;23	715;4	197;2	518;17	000;0	000;9	510;0	100;0	000;0	000;0	000;0	250;0	923;0	198;0	119;0	000;15	000;0	000;15	00	
16		43 000;0	000;0	000;0	000;15	000;23	715;0	000;0	000;43	000;0	000;9	510;0	100;0	000;0	000;0	000;0	250;0	121;0	577;0	346;0	000;15	000;0	000;15	00	
17		43 000;26	875;16	125;24	000;15	000;23	715;0	000;0	000;43	000;65	317;9	510;0	250;0	000;0	000;0	000;0	000;5	660;0	076;0	045;0	000;15	000;0	000;15	00	
18		43 000;0	000;0	000;24	000;15	000;23	715;0	000;0	000;43	000;65	317;10	630;0	250;0	133;0	094;0	023;0	110;2	171;10	181;6	109;0	000;5	583;0	056;10	63	
19		43 000;0	000;0	000;24	000;15	000;23	715;0	000;0	000;43	000;65	317;10	630;0	250;0	000;0	000;0	000;0	110;0	292;1	357;0	814;0	000;3	670;0	037;10	63	
20		43 000;0	000;0	000;24	000;15	000;23	715;0	000;0	000;43	000;65	317;10	630;0	250;0	000;0	000;0	000;0	110;0	043;0	183;0	110;0	000;3	387;0	034;10	63	
21		43 000;0	000;0	000;24	000;15	000;23	715;0	000;0	000;43	000;65	317;10	630;0	110;0	000;0	000;0	000;0	000;0	010;0	027;0	016;0	000;3	321;0	033;10	63	
22		43 000;0	000;0	000;24	000;15	000;23	715;0	000;0	000;43	000;65	317;10	630;0	110;0	058;0	041;0	010;0	500;0	020;0	006;0	004;0	000;3	309;0	033;10	63	
23		43 000;0	000;0	000;24	000;15	000;23	715;0	000;0	000;43	000;65	317;10	630;0	110;0	000;0	000;0	000;0	500;0	007;0	013;0	008;0	000;3	263;0	033;10	63	
24		43 000;0	000;0	000;24	000;15	000;23	715;0	000;0	000;43	000;65	317;10	630;0	110;0	000;0	000;0	000;0	500;0	005;0	004;0	003;0	000;3	229;0	032;10	63	
25		43 000;0	000;0	000;24	000;15	000;23	715;0	000;0	000;43	000;65	317;10	630;0	500;0	000;0	000;0	000;0	000;0	005;0	003;0	002;0	000;3	197;0	032;10	63	
26		43 000;0	000;0	000;24	000;15	000;23	715;0	000;0	000;43	000;65	317;10	630;0	500;0	266;0	188;0	047;0	550;0	071;0	003;0	002;0	000;3	282;0	033;10	63	
27		43 000;0	000;0	000;24	000;15	000;65	317;0	000;0	000;43	000;0	000;3	820;0	500;0	000;0	000;0	000;0	550;0	515;2	432;1	459;0	000;3	282;0	033;10	82	
28		24 000;0	000;0	000;0	000;15	000;65	317;13	948;8	369;24	000;0	000;3	820;0	500;0	000;0	000;0	000;0	550;4	627;0	322;0	193;0	000;3	282;0	033;10	82	
29		24 000;15	000;9	000;55	000;15	000;65	317;0	000;0	000;24	000;39	192;3	820;0	550;0	000;0	000;0	000;0	600;3	757;2	892;1	735;0	000;3	282;0	033;10	82	
30		24 000;0	000;0	000;55	000;15	000;39	192;0	000;0	000;24	000;0	000;3	820;0	550;0	292;0	206;0	052;0	600;0	565;2	348;1	409;0	000;0	051;0	001;3	82	
31		24 000;0	000;0	000;55	000;12	000;39	192;0	495;5	697;24	000;0	000;3	820;0	550;0	000;0	000;0	000;0	600;2	068;0	353;0	212;0	000;12	000;0	001;12	00	
32		55 000;0	000;0	000;0	000;12	000;39	192;0	000;0	000;55	000;0	000;3	820;0	550;0	000;0	000;0	000;0	600;0	271;1	293;0	776;0	000;12	000;0	001;12	00	
33		55 000;34	375;20	625;73	000;12	000;39	192;0	000;0	000;55	000;70	675;3	820;0	600;0	000;0	000;0	000;0	000;7	254;0	170;0	107;0	000;12	000;0	001;12	00	
34		55 000;0	000;0	000;73	000;12	000;70	675;0	000;0	000;55	000;0	000;3	820;0	600;0	319;0	225;0	056;0	510;1	031;4	534;2	720;0	000;0	056;0	001;12	00	
35		55 000;0	000;0	000;73	000;18	000;70	675;9	797;5	878;55	000;0	000;3	820;0	600;0	000;0	000;0	000;0	510;2	193;0	644;0	387;0	000;18	000;0	001;18	00	
36		73 000;0	000;0	000;0	000;18	000;70	675;0	000;0	000;73	000;0	000;3	820;0	600;0	000;0	000;0	000;0	510;0	288;1	370;0	822;0	000;18	000;0	001;18	00	
37		73 000;45	625;27	375;31	000;18	000;70	675;0	000;0	000;73	000;99	718;3	820;0	510;0	000;0	000;0	000;0	000;9	619;0	180;0	108;0	000;18	000;0	001;18	00	
38		73 000;0	000;0	000;31	000;18	000;70	675;0	000;0	000;73	000;99	718;3	820;0	510;0	271;0	191;0	048;0	600;1	329;6	012;3	607;0	000;5	891;0	059;18	00	
39		73 000;0	000;0	000;31	000;18	000;70	675;0	000;0	000;73	000;99	718;3	820;0	510;0	000;0	000;0	000;0	600;0	182;0	831;0	499;0	000;4	698;0	047;18	00	
40		73 000;0	000;0	000;31	000;18	000;70	675;0	000;0	000;73	000;99	718;3	820;0	510;0	000;0	000;0	000;0	600;0	030;0	114;0	068;0	000;4	501;0	045;18	00	
41		73 000;0	000;0	000;31	000;18	000;70	675;0	000;0	000;73	000;99	718;3	820;0	510;0	000;0	000;0	000;0	600;0	030;0	010;0	010;0	000;4	426;0	046;18	00	
42		Year 1																							

Рисунок 1.5 – Файл з розрахунками результатів моделювання

## Висновок до розділу 1

Як видно із прикладів роботи програми, є недоліки у її роботі, такі як: некоректне відображення більшості символів тексту, із за чого стає неможливим зрозуміти до чого відноситься та чи інша інформація, а також вихідний файл з розрахунками не має заголовків у колонках, що теж унеможлиблює розуміння природу даних та до чого вони відносяться.

## **2 ЗОВНІШНЄ ПРОЕКТУВАННЯ**

### **2.1 Функціональне призначення**

Програма моделює макроекономічну систему на основі базової ринкової моделі [1, 2]. У програмі представлені 8 економічних підприємств, програма моделює взаємодію між ними та виводить інформацію про представлені підприємства у вигляді стовпчикових діаграм.

### **2.2 Експлуатаційне призначення**

Дана програма допомагає користувачу краще зрозуміти механізм функціонування сучасної макроекономічної моделі [3, 4]. У кінці моделювання користувач може переглянути результати програми, де показані більш детальні події, які призвели до даних результатів.

### **2.3 Функціональні вимоги**

Завданням даного ПЗ є демонстрація сучасної макроекономічної моделі.

Для спрощення розрахунків було прийнято агрегований спосіб моделювання [5, 6]. Кожен агрегований суб'єкт економічної діяльності (АСЕД) являє собою величезну кількість схожих об'єктів, але які виробляють абсолютно різну продукцію (різна ціна, різна тривалість технологічного циклу тощо) [7, 8].

Процес моделювання охоплює якийсь календарний проміжок часу, тому потрібен лічильник днів. Вважатимемо, що кожен день щось послідовно відбувається (взяття кредиту, покупка сировини, виготовлення продукції, продаж продукції, виплата заробітної плати, купівля сировини тощо). У даному моделюванні нас цікавить не кількість виробленого товару (що практично неможливо підрахувати), а умовну кількість грошей, яка обертається в макроекономічній системі та як воно впливає на макроекономічні показники.

У кожному АСЕД мають бути лічильники, які вважають прихід грошей і витрати грошей, а також показують скільки грошей знаходиться зараз і куди були спрямовані гроші.

Програма має щодня переходити від одного АСЭД до іншого та робити одну операцію згідно загальним вхідним даним.

## **2.4 Загальні вхідні дані**

Перелік загальних вхідних даних:

1. задається відсоток грошей на сплату податку, %, зберігається у змінній типу даних int;
2. задається відсоток грошей, які спрямовуються на виплату заробітної плати, %, зберігається у змінній типу даних int;
3. задаються відсотки за кредитами та депозитами, %, зберігається у змінній типу даних int;
4. задається ставка Національного банку, %, зберігається у змінній типу даних int;
5. задається період повернення кредиту, зберігається у змінній типу даних int;
6. задається інтервал періоду моделювання на днях або вказується – 1 рік, 2 роки, зберігається у змінній типу даних int;
7. задається період часу, через який виплачується зарплата та податок (або конкретна кількість днів, або можливий період), зберігається у змінній типу даних int.

## **2.5 Окремі вхідні дані суб'єктів економічної діяльності**

### **2.5.1 Виробники сировини (ВС)**

Загальний лічильник фіксує 1-й день роботи.

Визначається початкова кількість грошей в умовних одиницях (задається оператором). Це може бути або конкретне число, або вказано можливий інтервал, з якого програма випадковим чином згідно нормальному закону розподілу сама має обрати якесь число (наприклад, 10-100). (мабуть другий варіант більш зручним). (наприклад, програма обрала 65) Це означає, що цю кількість грошей треба витратити на закупівлю необхідних товарів для випуску сировини.

Задається можливий період циклу технологічного процесу днями (наприклад, 1-20). Програма випадково вибирає тривалість технологічного циклу (наприклад, 15). Це означає що через 15 днів буде виготовлено товар.

Визначається відсоток торгової націнки в деяких межах (наприклад, 20-100). Наступного дня передбачається та сама процедура за тими ж вихідними даними.

Завдання вихідних даних по кожному АСЭД бачиться у вигляді форми, куди вводяться потрібні числа.

### **2.5.2 Виробники засобів виробництва (ВЗВ)**

Визначається початкова кількість грошей в умовних одиницях (задається оператором). Це може бути або конкретне число, або вказано можливий інтервал, з якого програма випадковим чином згідно нормальному закону розподілу сама має обрати якесь число (наприклад, 20-50). (мабуть другий варіант більш зручним). (наприклад, програма вибрала 45) Це означає, що цю кількість грошей треба витратити на закупівлю необхідних товарів для випуску сировини

Задається можливий період циклу технологічного процесу днями (наприклад, 1-30). Програма випадково вибирає тривалість технологічного циклу (наприклад, 20). Це означає, що через 20 днів буде виготовлено товар.

Визначається відсоток торгової націнки в деяких межах(наприклад, 20-100).

### **2.5.3 Виробники споживчих товарів (ВСТ)**

Визначається початкова кількість грошей в умовних одиницях (задається оператором). Це може бути або конкретне число, або вказано можливий інтервал, з якого програма випадковим чином згідно нормальному закону розподілу сама має обрати якесь число (наприклад, 20-50). (мабуть другий варіант більш зручним). (наприклад, програма обрала 30) Це означає, що цю кількість грошей треба витратити на закупівлю необхідних товарів для випуску сировини.

Задається можливий період реалізації товару днями (наприклад, 1-10). Програма випадково вибирає тривалість технологічного циклу (наприклад, 5). Це означає, що через 5 днів товар буде реалізовано.

Визначається відсоток торгової націнки в деяких межах (наприклад, 20-100)

#### **2.5.4 Торгівля та Послуги (ТП)**

Визначається початкова кількість грошей в умовних одиницях (задається оператором). Це може бути або конкретне число, або вказано можливий інтервал, з якого програма випадковим чином згідно нормальному закону розподілу сама має обрати якесь число (наприклад, 20-50). (мабуть другий варіант більш зручним). (наприклад, програма обрала 30) Це означає, що цю кількість грошей треба витратити на закупівлю необхідних товарів для випуску сировини.

Послуги надаються на ті суми, на які до них звертаються.

#### **2.5.5 Покупці споживчих товарів (ПСТ)**

Задаються відсотки витрати грошей на «Послуги» та «Торгівлю».

#### **2.5.6 Бюджет (Держава) (Б)**

Задаються відсотки витрати бюджетних грошей на «Послуги», «Торгівлю» та оплату бюджетників та соціально захищених верств населення.

Задається ймовірність настання Надзвичайної Ситуації (стихійне лихо, техногенна катастрофа та т.п.). Задається діапазон потреб у кредитах (емісії грошей).

#### **2.5.7 Комерційні банки (КБ)**

Задається кількість грошей в умовних одиницях.

#### **2.5.8 Нацбанк (НБ)**

Задається загальна кількість грошей в умовних одиницях.

### **2.6 Вихідні дані**

Вихідними даними програми є результати моделювання макроекономічної системи на основі базової ринкової моделі у вигляді

стовпчикових діаграм. Також вихідними даними є вікно подій, які траплялися під час моделювання та файл з результатами моделювання.

### **Висновки до розділу 2**

У даному розділі було визначено функціональне та експлуатаційне призначення програми. Були описані функціональні вимоги. Також були визначені вхідні та вихідні дані програми.

## 3 ВНУТРІШНЄ ПРОЕКТУВАННЯ

### 3.1 Об'єктно-орієнтоване програмування

Об'єктно-орієнтоване програмування (ООП) — це парадигма програмування, яка спирається на концепцію класів та об'єктів [9]. Вона використовується для структурування програми на прості фрагменти коду, які можна використовувати повторно (зазвичай їх називають класами), які використовуються для створення окремих екземплярів об'єктів. Існує багато об'єктно-орієнтованих мов програмування, наприклад C#, C++, Java та Python.

Клас — це абстрактний план, який використовується для створення більш конкретних об'єктів. Класи часто представляють широкі категорії, наприклад автомобіль чи собака, які мають спільні атрибути [10]. Ці класи визначають, які атрибути матиме екземпляр цього типу, наприклад колір, але не значення цих атрибутів для конкретного об'єкта.

Класи також можуть містити функції, які називаються методами, доступними лише для об'єктів цього типу. Ці функції визначаються всередині класу і виконують певну дію, корисну для цього конкретного типу об'єкта. Шаблони класів використовуються як план для створення окремих об'єктів. Вони представляють конкретні приклади абстрактного класу. Кожен об'єкт може мати унікальні значення властивостей, визначених у класі.

Переваги ООП:

- 1) ООП моделює складні речі як прості структури;
- 2) багаторазові об'єкти ООП можна використовувати в різних програмах;
- 3) дозволяє специфічну для класу поведінку через поліморфізм;
- 4) простіше налагоджувати, класи часто містять всю відповідну інформацію;
- 5) безпечність, захищає інформацію за допомогою інкапсуляції [11].

### 3.2 Опис відповідальності класів

Клас Program відповідає за запуск програми.

Клас Form1 відповідає за відображення інформації про моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

Клас Form2 відповідає за відображення інформації про загальні вхідні дані програми для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

Клас Form3 відповідає за відображення інформації про саму програму.

Клас Form4 відповідає за відображення подій, які відбуваються протягом процесу моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

Клас InputData відповідає за зберігання вхідних даних програми.

Клас Characteristics відповідає за зберігання характеристик макроекономічної системи держави на базі класичної ринкової моделі.

Клас Economy\_events відповідає за зберігання можливих подій під час моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

Клас Output\_data відповідає за зберігання вихідних даних після моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

Клас Form5 відповідає за відображення інформації про вхідні дані виробників сировини програми для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

Клас Form6 відповідає за відображення інформації про вхідні дані виробників засобів виробництва програми для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

Клас Form7 відповідає за відображення інформації про вхідні дані виробників споживчих товарів програми для моделювання механізму

функціонування макроекономічної системи держави на базі класичної ринкової моделі.

Клас Form8 відповідає за відображення інформації про вхідні дані торгівлі та послуг програми для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

Клас Form9 відповідає за відображення інформації про інші вхідні дані програми для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

Клас Form10 відповідає за відображення інформації про вихідні дані після моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

### **3.3 Опис зв'язків класів**

Описані вище класи мають наступні зв'язки.

Об'єкти сутності класу Program пов'язані з об'єктами класу Form1. Тип зв'язку – асоціація.

Об'єкти сутності класу Form1 пов'язані з об'єктами класів Form2, Form3, Form4, Form5, Form6, Form7, Form8, Form9, Form9, Characteristics. Тип зв'язку – асоціація.

Об'єкти сутності класу Form2, Form5, Form6, Form7, Form8, Form9 пов'язані з об'єктами класу InputData. Тип зв'язку – асоціація.

Об'єкти сутності класу Form4 пов'язані з об'єктами класу Economy\_events. Тип зв'язку – асоціація.

Об'єкти сутності класу Form4 пов'язані з об'єктами класу Output\_data. Тип зв'язку – асоціація.

### **3.4 Діаграма класів програми**

На основі вищеописаної інформації про об'єктно-орієнтовану модель програми була побудована діаграма класів. Діаграма класів представлена на рис. 3.1. Пояснення до цієї діаграми (специфікації класів) описані у наступному пункті розділу.

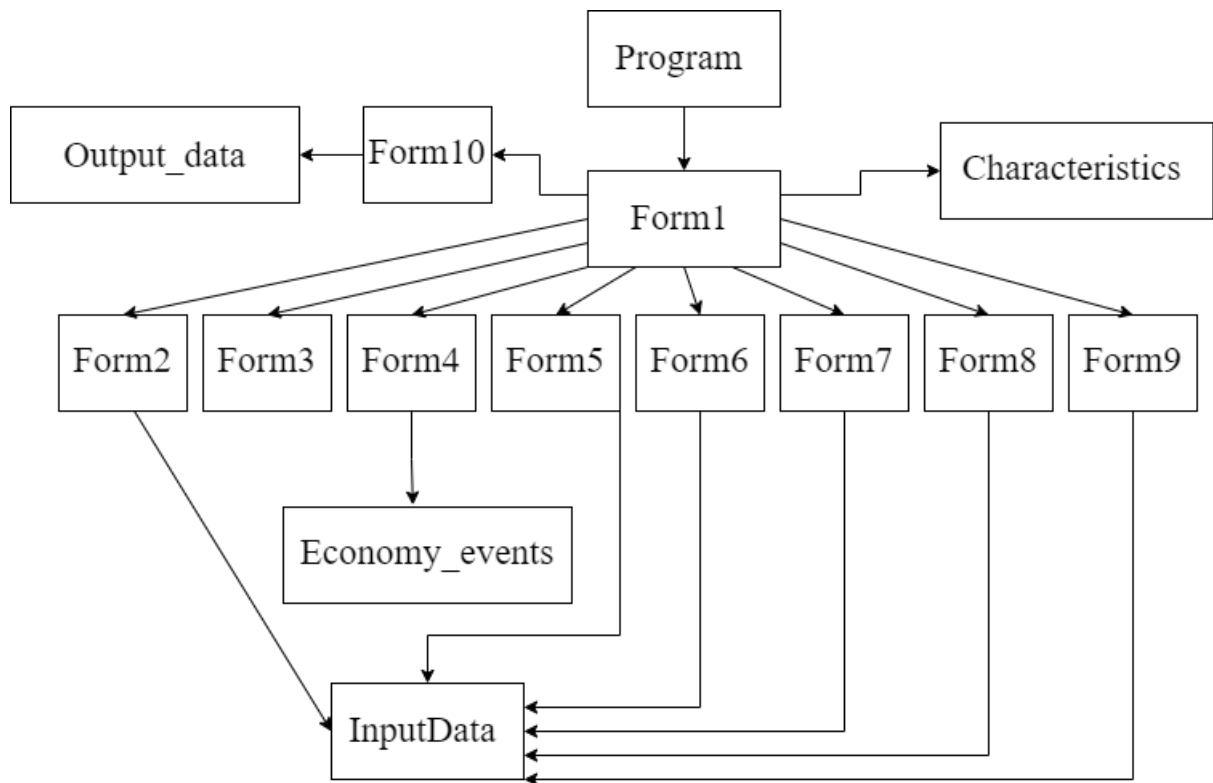


Рисунок 3.1 – Діаграма класів програми

### 3.5 Опис специфікацій класів

#### 3.5.1 Опис специфікацій класу Program

Клас реалізовує наступні операції – запуск програми.

#### 3.5.2 Опис специфікацій класу Form1

Клас реалізовує наступні операції:

- проведення розрахунків по моделюванню механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі та виведенні інформації на екран;
- моделюванню механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі у часі;
- перехід на форму введення загальних вхідних даних;
- перехід на форму введення вхідних даних виробників сировини;
- перехід на форму введення вхідних даних виробників засобів виробництва;
- перехід на форму введення вхідних даних виробників споживчих товарів;
- перехід на форму введення вхідних даних торгівлі та послуг;

- перехід на форму введення інших вхідних даних;
- перехід на форму перегляду вікна подій;
- перехід на форму перегляду детальної інформації про програму.

### **3.5.3 Опис специфікацій класу Form2**

Клас реалізовує наступні операції:

- збереження інформації про загальні вхідні дані, введені користувачем програми;
- перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

### **3.5.4 Опис специфікацій класу Form3**

Клас реалізовує наступні операції:

- відображення інформації про саму програму;
- перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

### **3.5.5 Опис специфікацій класу Form4**

Клас реалізовує наступні операції:

- відображення інформації про події протягом моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі;
- оновлення інформації про події протягом моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі;
- перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

### **3.5.6 Опис специфікацій класу Form5**

Клас реалізовує наступні операції:

- збереження інформації про вхідні дані виробників сировини, введені користувачем програми;
- перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

### **3.5.7 Опис специфікацій класу Form6**

Клас реалізовує наступні операції:

- збереження інформації про вхідні дані виробників засобів виробництва, введені користувачем програми;
- перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

### **3.5.8 Опис специфікацій класу Form7**

Клас реалізовує наступні операції:

- збереження інформації про вхідні дані виробників споживчих товарів, введені користувачем програми;
- перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

### **3.5.9 Опис специфікацій класу Form8**

Клас реалізовує наступні операції:

- збереження інформації про вхідні дані торгівлі та послуг, введені користувачем програми;
- перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

### **3.5.10 Опис специфікацій класу Form9**

Клас реалізовує наступні операції:

- збереження інформації про інші вхідні дані, введені користувачем програми;

- перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

### 3.5.11 Опис специфікацій класу InputData

Атрибути класу:

- змінна, що відповідає за швидкість покрокового моделювання;
- змінна, що відповідає за відсоток грошей на сплату податку;
- змінна, що відповідає за відсоток грошей на виплату заробітної плати;
- змінна, що відповідає за відсотки за кредитами та депозитами;
- змінна, що відповідає за ставку Національного банку;
- змінна, що відповідає за період часу повернення кредиту;
- змінна, що відповідає за інтервал періоду моделювання;
- змінна, що відповідає за первісну мінімальну кількість грошей виробників сировини;
- змінна, що відповідає за первісну максимальну кількість грошей виробників сировини;
- змінна, що відповідає за мінімальний період циклу технологічного процесу виробників сировини;
- змінна, що відповідає за максимальний період циклу технологічного процесу виробників сировини;
- змінна, що відповідає за мінімальний відсоток торгової націнки виробників сировини;
- змінна, що відповідає за максимальний відсоток торгової націнки виробників сировини;
- змінна, що відповідає за первісну мінімальну кількість грошей виробників засобів виробництва;
- змінна, що відповідає за первісну максимальну кількість грошей виробників засобів виробництва;

- змінна, що відповідає за мінімальний період циклу технологічного процесу виробників засобів виробництва;
- змінна, що відповідає за максимальний період циклу технологічного процесу виробників засобів виробництва;
- змінна, що відповідає за мінімальний відсоток торгової націнки виробників засобів виробництва;
- змінна, що відповідає за максимальний відсоток торгової націнки виробників засобів виробництва;
- змінна, що відповідає за первісну мінімальну кількість грошей виробників споживчих товарів;
- змінна, що відповідає за первісну максимальну кількість грошей виробників споживчих товарів;
- змінна, що відповідає за мінімальний період циклу технологічного процесу виробників споживчих товарів;
- змінна, що відповідає за максимальний період циклу технологічного процесу виробників споживчих товарів;
- змінна, що відповідає за мінімальний відсоток торгової націнки виробників споживчих товарів;
- змінна, що відповідає за максимальний відсоток торгової націнки виробників споживчих товарів;
- змінна, що відповідає за первісну мінімальну кількість грошей торгівлі та послуг;
- змінна, що відповідає за первісну максимальну кількість грошей торгівлі та послуг;
- змінна, що відповідає за відсоткове співвідношення купівлі товарів торгівлі та послуг;
- змінна, що відповідає за відсотки витрати грошей на «Послуги та торгівлю» ;
- змінна, що відповідає за відсотки витрачання бюджетних грошей на «Послуги та торгівлю» ;

- змінна, що відповідає за ймовірність настання Надзвичайної Ситуації;
- змінна, що відповідає за кількість грошей комерційного банку;
- змінна, що відповідає за загальну кількість грошей у Нацбанку.

### **3.5.12 Опис специфікацій класу Characteristics**

Атрибути класу:

- змінна, що відповідає за прихід виробників сировини;
- змінна, що відповідає за витрати виробників сировини;
- змінна, що відповідає за сировина виробників сировини;
- змінна, що відповідає за остача виробників сировини;
- змінна, що відповідає за прихід виробників засобів виробництва;
- змінна, що відповідає за витрати виробників засобів виробництва;
- змінна, що відповідає за сировина виробників засобів виробництва;
- змінна, що відповідає за остача виробників засобів виробництва;
- змінна, що відповідає за прихід виробників споживчих товарів;
- змінна, що відповідає за витрати виробників споживчих товарів;
- змінна, що відповідає за сировина виробників споживчих товарів;
- змінна, що відповідає за остача виробників споживчих товарів;
- змінна, що відповідає за прихід торгівлі та послуг;
- змінна, що відповідає за витрати торгівлі та послуг;
- змінна, що відповідає за сировина торгівлі та послуг;
- змінна, що відповідає за остача торгівлі та послуг;
- змінна, що відповідає за прихід бюджету;
- змінна, що відповідає за витрати бюджету;
- змінна, що відповідає за остача бюджету;
- змінна, що відповідає за прихід комерційних банків;
- змінна, що відповідає за витрати комерційних банків;
- змінна, що відповідає за остача комерційних банків;
- змінна, що відповідає за прихід покупців споживчих товарів;
- змінна, що відповідає за витрати покупців споживчих товарів;
- змінна, що відповідає за остача покупців споживчих товарів;

- змінна, що відповідає за запас Нацбанку;
- змінна, що відповідає за витрати Нацбанку;
- змінна, що відповідає за емісія Нацбанку;
- змінна, що відповідає за остача Нацбанку.

### 3.5.13 Опис специфікацій класу Economy\_events

Атрибути класу:

- змінна, що відповідає за лічильник днів;
- змінна, що відповідає за підсумковий рядок подій;
- змінні, що відповідають за можливі підприємства;
- змінні, що відповідають за можливі дії.

### 3.5.14 Опис специфікацій класу Form10

Клас реалізовує наступну операцію:

- відображення інформації про вихідні дані після моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі;

### 3.5.15 Опис специфікацій класу Output\_data

Атрибути класу:

- двомірний масив, що відповідає за збереження всієї інформації про суб'єкти економічної діяльності під час моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі;
- змінна, що відповідає за назву файлу для експорту вихідних даних.

## 3.6 Опис інтерфейсної частини класів

### 3.6.1 Клас Program

Члени класу описані у табл. 3.1.

Таблиця 3.1 – Члени класу Program

Назва	Опис	Вхід	Вихід
static void Main()	Функція, що запускає програму		

### 3.6.2 Клас Form1

Члени класу описані у табл. 3.2.

Таблиця 3.2 – Члени класу Form1

Назва	Опис	Вхід	Вихід
1	2	3	4
void Grafic_Update( )	Функція, що проводить розрахунки по моделюванню механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі та виводить інформацію на екран		Результати розрахунків на стовпчикових діаграмах
private void timer1_Tick(object sender, EventArgs e)	Функція, що проводить розрахунки по моделюванню механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі із плином часу	Інформація про те, що таймер активний	Результати розрахунків на стовпчикових діаграмах
private void зупинитиПокр оковеМоделюв анняToolStrip MenuItem_Click(object sender, EventArgs e)	Функція, що зупиняє роботу таймера	Інформація про те, що кнопка була натиснута	Зупинка роботи таймера

1	2	3	4
private void розпочатиПок роковеМодел юванняToolStrip MenuItem_Cli ck(object sender, EventArgs e)	Функція, що розпочинає роботу таймера	Інформаці я про те, що кнопка була натиснута	Початок роботи таймера
private void переглянутиТо olStripMenuIte m_  Click(object sender, EventArgs e)	Функція, що реалізовує перехід на форму для перегляду вікна подій під час моделювання  механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі	Інформаці я про те, що кнопка була натиснута	Перехід на форму для перегляду вікна подій
private void змоделювати1 КрокToolStrip MenuItem_Clic k(object sender, EventArgs e)	Функція, що проводить розрахунки по моделюванню механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі та виводить інформацію на екран	Інформаці я про те, що кнопка була натиснута	Результати розрахункі в на стовпчиков их діаграмах
private void змінитиВхідні ДаніToolStrip MenuItem_  Click(object sender, EventArgs e)	Функція, що реалізовує перехід на форму для введення загальних вхідних даних	Інформаці я про те, що кнопка була	Перехід на форму для введення загальних

1	2	3	4
Click(object sender, EventArgs e)		натиснута	вхідних даних
private void вихідToolStripMenuItem_Click(object sender, EventArgs e)	Функція, що реалізовує завершення роботи програми	Інформація про те, що кнопка була натиснута	Завершення роботи програми
private void Form1_Load(object sender, EventArgs e)	Функція, що реалізовує завантаження даних при переході на дану форму	Інформація про те, що користувач перейшов на дану форму	Завантаження даних при переході на дану форму
private void дузнатисяБільшеПроПрограмуToolStripMenuItem_Click(object sender, EventArgs e)	Функція, що реалізовує перехід на форму для перегляду детальнішої інформації про програму	Інформація про те, що кнопка була натиснута	Перехід на форму для перегляду детальнішої інформації про програму
private void виробникиСировиниToolStripMenuItem_	Функція, що реалізовує перехід на форму для введення вхідних даних виробників сировини	Інформація про те, що кнопка була	Перехід на форму для введення вхідних

1	2	3	4
Click(object sender, EventArgs e)		натиснута	даних виробників сировини
private void виробникиЗасобівВиробництваToolStripMenuItem_Click(object sender, EventArgs e)	Функція, що реалізовує перехід на форму для введення вхідних даних виробників засобів виробництва	Інформація про те, що кнопка була натиснута	Перехід на форму для введення вхідних даних виробників засобів виробництва
private void виробникиСпоживчихТоварівToolStripMenuItem_Click(object sender, EventArgs e)	Функція, що реалізовує перехід на форму для введення вхідних даних виробників споживчих товарів	Інформація про те, що кнопка була натиснута	Перехід на форму для введення вхідних даних виробників споживчих товарів
private void торгівляТаПослугиToolStripMenuItem_Click(object sender, EventArgs e)	Функція, що реалізовує перехід на форму для введення вхідних даних торгівлі та послуг	Інформація про те, що кнопка була натиснута	Перехід на форму для введення вхідних даних торгівлі та послуг

1	2	3	4
private void іншіToolStrip MenuItem_Click(object sender, EventArgs e)	Функція, що реалізовує перехід на форму для введення інших вхідних даних	Інформація про те, що кнопка була натиснута	Перехід на форму для введення інших вхідних даних

### 3.6.3 Клас Form2

Члени класу описані у табл. 3.3.

Таблиця 3.3 – Члени класу Form2

Назва	Опис	Вхід	Вихід
private void button1_Click (object sender, EventArgs e)	Функція, що зберігає вхідні дані, введені користувачем та реалізовує перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі	Інформація про те, що кнопка була натиснута	Перехід на форму для моделювання механізму функціонуван ня макроекономіч ної системи держави на базі класичної ринкової моделі
private void Form2_Load( object sender, EventArgs e)	Функція, що реалізовує завантаження даних при переході на дану форму	Інформація про те, що користувач перейшов на дану форму	Завантаження даних при переході на дану форму

### 3.6.4 Клас Form3

Члени класу описані у табл. 3.4.

Таблиця 3.4 – Члени класу Form3

Назва	Опис	Вхід	Вихід
private void Form3_Load(object sender, EventArgs e)	Функція, що реалізовує завантаження даних при переході на дану форму	Інформація про те, що користувач перейшов на дану форму	Завантаження даних при переході на дану форму

### 3.6.5 Клас Form4

Члени класу описані у табл. 3.5.

Таблиця 3.5 – Члени класу Form4

Назва	Опис	Вхід	Вихід
private void button1_Click(object sender, EventArgs e)	Функція, що оновлює вміст поля, для виведення інформації про події під час моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі	Інформація про те, що кнопка була натиснута	Оновлення вмісту поля
private void Form2_Load(object sender, EventArgs e)	Функція, що реалізовує завантаження даних при переході на дану форму	Інформація про те, що користувач перейшов на дану форму	Завантаження даних при переході на дану форму

### 3.6.6 Клас Form5

Члени класу описані у табл. 3.6.

Таблиця 3.6 – Члени класу Form5

Назва	Опис	Вхід	Вихід
private void button1_Click(object sender, EventArgs e)	Функція, що зберігає вхідні дані, введені користувачем та реалізовує перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі	Інформація про те, що кнопка була натиснута	Перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі
private void Form5_Load(object sender, EventArgs e)	Функція, що реалізовує завантаження даних при переході на дану форму	Інформація про те, що користувач перейшов на дану форму	Завантаження даних при переході на дану форму

### 3.6.7 Клас Form6

Члени класу описані у табл. 3.7.

Таблиця 3.7 – Члени класу Form6

Назва	Опис	Вхід	Вихід
1	2	3	4
private void button1_Click(object sender, EventArgs e)	Функція, що зберігає вхідні дані, введені користувачем та реалізовує перехід на форму для моделювання механізму функціонування макроекономічної системи	Інформація про те, що кнопка була натиснута	Перехід на форму для моделювання механізму функціонування

1	2	3	4
	держави на базі класичної ринкової моделі		макроекономічній системі держави на базі класичної ринкової моделі
private void Form6_Load(object sender, EventArgs e)	Функція, що реалізовує завантаження даних при переході на дану форму	Інформація про те, що користувач перейшов на дану форму	Завантаження даних при переході на дану форму

### 3.6.8 Клас Form7

Члени класу описані у табл. 3.8.

Таблиця 3.8 – Члени класу Form7

Назва	Опис	Вхід	Вихід
1	2	3	4
private void button1_Click(object sender, EventArgs e)	Функція, що зберігає вхідні дані, введені користувачем та реалізовує перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі	Інформація про те, що кнопка була натиснута	Перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі

1	2	3	4
private void Form6_Load( object sender, EventArgs e)	Функція, що реалізовує завантаження даних при переході на дану форму	Інформація про те, що користувач перейшов на дану форму	Завантаження даних при переході на дану форму

### 3.6.9 Клас Form8

Члени класу описані у табл. 3.9.

Таблиця 3.9 – Члени класу Form8

Назва	Опис	Вхід	Вихід
private void button1_Click(object sender, EventArgs e)	Функція, що зберігає вхідні дані, введені користувачем та реалізовує перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі	Інформація про те, що кнопка була натиснута	Перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі
private void Form6_Load( object sender, EventArgs e)	Функція, що реалізовує завантаження даних при переході на дану форму	Інформація про те, що користувач перейшов на дану форму	Завантаження даних при переході на дану форму

### 3.6.10 Клас Form9

Члени класу описані у табл. 3.10.

Таблиця 3.10 – Члени класу Form9

Назва	Опис	Вхід	Вихід
private void button1_Click(object sender, EventArgs e)	Функція, що зберігає вхідні дані, введені користувачем та реалізовує перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі	Інформація про те, що кнопка була натиснута	Перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі
private void Form6_Load(object sender, EventArgs e)	Функція, що реалізовує завантаження даних при переході на дану форму	Інформація про те, що користувач перейшов на дану форму	Завантаження даних при переході на дану форму

### 3.6.10 Клас Form10

Члени класу описані у табл. 3.11.

Таблиця 3.11 – Члени класу Form10

Назва	Опис	Вхід	Вихід
private void Form10_Load(object sender, EventArgs e)	Функція, що реалізовує завантаження даних при переході на дану форму	Інформація про те, що користувач перейшов на дану форму	Завантаження даних при переході на дану форму

### 3.7 Проектування інтерфейсу користувача

Інтерфейс – це набір інструментів взаємодії людини і комп'ютерної системи. Цими інструментами найчастіше є кнопки, галочки, текстові поля, підказки, перемикачі, списки, що випадають, та інші елементи [12].

Принципи проектування інтерфейсів користувача базуються на людських можливостях. Наприклад:

- інтерфейс має бути інтуїтивно зрозумілим. Таким, щоб користувачу не потрібно було пояснювати, як ним користуватися;
- для спрощення процесу вивчення потрібна довідка. Буквально - графічна підказка, яка пояснює значення того чи іншого елементу інтерфейсу. Повний посібник має бути частиною інтерфейсу, доступної будь-якої миті;
- найчастіше, користувачі в інтерфейсі спочатку шукають сутність (іменник), а потім дію (дієслово) до неї;
- чим швидше людина побачить результат – тим краще;
- чим більше користувач працює з якимось конкретним завданням, тим більше він на ньому концентрується і тим менше перестає помічати підказки та повідомлення, що виводяться програмою [13]. Чим більш критичним є завдання, тим менша ймовірність того, що користувач помітить попередження щодо тих чи інших потенційно небезпечних дій.

До візуальних атрибутів відображення інформації відносяться:

- палітра кольорів;
- взаємне розташування і розміри об'єктів, що відображаються;
- засоби залучення уваги користувача.

Колір завжди був важливим інструментом для художників, фотографів та дизайнерів, тому що він здатний миттєво викликати певні емоції та асоціації. У дизайні інтерфейсів колір теж відіграє важливу роль. Тому при виборі кольорів інтерфейсу були використані наступні принципи:

- певні кольори пов'язані з певними емоціями, але завжди. Відразу після народження ми починаємо знайомитися з різними відтінками, і згодом пам'ять пов'язує їх із певними переживаннями та об'єктами. Саме тому кольори можуть дати користувачеві певне уявлення про ціль інтерфейсу;
- гармонуючі кольори роблять елементи дизайну логічними. Існує багато способів створення колірних гармоній. Один з них – використовувати відтінки, розташовані поруч один з одним у колірному колі;
- контрастні кольори роблять інтерфейс динамічним. Це інша форма гармонії, яка досягається за рахунок використання кольорів, що знаходяться один навпроти одного в колірному колі;
- застосування однієї гами кольорів знижує когнітивне навантаження на користувача. Ієрархію контенту можна створити за допомогою близьких відтінків. Вони дозволять поділити елементи, не перевантажуючи композицію.

Таким чином, непродумане використання кольору може викликати у користувача негативний емоційний стан або навіть відволікати його від роботи, заважаючи зосередитися на виконуваному завданні.

### **3.8 Аналіз проекту**

Аналізуючи розроблений проект, можна зробити висновок, що усі класи мають функціональну зв'язність.

### **3.9 Вибір мови програмування**

Програма була написана на мові C#.

C# – це об'єктно-орієнтована, проста і водночас потужна мова програмування, яка дозволяє розробникам створювати багатofункціональні програми.

C# ставитися до мов компілюваного типу, тому він має всі переваги таких мов. Він поєднує найкращі ідеї сучасних мов програмування Java, C++, Visual Basic тощо. Через велику різноманітність синтаксичних конструкцій та

можливості працювати з платформою .Net, C# дозволяє швидше, ніж будь-яку іншу мову, розробляти програмні рішення. Також він відрізняється надійністю. Великої надійності було досягнуто через роботу CLR машини, адже на відміну від інших компілятор CLR запускає розроблений додаток на віртуальному процесорі. Тому у разі виникнення будь-яких помилок це ніяк не вплине на роботу інших програм у системі, але це також означає, що для запуску програми потрібен додатковий час. Відповідно програми написані мовою програмування C# надійніші, але менш швидкі (ніж самі програми написані на C++).

### **Висновки до розділу 3**

У даному розділі була описана відповідальність класів програми та зв'язки між цими класами. Також була побудована діаграма класів та визначена їх специфікація та інтерфейсна частина.

## 4 ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ ПРОГРАМИ

### 4.1 Стратегія тестування

Стратегія тестування — це план проведення робіт із тестування системи або її модуля, що враховує специфіку функціональності та залежності з іншими компонентами системи та платформи. Стратегія визначає типи тестів, які потрібно виконувати для даного функціоналу системи, включає опис необхідних підходів з точки зору цілей тестування та може задавати описи або вимоги до необхідних для проведення тестування інструментів та інфраструктури.

Стратегія відповідає на запитання:

- 1) як тестування дасть відповідь, що цей функціонал працює?
- 2) що потрібно зробити і чим користуватися з інструментальних засобів для досягнення цілей тестування?
- 3) коли певний функціонал тестуватиметься і коли очікувати отримання результатів?
- 4) як додаткове завдання, яке вирішується в процесі розуміння стратегії тестування, можна розглядати задачу мінімізації витрат на тестування.

Існує багато способів тестування програмного забезпечення. Для даної програми були обрані методи тестування «чорною скринькою» та «білою скринькою».

Тестування «чорною скринькою» – це функціональне та нефункціональне тестування без доступу до внутрішньої структури компонентів системи. Метод тестування «чорної скриньки» – процедура отримання та вибору тестових випадків на основі аналізу специфікації (функціональної або нефункціональної), компонентів або системи без посилання на їх внутрішній пристрій.

Переваги:

- 1) тестування методом «чорної скриньки» дозволяє знайти помилки, які неможливо виявити методом «білої скриньки». Найпростіший приклад:

розробник забув додати якусь функціональність. З погляду коду все працює ідеально, але з погляду специфікації це надкритичний баг;

2) «Чорна скринька» дозволяє швидко виявити помилки у функціональних специфікаціях (у них описані не лише вхідні значення, а й те, що ми маємо отримати). Якщо отриманий під час тестування результат відрізняється від заявленого специфікації, ми маємо привід спілкування з аналітиком уточнення кінцевого результату;

3) тестувальнику не потрібна додаткова кваліфікація. Часто ми користуємося різними сервісами та програмами, не дуже в них розбираючись. Для того, щоб відкрити інстаграм і обробити свою фотографію, нам не потрібно знати спосіб реалізації фільтрів. Ми хочемо відкрити фотографію, вибрати фільтр та отримати гарну картинку на виході. Завдання тестувальника, який тестує цю функцію в інстаграм, – переконатися, що користувач отримає цю найкрасивішу картинку відповідно до вибраного фільтра. При цьому нам зовсім не обов'язково мати якусь спеціалізацію – потрібні лише телефон та інстаграм;

4) тестування відбувається «з позиції користувача». Користувач завжди має рацію, він кінцевий споживач практично будь-якого ПЗ, а значить, йому має бути зручно, комфортно і зрозуміло;

5) складати тест-кейси можна одразу після підготовки специфікації. Це значно скорочує час на тестування: до того моменту, як продукт готовий до тестування, тест-кейси вже розроблені і тестувальник може відразу приступати до перевірки.

Недоліки:

1) основним недоліком методу «чорної скриньки» є можливість пропуску кордонів та переходів, які не очевидні зі специфікації, але є в реалізації коду (власне, це і змушує тестувальників використовувати метод «білої скриньки»). Згадується випадок, коли система отримувала котирування валют із біржі Forex та округляла до 3 знаків після коми. Система успішно пройшла тестування методом «чорної скриньки» (оскільки

жодна валюта не виходила за відповідні кордони) і добре працювала доти, поки курс долара до біткоїну не вийшов за межі 1000 доларів. Тестування «білою скринькою» виявило б помилку: фахівець побачив би, що коефіцієнт конверсії валюти обмежений 3 знаками;

2) можна протестувати лише невелику кількість можливих вступних (вхідних) значень; багато варіантів залишаються без перевірки;

3) тести можуть бути надмірними, якщо розробник вже перевіряв цю функціональність (наприклад, Unit-тестом);

4) за відсутності чіткої та повної специфікації проектувати тести та тест-сценарії виявляється важко.

Тестування «білою скринькою» – це тестування внутрішньої структури, дизайну та кодування програмного рішення. У цьому типі тестування код видно тестеру. Основна увага приділяється перевірці потоку вхідних та вихідних даних через додаток, покращення дизайну та зручності використання, посилення безпеки. Тестування білої скриньки також відоме як тестування Clear Box, тестування Open Box, структурне тестування, тестування прозорого боксу, тестування на основі коду та тестування Glass Box. Це зазвичай виконується розробниками.

Переваги:

1) оптимізація коду шляхом пошуку прихованих помилок;

2) тестові випадки «білої скриньки» можуть легко автоматизуватися;

3) тестування є ретельнішим, оскільки зазвичай покриваються всі шляхи коду;

4) тестування може розпочатися рано в SDLC, навіть якщо GUI недоступний.

Недоліки:

1) тестування білої скриньки може бути досить складним та дорогим;

2) розробники, які зазвичай виконують тести «білої скриньки», ненавидять це. Тестування білої скриньки розробниками не може детально призвести до виробничих помилок;

3) тестування білої скриньки потребує професійних ресурсів із докладним розумінням програмування та реалізації;

4) тестування білої скриньки забирає багато часу, більші програми для програмування вимагають часу для повного тестування.

## **4.2 Специфікація та текст модулів**

### **4.2.1 Опис специфікацій та текст функції `Grafic_Update`**

Атрибути функції:

- змінна, що відповідає за відсоток грошей на сплату податку;
- змінна, що відповідає за відсоток грошей на виплату заробітної плати;
- змінна, що відповідає за відсотки за кредитами та депозитами;
- змінна, що відповідає за ставку Національного банку;
- змінна, що відповідає за період часу повернення кредиту;
- змінна, що відповідає за інтервал періоду моделювання;
- первісна мінімальна кількість грошей виробників сировини;
- змінна, що відповідає за первісну максимальну кількість грошей виробників сировини;
- змінна, що відповідає за мінімальний період циклу технологічного процесу виробників сировини;
- змінна, що відповідає за максимальний період циклу технологічного процесу виробників сировини;
- мінімальний відсоток торгової націнки виробників сировини;
- максимальний відсоток торгової націнки виробників сировини;
- змінна, що відповідає за первісну мінімальну кількість грошей виробників засобів виробництва;
- змінна, що відповідає за первісну максимальну кількість грошей виробників засобів виробництва;
- змінна, що відповідає за мінімальний період циклу технологічного процесу виробників засобів виробництва;
- змінна, що відповідає за максимальний період циклу технологічного процесу виробників засобів виробництва;

- змінна, що відповідає за мінімальний відсоток торгової націнки виробників засобів виробництва;
- змінна, що відповідає за максимальний відсоток торгової націнки виробників засобів виробництва;
- змінна, що відповідає за первісну мінімальну кількість грошей виробників споживчих товарів;
- змінна, що відповідає за первісна максимальну кількість грошей виробників споживчих товарів;
- змінна, що відповідає за мінімальний період циклу технологічного процесу виробників споживчих товарів;
- змінна, що відповідає за максимальний період циклу технологічного процесу виробників споживчих товарів;
- змінна, що відповідає за мінімальний відсоток торгової націнки виробників споживчих товарів;
- змінна, що відповідає за максимальний відсоток торгової націнки виробників споживчих товарів;
- первісна мінімальна кількість грошей торгівлі та послуг;
- первісна максимальна кількість грошей торгівлі та послуг;
- відсоткове співвідношення купівлі товарів торгівлі та послуг;
- відсотки витрати грошей на «Послуги та торгівлю»;
- відсотки витрачання бюджетних грошей на «Послуги та торгівлю»;
- змінна, що відповідає за ймовірність настання Надзвичайної Ситуації;
- змінна, що відповідає за кількість грошей комерційного банку;
- змінна, що відповідає за загальну кількість грошей у Нацбанку.
- змінна, що відповідає за прихід виробників сировини;
- змінна, що відповідає за витрати виробників сировини;
- змінна, що відповідає за сировина виробників сировини;
- змінна, що відповідає за остача виробників сировини;
- змінна, що відповідає за прихід виробників засобів виробництва;
- змінна, що відповідає за витрати виробників засобів виробництва;

- змінна, що відповідає за сировина виробників засобів виробництва;
- змінна, що відповідає за остача виробників засобів виробництва;
- змінна, що відповідає за прихід виробників споживчих товарів;
- змінна, що відповідає за витрати виробників споживчих товарів;
- змінна, що відповідає за сировина виробників споживчих товарів;
- змінна, що відповідає за остача виробників споживчих товарів;
- змінна, що відповідає за прихід торгівлі та послуг;
- змінна, що відповідає за витрати торгівлі та послуг;
- змінна, що відповідає за сировина торгівлі та послуг;
- змінна, що відповідає за остача торгівлі та послуг;
- змінна, що відповідає за прихід бюджету;
- змінна, що відповідає за витрати бюджету;
- змінна, що відповідає за остача бюджету;
- змінна, що відповідає за прихід комерційних банків;
- змінна, що відповідає за витрати комерційних банків;
- змінна, що відповідає за остача комерційних банків;
- змінна, що відповідає за прихід покупців споживчих товарів;
- змінна, що відповідає за витрати покупців споживчих товарів;
- змінна, що відповідає за остача покупців споживчих товарів;
- змінна, що відповідає за запас Нацбанку;
- змінна, що відповідає за витрати Нацбанку;
- змінна, що відповідає за емісія Нацбанку;
- змінна, що відповідає за остача Нацбанку;
- змінна, що відповідає за лічильник днів;
- змінна, що відповідає за підсумковий рядок подій;
- змінні, що відповідає за можливі підприємства;
- змінні, що відповідає за можливі дії.

Функція проводить розрахунки по моделюванню механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі та виводить інформацію на екран.

### 4.3 Тестування методом «білої скриньки»

Блок-схема функції `Grafic_Update` зображена на рис. 4.1.

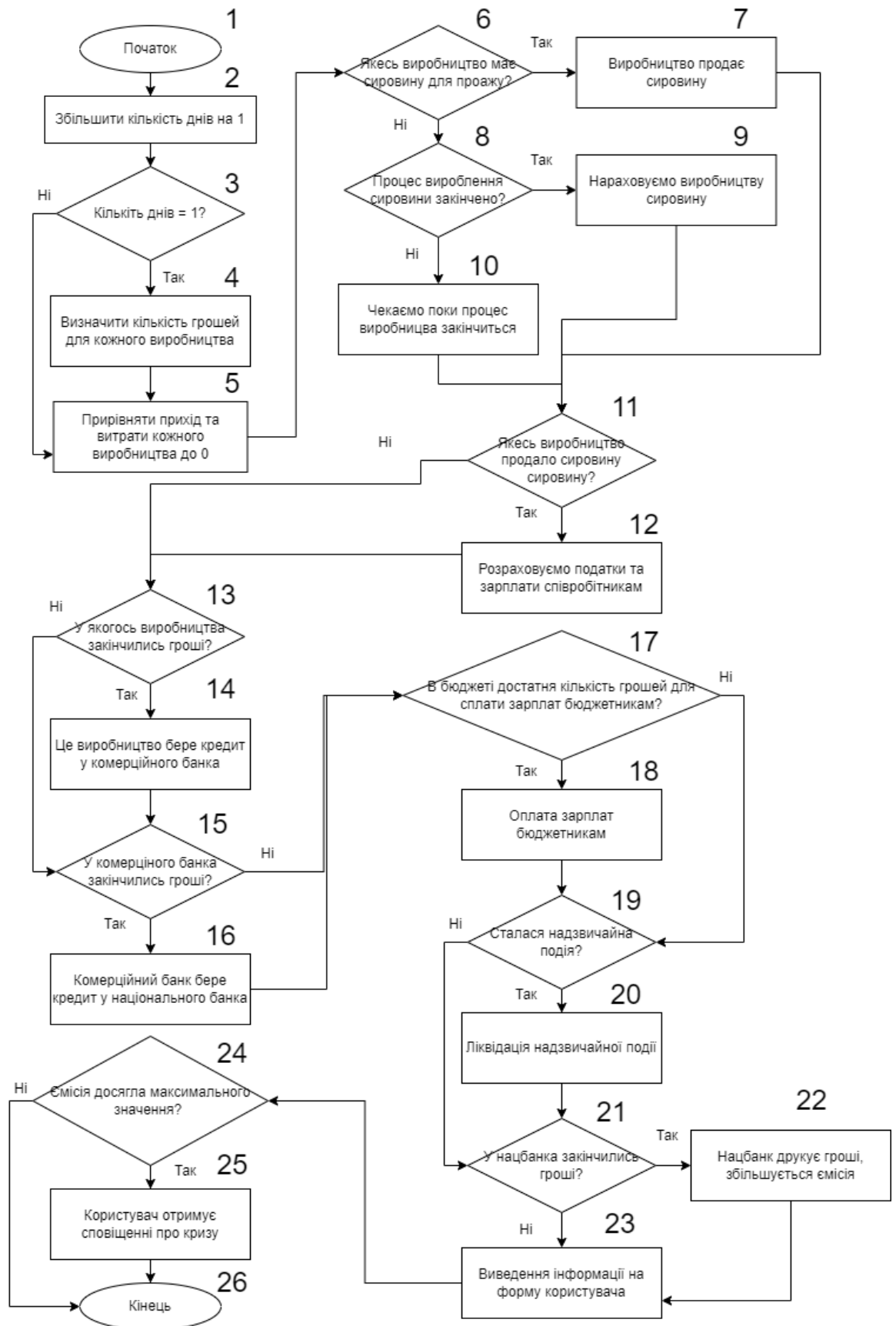


Рисунок 4.1 – Блок-схема функції `Grafic_Update`

Тести функції Grafic\_Update описані у табл. 4.1.

Вхідні та вихідні дані представлені у такому порядку:

1. Кількість днів;
2. Остача кожного виробництва (виробника сировини, виробника засобів виробництва, виробника споживчих товарів, торгівлі та послуг відповідно);
3. Кількість сировини кожного виробництва (виробника сировини, виробника засобів виробництва, виробника споживчих товарів, торгівлі та послуг відповідно);
4. Кількість днів до кінця виготовлення сировини кожного виробництва (виробника сировини, виробника засобів виробництва, виробника споживчих товарів, торгівлі та послуг відповідно);
5. Остача комерційного банку;
6. Остача бюджету;
7. Остача нацбанку;
8. Емісія нацбанку.

Таблиця 4.1 – Тести функції Grafic\_Update

Номер тесту	Вхідні значення	Вихідні значення
1	2	3
1	1. 0 2. 0 0 0 0 3. 0 0 0 0 4. 3 5 7 1 5. 0 6. 0 7. 0 8. 0	1. 1 2. 628 546 894 386 3. 0 0 0 0 4. 2 4 6 0 5. 800 6. 0 7. 1000 8. 0
2	1. 1 2. 628 546 894 386 3. 0 0 0 0	1. 2 2. 628 546 894 353 3. 0 0 0 394

1	2	3
	4. 2 4 6 0 5. 800 6. 0 7. 1000 8. 0	4. 1 3 5 1 5. 800 6. 0 7. 1000 8. 0
3	1. 2 2. 628 546 894 353 3. 0 0 0 394 4. 1 3 5 1 5. 800 6. 0 7. 1000 8. 0	1. 3 2. 617 535 883 379 3. 0 0 952 0 4. 0 2 4 0 5. 800 6. 16 7. 989 8. 0
4	1. 1050 2. 597 228 453 58 3. 0 0 0 0 4. 3 5 7 0 5. 16 6. 329 7. 57 8. 0	1. 1051 2. 597 228 453 384 3. 0 0 0 405 4. 2 4 6 1 5. 298 6. 24 7. 106 8. 100
5	1. 2021 2. 17 227 243 307 3. 0 0 0 354 4. 3 5 7 1 5. 25 6. 175 7. 18 8. 900	1. 2022 2. 189 216 232 296 3. 0 0 0 0 4. 2 4 6 0 5. 153 6. 184 7. 166 8. 1000

Тестування методом покриття операторів описано у табл. 4.2.

Таблиця 4.2 – Тестування методом покриття операторів

Номер тесту	Номер оператора																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	+	+	+	+	+	+	-	+	-	+	+	-	+	-	+	-	+	-	+	-	+	-	+	+	-	+
2	+	+	+	-	+	+	-	+	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	+	-	+
3	+	+	+	-	+	+	+	-	-	+	+	+	+	-	+	-	+	-	+	-	+	-	+	+	-	+
4	+	+	+	-	+	+	-	+	+	-	+	-	+	+	+	+	+	+	+	+	+	+	+	+	-	+
5	+	+	+	-	+	+	+	-	-	+	+	+	+	-	+	-	+	-	+	-	+	+	+	+	+	+

При виконанні тестів, кожен оператор виконався хоча б один раз.

Тестування методом покриття рішень описано у табл. 4.3.

Таблиця 4.3 – Тестування методом покриття рішень

Номер тесту	Номер оператора									
	3	6	8	11	13	15	17	19	21	24
1	+	-	-	-	-	-	-	-	-	-
2	-	-	+	-	-	-	-	-	-	-
3	-	+	-	+	-	-	-	-	-	-
4	-	-	+	-	+	+	+	+	+	-
5	-	+	-	+	+	+	-	-	+	+

При виконанні тестів, кожен напрямок переходу був реалізований хоча б раз.

## 4.4 Тестування методом «чорної скриньки»

1. Якщо виробництво виробило сировину, то воно має продати його іншим виробництвам та отримати гроші. Програма до виконання функції зображена на рис. 4.2, а після виконання функції на рис 4.3.

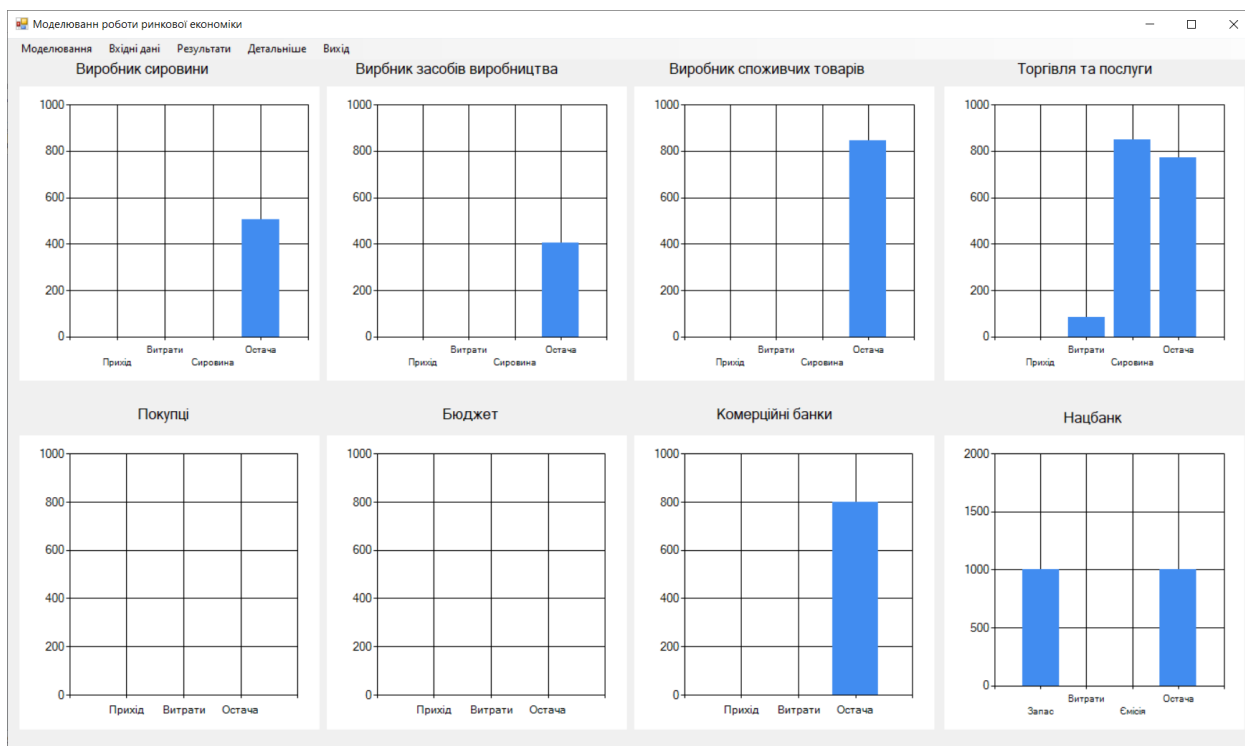


Рисунок 4.2 – Програма до виконання функції

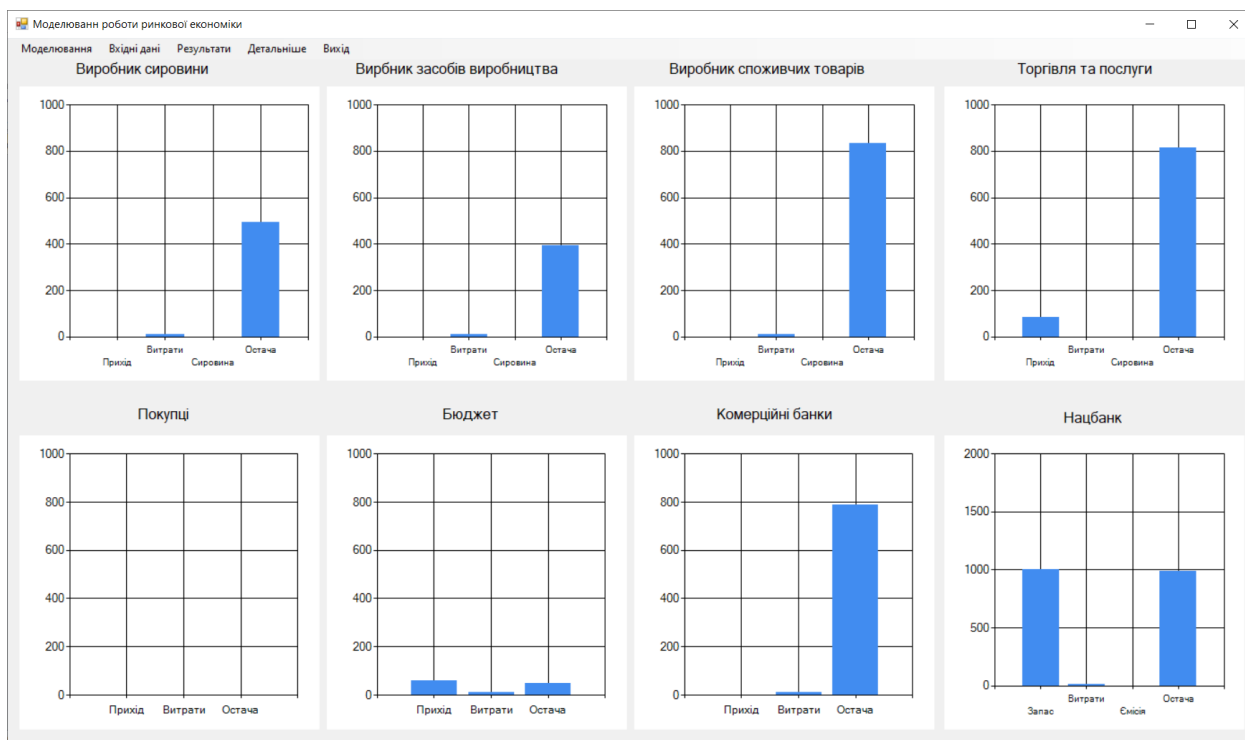


Рисунок 4.3 – Програма після виконання функції

2. Якщо у виробництва закінчились гроші, то воно має взяти кредит у комерційного банку. Програма до виконання функції зображена на рис. 4.4, а після виконання функції на рис 4.5.

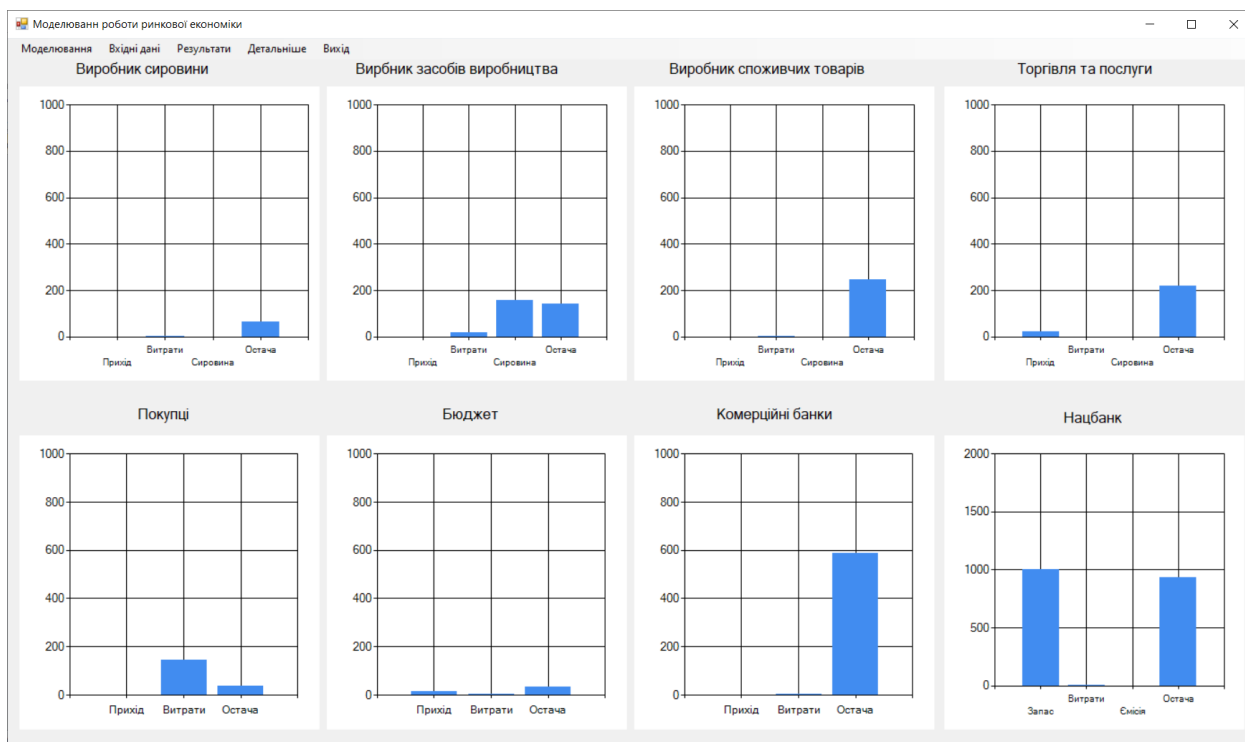


Рисунок 4.4 – Програма до виконання функції

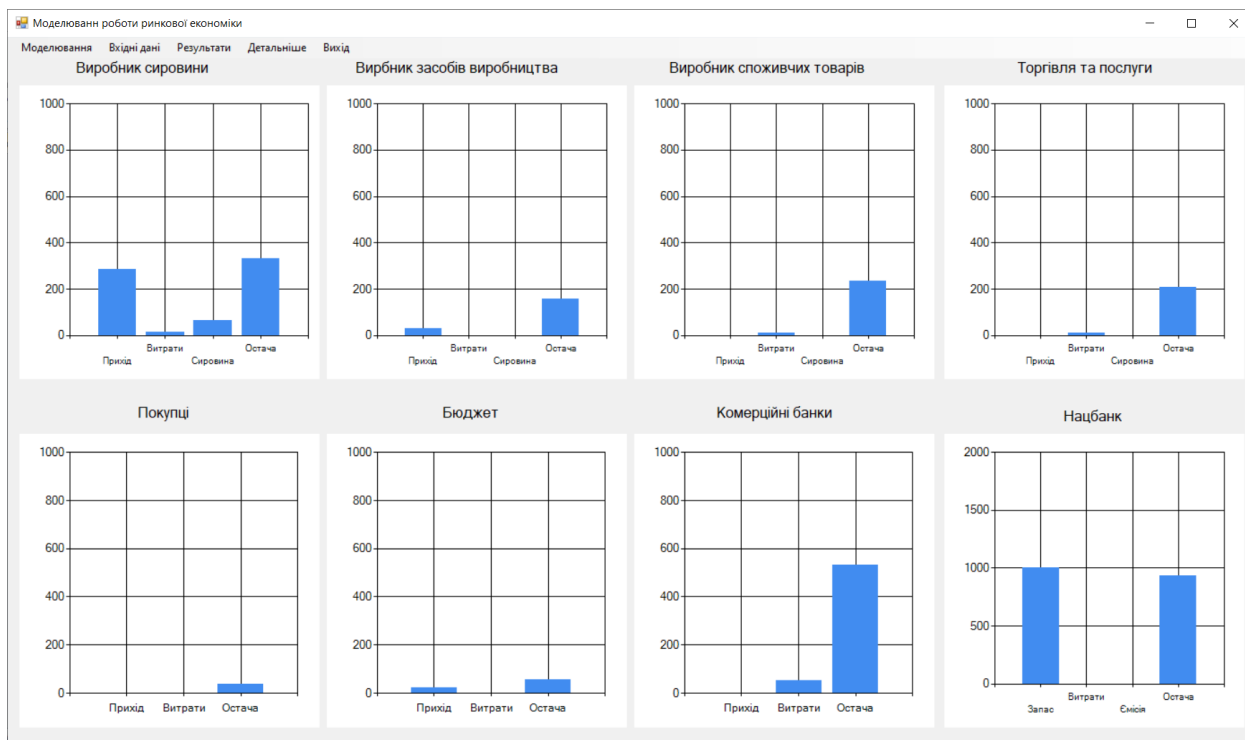


Рисунок 4.5 – Програма після виконання функції

3. Якщо у комерційного банку закінчились гроші, то він має взяти кредит у Національного банку. Програма до виконання функції зображена на рис. 4.6, а після виконання функції на рис 4.7.

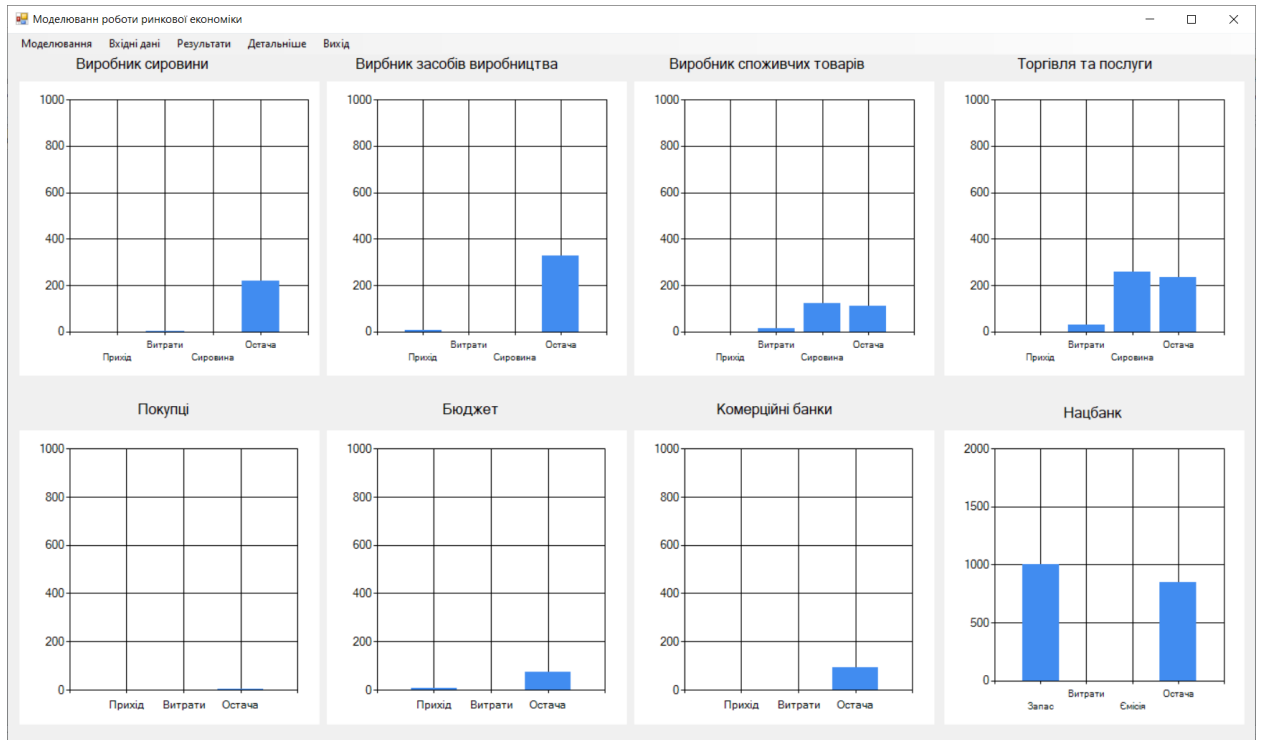


Рисунок 4.6 – Програма до виконання функції

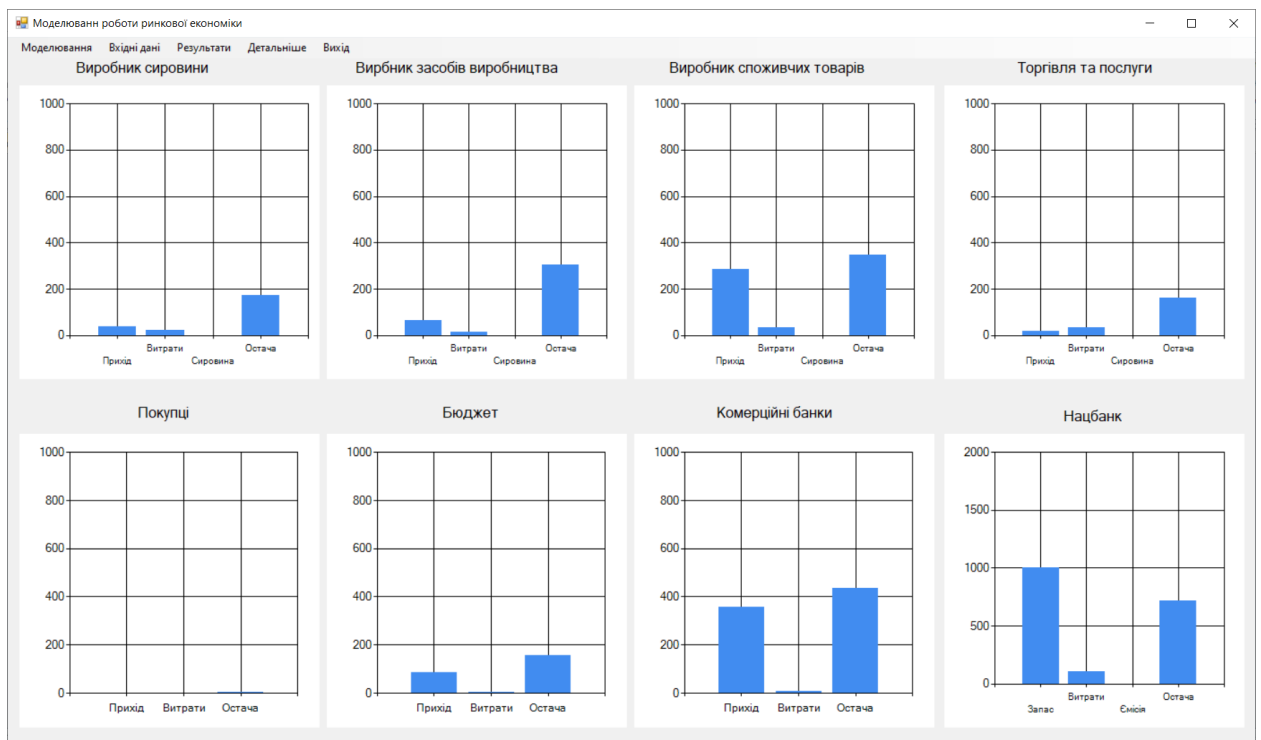


Рисунок 4.7 – Програма після виконання функції

4. Якщо у Національного банку закінчились гроші, то він має надрукувати гроші та збільшити емісію. Програма до виконання функції зображена на рис. 4.8, а після виконання функції на рис 4.9.

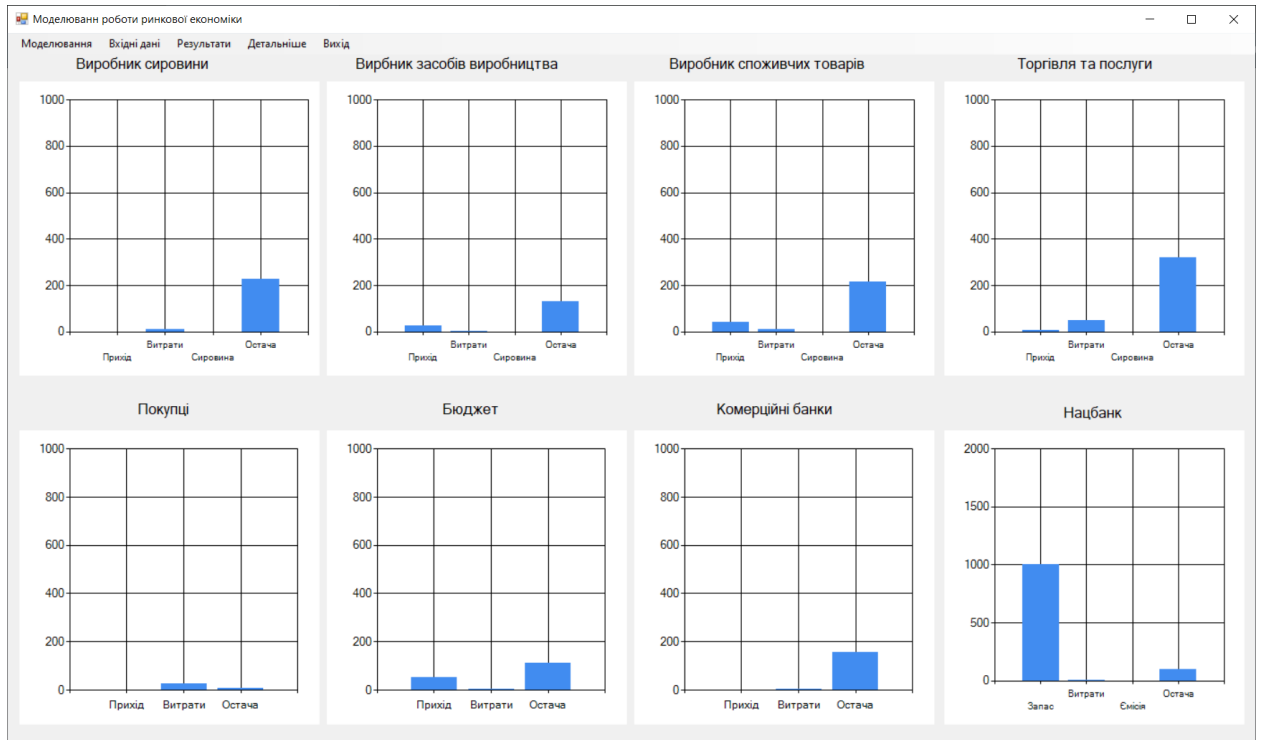


Рисунок 4.8 – Програма до виконання функції

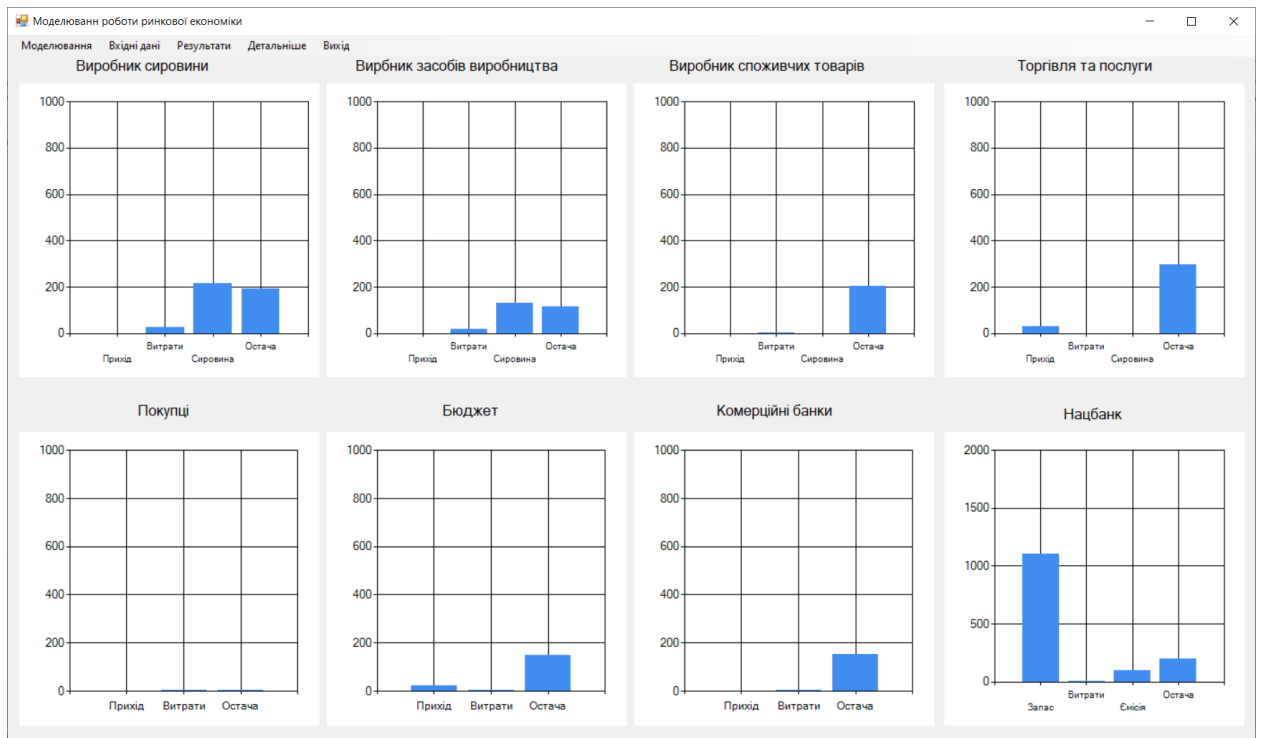


Рисунок 4.9 – Програма після виконання функції

5. Якщо емісія досягла максимального показника, то у країні настає криза. Програма до виконання функції зображена на рис. 4.10, а після виконання функції на рис 4.11.

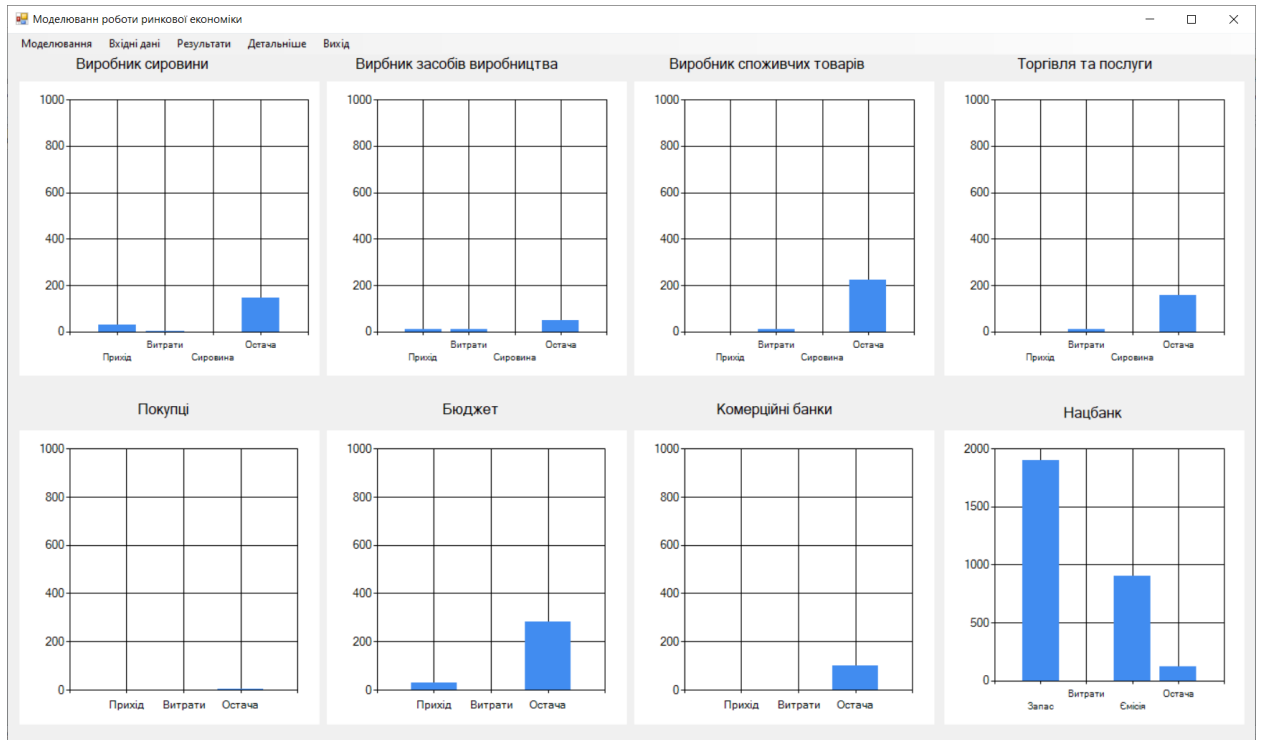


Рисунок 4.10 – Програма до виконання функції

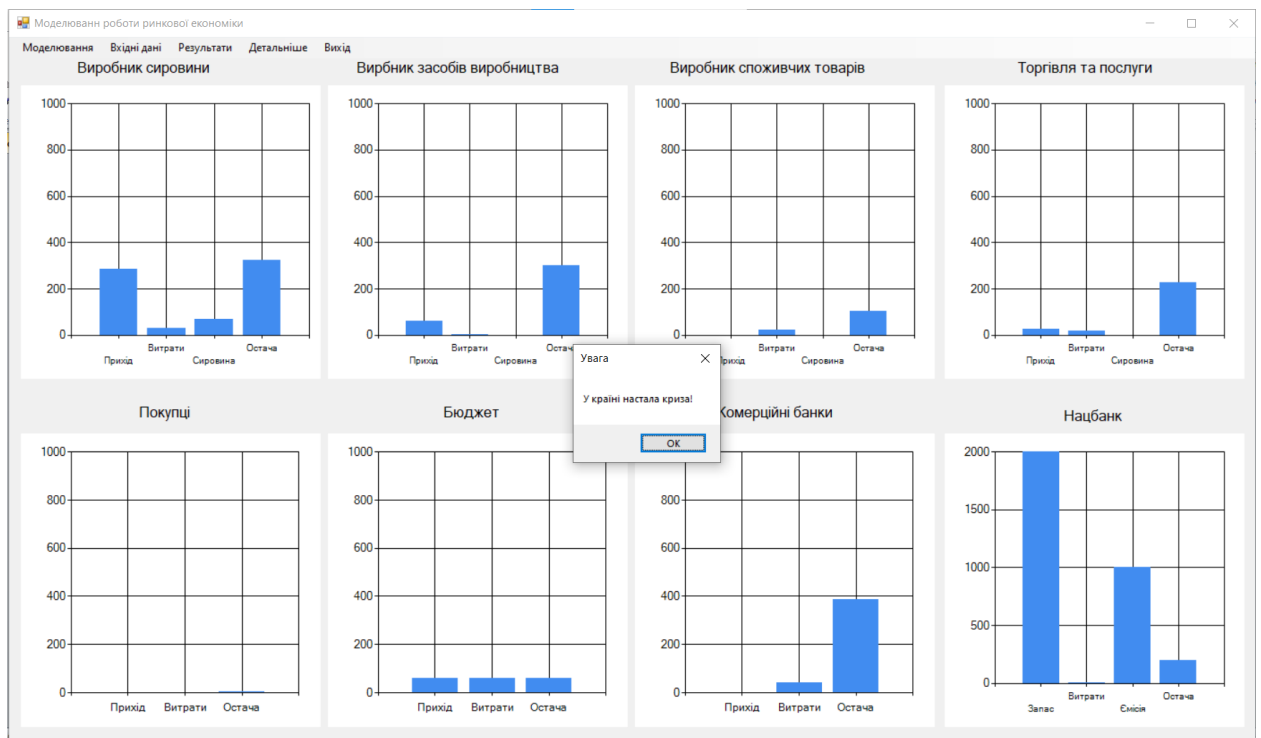


Рисунок 4.11 – Програма після виконання функції

#### 4.5 Вибір стратегії налагодження

Налагодження програми так чи інакше приймає розгляд та логічне вирішення доступної інформації про помилки. Більшість помилок може бути впізнана до непрямих знаків за допомогою ретельного аналізу текстів програм та результатів тестування, не отримуючи додаткової інформації. Таким чином використовуються різні методи налагодження програми:

- 1) метод ручного тестування;
- 2) метод пролога;
- 3) метод зворотного трасування.

Метод ручного тестування - найпростіший і найприродніший спосіб налагодження програмного забезпечення. У виявлених помилках необхідно виконати протестовану програму вручну, використовуючи тест, що комується, роботою, з якої була пізнана помилка. Метод дуже ефективний, але не застосовний для великих програм, програм із важкими розрахунками. Цей метод часто використовується як компонент інших методів усунення несправностей.

Метод прологу ґрунтується на ретельному аналізі симптомів помилки, які можна показати як неправильні результати розрахунків або як повідомлення про помилку. Якщо комп'ютер просто "зависає", причину помилки обчислюють на основі останніх отриманих результатів та дій користувача. Інформація, отримана таким чином, організовується та обережно вивчається, переглядається адекватний фрагмент програми. Внаслідок перегляду та вивчення інформації, програмний код удосконалюється.

Для малих програм ефективним є метод зворотного трасування. Він починається із точки виведення неправильного результату. Для цієї точки знаходиться гіпотеза про значення основних змінних, які могли призвести до отримання доступного результату. Далі, на основі цієї гіпотези, робляться пропозиції про значення змінних у попередній точці. Процес триває, поки не знаходиться причина помилки.

Програм без помилок не існує, тому під час та після написання програми проводилось її налагодження.

#### 4.6 Результати налагодження

При налагодженні програми були виявлені помилки. Наприклад, при відображенні результатів моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі, кількість днів відображалась некоректно.

Помилки виявлялись на ранній стадії та усувались з моменту їх виявлення. Більшість помилок усувалось за допомогою точок переривань програми. Приклад налагодження програми за допомогою точок переривань приведений на рис. 4.12.

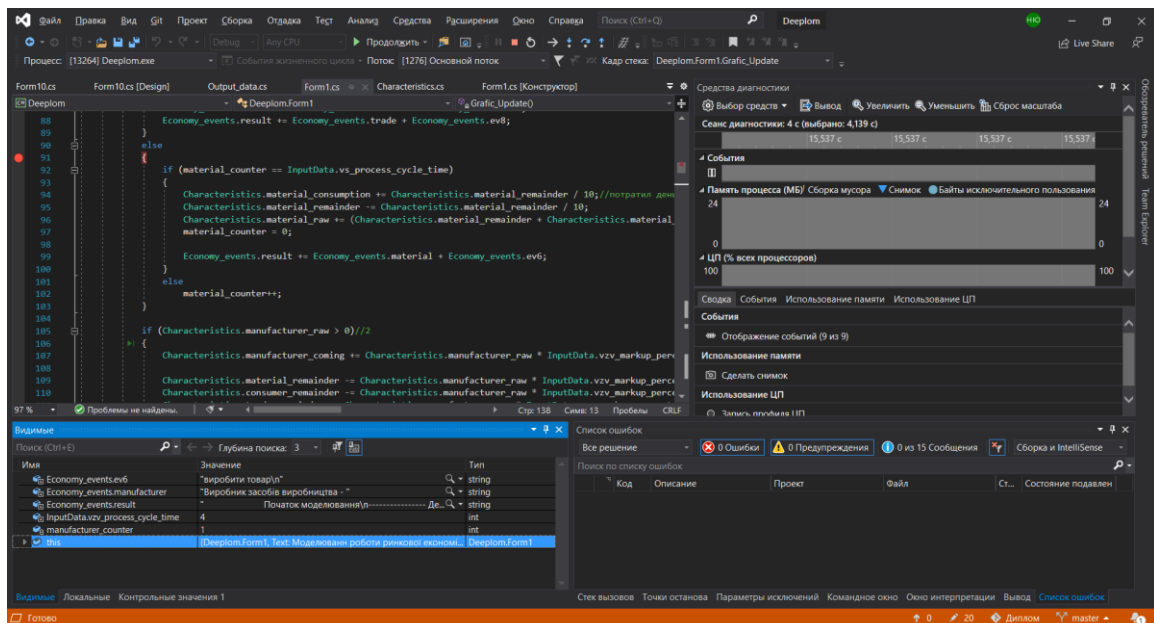


Рисунок 4.12 – Налагодження програми за допомогою точок переривань

#### Висновки до розділу 4

У даному розділі було проведено тестування та налагодження програми. Були виявлені та усунені помилки, які призводили до некоректної роботи програми.

## ВИСНОВКИ

В результаті виконання дипломної роботи було розроблено програму, що реалізує моделювання макроекономічної системи на основі базової ринкової моделі. Розроблено інтерфейс, який інтуїтивно зрозумілий і не вимагає додаткових знань чи навичок при роботі з ним. Було проведено тестування та налагодження даної програми.

Користувачами програми можуть виступати будь-які люди, які хочуть дізнатися механізм функціонування держави на основі базової ринкової моделі. Перед проектуванням було проведено аналіз аналогів та збір вимог з метою збільшення ефективності.

Дана програма імітує роботу ринкової економіки, демонструє рух грошей і демонструє причини виникнення економічних криз.

## ЛІТЕРАТУРА

1. Шпакович О. Ринкова економіка і демократичне суспільство в ОЕСР / О. Шпакович // Віче. – 2010. – 21 (282). – С. 30–31.
2. Aspers P. The worth of goods: Valuation and pricing in the economy / Patrik Aspers, Jens Beckert. – New York : Oxford University Press, 2011. – 346 с.
3. Cowen T. Michael F. Brown: 2003, Who Owns Native Culture?, Harvard University Press, Cambridge, MA, 315 pp., ISBN 0-67401-171-6 [Електронний ресурс] / Tyler Cowen // Journal of Cultural Economics. – 2004. – Т. 28, № 4. – С. 317–323. – Режим доступу: <https://doi.org/10.1007/s10824-004-5547-9>.
4. Smith N. H. Charles Taylor : The Language Animal: The Full Shape of the Human Linguistic Capacity. (Cambridge, MA: Belknap Press of Harvard University Press, 2016. Pp. x, 352.) [Електронний ресурс] / Nicholas H. Smith // The Review of Politics. – 2017. – Т. 79, № 3. – С. 500–502. – Режим доступу: <https://doi.org/10.1017/s0034670517000122>.
5. Mao W. Book Review: China's Spatial Economy: Recent Developments and Reforms G. J. R. LINGE and D. K. FORBES (Eds), 1991 Oxford: Oxford University Press 225 pp. \$20.00 hardback [Електронний ресурс] / Wu Mao // Urban Studies. – 1992. – Т. 29, № 7. – С. 1198–1201. – Режим доступу: <https://doi.org/10.1080/00420989220081161>.
6. Porter P. K. Varieties of Capitalism: The Institutional Foundations of Comparative Advantage [Електронний ресурс] / Philip K. Porter, Peter Hall, David Soskice // The Academy of Management Review. – 2003. – Т. 28, № 3. – С. 515. – Режим доступу: <https://doi.org/10.2307/30040740>.
7. Melé D. Aquinas and the Market: Toward a Humane Economy, by Mary L. Hirschfeld. Cambridge: Harvard University Press, 2018. 288 pp. [Електронний ресурс] / Domènec Melé, Alfredo Pastor // Business Ethics Quarterly. – 2019. – Т. 29, № 03. – С. 425–428. – Режим доступу: <https://doi.org/10.1017/beq.2019.17>.

8. Cachia P. M. M. Badawi, A Critical Introduction to Modern Arabic Poetry (Cambridge, Cambridge University Press, 1975). Pp. 289. [Электронный ресурс] / P. Cachia // International Journal of Middle East Studies. – 1977. – Т. 8, № 4. – С. 571–573. – Режим доступа: <https://doi.org/10.1017/s0020743800026155>.
9. Armstrong D. J. The quarks of object-oriented development [Электронный ресурс] / Deborah J. Armstrong // Communications of the ACM. – 2006. – Т. 49, № 2. – С. 123–128. – Режим доступа: <https://doi.org/10.1145/1113034.1113040>.
10. Meyer B. Object-oriented software construction. / Bertrand Meyer. – Goleta, MA : ISE, 1993.
11. Weisfeld M. A. The object-oriented thought process / Matt A. Weisfeld. – 2-ге вид. – Indianapolis, Ind : Sams Pub., 2004. – 271 с.
12. Dissociated Representations of Irritation and Valence in Human Primary Olfactory Cortex [Электронный ресурс] / C. Zelano [та ін.] // Journal of Neurophysiology. – 2007. – Т. 97, № 3. – С. 1969–1976. – Режим доступа: <https://doi.org/10.1152/jn.01122.2006>.
13. Dexterous Control of a Prosthetic Hand Using Fine-Wire Intramuscular Electrodes in Targeted Extrinsic Muscles [Электронный ресурс] / Christian Cipriani [та ін.] // IEEE Transactions on Neural Systems and Rehabilitation Engineering. – 2014. – Т. 22, № 4. – С. 828–836. – Режим доступа: <https://doi.org/10.1109/tnsre.2014.2301234>.

**ДОДАТКИ**

## ТЕКСТ ПРОГРАМИ

**Файл Form1.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Deeplom
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            int material_counter = 0;
            int manufacturer_counter = 0;
            int consumer_counter = 0;
            int trade_counter = 0;

            void Grafic_Update()
            {
                Economy_events.day++;
                Random rand = new Random();
                Economy_events.result += "-----
----- День " + Economy_events.day + " ----
-----\n";

                if (Economy_events.day == 1)//1
                {
                    InputData.vs_initial_amount_of_money =
                    rand.Next(InputData.vs_initial_amount_of_
                    money_min,
                    InputData.vs_initial_amount_of_money_ma
                    x + 1);

                    InputData.vzv_initial_amount_of_money =
                    rand.Next(InputData.vzv_initial_amount_of_
                    _money_min,
                    InputData.vzv_initial_amount_of_money_m
                    ax + 1);
                    InputData.vst_initial_amount_of_money =
                    rand.Next(InputData.vst_initial_amount_of_

```

```

                    money_min,
                    InputData.vst_initial_amount_of_money_m
                    ax + 1);

                    InputData.ttp_initial_amount_of_money =
                    rand.Next(InputData.ttp_initial_amount_of_
                    money_min,
                    InputData.ttp_initial_amount_of_money_ma
                    x + 1);

                    InputData.vs_process_cycle_time
                    =
                    rand.Next(InputData.vs_process_cycle_time
                    _min,
                    InputData.vs_process_cycle_time_max + 1);
                    InputData.vs_markup_percentage
                    =
                    rand.Next(InputData.vs_markup_percentage
                    _min,
                    InputData.vs_markup_percentage_max + 1);

                    InputData.vzv_process_cycle_time =
                    rand.Next(InputData.vzv_process_cycle_tim
                    e_min,
                    InputData.vzv_process_cycle_time_max +
                    1);

                    InputData.vzv_markup_percentage =
                    rand.Next(InputData.vzv_markup_percentag
                    e_min,
                    InputData.vzv_markup_percentage_max +
                    1);
                    InputData.vst_process_cycle_time
                    =
                    rand.Next(InputData.vst_process_cycle_tim
                    e_min,
                    InputData.vst_process_cycle_time_max +
                    1);
                    InputData.vst_markup_percentage
                    =
                    rand.Next(InputData.vst_markup_percentag
                    e_min,
                    InputData.vst_markup_percentage_max +
                    1);

                    Characteristics.material_remainder =
                    InputData.vs_initial_amount_of_money;

                    Characteristics.manufacturer_remainder =
                    InputData.vzv_initial_amount_of_money;

```

```

Characteristics.consumer_remainder =
InputData.vst_initial_amount_of_money;
    Characteristics.trade_remainder =
InputData.ttp_initial_amount_of_money;

```

```

Characteristics.commercial_remainder =
InputData.commercial_bank_amount;
    Characteristics.bank_stock =
InputData.national_Bank_total_amount;
    Characteristics.bank_remainder =
InputData.national_Bank_total_amount;
    }

```

```

    Characteristics.material_coming = 0;
Characteristics.manufacturer_coming = 0;
    Characteristics.consumer_coming =
0;
    Characteristics.trade_coming = 0;
    Characteristics.budget_coming = 0;
    Characteristics.commercial_coming
= 0;
    Characteristics.buyers_coming = 0;

```

```

Characteristics.material_consumption = 0;
Characteristics.manufacturer_consumption =
0;
Characteristics.consumer_consumption = 0;
    Characteristics.trade_consumption =
0;
    Characteristics.budget_consumption
= 0;
Characteristics.commercial_consumption =
0;
    Characteristics.buyers_consumption
= 0;
    Characteristics.bank_consumption =
0;
    if (Characteristics.material_raw >
0)//2
    {
        Characteristics.material_coming =
Characteristics.material_raw *
InputData.vs_markup_percentage /
100;//заработал деньги за товар

```

```

Characteristics.manufacturer_remainder -=
Characteristics.material_raw *
InputData.vs_markup_percentage /
300;//отнимаем деньги у купивших товар

```

```

Characteristics.consumer_remainder -=
Characteristics.material_raw *
InputData.vs_markup_percentage / 300;
    Characteristics.trade_remainder -=
Characteristics.material_raw *
InputData.vs_markup_percentage / 300;

```

```

Characteristics.manufacturer_consumption
+= Characteristics.material_raw *
InputData.vs_markup_percentage /
300;//делаем расход купившим товар

```

```

Characteristics.consumer_consumption +=
Characteristics.material_raw *
InputData.vs_markup_percentage / 300;
    Characteristics.trade_consumption
+= Characteristics.material_raw *
InputData.vs_markup_percentage / 300;

```

```

        Characteristics.material_raw =
0;//продал товар, его не стало

```

```

        Economy_events.result +=
Economy_events.material +
Economy_events.ev10;
        Economy_events.result +=
Economy_events.manufacturer +
Economy_events.ev8;
        Economy_events.result +=
Economy_events.consumer +
Economy_events.ev8;
        Economy_events.result +=
Economy_events.trade +
Economy_events.ev8;
    }
    else
    {
        if (material_counter ==
InputData.vs_process_cycle_time)
        {

```

```

Characteristics.material_consumption +=
Characteristics.material_remainder /
10;//потратил деньги на производство
сырья

```

```

Characteristics.material_remainder -=
Characteristics.material_remainder / 10;
    Characteristics.material_raw +=
(Characteristics.material_remainder +
Characteristics.material_remainder /
10);//произвели товар(думаю)
    material_counter = 0;

    Economy_events.result +=
Economy_events.material +
Economy_events.ev6;
    }
    else
        material_counter++;
    }

    if (Characteristics.manufacturer_raw
> 0)//2
    {
Characteristics.manufacturer_coming +=
Characteristics.manufacturer_raw *
InputData.vzv_markup_percentage /
100;//заработал деньги за товар

Characteristics.material_remainder -=
Characteristics.manufacturer_raw *
InputData.vzv_markup_percentage / 300;

Characteristics.consumer_remainder -=
Characteristics.manufacturer_raw *
InputData.vzv_markup_percentage / 300;
    Characteristics.trade_remainder -=
Characteristics.manufacturer_raw *
InputData.vzv_markup_percentage / 300;

Characteristics.material_consumption +=
Characteristics.manufacturer_raw *
InputData.vzv_markup_percentage / 300;

Characteristics.consumer_consumption +=
Characteristics.manufacturer_raw *
InputData.vzv_markup_percentage / 300;
    Characteristics.trade_consumption
+= Characteristics.manufacturer_raw *
InputData.vzv_markup_percentage / 300;

    Characteristics.manufacturer_raw
= 0;//продал товар, его не стало

```

```

    Economy_events.result +=
Economy_events.manufacturer +
Economy_events.ev10;
    Economy_events.result +=
Economy_events.material +
Economy_events.ev8;
    Economy_events.result +=
Economy_events.consumer +
Economy_events.ev8;
    Economy_events.result +=
Economy_events.trade +
Economy_events.ev8;
    }
    else
    {
        if (manufacturer_counter ==
InputData.vzv_process_cycle_time)
        {
Characteristics.manufacturer_consumption
+= Characteristics.manufacturer_remainder /
10;//потратил деньги на производство
сырья

Characteristics.manufacturer_remainder -=
Characteristics.manufacturer_remainder /
10;

Characteristics.manufacturer_raw +=
(Characteristics.manufacturer_remainder +
Characteristics.manufacturer_remainder /
10);//произвели товар(думаю)

Characteristics.manufacturer_coming = 0;

    Economy_events.result +=
Economy_events.manufacturer +
Economy_events.ev6;
    }
    else
        manufacturer_counter++;
    }

    if (Characteristics.consumer_raw >
0)//2
    {
        Characteristics.consumer_coming
+= Characteristics.consumer_raw *
InputData.vst_markup_percentage /
100;//заработал деньги за товар

```

```

    Characteristics.trade_remainder -=
Characteristics.consumer_raw *
InputData.vst_markup_percentage / 100;

```

```

    Characteristics.trade_consumption
+= Characteristics.consumer_raw *
InputData.vst_markup_percentage / 100;

```

```

    Characteristics.consumer_raw =
0;//продал товар, его не стало

```

```

    Economy_events.result +=
Economy_events.consumer +
Economy_events.ev10;
    Economy_events.result +=
Economy_events.trade +
Economy_events.ev8;
    }
    else
    {
        if (consumer_counter ==
InputData.vst_process_cycle_time)
        {

```

```

Characteristics.consumer_consumption +=
Characteristics.consumer_remainder /
10;//потратил деньги на производство
сырья

```

```

Characteristics.consumer_remainder -=
Characteristics.consumer_remainder / 10;
    Characteristics.consumer_raw
+= (Characteristics.consumer_remainder +
Characteristics.consumer_remainder /
10);//произвели товар(думаю)
    consumer_counter = 0;

```

```

    Economy_events.result +=
Economy_events.consumer +
Economy_events.ev6;
    }
    else
        consumer_counter++;
    }

```

```

    if (Characteristics.trade_raw > 0)//2
    {
        Characteristics.trade_coming +=
Characteristics.trade_raw / 10;//заработал
деньги за товар

```

```

Characteristics.material_consumption +=
Characteristics.trade_raw / 70;

```

```

Characteristics.consumer_consumption +=
Characteristics.trade_raw / 70;

```

```

Characteristics.manufacturer_consumption
+= Characteristics.trade_raw / 70;

```

```

Characteristics.budget_consumption +=
Characteristics.trade_raw / 70;

```

```

Characteristics.commercial_consumption +=
Characteristics.trade_raw / 70;

```

```

Characteristics.buyers_consumption +=
Characteristics.buyers_remainder *
InputData.money_spending_percentage /
100;

```

```

    Characteristics.bank_consumption
+= Characteristics.trade_raw / 70;

```

```

Characteristics.material_remainder -=
Characteristics.trade_raw / 70;

```

```

Characteristics.consumer_remainder -=
Characteristics.trade_raw / 70;

```

```

Characteristics.manufacturer_remainder -=
Characteristics.trade_raw / 70;
    Characteristics.budget_remainder
-= Characteristics.trade_raw / 70;

```

```

Characteristics.commercial_remainder -=
Characteristics.trade_raw / 70;
    Characteristics.buyers_remainder -
= Characteristics.buyers_remainder *
InputData.money_spending_percentage /
100;
    Characteristics.bank_remainder -=
Characteristics.trade_raw / 70;

```

```

    Characteristics.trade_raw =
0;//продал товар, его не стало

```

```

    Economy_events.result +=
Economy_events.trade +
Economy_events.ev10;
    Economy_events.result +=
Economy_events.material +
Economy_events.ev8;

```

```

        Economy_events.result +=
Economy_events.consumer +
Economy_events.ev8;
        Economy_events.result +=
Economy_events.manufacturer +
Economy_events.ev8;
        Economy_events.result +=
Economy_events.budget +
Economy_events.ev8;
        Economy_events.result +=
Economy_events.commercial +
Economy_events.ev8;
        Economy_events.result +=
Economy_events.buyers +
Economy_events.ev8;
        Economy_events.result +=
Economy_events.bank +
Economy_events.ev8;
    }
    else
    {
        if (trade_counter == 1)
        {
            Characteristics.trade_consumption +=
            Characteristics.trade_remainder /
            10;//потратил деньги на производство
            сырья
            Characteristics.trade_remainder
            -= Characteristics.trade_remainder / 10;
            Characteristics.trade_raw +=
            (Characteristics.trade_remainder +
            Characteristics.trade_remainder /
            10);//произвели товар(думаю)
            trade_counter = 0;

            Economy_events.result +=
            Economy_events.trade +
            Economy_events.ev6;
        }
        else
            trade_counter++;
    }

    if (Characteristics.material_coming >
0)//3
    {
        Characteristics.material_remainder +=
        Characteristics.material_coming -
        (Characteristics.material_coming *
        InputData.tax_money_percentage / 100

```

```

        +
        Characteristics.material_coming *
        InputData.pay_wages_percentage /
        100);//заплатил налоги и зп

        Characteristics.budget_coming +=
        Characteristics.material_coming -
        Characteristics.material_coming *
        InputData.tax_money_percentage / 100;
        Characteristics.budget_remainder
        += Characteristics.material_coming -
        Characteristics.material_coming *
        InputData.tax_money_percentage / 100;

        Economy_events.result +=
        Economy_events.material +
        Economy_events.ev1;
        Economy_events.result +=
        Economy_events.material +
        Economy_events.ev2;
    }

    if
    (Characteristics.manufacturer_coming >
0)//3 5
    {
        Characteristics.manufacturer_remainder +=
        Characteristics.manufacturer_coming -
        (Characteristics.manufacturer_coming *
        InputData.tax_money_percentage / 100
        +
        Characteristics.manufacturer_coming *
        InputData.pay_wages_percentage /
        100);//заплатил налоги и зп

        Characteristics.budget_coming +=
        Characteristics.manufacturer_coming -
        Characteristics.manufacturer_coming *
        InputData.tax_money_percentage / 100;
        Characteristics.budget_remainder
        += Characteristics.manufacturer_coming -
        Characteristics.manufacturer_coming *
        InputData.tax_money_percentage / 100;

        Economy_events.result +=
        Economy_events.manufacturer +
        Economy_events.ev1;
        Economy_events.result +=
        Economy_events.manufacturer +
        Economy_events.ev2;
    }

```

```

        if (Characteristics.consumer_coming
> 0)//3
        {
Characteristics.consumer_remainder +=
Characteristics.consumer_coming -

(Characteristics.consumer_coming *
InputData.tax_money_percentage / 100
+
Characteristics.consumer_coming *
InputData.pay_wages_percentage /
100);//заплатил налоги и зп

        Characteristics.budget_coming +=
Characteristics.consumer_coming -
Characteristics.consumer_coming *
InputData.tax_money_percentage / 100;
        Characteristics.budget_remainder
+= Characteristics.consumer_coming -
Characteristics.consumer_coming *
InputData.tax_money_percentage / 100;

        Economy_events.result +=
Economy_events.consumer +
Economy_events.ev1;
        Economy_events.result +=
Economy_events.consumer +
Economy_events.ev2;
        }

        if (Characteristics.trade_coming >
0)//3
        {
        Characteristics.trade_remainder
+= Characteristics.trade_coming -
        (Characteristics.trade_coming *
InputData.tax_money_percentage / 100
+ Characteristics.trade_coming
* InputData.pay_wages_percentage /
100);//заплатил налоги и зп

        Characteristics.budget_coming +=
Characteristics.trade_coming -
Characteristics.trade_coming *
InputData.tax_money_percentage / 100;
        Characteristics.budget_remainder
+= Characteristics.trade_coming -
Characteristics.trade_coming *
InputData.tax_money_percentage / 100;

```

```

        Economy_events.result +=
Economy_events.trade +
Economy_events.ev1;
        Economy_events.result +=
Economy_events.trade +
Economy_events.ev2;
        }

        if
(Characteristics.material_remainder <
(InputData.vs_initial_amount_of_money /
10))//кредит у комбанка
        {
Characteristics.material_remainder +=
InputData.commercial_bank_amount *
InputData.interest_on_loans / 70;

Characteristics.commercial_remainder -=
Characteristics.commercial_remainder / 10;

Characteristics.commercial_consumption +=
Characteristics.commercial_remainder / 10;
        Characteristics.material_coming
+= InputData.commercial_bank_amount *
InputData.interest_on_loans / 70;

        Economy_events.result +=
Economy_events.material +
Economy_events.ev3;
        }

        if
(Characteristics.manufacturer_remainder <
(InputData.vzv_initial_amount_of_money /
10))//кредит у комбанка
        {
Characteristics.manufacturer_remainder +=
InputData.commercial_bank_amount *
InputData.interest_on_loans / 70;

Characteristics.commercial_remainder -=
Characteristics.commercial_remainder / 10;

Characteristics.commercial_consumption +=
Characteristics.commercial_remainder / 10;

Characteristics.manufacturer_coming +=
InputData.commercial_bank_amount *
InputData.interest_on_loans / 70;

```

```

        Economy_events.result +=
Economy_events.manufacturer +
Economy_events.ev3;
    }

    if
(Characteristics.consumer_remainder <
(InputData.vst_initial_amount_of_money /
10))//кредит у комбанка
    {

Characteristics.consumer_remainder +=
InputData.commercial_bank_amount *
InputData.interest_on_loans / 70;

Characteristics.commercial_remainder -=
Characteristics.commercial_remainder / 10;

Characteristics.commercial_consumption +=
Characteristics.commercial_remainder / 10;
    Characteristics.consumer_coming
+= InputData.commercial_bank_amount *
InputData.interest_on_loans / 70;

        Economy_events.result +=
Economy_events.consumer +
Economy_events.ev3;
    }

    if (Characteristics.trade_remainder <
(InputData.ttp_initial_amount_of_money /
10))//кредит у комбанка
    {
        Characteristics.trade_remainder
+= InputData.commercial_bank_amount *
InputData.interest_on_loans / 70;

Characteristics.commercial_remainder -=
Characteristics.commercial_remainder / 10;

Characteristics.commercial_consumption +=
Characteristics.commercial_remainder / 10;
        Characteristics.trade_coming +=
InputData.commercial_bank_amount *
InputData.interest_on_loans / 70;

        Economy_events.result +=
Economy_events.trade +
Economy_events.ev3;
    }

```

```

    if
(Characteristics.commercial_remainder <
(InputData.commercial_bank_amount /
10))//кредит у нацбанка
    {

Characteristics.commercial_remainder +=
InputData.national_Bank_total_amount *
InputData.national_Bank_rate / 70;
        Characteristics.bank_remainder -=
Characteristics.bank_remainder / 7;

        Characteristics.bank_consumption
+= Characteristics.bank_remainder / 7;

Characteristics.commercial_coming +=
InputData.national_Bank_total_amount *
InputData.national_Bank_rate / 70;

        Economy_events.result +=
Economy_events.commercial +
Economy_events.ev3;
    }

    if (Characteristics.budget_remainder
> 300)
    {
        Characteristics.buyers_coming +=
Characteristics.budget_remainder / 2;
        Characteristics.buyers_remainder
+= Characteristics.budget_remainder / 2;

Characteristics.budget_consumption +=
Characteristics.budget_remainder;
        Characteristics.budget_remainder
= 0;

        Economy_events.result +=
Economy_events.budget +
Economy_events.ev7;
        Economy_events.result +=
Economy_events.budget +
Economy_events.ev9;
    }

    if (rand.Next(1, 100) <
InputData.likelihood_of_an_emergency)
    {

Characteristics.budget_consumption +=
Characteristics.budget_remainder / 2;

```

```

        Characteristics.budget_remainder
    -= Characteristics.budget_remainder / 2;

        Economy_events.result +=
    Economy_events.budget +
    Economy_events.ev5;
    }

    if (Characteristics.bank_remainder <
    (InputData.national_Bank_total_amount /
    10))//закончились деньги в нацбанке
    {
        Characteristics.bank_stock +=
    100;
        Characteristics.bank_emission +=
    100;
        Characteristics.bank_remainder
    += 100;

        Economy_events.result +=
    Economy_events.bank +
    Economy_events.ev4;
    }

    chart1.Series["Data"].Points.DataBindXY(n
    ew string[] { "Прихід", "Витрати",
    "Сировина", "Остача" }, new int[] {
    Characteristics.material_coming,
    Characteristics.material_consumption,
    Characteristics.material_raw,
    Characteristics.material_remainder });

    //chart1.Series["Data"].Points.DataBindXY(
    new string[] { "Прихід" }, new int[] {
    Characteristics.material_coming });

    //chart1.Series["Data1"].Points.DataBindXY
    (new string[] { "Витрати" }, new int[] {
    Characteristics.material_consumption });

    //chart1.Series["Data2"].Points.DataBindXY
    (new string[] { "Сировина" }, new int[] {
    Characteristics.material_raw });

    //chart1.Series["Data3"].Points.DataBindXY
    (new string[] { "Остача" }, new int[] {
    Characteristics.material_remainder });

    chart2.Series["Data"].Points.DataBindXY(n
    ew string[] { "Прихід", "Витрати",
    "Сировина", "Остача" }, new int[] {
    Characteristics.manufacturer_coming,
    Characteristics.manufacturer_consumption,
    Characteristics.manufacturer_raw,
    Characteristics.manufacturer_remainder });

    chart3.Series["Data"].Points.DataBindXY(n
    ew string[] { "Прихід", "Витрати",
    "Сировина", "Остача" }, new int[] {
    Characteristics.consumer_coming,
    Characteristics.consumer_consumption,
    Characteristics.consumer_raw,
    Characteristics.consumer_remainder });

    chart4.Series["Data"].Points.DataBindXY(n
    ew string[] { "Прихід", "Витрати",
    "Сировина", "Остача" }, new int[] {
    Characteristics.trade_coming,
    Characteristics.trade_consumption,
    Characteristics.trade_raw,
    Characteristics.trade_remainder });

    chart6.Series["Data"].Points.DataBindXY(n
    ew string[] { "Прихід", "Витрати",
    "Остача" }, new int[] {
    Characteristics.budget_coming,
    Characteristics.budget_consumption,
    Characteristics.budget_remainder });

    chart7.Series["Data"].Points.DataBindXY(n
    ew string[] { "Прихід", "Витрати",
    "Остача" }, new int[] {
    Characteristics.commercial_coming,
    Characteristics.commercial_consumption,
    Characteristics.commercial_remainder });

    chart5.Series["Data"].Points.DataBindXY(n
    ew string[] { "Прихід", "Витрати",
    "Остача" }, new int[] {
    Characteristics.buyers_coming,
    Characteristics.buyers_consumption,
    Characteristics.buyers_remainder });

    chart8.Series["Data"].Points.DataBindXY(n
    ew string[] { "Запас", "Витрати", "Ємісія",
    "Остача" }, new int[] {
    Characteristics.bank_stock,
    Characteristics.bank_consumption,
    Characteristics.bank_emission,
    Characteristics.bank_remainder });

    if (Characteristics.bank_emission ==
    1000)//кризис

```

```

    {
        Economy_events.result += "-----
----- Кінець програми -----\n";
        timer1.Stop();
        MessageBox.Show("У країні
настала криза!", "Увага",
MessageBoxButtons.OK);
    }

```

```

Output_data.output_array[Economy_events.
day - 1, 0] = Economy_events.day;

```

```

Output_data.output_array[Economy_events.
day - 1, 1] =
Characteristics.manufacturer_coming;

```

```

Output_data.output_array[Economy_events.
day - 1, 2] =
Characteristics.manufacturer_consumption;

```

```

Output_data.output_array[Economy_events.
day - 1, 3] = Characteristics.material_raw;

```

```

Output_data.output_array[Economy_events.
day - 1, 4] =
Characteristics.material_remainder;

```

```

Output_data.output_array[Economy_events.
day - 1, 5] =
Characteristics.manufacturer_coming;

```

```

Output_data.output_array[Economy_events.
day - 1, 6] =
Characteristics.manufacturer_consumption;

```

```

Output_data.output_array[Economy_events.
day - 1, 7] =
Characteristics.manufacturer_raw;

```

```

Output_data.output_array[Economy_events.
day - 1, 8] =
Characteristics.manufacturer_remainder;

```

```

Output_data.output_array[Economy_events.
day - 1, 9] =
Characteristics.consumer_coming;

```

```

Output_data.output_array[Economy_events.
day - 1, 10] =
Characteristics.consumer_consumption;

```

```

Output_data.output_array[Economy_events.

```

```

day - 1, 11] =
Characteristics.consumer_raw;

```

```

Output_data.output_array[Economy_events.
day - 1, 12] =
Characteristics.consumer_remainder;

```

```

Output_data.output_array[Economy_events.
day - 1, 13] = Characteristics.trade_coming;

```

```

Output_data.output_array[Economy_events.
day - 1, 14] =
Characteristics.trade_consumption;

```

```

Output_data.output_array[Economy_events.
day - 1, 15] = Characteristics.trade_raw;

```

```

Output_data.output_array[Economy_events.
day - 1, 16] =
Characteristics.trade_remainder;

```

```

Output_data.output_array[Economy_events.
day - 1, 17] =
Characteristics.buyers_coming;

```

```

Output_data.output_array[Economy_events.
day - 1, 18] =
Characteristics.buyers_consumption;

```

```

Output_data.output_array[Economy_events.
day - 1, 19] =
Characteristics.buyers_remainder;

```

```

Output_data.output_array[Economy_events.
day - 1, 20] =
Characteristics.budget_coming;

```

```

Output_data.output_array[Economy_events.
day - 1, 21] =
Characteristics.budget_consumption;

```

```

Output_data.output_array[Economy_events.
day - 1, 22] =
Characteristics.budget_remainder;

```

```

Output_data.output_array[Economy_events.
day - 1, 23] =
Characteristics.commercial_coming;

```

```

Output_data.output_array[Economy_events.
day - 1, 24] =
Characteristics.commercial_consumption;

```

```

Output_data.output_array[Economy_events.
day - 1, 25] =
Characteristics.commercial_remainder;

Output_data.output_array[Economy_events.
day - 1, 26] = Characteristics.bank_stock;

Output_data.output_array[Economy_events.
day - 1, 27] =
Characteristics.bank_consumption;

Output_data.output_array[Economy_events.
day - 1, 28] =
Characteristics.bank_emission;

Output_data.output_array[Economy_events.
day - 1, 29] =
Characteristics.bank_remainder;
    }

    private void timer1_Tick(object sender,
EventArgs e)
    {
        Grafic_Update();
    }

    private void
переглянутиToolStripMenuItem_Click(obj
ect sender, EventArgs e)
    {
        Form4 newForm = new Form4();
        newForm.Show();
    }

    private void
розпочатиПокроковеМоделюванняToolSt
ripMenuItem_Click(object sender,
EventArgs e)
    {
        timer1.Start();
    }

розпочатиПокроковеМоделюванняToolSt
ripMenuItem.Enabled = false;

зупинитиПокроковеМоделюванняToolStri
pMenuItem.Enabled = true;
    }

    private void
зупинитиПокроковеМоделюванняToolStri
pMenuItem_Click(object sender, EventArgs
e)

```

```

    {
        timer1.Stop();
    }

розпочатиПокроковеМоделюванняToolSt
ripMenuItem.Enabled = true;

зупинитиПокроковеМоделюванняToolStri
pMenuItem.Enabled = false;
    }

    private void
змоделювати1КрокToolStripMenuItem_Cli
ck(object sender, EventArgs e)
    {
        Grafic_Update();
    }

    private void
змінитиВхідніДаніToolStripMenuItem_Cli
ck(object sender, EventArgs e)
    {
        if (timer1.Enabled == true)
        {
            timer1.Stop();
            DialogResult result =
MessageBox.Show("Зупинити покрокове
моделювання та змінити вхідні дані?",
"Увага", MessageBoxButtons.YesNo);
            if (result == DialogResult.Yes)
            {
                Form2 newForm = new
Form2();
                newForm.Show();
                this.Hide();
            }
            else
                timer1.Start();
        }
        else
        {
            Form2 newForm = new Form2();
            newForm.Show();
            this.Hide();
        }
    }

    private void
вихідToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        Application.Exit();
    }

```

```

private void Form1_Load(object
sender, EventArgs e)
{
    //виробник сировини

chart1.Series["Data"].Points.AddXY("Прихід", Characteristics.material_coming);

chart1.Series["Data"].Points.AddXY("Витрати",
Characteristics.material_consumption);

chart1.Series["Data"].Points.AddXY("Сировина", Characteristics.material_raw);

chart1.Series["Data"].Points.AddXY("Остача", Characteristics.material_remainder);

//chart1.Series["Data"].Points[1].Color =
Color.Red;

chart1.ChartAreas[0].AxisY.Maximum =
1000;

    //Виробник засобів виробництва

chart2.Series["Data"].Points.AddXY("Прихід", Characteristics.manufacturer_coming);

chart2.Series["Data"].Points.AddXY("Витрати",
Characteristics.manufacturer_consumption);

chart2.Series["Data"].Points.AddXY("Сировина", Characteristics.manufacturer_raw);

chart2.Series["Data"].Points.AddXY("Остача",
Characteristics.manufacturer_remainder);

chart2.ChartAreas[0].AxisY.Maximum =
1000;

    //Виробник споживчих товарів

chart3.Series["Data"].Points.AddXY("Прихід", Characteristics.consumer_coming);

chart3.Series["Data"].Points.AddXY("Витрати",
Characteristics.consumer_consumption);

```

```

chart3.Series["Data"].Points.AddXY("Сировина", Characteristics.consumer_raw);

chart3.Series["Data"].Points.AddXY("Остача", Characteristics.consumer_remainder);

chart3.ChartAreas[0].AxisY.Maximum =
1000;

    //Торгівля та Послуги

chart4.Series["Data"].Points.AddXY("Прихід", Characteristics.trade_coming);

chart4.Series["Data"].Points.AddXY("Витрати", Characteristics.trade_consumption);

chart4.Series["Data"].Points.AddXY("Сировина", Characteristics.trade_raw);

chart4.Series["Data"].Points.AddXY("Остача", Characteristics.trade_remainder);

chart4.ChartAreas[0].AxisY.Maximum =
1000;

    //Покупці споживчих товарів

chart5.Series["Data"].Points.AddXY("Прихід", Characteristics.buyers_coming);

chart5.Series["Data"].Points.AddXY("Витрати", Characteristics.buyers_consumption);

chart5.Series["Data"].Points.AddXY("Остача", Characteristics.buyers_remainder);

chart5.ChartAreas[0].AxisY.Maximum =
1000;

    //Бюджет

chart6.Series["Data"].Points.AddXY("Прихід", Characteristics.budget_coming);

chart6.Series["Data"].Points.AddXY("Витрати", Characteristics.budget_consumption);

chart6.Series["Data"].Points.AddXY("Остача", Characteristics.budget_remainder);

```

```

chart6.ChartAreas[0].AxisY.Maximum =
1000;

        //Комерційні банки

chart7.Series["Data"].Points.AddXY("Прихі
д", Characteristics.commercial_coming);

chart7.Series["Data"].Points.AddXY("Витр
ати",
Characteristics.commercial_consumption);

chart7.Series["Data"].Points.AddXY("Оста
ча", Characteristics.commercial_remainder);

chart7.ChartAreas[0].AxisY.Maximum =
1000;

        //Нацбанк

chart8.Series["Data"].Points.AddXY("Запас
", Characteristics.bank_stock);

chart8.Series["Data"].Points.AddXY("Витр
ати", Characteristics.bank_consumption);

chart8.Series["Data"].Points.AddXY("Ємісі
я", Characteristics.bank_emission);

chart8.Series["Data"].Points.AddXY("Оста
ча", Characteristics.bank_remainder);

chart8.ChartAreas[0].AxisY.Maximum =
2000;

розпочатиПокроковеМоделюванняToolStrip
MenuItem.Enabled = true;

зупинитиПокроковеМоделюванняToolStrip
MenuItem.Enabled = false;

        timer1.Interval = 1000 /
InputData.speed;
    }

    private void chart1_Click(object
sender, EventArgs e)
    {

    }

```

```

        private void
дзнатисяБільшеПроПрограмуToolStripM
enuItem_Click(object sender, EventArgs e)
    {
        Form3 newForm = new Form3();
        newForm.Show();
    }

        private void
виробникиСировиниToolStripMenuItem_
Click(object sender, EventArgs e)
    {
        Form5 newForm = new Form5();
        newForm.Show();
        this.Hide();
    }

        private void
виробникиЗасобівВиробництваToolStrip
MenuItem_Click(object sender, EventArgs
e)
    {
        Form6 newForm = new Form6();
        newForm.Show();
        this.Hide();
    }

        private void
виробникиСпоживчихТоварівToolStripMe
nuItem_Click(object sender, EventArgs e)
    {
        Form7 newForm = new Form7();
        newForm.Show();
        this.Hide();
    }

        private void
торгівляТаПослугиToolStripMenuItem_Cli
ck(object sender, EventArgs e)
    {
        Form8 newForm = new Form8();
        newForm.Show();
        this.Hide();
    }

        private void
іншіToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        Form9 newForm = new Form9();
        newForm.Show();
        this.Hide();
    }

```

```

        private void
переглянутиРезультатиМоделюванняTool
StripMenuItem_Click(object sender,
EventArgs e)
    {
        Form10 newForm = new Form10();
        newForm.Show();
    }
}

```

### Файл Form2.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Deepлом
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void button1_Click(object
sender, EventArgs e)
        {
            trackBar1.Value = InputData.speed;
            InputData.tax_money_percentage =
int.Parse(textBox1.Text);
            InputData.pay_wages_percentage =
int.Parse(textBox2.Text);
            InputData.interest_on_loans =
int.Parse(textBox3.Text);
            InputData.national_Bank_rate =
int.Parse(textBox4.Text);
            InputData.loan_repayment_period =
int.Parse(textBox5.Text);

            InputData.simulation_period_interval =
int.Parse(textBox6.Text);

            Form1 newForm = new Form1();
            newForm.Show();
            this.Hide();
        }
    }
}

```

```

    }

    private void trackBar1_Scroll(object
sender, EventArgs e)
    {
        InputData.speed = trackBar1.Value;
    }

    private void Form2_Load(object
sender, EventArgs e)
    {
        trackBar1.Value = InputData.speed;
        textBox1.Text =
InputData.tax_money_percentage.ToString()
;
        textBox2.Text =
InputData.pay_wages_percentage.ToString()
;
        textBox3.Text =
InputData.interest_on_loans.ToString();
        textBox4.Text =
InputData.national_Bank_rate.ToString();
        textBox5.Text =
InputData.loan_repayment_period.ToString(
);
        textBox6.Text =
InputData.simulation_period_interval.ToStri
ng();
    }
}

```

### Файл Form3.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Deepлом
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void Form3_Load(object
sender, EventArgs e)

```

```

    {
        int newFontSize = 14;
        richTextBox1.Select(0, 37);
        richTextBox1.SelectionAlignment =
HorizontalAlignment.Center;
        richTextBox1.SelectionFont = new
Font(richTextBox1.Font.FontFamily,
(float)newFontSize, FontStyle.Bold);

        richTextBox1.Select(0, 0);
    }
}

```

#### Файл Form4.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Deeplom
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();

            private void Form4_Load(object
sender, EventArgs e)
            {
                richTextBox1.Text =
Economy_events.result;
            }

            private void button1_Click(object
sender, EventArgs e)
            {
                richTextBox1.Text =
Economy_events.result;
                richTextBox1.Update();
            }
        }
    }
}

```

#### Файл Form5.cs

```

using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```
namespace Deeplom
```

```

{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();

            private void label1_Click(object
sender, EventArgs e)
            {

            }

            private void button1_Click(object
sender, EventArgs e)
            {

            }

            InputData.vs_initial_amount_of_money_min =
int.Parse(textBox1.Text);

            InputData.vs_initial_amount_of_money_max =
int.Parse(textBox2.Text);

            InputData.vs_process_cycle_time_min =
int.Parse(textBox4.Text);

            InputData.vs_process_cycle_time_max =
int.Parse(textBox5.Text);

            InputData.vs_markup_percentage_min =
int.Parse(textBox6.Text);

            InputData.vs_markup_percentage_max =
int.Parse(textBox7.Text);

            Random rand = new Random();
            InputData.vs_process_cycle_time =
rand.Next(InputData.vs_process_cycle_time_min,
InputData.vs_process_cycle_time_max + 1);
            InputData.vs_markup_percentage =
rand.Next(InputData.vs_markup_percentage

```

```

_min,
InputData.vs_markup_percentage_max + 1);

    Form1 newForm = new Form1();
    newForm.Show();
    this.Hide();
}

private void Form5_Load(object
sender, EventArgs e)
{
    textBox1.Text =
InputData.vs_initial_amount_of_money_mi
n.ToString();
    textBox2.Text =
InputData.vs_initial_amount_of_money_ma
x.ToString();
    textBox4.Text =
InputData.vs_process_cycle_time_min.ToSt
ring();
    textBox5.Text =
InputData.vs_process_cycle_time_max.ToSt
ring();
    textBox6.Text =
InputData.vs_markup_percentage_min.ToSt
ring();
    textBox7.Text =
InputData.vs_markup_percentage_max.ToSt
ring();
}
}
}
}

```

### Файл Form6.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Deepлом
{
    public partial class Form6 : Form
    {
        public Form6()
        {
            InitializeComponent();
        }
    }
}

```

```

private void label3_Click(object
sender, EventArgs e)
{
}

private void button1_Click(object
sender, EventArgs e)
{
    InputData.vzv_initial_amount_of_money_m
in = int.Parse(textBox3.Text);

    InputData.vzv_initial_amount_of_money_m
ax = int.Parse(textBox4.Text);

    InputData.vzv_process_cycle_time_min =
int.Parse(textBox5.Text);

    InputData.vzv_process_cycle_time_max =
int.Parse(textBox6.Text);

    InputData.vzv_markup_percentage_min =
int.Parse(textBox7.Text);

    InputData.vzv_markup_percentage_max =
int.Parse(textBox8.Text);

    Random rand = new Random();
    InputData.vzv_process_cycle_time =
rand.Next(InputData.vzv_process_cycle_tim
e_min,
InputData.vzv_process_cycle_time_max +
1);

    InputData.vzv_markup_percentage =
rand.Next(InputData.vzv_markup_percentag
e_min,
InputData.vzv_markup_percentage_max +
1);

    Form1 newForm = new Form1();
    newForm.Show();
    this.Hide();
}

private void Form6_Load(object
sender, EventArgs e)
{
    textBox3.Text =
InputData.vzv_initial_amount_of_money_m
in.ToString();
}

```

```

        textBox4.Text =
InputData.vzv_initial_amount_of_money_max.ToString();
        textBox5.Text =
InputData.vzv_process_cycle_time_min.ToString();
        textBox6.Text =
InputData.vzv_process_cycle_time_max.ToString();
        textBox7.Text =
InputData.vzv_markup_percentage_min.ToString();
        textBox8.Text =
InputData.vzv_markup_percentage_max.ToString();
    }
}

```

### Файл Form7.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Deepлом
{
    public partial class Form7 : Form
    {
        public Form7()
        {
            InitializeComponent();

            private void button1_Click(object sender, EventArgs e)
            {
                InputData.vst_initial_amount_of_money_min = int.Parse(textBox3.Text);

                InputData.vst_initial_amount_of_money_max = int.Parse(textBox4.Text);

                InputData.vst_process_cycle_time_min = int.Parse(textBox5.Text);

                InputData.vst_process_cycle_time_max = int.Parse(textBox6.Text);

```

```

InputData.vst_markup_percentage_min = int.Parse(textBox7.Text);

```

```

InputData.vst_markup_percentage_max = int.Parse(textBox8.Text);

```

```

        Random rand = new Random();
        InputData.vst_process_cycle_time = rand.Next(InputData.vst_process_cycle_time_min,
InputData.vst_process_cycle_time_max + 1);
        InputData.vst_markup_percentage = rand.Next(InputData.vst_markup_percentage_min,
InputData.vst_markup_percentage_max + 1);

```

```

        Form1 newForm = new Form1();
        newForm.Show();
        this.Hide();
    }

```

```

        private void Form7_Load(object sender, EventArgs e)
        {
            textBox3.Text =
InputData.vst_initial_amount_of_money_min.ToString();
            textBox4.Text =
InputData.vst_initial_amount_of_money_max.ToString();
            textBox5.Text =
InputData.vst_process_cycle_time_min.ToString();
            textBox6.Text =
InputData.vst_process_cycle_time_max.ToString();
            textBox7.Text =
InputData.vst_markup_percentage_min.ToString();
            textBox8.Text =
InputData.vst_markup_percentage_max.ToString();
        }
    }
}

```

### Файл Form8.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;

```

```

using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Deepplom
{
    public partial class Form8 : Form
    {
        public Form8()
        {
            InitializeComponent();

            private void
textBox3_TextChanged(object sender,
EventArgs e)
        {

        }

        private void button1_Click(object
sender, EventArgs e)
        {

        }

        InputData.ttp_initial_amount_of_money_mi
n = int.Parse(textBox3.Text);

        InputData.ttp_initial_amount_of_money_ma
x = int.Parse(textBox4.Text);

        InputData.ttp_percentage_of_purchases_goo
ds = int.Parse(textBox9.Text);

        Form1 newForm = new Form1();
        newForm.Show();
        this.Hide();
    }

    private void Form8_Load(object
sender, EventArgs e)
    {
        textBox3.Text =
InputData.ttp_initial_amount_of_money_mi
n.ToString();
        textBox4.Text =
InputData.ttp_initial_amount_of_money_ma
x.ToString();
        textBox9.Text =
InputData.ttp_percentage_of_purchases_goo
ds.ToString();
    }
}

```

```

    }
}
}
Файл Form9.cs
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Deepplom
{
    public partial class Form9 : Form
    {
        public Form9()
        {
            InitializeComponent();

            private void button1_Click(object
sender, EventArgs e)
        {

        }

        InputData.money_spending_percentage =
int.Parse(textBox1.Text);

        InputData.percentage_of_spending_budget_
money = int.Parse(textBox2.Text);

        InputData.likelihood_of_an_emergency =
int.Parse(textBox3.Text);
        InputData.commercial_bank_amount
= int.Parse(textBox6.Text);

        InputData.national_Bank_total_amount =
int.Parse(textBox8.Text);

        Form1 newForm = new Form1();
        newForm.Show();
        this.Hide();
    }

    private void Form9_Load(object
sender, EventArgs e)
    {
        textBox1.Text =
InputData.money_spending_percentage.ToS
tring();
    }
}

```

```

        textBox2.Text =
InputData.percentage_of_spending_budget_
money.ToString();
        textBox3.Text =
InputData.likelihood_of_an_emergency.ToS
tring();
        textBox6.Text =
InputData.commercial_bank_amount.ToStri
ng();
        textBox8.Text =
InputData.national_Bank_total_amount.ToS
tring();
    }

```

```

        private void
textBoX8_TextChanged(object sender,
EventArgs e)
        {

        }
    }
}

```

### Файл Form10.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Deplom
{
    public partial class Form10 : Form
    {
        public Form10()
        {
            InitializeComponent();

            private void Form10_Load(object
sender, EventArgs e)
            {
                if (Economy_events.day != 0)
                {
                    dataGridView1.RowCount =
Economy_events.day;
                    dataGridView1.ColumnCount =
30;

```

```

dataGridView1.Columns[0].HeaderText =
"День";

```

```

dataGridView1.Columns[1].HeaderText =
"ВС прихід";

```

```

dataGridView1.Columns[2].HeaderText =
"ВС витрати";

```

```

dataGridView1.Columns[3].HeaderText =
"ВС сировина";

```

```

dataGridView1.Columns[4].HeaderText =
"ВС остача";

```

```

dataGridView1.Columns[5].HeaderText =
"ВЗВ прихід";

```

```

dataGridView1.Columns[6].HeaderText =
"ВЗВ витрати";

```

```

dataGridView1.Columns[7].HeaderText =
"ВЗВ сировина";

```

```

dataGridView1.Columns[8].HeaderText =
"ВЗВ остача";

```

```

dataGridView1.Columns[9].HeaderText =
"ВСТ прихід";

```

```

dataGridView1.Columns[10].HeaderText =
"ВСТ витрати";

```

```

dataGridView1.Columns[11].HeaderText =
"ВСТ сировина";

```

```

dataGridView1.Columns[12].HeaderText =
"ВСТ остача";

```

```

dataGridView1.Columns[13].HeaderText =
"ТПП прихід";

```

```

dataGridView1.Columns[14].HeaderText =
"ТПП витрати";

```

```

dataGridView1.Columns[15].HeaderText =
"ТПП сировина";

```

```

dataGridView1.Columns[16].HeaderText =
"ТПП остача";

dataGridView1.Columns[17].HeaderText =
"П прихід";

dataGridView1.Columns[18].HeaderText =
"П витрати";

dataGridView1.Columns[19].HeaderText =
"П остача";

dataGridView1.Columns[20].HeaderText =
"Б прихід";

dataGridView1.Columns[21].HeaderText =
"Б витрати";

dataGridView1.Columns[22].HeaderText =
"Б остача";

dataGridView1.Columns[23].HeaderText =
"КБ прихід";

dataGridView1.Columns[24].HeaderText =
"КБ витрати";

dataGridView1.Columns[25].HeaderText =
"КБ остача";

dataGridView1.Columns[26].HeaderText =
"НБ запас";

dataGridView1.Columns[27].HeaderText =
"НБ витрати";

dataGridView1.Columns[28].HeaderText =
"НБ емісія";

dataGridView1.Columns[29].HeaderText =
"НБ остача";
    }

    for (int i = 0; i <
Economy_events.day; i++)
    {
        for (int j = 0; j < 30; j++)

```

```

    {
        dataGridView1.Rows[i].Cells[j].Value =
Output_data.output_array[i, j];
    }
}
}
}
Файл Economy_events.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Deplom
{
    static class Economy_events//класс для
хранения возможных событий
    {
        public static int day = 0;//счетчик дней

        //итоговая строка событий
        public static string result = "
Початок моделювання\n";

        //возможные предприятия
        public static string material =
"Виробник сировини - ";
        public static string manufacturer =
"Виробник засобів виробництва - ";
        public static string consumer =
"Виробник споживчих товарів - ";
        public static string trade = "Торгівля та
Послуги - ";
        public static string budget = "Бюджет -
";
        public static string commercial =
"Комерційні банки - ";
        public static string buyers = "Покупці
споживчих товарі - ";
        public static string bank = "Нацбанк -
";

        //возможные действия
        public static string ev1 = "розрахувати
податки та зарплати та виробництво
сировини\n";//+
        public static string ev2 = "виплатити
податки та зарплати\n";//+
        public static string ev3 = "взяти
позику\n";//+
    }
}

```

```

    public static string ev4 = "підвищити
емісію\n";//+
    public static string ev5 = "ліквідувати
наслідки НС\n";//+
    public static string ev6 = "виробити
товар\n";//+
    public static string ev7 = "розрахувати
виплати\n";//+
    public static string ev8 = "купити
товар\n";//+
    public static string ev9 = "виплатити
пенсії та зарплати бюджетникам\n";//+
    public static string ev10 = "продати
товар\n";//+
    }
}

```

### Файл Input\_data.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Deplom
{
    static class InputData //клас для
хранения входных данных программы
    {
        //загальні вхідні дані
        public static int speed = 7;//Швидкість
покрокового моделювання
        public static int tax_money_percentage
= 30;//Відсоток грошей на сплату податку
        public static int pay_wages_percentage
= 25;//Відсоток грошей на виплату
заробітної плати
        public static int interest_on_loans =
25;//Відсотки за кредитами та депозитами
        public static int national_Bank_rate =
25;//Ставка Національного банку
        public static int loan_repayment_period
= 30;//Період часу повернення кредиту
        public static int
simulation_period_interval = 365;//Інтервал
періоду моделювання

        //Виробники сировини
        public static int
vs_initial_amount_of_money_min =
100;//Первісна мінімальна кількість
грошей
        public static int
vs_initial_amount_of_money_max =

```

```

900;//Первісна максимальна кількість
грошей
        public static int
vs_initial_amount_of_money =
0;//Первісна кількість грошей
        public static int
vs_process_cycle_time_min =
1;//мінімальний Період циклу
технологічного процесу
        public static int
vs_process_cycle_time_max =
10;//максимальний Період циклу
технологічного процесу
        public static int vs_process_cycle_time
= 0;//Період циклу технологічного
процесу
        public static int
vs_markup_percentage_min =
10;//мінімальний Відсоток торгової
націнки
        public static int
vs_markup_percentage_max =
20;//максимальний Відсоток торгової
націнки
        public static int vs_markup_percentage
= 0;//Відсоток торгової націнки

        //Виробники засобів виробництва
        public static int
vzv_initial_amount_of_money_min =
100;//Первісна мінімальна кількість
грошей
        public static int
vzv_initial_amount_of_money_max =
900;//Первісна максимальна кількість
грошей
        public static int
vzv_initial_amount_of_money =
0;//Первісна кількість грошей
        public static int
vzv_process_cycle_time_min =
1;//мінімальний Період циклу
технологічного процесу
        public static int
vzv_process_cycle_time_max =
10;//максимальний Період циклу
технологічного процесу
        public static int
vzv_process_cycle_time = 0;//Період циклу
технологічного процесу
        public static int
vzv_markup_percentage_min =

```

```

10;//мінімальний Відсоток торгової
націнки
    public static int
vzv_markup_percentage_max =
20;//максимальний Відсоток торгової
націнки
    public static int
vzv_markup_percentage = 0;//Відсоток
торгової націнки

    //Виробники споживчих товарів
    public static int
vst_initial_amount_of_money_min =
100;//Первісна мінімальна кількість
грошей
    public static int
vst_initial_amount_of_money_max =
900;//Первісна максимальна кількість
грошей
    public static int
vst_initial_amount_of_money =
0;//Первісна кількість грошей
    public static int
vst_process_cycle_time_min =
1;//мінімальний Період циклу
технологічного процесу
    public static int
vst_process_cycle_time_max =
10;//максимальний Період циклу
технологічного процесу
    public static int vst_process_cycle_time
= 0;//Період циклу технологічного
процесу
    public static int
vst_markup_percentage_min =
10;//мінімальний Відсоток торгової
націнки
    public static int
vst_markup_percentage_max =
20;//максимальний Відсоток торгової
націнки
    public static int vst_markup_percentage
= 0;//Відсоток торгової націнки

    //Торгівля та послуги
    public static int
ttp_initial_amount_of_money_min =
100;//Первісна мінімальна кількість
грошей
    public static int
ttp_initial_amount_of_money_max =
900;//Первісна максимальна кількість
грошей

```

```

    public static int
ttp_initial_amount_of_money =
0;//Первісна кількість грошей
    public static int
ttp_percentage_of_purchases_goods =
50;//Відсоткове співвідношення купівлі
товарів

    //Інші
    public static int
money_spending_percentage =
80;//Відсотки витрати грошей на
«Послуги та торгівлю»
    public static int
percentage_of_spending_budget_money =
20;//Відсотки витрачання бюджетних
грошей на «Послуги та торгівлю»
    public static int
likelihood_of_an_emergency =
15;//Ймовірність настання Надзвичайної
Ситуації
    public static int
commercial_bank_amount =
800;//Кількість грошей комерційного
банку
    public static int
national_Bank_total_amount =
1000;//Загальна кількість грошей у
Нацбанку
    }
}

```

### Файл Output\_data.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Deplom
{
    static class Output_data//клас для
хранения выходных данных
    {
        public static int[,] output_array = new
int[10000, 30];
        public static string filename =
"result.txt";
    }
}

```

### Файл Characteristics.cs

```

using System;
using System.Collections.Generic;
using System.Linq;

```

```

using System.Text;
using System.Threading.Tasks;

namespace Deeplom
{
    static class Characteristics//класс для
    хранения характеристик модели
    {
        //виробник сировини
        public static int material_coming =
0;//приход
        public static int material_consumption
= 0;//расход
        public static int material_raw =
0;//сырье
        public static int material_remainder =
0;//остаток

        //Виробник засобів виробництва
        public static int manufacturer_coming
= 0;//приход
        public static int
manufacturer_consumption = 0;//расход
        public static int manufacturer_raw =
0;//сырье
        public static int
manufacturer_remainder = 0;//остаток

        //Виробник споживчих товарів
        public static int consumer_coming =
0;//приход
        public static int consumer_consumption
= 0;//расход
        public static int consumer_raw =
0;//сырье
        public static int consumer_remainder =
0;//остаток

        //Торгівля та Послуги
        public static int trade_coming =
0;//приход
        public static int trade_consumption =
0;//расход
        public static int trade_raw = 0;//сырье
        public static int trade_remainder =
0;//остаток

        //Бюджет
        public static int budget_coming =
0;//приход
        public static int budget_consumption =
0;//расход

```

```

        public static int budget_remainder =
0;//остаток

        //Комерційні банки
        public static int commercial_coming =
0;//приход
        public static int
commercial_consumption = 0;//расход
        public static int commercial_remainder
= 0;//остаток

        //Покупці споживчих товарі
        public static int buyers_coming =
0;//приход
        public static int buyers_consumption =
0;//расход
        public static int buyers_remainder =
0;//остаток

        //Нацбанк
        public static int bank_stock = 0;//запас
        public static int bank_consumption =
0;//расход
        public static int bank_emission =
0;//эмиссия
        public static int bank_remainder =
0;//остаток
    }
}

```

### Файл Program.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Deeplom
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для
        приложения.
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingD
efault(false);
            Application.Run(new Form1());
        }
    }
}

```

## ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАЧА ПРОГРАМИ

Після запуску програми користувач одразу знаходиться на головній формі програми, де він може бачити процес моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі, а також переходити на форми для введення вхідних даних, а також переходити на форми для введення вхідних даних програми, перегляду результатів моделювання програми та подій, які сталися під час моделювання.

Для початку моделювання користувачу потрібно натиснути на кнопку «Моделювання» у контекстному меню програми та вибрати підпункт «Змоделювати 1 крок» або «Розпочати покрокове моделювання». Для зупинки покрокового моделювання користувач має натиснути кнопку «Моделювання» у контекстному меню програми та вибрати підпункт «Зупинити покрокове моделювання».

Якщо користувач хоче змінити загальні вхідні дані програми він повинен натиснути на кнопку «Вхідні дані» у контекстному меню програми та вибрати підпункт «Загальні вхідні дані». Якщо користувач хоче змінити вхідні дані підприємств він повинен натиснути на кнопку «Вхідні дані» у контекстному меню програми та навести курсор на підпункт «Вхідні дані підприємств». У випадяючому підменю користувач має натиснути кнопку «Виробники сировини», якщо він хоче змінити вхідні дані виробників сировини; кнопку «Виробники засобів виробництва», якщо він хоче змінити вхідні дані виробників засобів виробництва; кнопку «Виробники споживчих товарів», якщо він хоче змінити вхідні дані виробників споживчих товарів; кнопку «Торгівля та послуги», якщо він хоче змінити вхідні дані торгівлі та послуг; кнопку «Інші», якщо він хоче змінити інші вхідні дані програми.

Під час та після моделювання користувач має можливість переглянути результати моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі. Для цього користувач має натиснути кнопку «Результати» у контекстному меню програми. Після цього, якщо користувач хоче переглянути список подій, які траплялися під час моделювання, він має натиснути кнопку «Переглянути вікно подій»; якщо користувач хоче переглянути загальні результати моделювання, він має натиснути кнопку «Переглянути результати моделювання»; якщо користувач хоче записати загальні результати моделювання у файл, він має натиснути кнопку «Записати результати у файл».

Якщо користувач хоче переглянути більш детальну інформацію про програму, він повинен натиснути кнопку «Детальніше» у контекстному меню програми та вибрати пункт «Дізнатися більше про програму».

Якщо користувач перейшов на будь-яку іншу форму і хоче повернутися на головну, він має натиснути кнопку «Назад».

Для виходу із програми користувач повинен натиснути кнопку «Вихід» у контекстному меню головної форми програми.