

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 безкризової моделі»


за освітньою програмою: «Інженерія програмного забезпечення»

зі спеціальності: «121 Інженерія програмного забезпечення»

Виконав: студент групи «ПЗ1812»

Керівник:

Нормоконтролер:

  
(підпис студента)

  
(підпис)

  
(підпис)

/Олександр СЕРЕДА/

(Ім'я ПРІЗВИЩЕ)

/доц. Вадим ГОРЯЧКІН/

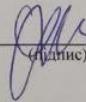
(посада, Ім'я ПРІЗВИЩЕ)

/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 a simulation program modeling of the mechanism of functioning macroeconomic system of the state on base of crisis-free model» according to educational curriculum in the Speciality: «121 Software engineering»

Done by the student of the group PZ1812:

/Oleksandr SEREDA/

Scientific Supervisor:


/Vadym HORIACHKIN/

Normative controller:

/Olena KUROIATNYK/

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  
 /Вадим ГОРЯЧКІН/  
(підпис)  
Дата 14.06.2022

**ЗАВДАННЯ**

на кваліфікаційну роботу бакалавра  
студенту Середі Олександр Андрійовичу

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

Керівник роботи: Горячкін Вадим Миколайович

затверджені наказом № 77 ст від 08.12.2021

2. Строк подання студентом роботи: 19.06.2022 р.

3. Вихідні дані до роботи: \_\_\_\_\_

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):  
збір та аналіз вимог, опис аналогів, внутрішнє проектування, зовнішнє проектування, тестування та відлагодження, висновки роботи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): Відеоролик з роботою програми, презентація з матеріалами по темі дипломної роботи, доповідь

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання теми роботи	08.12.21	
2	Збір та аналіз вимог	01.02.21	
3	Зовнішнє проектування	01.03.21	
4	Внутрішнє проектування	20.03.21	
5	Розробка програми	01.05.22	
6	Тестування та відлагодження	11.05.22	
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



Олександр СЕРЕДА

(Ім'я ПРІЗВИЩЕ)

Керівник роботи



доц. Вадим ГОРЯЧКІН

(Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Структура пояснювальної записки складається з восьми частин:

- вступ – даний розділ містить загальну інформацію про тематику виконуваної роботи. Містить одну сторінку;
- процес збору вимог до програмного забезпечення – цей розділ містить опис використаних методів збору та аналізу вимог, а також результати проведеної роботи. Складається з 2 сторінок;
- опис аналогів – у цьому розділі описуються аналоги програми, їх сильні та слабкі сторони, можливості до покращень. Містить 2 сторінок.
- зовнішнє і внутрішнє проектування – розділ містить огляд вхідних та вихідних даних, формалізацію задачі, приводиться опис об'єктно-орієнтованого програмування, проектування інтерфейсу користувача, ескізи форм, аналіз проекту, вибір мови програмування. Містить 17 сторінок;
- тестування та налагодження – містить опис вибраної стратегії тестування, приклади написаних тестів та їх результати. Містить 6 сторінок;
- висновки. Містить 1 сторінку;
- список літератури – включає в себе бібліографічний список використаної літератури. Містить 2 сторінки;
- додатки – містить керівництво користувача та код програми.

## ЗМІСТ

Вступ.....	2
1. Збір та аналіз вимог.....	3
2. Опис аналогів.....	5
2.1 SimpleModel .....	5
3. Проєктування.....	7
3.1 Зовнішнє проєктування .....	7
3.2.1 Функціональне призначення.....	7
3.2.2 Експлуатаційне призначення.....	7
3.2.3 Функціональні вимоги.....	7
3.2.4 Вхідні дані .....	10
3.2.5 Вихідні дані .....	10
3.2 Внутрішнє проєктування.....	11
3.2.1 Об'єктно-орієнтоване програмування .....	11
3.2.2 Опис відповідальності класів .....	13
3.2.3 Опис зв'язків класів.....	14
3.2.4 Діаграма класів програми .....	14
3.2.5 Опис специфікацій класів .....	15
3.3 Проєктування інтерфейсу користувача .....	19
3.4 Вибір мови програмування .....	20
4. Тестування та відлагодження.....	22
Висновки .....	27
Список використаної літератури .....	28
Додаток А.....	30
Додаток Б .....	33

## ВСТУП

Ця робота присвячена розробці програмного забезпечення[1-3], що моделює механізм функціонування макроекономічної системи на базі безкризової моделі.

Макроекономіка — це теоретична наука, що вивчає поведінку великих груп господарюючих суб'єктів і національної економіки в цілому з погляду ринкового механізму і державного регулювання.

На відміну від мікроекономіки, яка вивчає економічну поведінку окремих (індивідуальних) господарюючих суб'єктів (споживача або виробника) на індивідуальних ринках, макроекономіка вивчає економіку як єдине ціле, досліджує проблеми, які загальні для всієї економіки, і оперує такими агрегованими величинами, як валовий внутрішній продукт, національний дохід, сукупний попит, сукупна пропозиція, сукупне споживання, інвестиції, загальний рівень цін, рівень безробіття, державний борг тощо.

Основними проблемами, які досліджує макроекономіка, є: економічне зростання і його темпи; економічний цикл і його причини; рівень зайнятості та проблема безробіття; загальний рівень цін і проблема інфляції; рівень процентної ставки і проблеми грошового обігу; стан державного бюджету, проблеми фінансування бюджетного дефіциту і державного боргу; стан платіжного балансу і проблеми валютного курсу; проблеми макроекономічної політики.

Ця робота базується на циклі наукових праць, спрямованих на створення безкризової(високоєфективної) макроекономічної системи, яка докорінно відрізняється від того, що пропонує сучасна економічна теорія, тому що вона функціонує на основі зовсім інших принципів, головним з яких є відмова від прибутку. Тому для глибшого розуміння її функціонування, цю роботу необхідно розглядати не ізольовано, а в контексті з іншими роботами на цю тему [4-14].

Отже виникає необхідність розробити відповідне програмне забезпечення, яке зможе продемонструвати сутність макроекономічної системи на основі безкризової ринкової моделі. Розроблювана програма зможе змодельовати цю систему і показати розрахунки, на основі яких проводилось моделювання.

## 1. ЗБІР ТА АНАЛІЗ ВИМОГ

Коли розробляється нове програмне забезпечення, існують ризики отримання результату, що не відповідає меті технічного завдання. Для усунення таких ризиків необхідно звернути увагу на вимоги до програмного забезпечення (ПЗ) [15–16]. У даному розділі розглянуті питання збору та аналізу вимог до програмного забезпечення для імітаційного моделювання механізму функціонування макроекономічної системи держави на базі безкризової моделі.

Розроблювана програма імітаційного моделювання механізму функціонування макроекономічної системи держави на базі безкризової моделі відзначається тим, що уже існує програмне забезпечення на дану тематику, тобто воно може використовуватися як частина збору на аналізу вимог. Разом з цим для виначення вимог використовувалася така комбінація методів збору та аналізу вимог як метод діалогу та метод прототипування.

При створенні програми були використані такі методи збору та аналізу вимог як метод діалогу та метод еволюційного прототипування. Метод діалогу дозволяє швидко та надійно визначати необхідні вимоги та бізнес правила, а метод еволюційного прототипування допомагає розроблювати програмне забезпечення покроково, не зупиняючись на переписування ужеготової частини програми. Комбінація методів діалогу та еволюційного прототипування зробило розробку програми більш зручною та ефективною з точки зору визначення за змін у вимогах.

На основі уже існуючої програми було розроблено прототип дизайну, узгоджено із замовником, визначено у замовника наступні задачі для наступного прототипу, знову узгодження розробленої частини, і так вівся зручний та нечутливий до змін збір вимог та розробка програмного забезпечення.

Для розроблюваної програми був розроблений бізнес процес, який описує бізнес частину роботи програми з боку користувача. Він виконується наступним чином:

1. Користувач відкриває програму
2. Якщо потрібно змінює вхідні дані(за замовчуванням вхідні дані запрограмовані)



3. Користувач моделює макроекономічну систему покроково або(змінює вхідні дані якщо є необхідність між кроками)
4. Користувач продовжує автоматичне моделювання
5. Програма виводить на екран результати моделювання у виді намальованих схем
6. Програма записує дані усіх розрахунків після моделювання до файлу
7. Користувач закриває програму

Бізнес процес можна вважати завершеним, якщо користувач отримав усю необхідну інформацію про змодельовану систему у вигляді діаграм та у файлі.

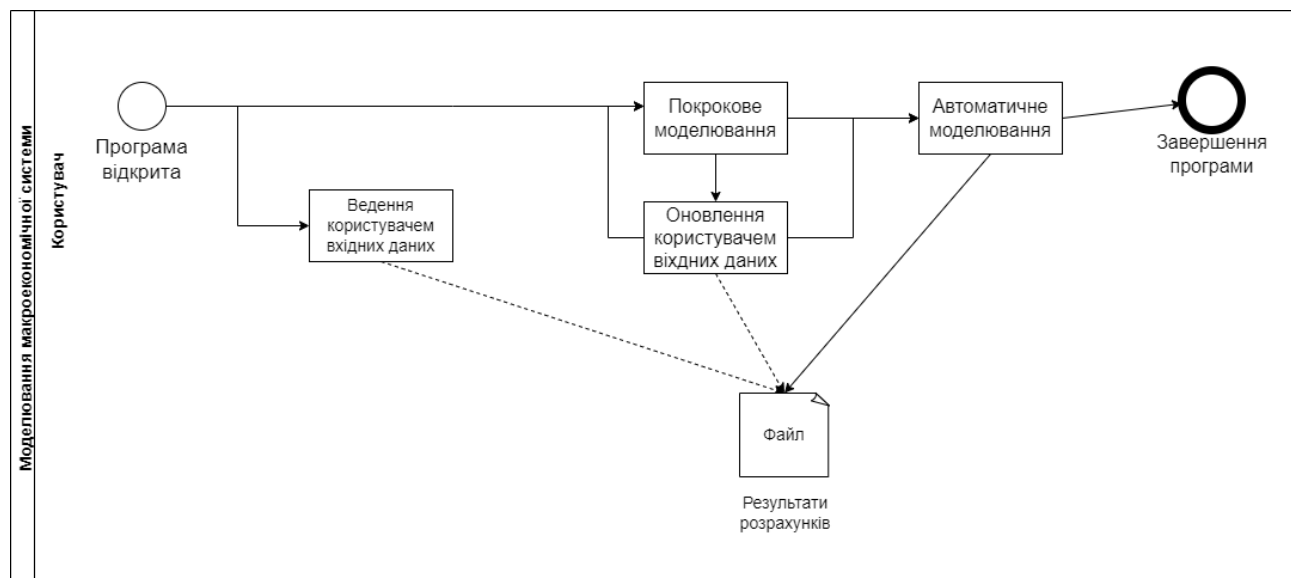


Рисунок 1.1 – Бізнес процес моделювання макроекономічної системи на основі безкризової моделі

### Висновки до збору та аналізу вимог

У ході виконання роботи та розробки програмного забезпечення збір та аналіз вимог відбувався протягом усього процесу. Спочатку був розроблений бізнес процес за допомогою методу діалогу, а потім розроблялася програма за допомогою методу еволюційного прототипування.

## 2. ОПИС АНАЛОГІВ

### 2.1 SimpleModel

SimpleModel – єдина версія програми, яка виконує імітаційне моделювання механізму функціонування макроекономічної системи держави на базі безкризової моделі. Дане програмне забезпечення має досить простий інтерфейс, але є певні проблеми з відображенням тексту на нових версіях систем Windows.

Також передбачалось, що програма буде підтримувати автоматичне та покрокове моделювання з можливістю для користувача вводити та оновлювати вхідні дані перед кожним кроком, представле програмне забезпечення цього не виконує.

Ще один недолік це неможливість переглянути розрахунки після моделювання, так як програма не записує дані у файл.

Приклади роботи програми:

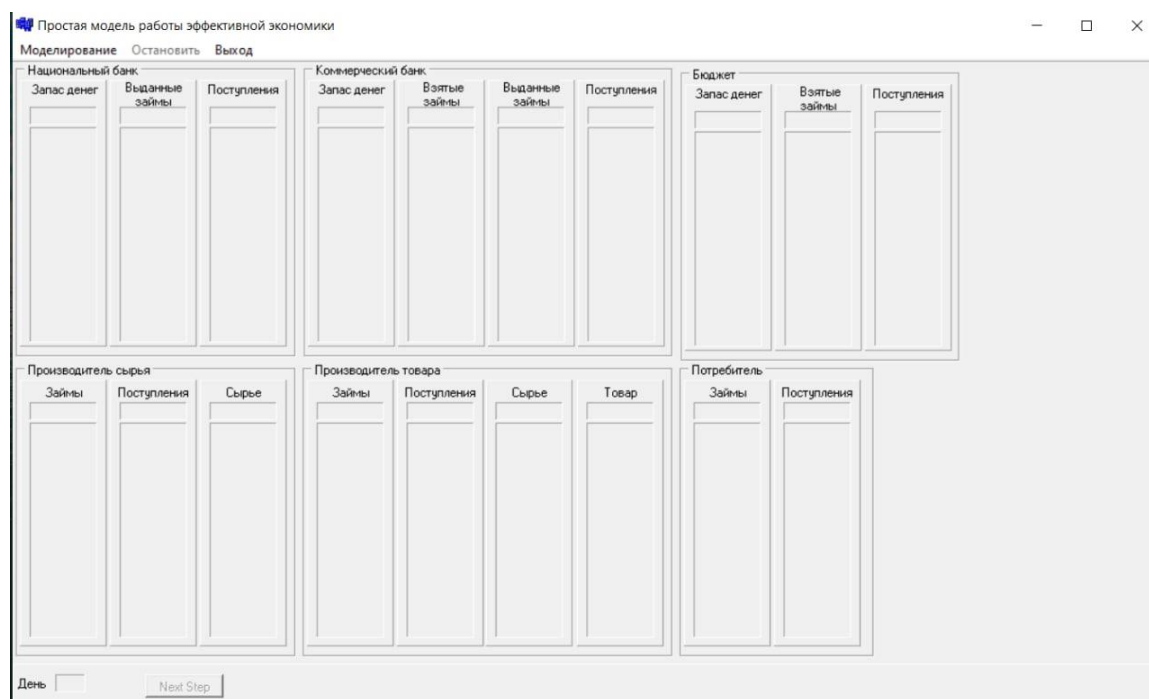


Рисунок 2.1 – Початковий вигляд програми після запуску

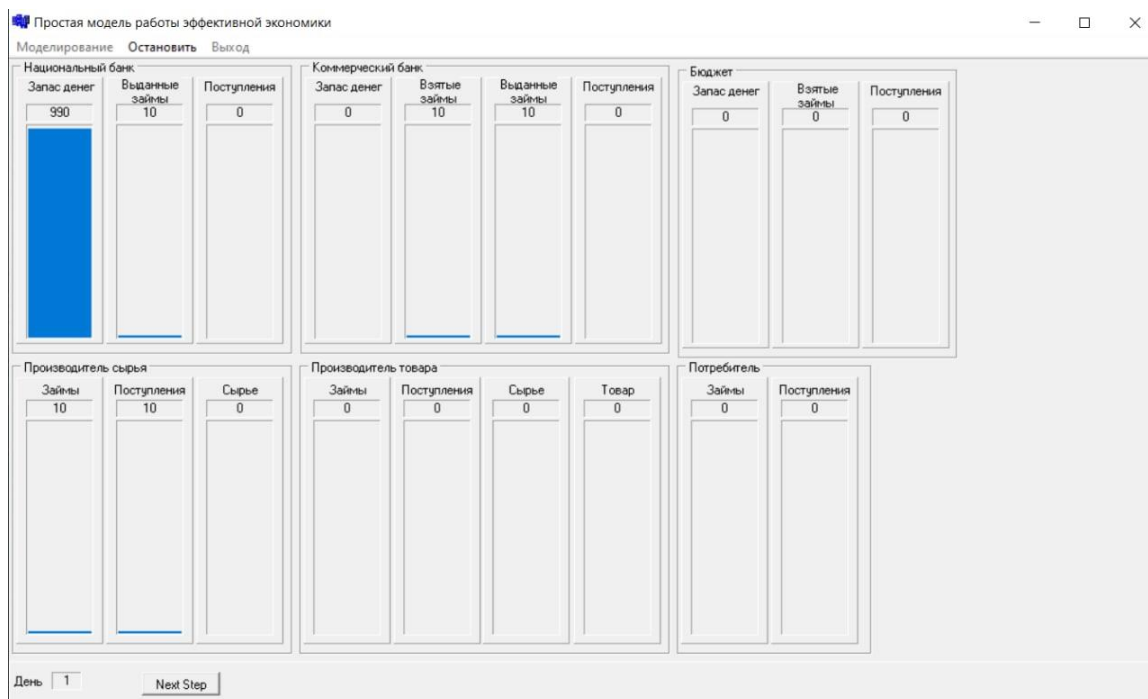


Рисунок 2.2 – Вигляд програми після початку моделювання

За результатами роботи програми можна сказати, що вона частково виконує свою функцію, так як через некорктне відображення тексту неможливо зрозуміти за що відповідає відповідний розрахунок.

Також програма дозволяє виконувати моделювання тільки покроково і не має обмежень у кількості кроків(повинно бути обмеження у 365 днів – 1 рік).

### 3. ПРОЄКТУВАННЯ

#### 3.1 Зовнішнє проєктування

##### 3.2.1 Функціональне призначення

Розроблюваний програмний продукт створює імітаційну модель макроекономічної системи на основі безкризової (ефективної) моделі.

Практичним завданням програми є ціль продемонструвати розрахунки макроекономічної системи у вигляді діаграм та побачити результати виконуваних дій. Ефективна (безкризова) економіка докорінно відрізняється від того, що пропонує сучасна економічна теорія, оскільки вона функціонує на основі зовсім інших принципів, головним у тому числі є відсутність прибутку.

##### 3.2.2 Експлуатаційне призначення

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

Програмне забезпечення, яке розробляється, допомагає побачити основу роботи макроекономічної системи на основі безкризової моделі.

##### 3.2.3 Функціональні вимоги

Завданням даного імітаційного моделювання є визначення правильних принципів функціонування макроекономічної системи (Економіка окремої держави).

Для спрощення розрахунків було прийнято агрегований спосіб моделювання. 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).

Основним принципом ефективної економіки – відсутність прибутку у ціні товару, приналежність грошей тільки державі і безвідсоткові кредити (займи). В моделі, яка описується розглядаються 8 суб'єктів економічної діяльності:

1. Національний банк – є зберігачем усіх грошових коштів держави. Передає гроші комерційному банку на видачу позик суб'єктам економічної діяльності та бюджету на ліквідацію наслідків

надзвичайних ситуацій на безвідсотковій основі. У разі потреби здійснює емісію грошей.

2. Комерційний банк – видає позики суб'єктам економічної діяльності (в даній моделі всім, крім виробника сировини та виробника товару 3). Гроші видаються під відсоток (у даної моделі прийнято 1%).

3. Бюджет – фінансує пенсійний фонд, працівників бюджетної сфери, розвиток соціальної інфраструктури, закупівлі для державних підприємств та ліквідацію наслідків надзвичайних ситуацій.

4. Споживачі товарів – ними є населення.

5. Виробник сировини – галузь промисловості, зайнята в виробництві сировини та матеріалів для інших галузей промисловості.

6. Виробник товару 1 – виробники товарів споживання, до моменту початку моделювання вже мають основні фонди виробництва.

7. Виробник товару 2 – новостворені виробництва споживчих товарів за рахунок розширення виробничої бази виробника товарів 1.

8. Виробник товару 3 – виробник, який працює не на споживача, а в бюджет.

Основні парадигми моделі:

1. Одиницею вимірювання часу моделі прийнятий день.

2. За 1 день кожен суб'єкт економічної діяльності виконує як мінімум 1 дію.

3. Порядок моделювання роботи суб'єктів економічної діяльності наступний: виробник сировини, виробник товару 1, виробник товару 2, виробник товару 3, комерційний банк, бюджет, споживач.

4. Вартість товарів визначається так:

Виробник сировини – випадково з діапазону 1 – 100 одиниць.

Виробник товарів 1 – виходячи із вартості сировини помноженої на випадковий коефіцієнт діапазону 1,1 – 2.

Виробник товару 2 – див. п.6.

Виробник товару 3 працює на бюджет та вартість його товарів визначається наявними у розпорядженні бюджету коштами на розвиток соціальної інфраструктури та витратами на ліквідацію наслідків надзвичайних ситуацій (див. п.7).

5. Кошти, отримані суб'єктами економічної діяльності за реалізацію своїх товарів, розподіляються так:

Виробник сировини та Виробник товару 1 та 3 та Комерційний банк: податки – 37,5%, зарплата – 62,5%. Виробник товару 2: податки – 37,5%, зарплата якщо є заборгованість за взятими позиками – 53,125%,

погашення заборгованості із позик – 9,375% з урахуванням процентної ставки Комерційного банку (1%), якщо заборгованості немає – 62,5%.

#### 6. Виробник товару 2

Один раз на 9 днів визначає необхідність розширення виробництва. Ймовірність розширення виробництва – 0,6. У разі виникнення необхідності у розвитку виробництва бере позику. Сума кожної позики визначається випадковим чином діапазону 1 – 21 одиниць. Період випуску товарів – 3 дні. Вартість товарів визначається випадковим чином із діапазону 1 – 3% від суми раніше взятих позик.

#### 7. Надходження до бюджету розподіляються так

Пенсійний фонд – 50%. Оплата праці бюджетників – 12,5%. Розвиток соціальної інфраструктури – 35%. Кошти, що накопичуються на витрати на усунення наслідків надзвичайних ситуацій – 2,5%.

#### 8. Споживач товарів

Придбає товари за рахунок власних коштів та за рахунок взятих безвідсоткових позик. Власні кошти Споживача = Надходження до пенсійного фонду + Зарплата працівників виробничої сфери (Виробник сировини, Виробник товарів 1 – 3) + Зарплата працівників бюджетної сфери – Вартість раніше придбаних товарів - Вартість раніше виплачених заборгованостей по позикам. Виплачує заборгованості за взятими позиками лише з власних коштів з урахуванням процентної ставки комерційного банку (1%).

#### 9. Час виробництва 1-ї одиниці товару

Виробник сировини, Виробник товару 1 та 3 – 1 день. Виробник товару 2 – 3 дні.

10. Облік надзвичайних ситуацій провадиться по черзі через 3 – 4 дня (спочатку 3 дні, потім 4 дні, потім знову 3 дні тощо). Ймовірність настання надзвичайної ситуації при кожному випробуванні – 0,20. Якщо подія настала, то необхідно визначити категорію надзвичайної ситуації. Можливі п'ять категорій НС:

I – повінь (ймовірність – 0,40);

II – пожежа (ймовірність – 0,25);

III – транспортна катастрофа (ймовірність – 0,10);

IV - руйнування будівлі (ймовірність - 0,15);

V – інше (ймовірність – 0,10).

Потреба фінансових коштів на ліквідації різних категорій надзвичайних ситуацій визначається випадковим чином із заданого інтервалу можливих значень:

I – повінь (0,1 – 0,3);

II – пожежа (0,1 – 0,5);

III – транспортна катастрофа (0,1 – 5,1);

IV – руйнування будівлі (1,0 - 1,4);

V - інше (1-7).

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 АСЕД до іншого та робити одну операцію згідно загальним вхідним даним.

#### 3.2.4 Вхідні дані

Програма має наступні вхідні дані:

1. Задається відсоток грошей на сплату податку, %
2. Задається відсоток грошей, які спрямовуються на виплату
3. заробітної плати, %
4. Задаються відсотки за кредитами та депозитами, %
5. Задається ставка Національного банку, %
6. Задається період повернення кредиту
7. Задається інтервал періоду моделювання на днях або
8. Вказується – 1 рік, 2 роки.
9. Задається період часу, через який виплачується
10. Зарплата та податок

#### 3.2.5 Вихідні дані

Вихідними даними є намальовані у програмі діаграми за результатами моделювання, а також записані у файл усі розрахунки, які відбувалися при виконанні моделювання програми.

## 3.2 Внутрішнє проектування

### 3.2.1 Об'єктно-орієнтоване програмування

Об'єктно-орієнтоване програмування (ООП) — це модель комп'ютерного програмування, яка організовує проектування програмного забезпечення навколо даних або об'єктів, а не функцій і логіки. Об'єкт можна визначити як поле даних, яке має унікальні атрибути та поведінку.

ООП зосереджується на об'єктах, якими розробники хочуть маніпулювати, а не на логіці, необхідній для маніпулювання ними. Такий підхід до програмування добре підходить для великих, складних програм, які активно оновлюються або обслуговуються. Сюди входять програми для виробництва та дизайну, а також мобільні додатки; наприклад, ООП можна використовувати для виробництва програмного забезпечення для моделювання системи.

Організація об'єктно-орієнтованої програми також робить метод корисним для спільної розробки, коли проекти поділяються на групи. Додаткові переваги ООП включають можливість повторного використання коду, масштабованість та ефективність.

Перший крок в ООП — це зібрати всі об'єкти, якими програміст хоче маніпулювати, і визначити, як вони пов'язані один з одним — вправа, відома як моделювання даних.

Приклади об'єкта можуть варіюватися від фізичних сутностей, таких як людина, яка описується такими властивостями, як ім'я та адреса, до невеликих комп'ютерних програм, таких як віджети.

Як тільки об'єкт відомий, він позначається класом об'єктів, який визначає тип даних, які він містить, і будь-які логічні послідовності, які можуть ним маніпулювати. Кожна окрема логічна послідовність відома як метод. Об'єкти можуть спілкуватися за допомогою чітко визначених інтерфейсів, які називаються повідомленнями.

Структура або будівельні блоки об'єктно-орієнтованого програмування включають наступне:

Класи — це визначені користувачем типи даних, які діють як план окремих об'єктів, атрибутів і методів.



Об'єкти — це екземпляри класу, створеного зі спеціально визначеними даними. Об'єкти можуть відповідати об'єктам реального світу або абстрактній сутності. Коли клас визначено спочатку, опис є єдиним об'єктом, який визначається.

Методи — це функції, визначені всередині класу, що описують поведінку об'єкта. Кожен метод, що міститься у визначеннях класів, починається з посилання на об'єкт екземпляра. Крім того, підпрограми, що містяться в об'єкті, називаються методами екземпляра. Програмісти використовують методи для повторного використання або збереження функціональності, інкапсульованої всередині одного об'єкта за раз.

Атрибути визначаються в шаблоні класу і представляють стан об'єкта. Дані об'єктів зберігатимуться в полі атрибутів. Атрибути класу належать самому класу.

Об'єктно-орієнтоване програмування базується на наступних принципах:

Інкапсуляція. Цей принцип стверджує, що вся важлива інформація міститься всередині об'єкта, а доступна лише вибрана інформація. Реалізація та стан кожного об'єкта приватно зберігаються всередині визначеного класу. Інші об'єкти не мають доступу до цього класу або повноважень на внесення змін. Вони можуть викликати лише список відкритих функцій або методів.

Абстракція. Об'єкти розкривають лише внутрішні механізми, які є релевантними для використання інших об'єктів, приховуючи будь-який непотрібний код реалізації. Функціональність похідного класу може бути розширена. Ця концепція може допомогти розробникам легше вносити додаткові зміни або доповнення з часом.

Спадкування. Класи можуть повторно використовувати код інших класів. Між об'єктами можна призначати зв'язки та підкласи, що дозволяє розробникам повторно використовувати загальну логіку, зберігаючи при цьому унікальну ієрархію. Ця властивість ООП змушує проводити більш ретельний аналіз даних, скорочує час розробки та забезпечує більш високий рівень точності.

Поліморфізм. Об'єкти створені для спільної поведінки, і вони можуть приймати більше ніж одну форму. Програма визначатиме, яке значення або використання необхідно для кожного виконання цього об'єкта з батьківського класу, зменшуючи необхідність дублювання

коду. Потім створюється дочірній клас, який розширює функціональні можливості батьківського класу. Поліморфізм дозволяє різним типам об'єктів проходити через один інтерфейс.

### 3.2.2 Опис відповідальності класів

Клас Program являє собою точку входу у програму.

Клас ModelingForm відповідає за відображення графіків моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі.

Клас InputDataForm відповідає за відображення інформації про загальні вхідні дані програми для моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі.

Клас ProgramInfo відповідає за відображення довідкової інформації про програмне забезпечення.

Клас EventsForm відповідає за відображення подій, які відбуваються протягом процесу моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі.

Клас InputData відповідає за зберігання вхідних даних програми.

Клас Characteristics відповідає за зберігання характеристик макроекономічної системи держави на базі безкризової(ефективної) моделі.

Клас EconomyEvents відповідає за зберігання можливих подій під час моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі.

Клас RawProducerInputDataForm відповідає за відображення інформації про вхідні дані виробників сировини програми для моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі.

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

Клас ConsumerGoodsProducerInputDataForm відповідає за відображення інформації про вхідні дані виробників споживчих товарів програми для моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі.

Клас `TradeAndServicesInputDataForm` відповідає за відображення інформації про вхідні дані торгівлі та послуг програми для моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі.

Клас `GeneralInputDataForm` відповідає за відображення інформації про інші вхідні дані програми для моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі.

### 3.2.3 Опис зв'язків класів

Описані вище класи мають наступні зв'язки.

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

Об'єкти сутності класу `ModelingForm` пов'язані з об'єктами класів `InputDataForm`, `ProgramInfo`, `EventsForm`, `RawProducerInputDataForm`, `CapitalGoodsProducerInputDataForm`, `ConsumerGoodsProducerInputDataForm`, `TradeAndServicesInputDataForm`, `GeneralInputDataForm`, `Characteristics`. Тип зв'язку – асоціація.

Об'єкти сутності класу `InputDataForm`, `RawProducerInputDataForm`, `CapitalGoodsProducerInputDataForm`, `ConsumerGoodsProducerInputDataForm`, `TradeAndServicesInputDataForm`, `GeneralInputDataForm` пов'язані з об'єктами класу `InputData`. Тип зв'язку – асоціація.

Об'єкти сутності класу `EventsForm` пов'язані з об'єктами класу `EconomyEvents`. Тип зв'язку – асоціація.

### 3.2.4 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.

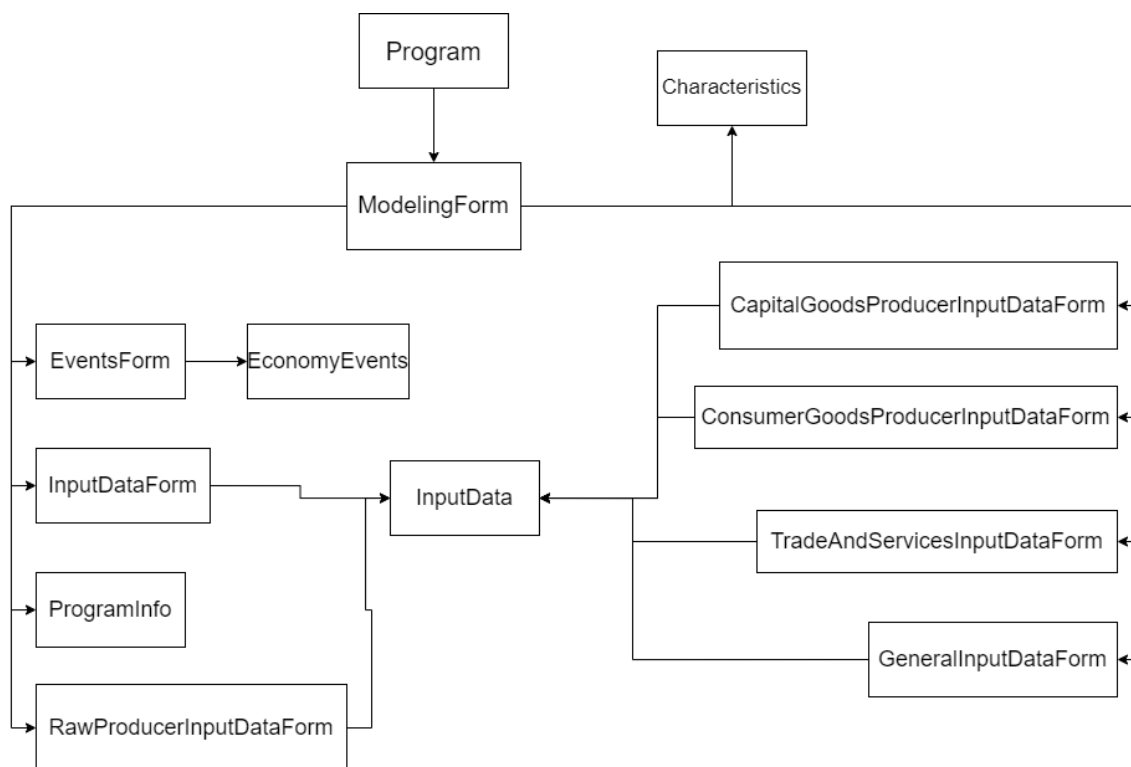


Рисунок 5.1 – Діаграма класів програми

### 3.2.5 Опис специфікацій класів

#### 3.2.5.1 Опис специфікацій класу Program

Клас відповідає за запуск програми.

#### 3.2.5.2 Опис специфікацій класу ModelingForm

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.

### 3.2.5.3 Опис специфікацій класу InputDataForm

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 безкризової(ефективної) моделі.

### 3.2.5.4 Опис специфікацій класу ProgramInfo

Клас реалізовує наступні операції:

- відображення інформації про саму програму;
- перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі.

### 3.2.5.5 Опис специфікацій класу EventsForm

Клас реалізовує наступні операції:

- відображення інформації про події протягом моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі;
- оновлення інформації про події протягом моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі;
- перехід на форму для моделювання механізму функціонування макроекономічної системи держави на базі безкризової(ефективної) моделі.

### 3.2.5.6 Опис специфікацій класу RawProducerInputDataForm

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 безкризової(ефективної) моделі.

### 3.2.5.7 Опис специфікацій класу CapitalGoodsProducerInputDataForm

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 безкризової(ефективної) моделі.

### 3.2.5.8 Опис специфікацій класу ConsumerGoodsProducerInputDataForm

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 безкризової(ефективної) моделі.

### 3.2.5.9 Опис специфікацій класу TradeAndServicesInputDataForm

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 безкризової(ефективної) моделі.

### 3.2.5.10 Опис специфікацій класу GeneralInputDataForm

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.

### 3.2.5.11 Опис специфікацій класу InputData

Атрибути класу:

- швидкість покрокового моделювання;
- відсоток грошей на сплату податку;
- відсоток грошей на виплату заробітної плати;
- відсотки за кредитами та депозитами;
- ставка Національного банку;
- час повернення кредиту;
- інтервал періоду моделювання;
- первісна мінімальна кількість грошей виробників сировини;
- первісна максимальна кількість грошей виробників сировини;
- мінімальний період циклу технологічного процесу виробників сировини;
- максимальний період циклу технологічного процесу виробників сировини;
- мінімальний відсоток торгової націнки виробників сировини;
- максимальний відсоток торгової націнки виробників сировини;
- первісна мінімальну кількість грошей виробників засобів виробництва;
- первісна максимальна кількість грошей виробників засобів виробництва;

- мінімальний період циклу технологічного процесу виробників засобів виробництва;
- максимальний період циклу технологічного процесу виробників засобів виробництва;
- мінімальний відсоток торгової націнки виробників засобів виробництва;
- максимальний відсоток торгової націнки виробників засобів виробництва;
- первісну мінімальну кількість грошей виробників споживчих товарів;
- первісна максимальна кількість грошей виробників споживчих товарів;
- мінімальний період циклу технологічного процесу виробників споживчих товарів;
- максимальний період циклу технологічного процесу виробників споживчих товарів;
- мінімальний відсоток торгової націнки виробників споживчих товарів;
- 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» ;
- ймовірність настання Надзвичайної Ситуації;
- кількість грошей комерційного банку;
- загальна кількість грошей у Нацбанку.

### 3.2.5.12 Опис специфікацій класу Characteristics

Атрибути класу:

- прихід виробників сировини;
- витрати виробників сировини;
- сировина виробників сировини;
- остача виробників сировини;
- прихід виробників засобів виробництва;
- витрати виробників засобів виробництва;
- сировина виробників засобів виробництва;
- остача виробників засобів виробництва;
- прихід виробників споживчих товарів;
- витрати виробників споживчих товарів;
- сировина виробників споживчих товарів;
- остача виробників споживчих товарів;
- за прихід торгівлі та послуг;
- витрати торгівлі та послуг;
- сировина торгівлі та послуг;

- остача торгівлі та послуг;
- прихід бюджету;
- витрати бюджету;
- остача бюджету;
- прихід комерційних банків;
- витрати комерційних банків;
- остача комерційних банків;
- прихід покупців споживчих товарів;
- витрати покупців споживчих товарів;
- остача покупців споживчих товарів;
- запас Нацбанку;
- витрати Нацбанку;
- емісія Нацбанку;
- остача Нацбанку.

### 3.2.5.13 Опис специфікацій класу EconomyEvents

Атрибути класу:

- лічильник днів;
- підсумковий рядок подій;
- можливі підприємства;
- можливі дії.

## 3.3 Проектування інтерфейсу користувача

Інтерфейс користувача (UI) – це точка, на якій люди взаємодіють з комп'ютером, веб-сайтом або програмою. Метою ефективного інтерфейсу є зробити роботу користувача простим та інтуїтивно зрозумілим, вимагаючи мінімальних зусиль з боку користувача для отримання максимального бажаного результату.

Інтерфейс користувача створюється у вигляді шарів взаємодії, які звертаються до органів чуття людини (зір, дотик, слух тощо). Вони включають обидва пристрої введення, такі як клавіатура, миша, трекпад, мікрофон, сенсорний екран, сканер відбитків пальців, електронне перо та камера, а також пристрої виведення, такі як монітори, колонки та принтери. Пристрої, які взаємодіють з кількома органами чуття, називаються «мультимедійними інтерфейсами користувача». Наприклад, повсякденний інтерфейс використовує комбінацію тактильного введення (клавіатура та миша) та візуального та слухового виходу (монітор і динаміки).

Інтерфейс користувача важливий для того, щоб відповідати очікуванням користувачів і підтримувати ефективну функціональність



вашого сайту. Добре виконаний користувальницький інтерфейс сприяє ефективній взаємодії між користувачем і програмою, додатком або машиною завдяки контрастним візуальним характеристикам, чистому дизайну та швидкості реагування. Розробляючи інтерфейс користувача для свого сайту, важливо враховувати очікування користувачів щодо доступності, візуальної естетики та простоти використання.

Оптимальне поєднання ефективних візуальних ефектів та ефективної швидкості реагування покращить коефіцієнт конверсії вашого сайту, оскільки воно передбачає потреби користувача, а потім задовольняє ці потреби.

У ході розробки програми було спроектовано простий та зручний інтерфейс на основі запропонованого аналогу. Програма має притаманні усім проектам, які розроблені на фреймворку Windows Forms графічні елементи такі як MenuStrip(Меню та елементи меню), Button(кнопка), TextBox(поле для вводу тексту) та інші.

### 3.4 Вибір мови програмування

Серед існуючих мов програмування є не дуже багато таких, які мають можливість створювати програмне забезпечення для десктопних комп'ютерів з гарним та функціональним інтерфейсом. Наприклад C++, Java, C#, Python.

C# - це об'єктно-орієнтована мова програмування, створена Microsoft, яка працює на .NET Framework. C# [17-18] має коріння з сімейства C, і мова близька до інших популярних мов, таких як C++ і Java. Мова програмування C# використовується для створення мобільних додатків, настільних програм, веб-додатків, веб-сервісів, веб-сайтів, ігор, програм для баз даних.

Ця була вибрана для розробки цього проекту так як має дуже зручний та простий для розробки фреймворк Windows Forms, що дозволяє швидко та надійно створювати програмне забезпечення з гарним користувацьким інтерфейсом.

### **Висновки до проєктування**

У ході виконання роботи було виконано внутрішнє за зовнішнє проєктування, визначено функціональні та експлуатаційні вимоги, вхідні та вихідні дані, описано програму з точки зору об'єктно орієнтованого програмування, спроектовано інтерфейс користувача та вибрано мову програмування.



#### 4. ТЕСТУВАННЯ ТА ВІДЛАГОДЖЕННЯ

Як і в будь-якій формі людської поведінки, у тестуванні програмного забезпечення переважає ставлення тестувальника до тестової діяльності. Таке ставлення до тестування можна узагальнити у формулюванні основних цілей тестування. Деякі з переважних цілей тестування зазначені нижче:

1. Продемонструвати, що помилок немає
2. Показати наявність призначених функцій
3. Бути впевненим у здатності програмного забезпечення робити те, що від нього вимагається

Однією з проблем цих цілей є те, що практично неможливо видалити всі помилки в нетривіальній програмі. Тому цілі для початку нереальні. Інша проблема полягає в тому, що, хоча програма може виконувати всі призначені їй функції, вона все одно може містити помилки, оскільки вона також виконує непередбачувані функції.

Набагато продуктивнішою метою тестування є мета дізнатися, які помилки присутні в програмному забезпеченні.

Ця мета передбачає, що в програмному забезпеченні присутні помилки, припущення, яке вірно практично для всього програмного забезпечення, і яке демонструє набагато більш продуктивне ставлення до тестування програмного забезпечення, тобто максимального навантаження на програмне забезпечення з метою пошуку помилок. Оскільки ця мета набагато більш сприятлива для пошуку помилок, вона також з більшою ймовірністю підвищить надійність програмного забезпечення.

Один із способів виявлення переважаючого ставлення до тестування — побачити, як слова «успішно» і «неуспішно» використовуються при описі результатів тестування. Якщо тестовий випадок, який не виявляє помилок, вважається успішним, це свідчить про те, що існує непродуктивне ставлення. Такий тестовий приклад не додає надійності програмному забезпеченню і, отже, є марною тратою часу та енергії. Успішний тестовий приклад має бути таким, який виявляє помилки. Насправді, чим більше виявлено помилок, тим краще тестовий приклад.

Два альтернативних і додаткових підходи до тестування називаються тестуванням за чорним ящиком і тестуванням за білим

ящиком. Тестування чорного ящика також називається тестуванням на основі даних (або на основі введення/виводу). Використовуючи цей підхід, тестувальник розглядає програму як чорний ящик і не турбується про внутрішню поведінку та структуру програми. Тестер зацікавлений лише в тому, щоб знайти обставини, за яких програма веде себе не відповідно до своїх специфікацій. Тестові дані отримують виключно зі специфікацій (тобто, не використовуючи знання внутрішньої структури програми).

Якщо необхідно знайти всі помилки в програмі, використовуючи цей підхід, критерієм є вичерпне тестування вхідних даних. Вичерпне тестування вхідних даних — це використання всіх можливих умов введення як тестового прикладу. Оскільки це зазвичай неможливо або непрактично з економічної точки зору, вичерпне тестування вхідних даних використовується рідко. Щоб максимізувати прибуток від інвестицій у тестування (тобто максимізувати кількість помилок, знайдених кінцевою кількістю тестових випадків), також використовується підхід білого ящика.

Інший підхід до тестування, білий ящик або структурне тестування на основі логіки, дозволяє вивчити внутрішню структуру програми. Використовуючи цю стратегію, тестувальник отримує дані тесту з огляду логіки та структури програми.

Аналог до вичерпного тестування вхідних даних підходу чорного ящика зазвичай вважається вичерпним тестуванням шляху виконання програми. Тобто, якщо виконати (через тестові випадки) всі можливі шляхи потоку керування програми, то, можливо, можна сказати, що програма повністю перевірена.

Після завершення розробки програми, було проведено тестування методом чорної скриньки за допомогою юніт тестів.

### **Специфікація та текст модулів**

Опис специфікації та текст методу RefreshCharts

Атрибути методу:

- відсоток грошей на сплату податку;
- відсоток грошей на виплату заробітної плати;
- відсоток за кредитами та депозитами;
- валютна ставка Національного банку;
- період часу повернення кредиту;

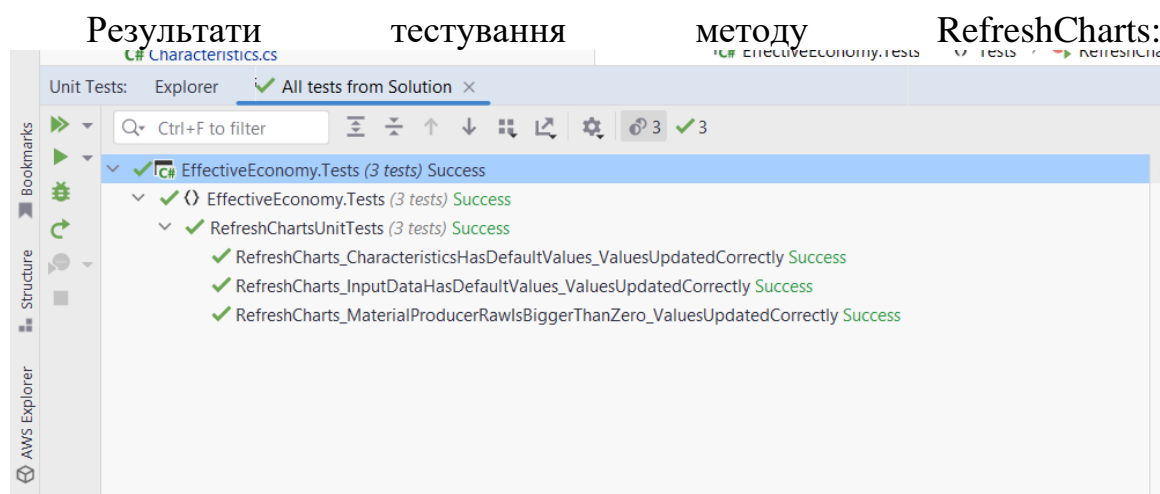
- інтервал періоду моделювання;
- первісна мінімальна кількість грошей виробників сировини;
- первісна максимальна кількість грошей виробників сировини;
- мінімальний період циклу технологічного процесу виробників сировини;
- максимальний період циклу технологічного процесу виробників сировини;
- мінімальний відсоток торгової націнки виробників сировини;
- максимальний відсоток торгової націнки виробників сировини;
- первісна мінімальна кількість грошей виробників засобів виробництва;
- первісна максимальна кількість грошей виробників засобів виробництва;
- мінімальний період циклу технологічного процесу виробників засобів виробництва;
- максимальний період циклу технологічного процесу виробників засобів виробництва;
- мінімальний відсоток торгової націнки виробників засобів виробництва;
- максимальний відсоток торгової націнки виробників засобів виробництва;
- первісна мінімальна кількість грошей виробників споживчих товарів;
- первісна максимальну кількість грошей виробників споживчих товарів;
- мінімальний період циклу технологічного процесу виробників споживчих товарів;
- максимальний період циклу технологічного процесу виробників споживчих товарів;
- мінімальний відсоток торгової націнки виробників споживчих товарів;
- 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;
- відсоток витрати грошей на «Послуги та торгівлю»;

- відсоткок витрачання бюджетних грошей на «Послуги та торгівлю»;

- кількість грошей комерційного банку;
- загальну кількість грошей у Нацбанку.
- кількість приходу виробників сировини;
- кількість витрати виробників сировини;
- кількість сировини виробників сировини;
- остача виробників сировини;
- прихід виробників засобів виробництва;
- витрати виробників засобів виробництва;
- сировина виробників засобів виробництва;
- остача виробників засобів виробництва;
- прихід виробників споживчих товарів;
- витрати виробників споживчих товарів;
- сировина виробників споживчих товарів;
- остача виробників споживчих товарів;
- прихід торгівлі та послуг;
- витрати торгівлі та послуг;
- сировина торгівлі та послуг;
- остача торгівлі та послуг;
- прихід бюджету;
- витрати бюджету;
- остача бюджету;
- прихід комерційних банків;
- витрати комерційних банків;
- остача комерційних банків;
- покупців споживчих товарів;
- витрати покупців споживчих товарів;
- остача покупців споживчих товарів;
- грошовий запас Нацбанку;
- витрати Нацбанку;
- емісія Нацбанку;
- остача Нацбанку;
- підсумковий рядок подій;
- масив назв підприємств;

- масив можливих дій.

Метод проводить розрахунки по моделюванню механізму функціонування макроекономічної системи держави на базі безкризової моделі, та на основі цих розрахунків малюються графіки на екрані користувача.



Таблиця 4.1 Протокол налагодження програми

Опис помилок	Опис ситуації	Способи усунення	Дії, що були застосовані для усунення
Деякі змінні не оновлюються програмою	Деякі змінні не оновлюються програмою, через що графіки малюються некоректно	Перевірити залучення усіх необхідних змінних до процесу розрахунків	Перевірено метод, у якому повинно оновлюватись відповідне поле
Графіки малюються некоректно	Графіки малюються некоректно або зовсім не малюються	Перевірити правильність передачі даних для процесу малювання графіку	Оновлено передачу змінних для малювання графіків
Тест не проходить	Тест не проходить, хоча всі вимоги виконані	Перевірити правильність написання тесту(правильність перевірки роботи методу, який тестується)	Було знайдено тест який не проходить та виправлено перевірку результату роботи методу, який

			тестується
--	--	--	------------

## ВИСНОВКИ

У ході виконання роботи було зроблено наступне:

### 1. Огляд аналогів

На основі запропонованої програми було зроблено огляд та аналіз інтерфейсу користувача та функціональних можливостей, після чого було визначено слабкі та сильні сторони аналогу.

### 2. Збір та аналіз вимог, розробка

Однією із найбільш важливіших частин розробки програмного забезпечення для моделювання економічної системи є визначення вимог від замовника. Збір та аналіз вимог відбувався паралельно із проектуванням і розробкою програми методами еволюційного прототипування та діалогу, що дало можливість розробляти програму без додаткових ризиків.

### 3. Тестування та відлагодження

Після розробки основної частини програми було виконано тестування програми, знайдені помилки були виправлені за допомогою відлагодження через вбудовані можливості інтегрованого середовища розробки.



### Список використаної літератури

1. Бородкіна І. Л. Інженерія програмного забезпечення : навч. посіб. / І. Л. Бородкіна, Г. О. Бородкін. – Київ : ЦУЛ, 2019. – 204 с. : іл
2. Левус, Є. В. Вступ до інженерії програмного забезпечення : навч. посіб. / Є. В. Левус, Н. Б. Мельник. – Львів : Видавництво Львівської політехніки, 2018. – 246 с. : іл.
3. Соммервіл І. Інженерія програмного забезпечення; пер.с англ./ І. Соммервіл.– Москва–Санкт–Петербург–Київ, 2002. – 623 с.
4. Мямлін В. В. Існуюча прибутково-фінансова модель господарювання - основна причина кризи світової економічної системи та краху фінансової системи Вісник ДНУЗТ. Випуск 25. Дніпропетровськ : Вид-во ДНУЗТ, 2008. с. 241-247.
5. Мямлін В. В. Теорія безприбуткової альтернативної економіки як основа нового економічного світогляду. Вісник ДНУЗТ. Випуск 26. Дніпропетровськ: Вид-во ДНУЗТ, 2009. с. 222-230.
6. Мямлін В. В. До питання про категорію прибутку. Вісник ДНУЗТ. Випуск 29. Дніпропетровськ: Вид-во ДНУЗТ, 2010. с. 268-279.
7. Мямлін В. В., Мямлін С. В. Основні наукові засади створення високоефективного економіко-фінансового «механізму» у державі Незалежне дослідження із позицій ІТ-технологій: монографія. Київ-Дніпро: Моноліт, 2019. с. 40-58.
8. Мямлін В., Мямлін С. Як Україні запустити потужний економічний "двигун"? Світ. 2018. № 3-4. с. 3.
9. Мямлін В. В. До питання про хибні категорії в економічній науці. Science and education: problems, prospects and innovations. Proceedings of the 7th International scientific and practical conference. CPN Publishing Group. Kyoto, Japan. 2021. Pp. 738-747.
10. Мямлін В. В. Що потрібно змінити в економічній теорії, щоб вона стала потужним творчим інструментарієм? Проблеми та перспективи розвитку залізничного транспорту: матер. 81 міжн. наук.-практ. конф. (м. Дніпро, 22.04-23.04.2021 р.). Дніпро, 2021. с. 274-276.
11. Мямлін В. В. Фінансова рівновага виробництва та споживання як основа нормального функціонування економіки Світ наукових досліджень. Випуск 1: матер. 81 міжн. наук.-практ. конф. (м. Тернопіль, 22.06.2021 р.). Тернопіль, 2021. с. 14-18.
12. Мямлін В. В. До питання про врегулювання рівноваги між загальною вартістю товарів та кількістю грошей, призначених для їх споживання.

- Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення: зб. тез допів. міжн. наук. інтернет-конф. Вип. 62. (м. Тернопіль, 12 жовтня 2021 р.). Тернопіль, 2021. С. 77-82.
13. Мямлін В. В. Місце та роль грошей у науково обґрунтованій економіці. Традиційні принципи сучасної науки, соціальної і освітньої діяльності: Proceedings of the 6th International scientific and practical conference. SPC-Sci-conf.com.ua. Kharkiv, Україна. 2021. Рр. 1560-1566.
  14. Мямлін В. В. Про абсолютну невідповідність економічних законів природному порядку речей у природі. Science, innovations and education: problems and prospects: Proceedings of the 7th International scientific and practical конференція. SPN Publishing Group. Tokyo, Japan. 2022. Рр. 780-789.
  15. Храбатин О.В. Вимоги до сучасного програмного забезпечення систем моніторингу навколишнього середовища об'єктів нафтогазової галузі / О.В. Храбатин, Р.І. Храбатин, Г.В. Григорчук, Л.В. Саманів // Науковий вісник ІФНТУНГ. – 2011. – № 3(29). – С. 100–103
  16. Грицюк, Ю. І. Аналіз вимог до програмного забезпечення : навч. посіб. / Ю. І. Грицюк. – Львів : Вид-во Львівської політехніки, 2018. – 453 с. : іл.
  17. С# 2005 и платформа .NET 3.0 для профессионалов. Кристиан Нейгел, Билл Ивсен, Джей Глинн, Морган Скиннер, Карли Уотсон. "ДИАЛЕКТИКА" Москва. Санкт-Петербург. Киев. 2008 р. - 1790 с.
  18. Мова С# Базовий курс. 2-видання. Подбельський В.В. - 408 с.

## ДОДАТОК А

### Опис кроків для виконання моделювання

Після запуску програми користувач може бачити наступний графічний інтерфейс зображений на Рис. 1.

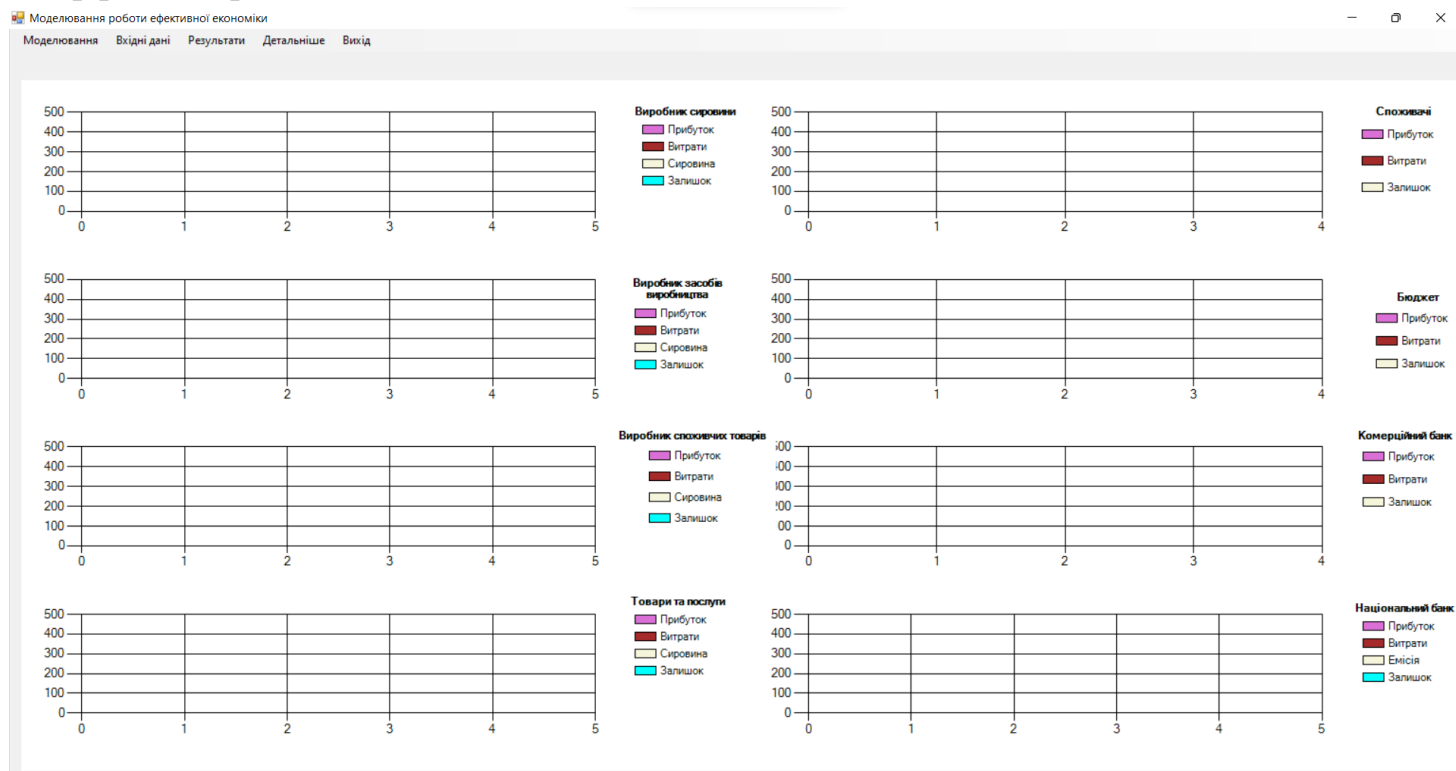


Рисунок 1. Інтерфейс програми

Для того щоб почати моделюванням необхідно обрати в меню пункт «Моделювання» зображений на Рис. 2, і в ному вибрати необхідний варіант:

- Змоделювати 1 крок
- Почати покрокове моделювання
- Зупинити покрокове моделювання

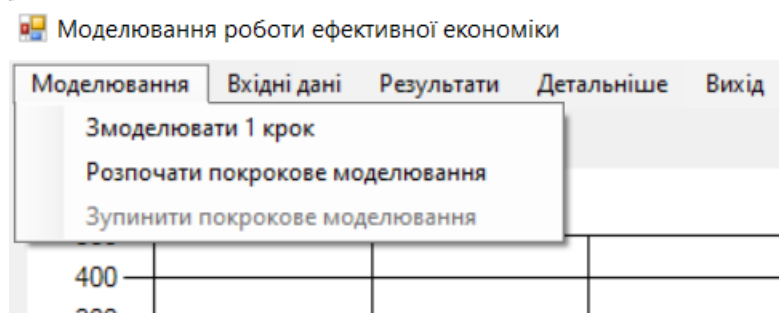


Рисунок 2. Меню моделювання

Для того щоб змінити діня для рорахунків використовується пункт меню «Вхідні дані» зображений на Рис 3, у якому можна ввести загальні дані та дані для конкретно вибраного підприємства.

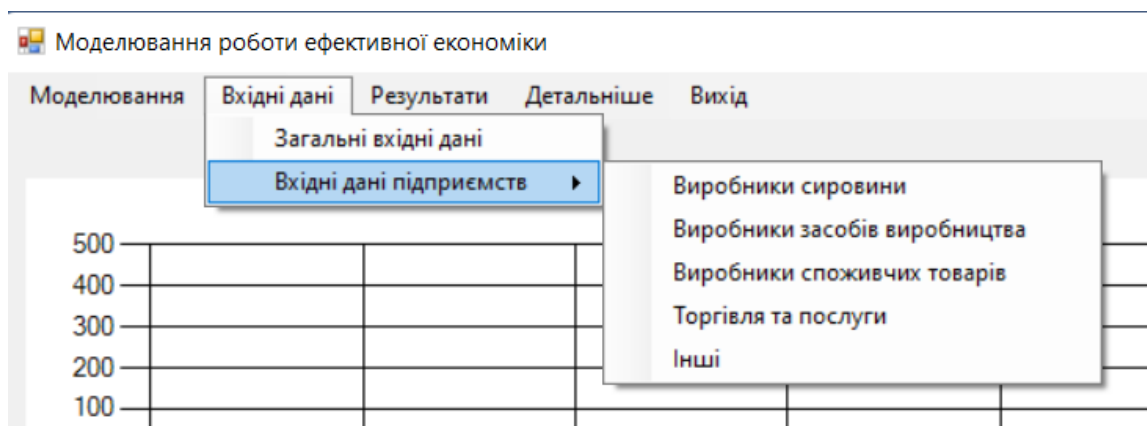


Рисунок 3. Меню вхідних даних підприємств

Вікно вводу загальних вхідних даних зображено на Рис. 4.

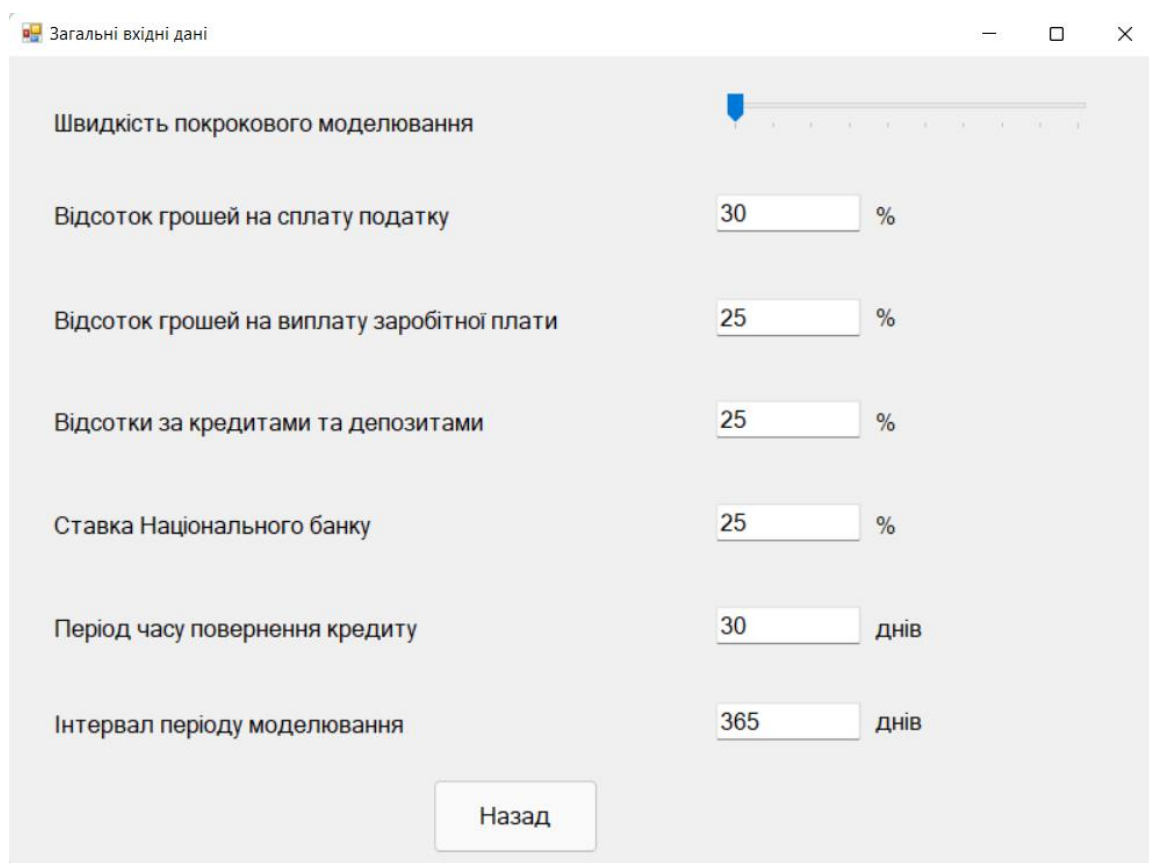


Рисунок 4. Вікно вводу загальних вхідних даних

Вікно вводу вхідних даних для виробника сировини зображено на Рис. 5.

Виробники сировини

Первісна кількість грошей від 100 до 900 у/о

Період циклу технологічного процесу від 1 до 10 днів

Відсоток торгової націнки від 10 до 20 %

Назад

Рисунок 5. Вікно вводу вхідних даних для виробника сировини

Для того щоб дізнатися які кроки були виконані при моделювання необхідно перейти до пункту меню «Результати», зображеного на Рис. 6, і обрати пункт «Переглянути вікно подій»

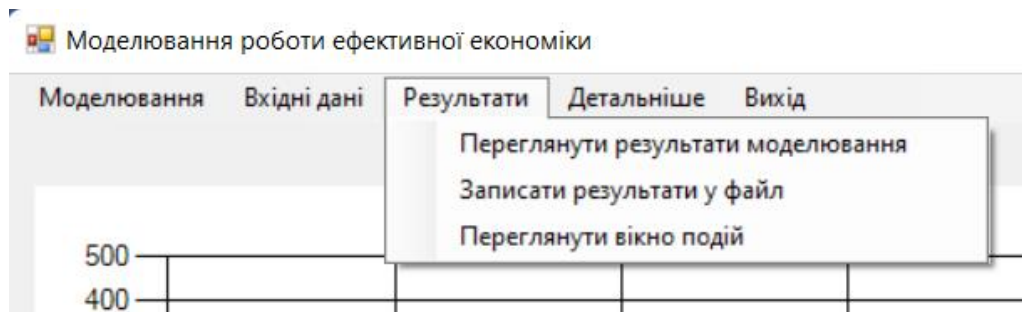


Рисунок 6. Меню результатів

## ДОДАТОК Б

### Текст програми

#### CapitalGoodsProducerInputDataForm.cs

```
using System;
using System.Windows.Forms;

namespace EffectiveEconomy
{
    public partial class CapitalGoodsProducerInputDataForm :
    Form
    {
        public CapitalGoodsProducerInputDataForm()
        {
            InitializeComponent();

            private void saveData_Click(object sender, EventArgs
            e)
            {
                InputData.VzvInitialAmountOfMoneyMin =
                int.Parse(textBox3.Text);
                InputData.VzvInitialAmountOfMoneyMax =
                int.Parse(textBox4.Text);
                InputData.VzvProcessCycleTimeMin =
                int.Parse(textBox5.Text);
                InputData.VzvProcessCycleTimeMax =
                int.Parse(textBox6.Text);
                InputData.VzvMarkupPercentageMin =
                int.Parse(textBox7.Text);
                InputData.VzvMarkupPercentageMax =
                int.Parse(textBox8.Text);

                var rand = new Random();
                InputData.VzvProcessCycleTime =
                rand.Next(InputData.VzvProcessCycleTimeMin,
                InputData.VzvProcessCycleTimeMax + 1);
                InputData.VzvMarkupPercentage =
                rand.Next(InputData.VzvMarkupPercentageMin,
                InputData.VzvMarkupPercentageMax + 1);

                var modelingForm = new ModelingForm();
                modelingForm.Show();
                this.Hide();
            }

            private void
            CapitalGoodsProducerInputDataForm_Load(object sender,
            EventArgs e)
            {
                textBox3.Text =
                InputData.VzvInitialAmountOfMoneyMin.ToString();
                textBox4.Text =
                InputData.VzvInitialAmountOfMoneyMax.ToString();
                textBox5.Text =
                InputData.VzvProcessCycleTimeMin.ToString();
                textBox6.Text =
                InputData.VzvProcessCycleTimeMax.ToString();
                textBox7.Text =
                InputData.VzvMarkupPercentageMin.ToString();
                textBox8.Text =
                InputData.VzvMarkupPercentageMax.ToString();
            }
        }
    }
}
```

```
textBox7.Text =
InputData.VzvMarkupPercentageMin.ToString();
textBox8.Text =
InputData.VzvMarkupPercentageMax.ToString();
}
```

#### CapitalGoodsProducerInputDataForm.Designer.cs

```
namespace EffectiveEconomy
{
    partial class CapitalGoodsProducerInputDataForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer
        components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed
        resources should be disposed; otherwise, false. </param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not
        modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.button1 = new System.Windows.Forms.Button();
            this.label5 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.label7 = new System.Windows.Forms.Label();
            this.label8 = new System.Windows.Forms.Label();
            this.textBox3 = new
            System.Windows.Forms.TextBox();
            this.textBox4 = new
            System.Windows.Forms.TextBox();
            this.textBox5 = new
            System.Windows.Forms.TextBox();
            this.textBox6 = new
            System.Windows.Forms.TextBox();
            this.textBox7 = new
            System.Windows.Forms.TextBox();
            this.textBox8 = new
            System.Windows.Forms.TextBox();
            this.label9 = new System.Windows.Forms.Label();
        }
    }
}
```

```

        this.label10 = new System.Windows.Forms.Label();
        this.label11 = new System.Windows.Forms.Label();
        this.label12 = new System.Windows.Forms.Label();
        this.textBox5 = new
System.Windows.Forms.TextBox();
        this.textBox6 = new
System.Windows.Forms.TextBox();
        this.label13 = new System.Windows.Forms.Label();
        this.label14 = new System.Windows.Forms.Label();
        this.label15 = new System.Windows.Forms.Label();
        this.label16 = new System.Windows.Forms.Label();
        this.textBox7 = new
System.Windows.Forms.TextBox();
        this.textBox8 = new
System.Windows.Forms.TextBox();
        this.SuspendLayout();
        //
        // button1
        //
        this.button1.Location = new
System.Drawing.Point(348, 211);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(123,
49);
        this.button1.TabIndex = 0;
        this.button1.Text = "Назад";
        this.button1.UseVisualStyleBackColor = true;
        this.button1.Click += new
System.EventHandler(this.saveData_Click);
        //
        // label5
        //
        this.label5.AutoSize = true;
        this.label5.Location = new System.Drawing.Point(18,
26);
        this.label5.Name = "label5";
        this.label5.Size = new System.Drawing.Size(253, 25);
        this.label5.TabIndex = 1;
        this.label5.Text = "Первісна кількість грошей";
        //
        // label6
        //
        this.label6.AutoSize = true;
        this.label6.Location = new
System.Drawing.Point(433, 26);
        this.label6.Name = "label6";
        this.label6.Size = new System.Drawing.Size(38, 25);
        this.label6.TabIndex = 1;
        this.label6.Text = "від";
        //
        // label7
        //
        this.label7.AutoSize = true;
        this.label7.Location = new
System.Drawing.Point(583, 26);
        this.label7.Name = "label7";
        this.label7.Size = new System.Drawing.Size(35, 25);
        this.label7.TabIndex = 1;
        this.label7.Text = "до";
        //
        // label8
        //
        this.label8.AutoSize = true;
        this.label8.Location = new
System.Drawing.Point(730, 26);
        this.label8.Name = "label8";

```

```

        this.label8.Size = new System.Drawing.Size(38, 25);
        this.label8.TabIndex = 1;
        this.label8.Text = "y/o";
        //
        // textBox3
        //
        this.textBox3.Location = new
System.Drawing.Point(477, 21);
        this.textBox3.Name = "textBox3";
        this.textBox3.Size = new System.Drawing.Size(100,
30);
        this.textBox3.TabIndex = 2;
        //
        // textBox4
        //
        this.textBox4.Location = new
System.Drawing.Point(624, 21);
        this.textBox4.Name = "textBox4";
        this.textBox4.Size = new System.Drawing.Size(100,
30);
        this.textBox4.TabIndex = 2;
        //
        // label9
        //
        this.label9.AutoSize = true;
        this.label9.Location = new System.Drawing.Point(18,
89);
        this.label9.Name = "label9";
        this.label9.Size = new System.Drawing.Size(361, 25);
        this.label9.TabIndex = 1;
        this.label9.Text = "Період циклу технологічного
процесу";
        //
        // label10
        //
        this.label10.AutoSize = true;
        this.label10.Location = new
System.Drawing.Point(433, 89);
        this.label10.Name = "label10";
        this.label10.Size = new System.Drawing.Size(38, 25);
        this.label10.TabIndex = 1;
        this.label10.Text = "від";
        //
        // label11
        //
        this.label11.AutoSize = true;
        this.label11.Location = new
System.Drawing.Point(583, 89);
        this.label11.Name = "label11";
        this.label11.Size = new System.Drawing.Size(35, 25);
        this.label11.TabIndex = 1;
        this.label11.Text = "до";
        //
        // label12
        //
        this.label12.AutoSize = true;
        this.label12.Location = new
System.Drawing.Point(730, 89);
        this.label12.Name = "label12";
        this.label12.Size = new System.Drawing.Size(49, 25);
        this.label12.TabIndex = 1;
        this.label12.Text = "днів";
        //
        // textBox5
        //
        this.textBox5.Location = new

```

```

System.Drawing.Point(477, 84);
    this.textBox5.Name = "textBox5";
    this.textBox5.Size = new System.Drawing.Size(100,
30);
    this.textBox5.TabIndex = 2;
    //
    // textBox6
    //
    this.textBox6.Location = new
System.Drawing.Point(624, 84);
    this.textBox6.Name = "textBox6";
    this.textBox6.Size = new System.Drawing.Size(100,
30);
    this.textBox6.TabIndex = 2;
    //
    // label13
    //
    this.label13.AutoSize = true;
    this.label13.Location = new
System.Drawing.Point(18, 148);
    this.label13.Name = "label13";
    this.label13.Size = new System.Drawing.Size(253,
25);
    this.label13.TabIndex = 1;
    this.label13.Text = "Відсоток торгової націнки";
    //
    // label14
    //
    this.label14.AutoSize = true;
    this.label14.Location = new
System.Drawing.Point(433, 148);
    this.label14.Name = "label14";
    this.label14.Size = new System.Drawing.Size(38, 25);
    this.label14.TabIndex = 1;
    this.label14.Text = "від";
    //
    // label15
    //
    this.label15.AutoSize = true;
    this.label15.Location = new
System.Drawing.Point(583, 148);
    this.label15.Name = "label15";
    this.label15.Size = new System.Drawing.Size(35, 25);
    this.label15.TabIndex = 1;
    this.label15.Text = "до";
    //
    // label16
    //
    this.label16.AutoSize = true;
    this.label16.Location = new
System.Drawing.Point(730, 148);
    this.label16.Name = "label16";
    this.label16.Size = new System.Drawing.Size(30, 25);
    this.label16.TabIndex = 1;
    this.label16.Text = "%";
    //
    // textBox7
    //
    this.textBox7.Location = new
System.Drawing.Point(477, 143);
    this.textBox7.Name = "textBox7";
    this.textBox7.Size = new System.Drawing.Size(100,
30);
    this.textBox7.TabIndex = 2;
    //
    // textBox8
    //
    this.textBox8.Location = new
System.Drawing.Point(624, 143);
    this.textBox8.Name = "textBox8";
    this.textBox8.Size = new System.Drawing.Size(100,
30);
    this.textBox8.TabIndex = 2;
    //
    // Form6
    //
    this.AutoScaleDimensions = new
System.Drawing.SizeF(12F, 25F);
    this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(843,
284);
    this.Controls.Add(this.textBox8);
    this.Controls.Add(this.textBox6);
    this.Controls.Add(this.textBox4);
    this.Controls.Add(this.textBox7);
    this.Controls.Add(this.textBox5);
    this.Controls.Add(this.label16);
    this.Controls.Add(this.textBox3);
    this.Controls.Add(this.label12);
    this.Controls.Add(this.label15);
    this.Controls.Add(this.label8);
    this.Controls.Add(this.label11);
    this.Controls.Add(this.label14);
    this.Controls.Add(this.label7);
    this.Controls.Add(this.label10);
    this.Controls.Add(this.label13);
    this.Controls.Add(this.label6);
    this.Controls.Add(this.label9);
    this.Controls.Add(this.label5);
    this.Controls.Add(this.button1);
    this.Font = new System.Drawing.Font("Microsoft
Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, (byte)(204));
    this.Margin = new
System.Windows.Forms.Padding(4, 5, 4, 5);
    this.Name = "CapitalGoodsProducerInputDataForm";
    this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
    this.Text = "Виробники засобів виробництва";
    this.Load += new
System.EventHandler(this.CapitalGoodsProducerInputDataF
orm_Load);
    this.ResumeLayout(false);
    this.PerformLayout();
}

#endregion

private System.Windows.Forms.Button button1;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.Label label8;
private System.Windows.Forms.TextBox textBox3;
private System.Windows.Forms.TextBox textBox4;
private System.Windows.Forms.Label label9;
private System.Windows.Forms.Label label10;
private System.Windows.Forms.Label label11;
private System.Windows.Forms.Label label12;
private System.Windows.Forms.TextBox textBox5;

```



```

        private System.Windows.Forms.TextBox textBox6;
        private System.Windows.Forms.Label label13;
        private System.Windows.Forms.Label label14;
        private System.Windows.Forms.Label label15;
        private System.Windows.Forms.Label label16;
        private System.Windows.Forms.TextBox textBox7;
        private System.Windows.Forms.TextBox textBox8;
    }
}

CapitalGoodsProducerInputDataForm.Designer.cs
using System;
using System.Windows.Forms;

namespace EffectiveEconomy
{
    public partial class
ConsumerGoodsProducerInputDataForm : Form
    {
        public ConsumerGoodsProducerInputDataForm()
        {
            InitializeComponent();
        }

        private void SaveData_Click(object sender, EventArgs
e)
        {
            InputData.VstInitialAmountOfMoneyMin =
int.Parse(textBox3.Text);
            InputData.VstInitialAmountOfMoneyMax =
int.Parse(textBox4.Text);
            InputData.VstProcessCycleTimeMin =
int.Parse(textBox5.Text);
            InputData.VstProcessCycleTimeMax =
int.Parse(textBox6.Text);
            InputData.VstMarkupPercentageMin =
int.Parse(textBox7.Text);
            InputData.VstMarkupPercentageMax =
int.Parse(textBox8.Text);

            Random rand = new Random();
            InputData.VstProcessCycleTime =
rand.Next(InputData.VstProcessCycleTimeMin,
InputData.VstProcessCycleTimeMax + 1);
            InputData.VstMarkupPercentage =
rand.Next(InputData.VstMarkupPercentageMin,
InputData.VstMarkupPercentageMax + 1);

            var newForm = new ModelingForm();
            newForm.Show();
            Hide();
        }

        private void
ConsumerGoodsProducerInputDataForm_Load(object
sender, EventArgs e)
        {
            textBox3.Text =
InputData.VstInitialAmountOfMoneyMin.ToString();
            textBox4.Text =
InputData.VstInitialAmountOfMoneyMax.ToString();
            textBox5.Text =
InputData.VstProcessCycleTimeMin.ToString();
            textBox6.Text =
InputData.VstProcessCycleTimeMax.ToString();
            textBox7.Text =

```

```

InputData.VstMarkupPercentageMin.ToString();
            textBox8.Text =
InputData.VstMarkupPercentageMax.ToString();
        }
    }
}

```

ConsumerGoodsProducerInputDataForm.Designer.cs

```

namespace EffectiveEconomy
{
    partial class ConsumerGoodsProducerInputDataForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer
components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources
should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not
modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.textBox8 = new
System.Windows.Forms.TextBox();
            this.textBox6 = new
System.Windows.Forms.TextBox();
            this.textBox4 = new
System.Windows.Forms.TextBox();
            this.textBox7 = new
System.Windows.Forms.TextBox();
            this.textBox5 = new
System.Windows.Forms.TextBox();
            this.label16 = new System.Windows.Forms.Label();
            this.textBox3 = new
System.Windows.Forms.TextBox();
            this.label12 = new System.Windows.Forms.Label();
            this.label15 = new System.Windows.Forms.Label();
            this.label8 = new System.Windows.Forms.Label();
            this.label11 = new System.Windows.Forms.Label();
            this.label14 = new System.Windows.Forms.Label();
            this.label7 = new System.Windows.Forms.Label();
            this.label10 = new System.Windows.Forms.Label();
            this.label13 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.label9 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
        }
    }
}

```

```

        this.button1 = new System.Windows.Forms.Button();
        this.SuspendLayout();
        //
        // textBox8
        //
        this.textBox8.Location = new
System.Drawing.Point(490, 134);
        this.textBox8.Name = "textBox8";
        this.textBox8.Size = new System.Drawing.Size(100,
30);
        this.textBox8.TabIndex = 27;
        //
        // textBox6
        //
        this.textBox6.Location = new
System.Drawing.Point(490, 75);
        this.textBox6.Name = "textBox6";
        this.textBox6.Size = new System.Drawing.Size(100,
30);
        this.textBox6.TabIndex = 25;
        //
        // textBox4
        //
        this.textBox4.Location = new
System.Drawing.Point(490, 12);
        this.textBox4.Name = "textBox4";
        this.textBox4.Size = new System.Drawing.Size(100,
30);
        this.textBox4.TabIndex = 24;
        //
        // textBox7
        //
        this.textBox7.Location = new
System.Drawing.Point(343, 134);
        this.textBox7.Name = "textBox7";
        this.textBox7.Size = new System.Drawing.Size(100,
30);
        this.textBox7.TabIndex = 23;
        //
        // textBox5
        //
        this.textBox5.Location = new
System.Drawing.Point(343, 75);
        this.textBox5.Name = "textBox5";
        this.textBox5.Size = new System.Drawing.Size(100,
30);
        this.textBox5.TabIndex = 22;
        //
        // label16
        //
        this.label16.AutoSize = true;
        this.label16.Location = new
System.Drawing.Point(596, 139);
        this.label16.Name = "label16";
        this.label16.Size = new System.Drawing.Size(30, 25);
        this.label16.TabIndex = 13;
        this.label16.Text = "%";
        //
        // textBox3
        //
        this.textBox3.Location = new
System.Drawing.Point(343, 12);
        this.textBox3.Name = "textBox3";
        this.textBox3.Size = new System.Drawing.Size(100,
30);
        this.textBox3.TabIndex = 21;
        //
        // label12
        //
        this.label12.AutoSize = true;
        this.label12.Location = new
System.Drawing.Point(596, 80);
        this.label12.Name = "label12";
        this.label12.Size = new System.Drawing.Size(49, 25);
        this.label12.TabIndex = 18;
        this.label12.Text = "днів";
        //
        // label15
        //
        this.label15.AutoSize = true;
        this.label15.Location = new
System.Drawing.Point(449, 139);
        this.label15.Name = "label15";
        this.label15.Size = new System.Drawing.Size(35, 25);
        this.label15.TabIndex = 9;
        this.label15.Text = "до";
        //
        // label8
        //
        this.label8.AutoSize = true;
        this.label8.Location = new
System.Drawing.Point(596, 17);
        this.label8.Name = "label8";
        this.label8.Size = new System.Drawing.Size(38, 25);
        this.label8.TabIndex = 8;
        this.label8.Text = "y/o";
        //
        // label11
        //
        this.label11.AutoSize = true;
        this.label11.Location = new
System.Drawing.Point(449, 80);
        this.label11.Name = "label11";
        this.label11.Size = new System.Drawing.Size(35, 25);
        this.label11.TabIndex = 7;
        this.label11.Text = "до";
        //
        // label14
        //
        this.label14.AutoSize = true;
        this.label14.Location = new
System.Drawing.Point(299, 139);
        this.label14.Name = "label14";
        this.label14.Size = new System.Drawing.Size(38, 25);
        this.label14.TabIndex = 5;
        this.label14.Text = "від";
        //
        // label7
        //
        this.label7.AutoSize = true;
        this.label7.Location = new
System.Drawing.Point(449, 17);
        this.label7.Name = "label7";
        this.label7.Size = new System.Drawing.Size(35, 25);
        this.label7.TabIndex = 4;
        this.label7.Text = "до";
        //
        // label10
        //
        this.label10.AutoSize = true;
        this.label10.Location = new
System.Drawing.Point(299, 80);

```

```

        this.label10.Name = "label10";
        this.label10.Size = new System.Drawing.Size(38, 25);
        this.label10.TabIndex = 10;
        this.label10.Text = "від";
        //
        // label13
        //
        this.label13.AutoSize = true;
        this.label13.Location = new
System.Drawing.Point(14, 139);
        this.label13.Name = "label13";
        this.label13.Size = new System.Drawing.Size(253,
25);
        this.label13.TabIndex = 11;
        this.label13.Text = "Відсоток торгової націнки";
        //
        // label6
        //
        this.label6.AutoSize = true;
        this.label6.Location = new
System.Drawing.Point(299, 17);
        this.label6.Name = "label6";
        this.label6.Size = new System.Drawing.Size(38, 25);
        this.label6.TabIndex = 12;
        this.label6.Text = "від";
        //
        // label9
        //
        this.label9.AutoSize = true;
        this.label9.Location = new System.Drawing.Point(14,
80);
        this.label9.Name = "label9";
        this.label9.Size = new System.Drawing.Size(239, 25);
        this.label9.TabIndex = 14;
        this.label9.Text = "Період реалізації товару";
        //
        // label5
        //
        this.label5.AutoSize = true;
        this.label5.Location = new System.Drawing.Point(14,
17);
        this.label5.Name = "label5";
        this.label5.Size = new System.Drawing.Size(253, 25);
        this.label5.TabIndex = 16;
        this.label5.Text = "Первісна кількість грошей";
        //
        // button1
        //
        this.button1.Location = new
System.Drawing.Point(251, 192);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(123,
49);
        this.button1.TabIndex = 3;
        this.button1.Text = "Назад";
        this.button1.UseVisualStyleBackColor = true;
        this.button1.Click += new
System.EventHandler(this.SaveData_Click);
        //
        // Form7
        //
        this.AutoScaleDimensions = new
System.Drawing.SizeF(12F, 25F);
        this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;

        this.ClientSize = new System.Drawing.Size(682,
268);
        this.Controls.Add(this.textBox8);
        this.Controls.Add(this.textBox6);
        this.Controls.Add(this.textBox4);
        this.Controls.Add(this.textBox7);
        this.Controls.Add(this.textBox5);
        this.Controls.Add(this.label16);
        this.Controls.Add(this.textBox3);
        this.Controls.Add(this.label12);
        this.Controls.Add(this.label15);
        this.Controls.Add(this.label8);
        this.Controls.Add(this.label11);
        this.Controls.Add(this.label14);
        this.Controls.Add(this.label7);
        this.Controls.Add(this.label10);
        this.Controls.Add(this.label13);
        this.Controls.Add(this.label6);
        this.Controls.Add(this.label9);
        this.Controls.Add(this.label5);
        this.Controls.Add(this.button1);
        this.Font = new System.Drawing.Font("Microsoft
Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.Margin = new
System.Windows.Forms.Padding(4, 5, 4, 5);
        this.Name =
"ConsumerGoodsProducerInputDataForm";
        this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
        this.Text = "Виробники споживчих товарів";
        this.Load += new
System.EventHandler(this.ConsumerGoodsProducerInputDat
aForm_Load);
        this.ResumeLayout(false);
        this.PerformLayout();
    }

    #endregion

    private System.Windows.Forms.TextBox textBox8;
    private System.Windows.Forms.TextBox textBox6;
    private System.Windows.Forms.TextBox textBox4;
    private System.Windows.Forms.TextBox textBox7;
    private System.Windows.Forms.TextBox textBox5;
    private System.Windows.Forms.Label label16;
    private System.Windows.Forms.TextBox textBox3;
    private System.Windows.Forms.Label label12;
    private System.Windows.Forms.Label label15;
    private System.Windows.Forms.Label label8;
    private System.Windows.Forms.Label label11;
    private System.Windows.Forms.Label label14;
    private System.Windows.Forms.Label label7;
    private System.Windows.Forms.Label label10;
    private System.Windows.Forms.Label label13;
    private System.Windows.Forms.Label label6;
    private System.Windows.Forms.Label label9;
    private System.Windows.Forms.Label label5;
    private System.Windows.Forms.Button button1;
}

}

EconomyEvents.cs
namespace EffectiveEconomy
{

```

```

public static class EconomyEvents
{
    //Result events string
    public static string ResultEventsString { get; set; } = "
Початок моделювання\n";

    //Enterprises
    public const string Material = "Виробник сировини -
";
    public const string Manufacturer = "Виробник
засобів виробництва - ";
    public const string Consumer = "Виробник
споживчих товарів - ";
    public const string Trade = "Торгівля та Послуги - ";
    public const string Budget = "Бюджет - ";
    public const string Commercial = "Комерційні банки -
";
    public const string Buyers = "Покупці споживчих
товарів - ";
    public const string Bank = "Нацбанк - ";

    //Possible events
    public const string CalculateSalariesEvent =
"розрахувати податки та зарплати та виробництво
сировини\n";/+
    public const string ProcessPaymentsEvent =
"виплатити податки та зарплати\n";/+
    public const string TakeCreditEvent = "взяти
позичку\n";/+
    public const string IncreaseEmissionEvent =
"підвищити еміцію\n";/+
    public const string SolveProblemEvent = "ліквідувати
наслідки НС\n";/+
    public const string ProduceGoodsEvent = "виробити
товар\n";/+
    public const string CalculatePaymentsEvent =
"розрахувати виплати\n";/+
    public const string BuyGoodsEvent = "купити
товар\n";/+
    public const string PaySalariesEvent = "виплатити
пенсії та зарплати бюджетникам\n";/+
    public const string SellGoodsEvent = "продати
товар\n";/+
}
}
EventsForm.cs
using System;
using System.Windows.Forms;

namespace EffectiveEconomy
{
    public partial class EventsForm : Form
    {
        public EventsForm()
        {
            InitializeComponent();
        }

        private void EventsForm_Load(object sender,
EventArgs e)
        {
            richTextBox1.Text =
EconomyEvents.ResultEventsString;
        }

        private void SaveData_Click(object sender, EventArgs

```

```

e)
{
    richTextBox1.Text =
EconomyEvents.ResultEventsString;
    richTextBox1.Update();
}
}

EventsForm.Designer.cs

namespace EffectiveEconomy
{
    partial class EventsForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer
components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed
resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not
modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager
resources = new
System.ComponentModel.ComponentResourceManager(typeof(EventsForm));
            this.richTextBox1 = new
System.Windows.Forms.RichTextBox();
            this.button1 = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // richTextBox1
            //
            this.richTextBox1.Location = new
System.Drawing.Point(12, 12);
            this.richTextBox1.Name = "richTextBox1";
            this.richTextBox1.ReadOnly = true;
            this.richTextBox1.Size = new
System.Drawing.Size(891, 502);
            this.richTextBox1.TabIndex = 1;
            this.richTextBox1.Text = "";
            //
            // button1

```

```

        //
        this.button1.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("button1.Bac
kgroundImage")));
        this.button1.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch;
        this.button1.ForeColor =
System.Drawing.SystemColors.Control;
        this.button1.Image =
((System.Drawing.Image)(resources.GetObject("button1.Ima
ge")));
        this.button1.Location = new
System.Drawing.Point(806, 12);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(75,
66);
        this.button1.TabIndex = 2;
        this.button1.UseVisualStyleBackColor = true;
        this.button1.Click += new
System.EventHandler(this.SaveData_Click);
        //
        // Form4
        //
        this.AutoScaleDimensions = new
System.Drawing.SizeF(12F, 25F);
        this.AutoScaleMode =
System.Windows.Forms.AutoScaleModeMode.Font;
        this.ClientSize = new System.Drawing.Size(915,
526);
        this.Controls.Add(this.button1);
        this.Controls.Add(this.richTextBox1);
        this.Font = new System.Drawing.Font("Microsoft
Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.Margin = new
System.Windows.Forms.Padding(4, 5, 4, 5);
        this.Name = "EventsForm";
        this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
        this.Text = "Вікно подій";
        this.TopMost = true;
        this.Load += new
System.EventHandler(this.EventsForm_Load);
        this.ResumeLayout(false);

    }

    #endregion

    private System.Windows.Forms.RichTextBox
richTextBox1;
    private System.Windows.Forms.Button button1;
}

GeneralInputDataForm.cs
using System;
using System.Windows.Forms;

namespace EffectiveEconomy
{
    public partial class GeneralInputDataForm : Form
    {
        public GeneralInputDataForm()
        {
            InitializeComponent();

```

```

    }

    private void SaveData_Click(object sender, EventArgs
e)
    {
        InputData.MoneySpendingPercentage =
int.Parse(textBox1.Text);
        InputData.PercentageOfSpendingBudgetMoney =
int.Parse(textBox2.Text);
        InputData.LikelihoodOfAnEmergency =
int.Parse(textBox3.Text);
        InputData.CommercialBankAmount =
int.Parse(textBox6.Text);
        InputData.NationalBankTotalAmount =
int.Parse(textBox8.Text);

        var newForm = new ModelingForm();
        newForm.Show();
        Hide();
    }

    private void GeneralInputDataForm_Load(object
sender, EventArgs e)
    {
        textBox1.Text =
InputData.MoneySpendingPercentage.ToString();
        textBox2.Text =
InputData.PercentageOfSpendingBudgetMoney.ToString();
        textBox3.Text =
InputData.LikelihoodOfAnEmergency.ToString();
        textBox6.Text =
InputData.CommercialBankAmount.ToString();
        textBox8.Text =
InputData.NationalBankTotalAmount.ToString();
    }
}

GeneralInputDataForm.Designer.cs

namespace EffectiveEconomy
{
    partial class GeneralInputDataForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer
components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources
should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

```

```

    /// <summary>
    /// Required method for Designer support - do not
    modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.textBox8 = new
System.Windows.Forms.TextBox();
        this.textBox6 = new
System.Windows.Forms.TextBox();
        this.label16 = new System.Windows.Forms.Label();
        this.textBox3 = new
System.Windows.Forms.TextBox();
        this.label12 = new System.Windows.Forms.Label();
        this.textBox2 = new
System.Windows.Forms.TextBox();
        this.label8 = new System.Windows.Forms.Label();
        this.label4 = new System.Windows.Forms.Label();
        this.label13 = new System.Windows.Forms.Label();
        this.label9 = new System.Windows.Forms.Label();
        this.label5 = new System.Windows.Forms.Label();
        this.label1 = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.label3 = new System.Windows.Forms.Label();
        this.textBox1 = new
System.Windows.Forms.TextBox();
        this.button1 = new System.Windows.Forms.Button();
        this.SuspendLayout();
        //
        // textBox8
        //
        this.textBox8.Location = new
System.Drawing.Point(550, 260);
        this.textBox8.Name = "textBox8";
        this.textBox8.Size = new System.Drawing.Size(100,
30);
        this.textBox8.TabIndex = 26;
        //
        // textBox6
        //
        this.textBox6.Location = new
System.Drawing.Point(550, 206);
        this.textBox6.Name = "textBox6";
        this.textBox6.Size = new System.Drawing.Size(100,
30);
        this.textBox6.TabIndex = 25;
        //
        // label16
        //
        this.label16.AutoSize = true;
        this.label16.Location = new
System.Drawing.Point(656, 206);
        this.label16.Name = "label16";
        this.label16.Size = new System.Drawing.Size(38, 25);
        this.label16.TabIndex = 17;
        this.label16.Text = "y/o";
        //
        // textBox3
        //
        this.textBox3.Location = new
System.Drawing.Point(550, 147);
        this.textBox3.Name = "textBox3";
        this.textBox3.Size = new System.Drawing.Size(100,
30);
        this.textBox3.TabIndex = 21;

```

```

//
// label12
//
this.label12.AutoSize = true;
this.label12.Location = new
System.Drawing.Point(656, 147);
this.label12.Name = "label12";
this.label12.Size = new System.Drawing.Size(30, 25);
this.label12.TabIndex = 16;
this.label12.Text = "%";
//
// textBox2
//
this.textBox2.Location = new
System.Drawing.Point(550, 81);
this.textBox2.Name = "textBox2";
this.textBox2.Size = new System.Drawing.Size(100,
30);
this.textBox2.TabIndex = 20;
//
// label8
//
this.label8.AutoSize = true;
this.label8.Location = new
System.Drawing.Point(656, 84);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(30, 25);
this.label8.TabIndex = 18;
this.label8.Text = "%";
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new
System.Drawing.Point(656, 26);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(30, 25);
this.label4.TabIndex = 9;
this.label4.Text = "%";
//
// label13
//
this.label13.AutoSize = true;
this.label13.Location = new
System.Drawing.Point(12, 206);
this.label13.Name = "label13";
this.label13.Size = new System.Drawing.Size(358,
25);
this.label13.TabIndex = 8;
this.label13.Text = "Кількість грошей
комерційного банку";
//
// label9
//
this.label9.AutoSize = true;
this.label9.Location = new System.Drawing.Point(12,
147);
this.label9.Name = "label9";
this.label9.Size = new System.Drawing.Size(431, 25);
this.label9.TabIndex = 6;
this.label9.Text = "Ймовірність настання
Надзвичайної Ситуації";
//
// label5
//
this.label5.AutoSize = true;

```

```

        this.label5.Location = new System.Drawing.Point(12,
84);
        this.label5.Name = "label5";
        this.label5.Size = new System.Drawing.Size(638, 25);
        this.label5.TabIndex = 4;
        this.label5.Text = "Відсотки витрачання
бюджетних грошей на «Послуги та торгівлю»";
        //
        // label1
        //
        this.label1.AutoSize = true;
        this.label1.Location = new System.Drawing.Point(12,
24);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(492, 25);
        this.label1.TabIndex = 14;
        this.label1.Text = "Відсотки витрати грошей на
«Послуги та торгівлю»";
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Location = new System.Drawing.Point(12,
263);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(361, 25);
        this.label2.TabIndex = 8;
        this.label2.Text = "Загальна кількість грошей у
Нацбанку";
        //
        // label3
        //
        this.label3.AutoSize = true;
        this.label3.Location = new
System.Drawing.Point(656, 263);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(38, 25);
        this.label3.TabIndex = 17;
        this.label3.Text = "y/o";
        //
        // textBox1
        //
        this.textBox1.Location = new
System.Drawing.Point(550, 23);
        this.textBox1.Name = "textBox1";
        this.textBox1.Size = new System.Drawing.Size(100,
30);
        this.textBox1.TabIndex = 26;
        //
        // button1
        //
        this.button1.Location = new
System.Drawing.Point(301, 319);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(123,
48);
        this.button1.TabIndex = 27;
        this.button1.Text = "Назад";
        this.button1.UseVisualStyleBackColor = true;
        this.button1.Click += new
System.EventHandler(this.SaveData_Click);
        //
        // Form9
        //
        this.AutoScaleDimensions = new
System.Drawing.SizeF(12F, 25F);

```

```

        this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(891,
379);
        this.Controls.Add(this.button1);
        this.Controls.Add(this.textBox1);
        this.Controls.Add(this.textBox8);
        this.Controls.Add(this.textBox6);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.label16);
        this.Controls.Add(this.textBox3);
        this.Controls.Add(this.label12);
        this.Controls.Add(this.textBox2);
        this.Controls.Add(this.label8);
        this.Controls.Add(this.label4);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label13);
        this.Controls.Add(this.label9);
        this.Controls.Add(this.label5);
        this.Controls.Add(this.label1);
        this.Font = new System.Drawing.Font("Microsoft
Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.Margin = new
System.Windows.Forms.Padding(4, 5, 4, 5);
        this.Name = "GeneralInputDataForm";
        this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
        this.Text = "Інші";
        this.Load += new
System.EventHandler(this.GeneralInputDataForm_Load);
        this.ResumeLayout(false);
        this.PerformLayout();

    }

    #endregion

    private System.Windows.Forms.TextBox textBox8;
    private System.Windows.Forms.TextBox textBox6;
    private System.Windows.Forms.Label label16;
    private System.Windows.Forms.TextBox textBox3;
    private System.Windows.Forms.Label label12;
    private System.Windows.Forms.TextBox textBox2;
    private System.Windows.Forms.Label label8;
    private System.Windows.Forms.Label label4;
    private System.Windows.Forms.Label label13;
    private System.Windows.Forms.Label label9;
    private System.Windows.Forms.Label label5;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.TextBox textBox1;
    private System.Windows.Forms.Button button1;
    }

InputData.cs
namespace EffectiveEconomy
{
    public static class InputData
    {
        public static int Speed { get; set; } = 7;

        public static int TaxMoneyPercentage { get; set; } = 30;
    }
}

```

```

public static int PayWagesPercentage { get; set; } = 25;

public static int InterestOnLoans { get; set; } = 25;

public static int NationalBankRate { get; set; } = 25;

public static int LoanRepaymentPeriod { get; set; } =
30;

public static int SimulationPeriodInterval { get; set; } =
365;

public static int VsInitialAmountOfMoneyMin { get;
set; } = 100;

public static int VsInitialAmountOfMoneyMax { get;
set; } = 900;

public static int VsInitialAmountOfMoney { get; set; }
= 0;

public static int VsProcessCycleTimeMin { get; set; } =
1;

public static int VsProcessCycleTimeMax { get; set; } =
10;

public static int VsProcessCycleTime { get; set; } = 0;

public static int VsMarkupPercentageMin { get; set; } =
10;

public static int VsMarkupPercentageMax { get; set; } =
20;

public static int VsMarkupPercentage { get; set; } = 0;

public static int VzInitialAmountOfMoneyMin = 100;
public static int VzInitialAmountOfMoneyMax = 900;
public static int VzInitialAmountOfMoney = 0;
public static int VzProcessCycleTimeMin = 1;
public static int VzProcessCycleTimeMax = 10;
public static int VzProcessCycleTime = 0;
public static int VzMarkupPercentageMin = 10;
public static int VzMarkupPercentageMax = 20;
public static int VzMarkupPercentage = 0;

public static int VstInitialAmountOfMoneyMin = 100;
public static int VstInitialAmountOfMoneyMax = 900;
public static int VstInitialAmountOfMoney = 0;
public static int VstProcessCycleTimeMin = 1;
public static int VstProcessCycleTimeMax = 10;
public static int VstProcessCycleTime = 0;
public static int VstMarkupPercentageMin = 10;
public static int VstMarkupPercentageMax = 20;
public static int VstMarkupPercentage = 0;

public static int TtpInitialAmountOfMoneyMin = 100;
public static int TtpInitialAmountOfMoneyMax = 900;
public static int TtpInitialAmountOfMoney = 0;
public static int TtpPercentageOfPurchasesGoods = 50;

public static int MoneySpendingPercentage = 80;
public static int PercentageOfSpendingBudgetMoney =
20;

public static int LikelihoodOfAnEmergency = 15;

```

```

public static int CommercialBankAmount = 800;
public static int NationalBankTotalAmount = 1000;
}
}

InputData.cs
using System;
using System.Windows.Forms;

namespace EffectiveEconomy
{
    public partial class InputDataForm : Form
    {
        public InputDataForm()
        {
            InitializeComponent();
        }

        private void SaveData_Click(object sender, EventArgs
e)
        {
            trackBar1.Value = InputData.Speed;
            InputData.TaxMoneyPercentage =
int.Parse(textBox1.Text);
            InputData.PayWagesPercentage =
int.Parse(textBox2.Text);
            InputData.InterestOnLoans =
int.Parse(textBox3.Text);
            InputData.NationalBankRate =
int.Parse(textBox4.Text);
            InputData.LoanRepaymentPeriod =
int.Parse(textBox5.Text);
            InputData.SimulationPeriodInterval =
int.Parse(textBox6.Text);

            var modelingForm = new ModelingForm();
            modelingForm.Show();
            this.Hide();
        }

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

        private void InputDataForm_Load(object sender,
EventArgs e)
        {
            trackBar1.Value = InputData.Speed;
            textBox1.Text =
InputData.TaxMoneyPercentage.ToString();
            textBox2.Text =
InputData.PayWagesPercentage.ToString();
            textBox3.Text =
InputData.InterestOnLoans.ToString();
            textBox4.Text =
InputData.NationalBankRate.ToString();
            textBox5.Text =
InputData.LoanRepaymentPeriod.ToString();
            textBox6.Text =
InputData.SimulationPeriodInterval.ToString();
        }
    }
}

```



InputDataForm.Designer.cs

```
namespace EffectiveEconomy
{
    partial class InputDataForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer
        components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources
        should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not
        modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.label1 = new System.Windows.Forms.Label();
            this.trackBar1 = new
            System.Windows.Forms.TrackBar();
            this.button1 = new System.Windows.Forms.Button();
            this.label2 = new System.Windows.Forms.Label();
            this.textBox1 = new
            System.Windows.Forms.TextBox();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.label7 = new System.Windows.Forms.Label();
            this.textBox2 = new
            System.Windows.Forms.TextBox();
            this.textBox3 = new
            System.Windows.Forms.TextBox();
            this.textBox4 = new
            System.Windows.Forms.TextBox();
            this.textBox5 = new
            System.Windows.Forms.TextBox();
            this.textBox6 = new
            System.Windows.Forms.TextBox();
            this.label8 = new System.Windows.Forms.Label();
            this.label9 = new System.Windows.Forms.Label();
            this.label10 = new System.Windows.Forms.Label();
            this.label11 = new System.Windows.Forms.Label();
            this.label12 = new System.Windows.Forms.Label();
            this.label13 = new System.Windows.Forms.Label();

            ((System.ComponentModel.ISupportInitialize)(this.trackBar1)).BeginInit();
```

```
this.SuspendLayout();
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(27,
36);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(363, 25);
this.label1.TabIndex = 0;
this.label1.Text = "Швидкість покровового
моделювання";
//
// trackBar1
//
this.trackBar1.Location = new
System.Drawing.Point(487, 24);
this.trackBar1.Minimum = 1;
this.trackBar1.Name = "trackBar1";
this.trackBar1.Size = new System.Drawing.Size(263,
56);
this.trackBar1.TabIndex = 1;
this.trackBar1.Value = 7;
this.trackBar1.Scroll += new
System.EventHandler(this.trackBar1_Scroll);
//
// button1
//
this.button1.Location = new
System.Drawing.Point(292, 498);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(114,
51);
this.button1.TabIndex = 2;
this.button1.Text = "Назад";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new
System.EventHandler(this.SaveData_Click);
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(27,
100);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(342, 25);
this.label2.TabIndex = 3;
this.label2.Text = "Відсоток грошей на сплату
податку";
//
// textBox1
//
this.textBox1.Location = new
System.Drawing.Point(487, 95);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(100,
30);
this.textBox1.TabIndex = 4;
this.textBox1.Text = "30";
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(27,
172);
this.label3.Name = "label3";
```

```

        this.label3.Size = new System.Drawing.Size(437, 25);
        this.label3.TabIndex = 3;
        this.label3.Text = "Відсоток грошей на виплату
заробітної плати";
        //
        // label4
        //
        this.label4.AutoSize = true;
        this.label4.Location = new System.Drawing.Point(27,
242);
        this.label4.Name = "label4";
        this.label4.Size = new System.Drawing.Size(379, 25);
        this.label4.TabIndex = 3;
        this.label4.Text = "Відсотки за кредитами та
депозитами";
        //
        // label5
        //
        this.label5.AutoSize = true;
        this.label5.Location = new System.Drawing.Point(27,
313);
        this.label5.Name = "label5";
        this.label5.Size = new System.Drawing.Size(272, 25);
        this.label5.TabIndex = 3;
        this.label5.Text = "Ставка Національного банку";
        //
        // label6
        //
        this.label6.AutoSize = true;
        this.label6.Location = new System.Drawing.Point(27,
384);
        this.label6.Name = "label6";
        this.label6.Size = new System.Drawing.Size(315, 25);
        this.label6.TabIndex = 3;
        this.label6.Text = "Період часу повернення
кредиту";
        //
        // label7
        //
        this.label7.AutoSize = true;
        this.label7.Location = new System.Drawing.Point(27,
450);
        this.label7.Name = "label7";
        this.label7.Size = new System.Drawing.Size(304, 25);
        this.label7.TabIndex = 3;
        this.label7.Text = "Інтервал періоду моделювання";
        //
        // textBox2
        //
        this.textBox2.Location = new
System.Drawing.Point(487, 167);
        this.textBox2.Name = "textBox2";
        this.textBox2.Size = new System.Drawing.Size(100,
30);
        this.textBox2.TabIndex = 4;
        this.textBox2.Text = "30";
        //
        // textBox3
        //
        this.textBox3.Location = new
System.Drawing.Point(487, 237);
        this.textBox3.Name = "textBox3";
        this.textBox3.Size = new System.Drawing.Size(100,
30);
        this.textBox3.TabIndex = 4;
        this.textBox3.Text = "30";

```

```

        //
        // textBox4
        //
        this.textBox4.Location = new
System.Drawing.Point(487, 308);
        this.textBox4.Name = "textBox4";
        this.textBox4.Size = new System.Drawing.Size(100,
30);
        this.textBox4.TabIndex = 4;
        this.textBox4.Text = "30";
        //
        // textBox5
        //
        this.textBox5.Location = new
System.Drawing.Point(487, 379);
        this.textBox5.Name = "textBox5";
        this.textBox5.Size = new System.Drawing.Size(100,
30);
        this.textBox5.TabIndex = 4;
        this.textBox5.Text = "30";
        //
        // textBox6
        //
        this.textBox6.Location = new
System.Drawing.Point(487, 445);
        this.textBox6.Name = "textBox6";
        this.textBox6.Size = new System.Drawing.Size(100,
30);
        this.textBox6.TabIndex = 4;
        this.textBox6.Text = "30";
        //
        // label8
        //
        this.label8.AutoSize = true;
        this.label8.Location = new
System.Drawing.Point(593, 100);
        this.label8.Name = "label8";
        this.label8.Size = new System.Drawing.Size(30, 25);
        this.label8.TabIndex = 3;
        this.label8.Text = "%";
        //
        // label9
        //
        this.label9.AutoSize = true;
        this.label9.Location = new
System.Drawing.Point(593, 170);
        this.label9.Name = "label9";
        this.label9.Size = new System.Drawing.Size(30, 25);
        this.label9.TabIndex = 3;
        this.label9.Text = "%";
        //
        // label10
        //
        this.label10.AutoSize = true;
        this.label10.Location = new
System.Drawing.Point(593, 242);
        this.label10.Name = "label10";
        this.label10.Size = new System.Drawing.Size(30, 25);
        this.label10.TabIndex = 3;
        this.label10.Text = "%";
        //
        // label11
        //
        this.label11.AutoSize = true;
        this.label11.Location = new
System.Drawing.Point(593, 313);

```

```

this.label11.Name = "label11";
this.label11.Size = new System.Drawing.Size(30, 25);
this.label11.TabIndex = 3;
this.label11.Text = "%";
//
// label12
//
this.label12.AutoSize = true;
this.label12.Location = new
System.Drawing.Point(593, 384);
this.label12.Name = "label12";
this.label12.Size = new System.Drawing.Size(49, 25);
this.label12.TabIndex = 3;
this.label12.Text = "днів";
//
// label13
//
this.label13.AutoSize = true;
this.label13.Location = new
System.Drawing.Point(593, 448);
this.label13.Name = "label13";
this.label13.Size = new System.Drawing.Size(49, 25);
this.label13.TabIndex = 3;
this.label13.Text = "днів";
//
// Form2
//
this.AutoScaleDimensions = new
System.Drawing.SizeF(12F, 25F);
this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(793,
561);
this.Controls.Add(this.textBox6);
this.Controls.Add(this.textBox5);
this.Controls.Add(this.textBox4);
this.Controls.Add(this.textBox3);
this.Controls.Add(this.textBox2);
this.Controls.Add(this.textBox1);
this.Controls.Add(this.label7);
this.Controls.Add(this.label6);
this.Controls.Add(this.label5);
this.Controls.Add(this.label4);
this.Controls.Add(this.label3);
this.Controls.Add(this.label13);
this.Controls.Add(this.label12);
this.Controls.Add(this.label11);
this.Controls.Add(this.label10);
this.Controls.Add(this.label9);
this.Controls.Add(this.label8);
this.Controls.Add(this.label2);
this.Controls.Add(this.button1);
this.Controls.Add(this.trackBar1);
this.Controls.Add(this.label1);
this.Font = new System.Drawing.Font("Microsoft
Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.Margin = new
System.Windows.Forms.Padding(4, 5, 4, 5);
this.Name = "InputDataForm";
this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Загальні вхідні дані";
this.Load += new
System.EventHandler(this.InputDataForm_Load);

```

```

((System.ComponentModel.ISupportInitialize)(this.trackBar
1)).EndInit();
this.ResumeLayout(false);
this.PerformLayout();
}

```

#endregion

```

private System.Windows.Forms.Label label1;
private System.Windows.Forms.TrackBar trackBar1;
private System.Windows.Forms.Button button1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.TextBox textBox1;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.TextBox textBox2;
private System.Windows.Forms.TextBox textBox3;
private System.Windows.Forms.TextBox textBox4;
private System.Windows.Forms.TextBox textBox5;
private System.Windows.Forms.TextBox textBox6;
private System.Windows.Forms.Label label8;
private System.Windows.Forms.Label label9;
private System.Windows.Forms.Label label10;
private System.Windows.Forms.Label label11;
private System.Windows.Forms.Label label12;
private System.Windows.Forms.Label label13;
}

```

ModelingForm.cs

```

using System;
using System.Drawing;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

```

namespace EffectiveEconomy

```

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

            public int _materialCounter;
            private int _manufacturerCounter;
            private int _consumerCounter;
            private int _tradeCounter;
            private int _daysCounter;

            public void RefreshCharts()
            {
                _daysCounter++;
                Random rand = new Random();
                EconomyEvents.ResultEventsString += "-----
- День " + _daysCounter + " -----\\n";

                if (_daysCounter == 1)//1
                {
                    InputData.VsInitialAmountOfMoney =
                    rand.Next(InputData.VsInitialAmountOfMoneyMin,
                    InputData.VsInitialAmountOfMoneyMax + 1);
                    InputData.VzvInitialAmountOfMoney =
                    rand.Next(InputData.VzvInitialAmountOfMoneyMin,
                    InputData.VzvInitialAmountOfMoneyMax + 1);
                }
            }
        }
    }
}

```

```

    InputData.VstInitialAmountOfMoney =
    rand.Next(InputData.VstInitialAmountOfMoneyMin,
    InputData.VstInitialAmountOfMoneyMax + 1);
    InputData.TtpInitialAmountOfMoney =
    rand.Next(InputData.TtpInitialAmountOfMoneyMin,
    InputData.TtpInitialAmountOfMoneyMax + 1);

    InputData.VsProcessCycleTime =
    rand.Next(InputData.VsProcessCycleTimeMin,
    InputData.VsProcessCycleTimeMax + 1);
    InputData.VsMarkupPercentage =
    rand.Next(InputData.VsMarkupPercentageMin,
    InputData.VsMarkupPercentageMax + 1);
    InputData.VzvProcessCycleTime =
    rand.Next(InputData.VzvProcessCycleTimeMin,
    InputData.VzvProcessCycleTimeMax + 1);
    InputData.VzvMarkupPercentage =
    rand.Next(InputData.VzvMarkupPercentageMin,
    InputData.VzvMarkupPercentageMax + 1);
    InputData.VstProcessCycleTime =
    rand.Next(InputData.VstProcessCycleTimeMin,
    InputData.VstProcessCycleTimeMax + 1);
    InputData.VstMarkupPercentage =
    rand.Next(InputData.VstMarkupPercentageMin,
    InputData.VstMarkupPercentageMax + 1);

    ModelingSettings.MaterialProducerRemainder =
    InputData.VsInitialAmountOfMoney;
    ModelingSettings.ManufacturerRemainder =
    InputData.VzvInitialAmountOfMoney;
    ModelingSettings.ConsumerRemainder =
    InputData.VstInitialAmountOfMoney;
    ModelingSettings.TradeAndServicesRemainder =
    InputData.TtpInitialAmountOfMoney;
    ModelingSettings.CommercialBankRemainder =
    InputData.CommercialBankAmount;
    ModelingSettings.NationalBankStock =
    InputData.NationalBankTotalAmount;
    ModelingSettings.NationalBankRemainder =
    InputData.NationalBankTotalAmount;
}

ModelingSettings.MaterialProducerComing = 0;
ModelingSettings.ManufacturerComing = 0;
ModelingSettings.ConsumerComing = 0;
ModelingSettings.TradeAndServicesComing = 0;
ModelingSettings.BudgetComing = 0;
ModelingSettings.CommercialBankComing = 0;
ModelingSettings.BuyersComing = 0;

ModelingSettings.MaterialProducerConsumption = 0;
ModelingSettings.ManufacturerConsumption = 0;
ModelingSettings.ConsumerConsumption = 0;
ModelingSettings.TradeAndServicesConsumption =
0;
ModelingSettings.BudgetConsumption = 0;
ModelingSettings.CommercialBankConsumption = 0;
ModelingSettings.BuyersConsumption = 0;
ModelingSettings.NationalBankConsumption = 0;

if (ModelingSettings.MaterialProducerRaw > 0)
{
    ModelingSettings.MaterialProducerComing =
    ModelingSettings.MaterialProducerRaw *
    InputData.VsMarkupPercentage / 100; //заработал деньги за
    товар

    ModelingSettings.ManufacturerRemainder -=
    ModelingSettings.MaterialProducerRaw *
    InputData.VsMarkupPercentage / 300; //отнимаем деньги у
    купивших товар
    ModelingSettings.ConsumerRemainder -=
    ModelingSettings.MaterialProducerRaw *
    InputData.VsMarkupPercentage / 300;
    ModelingSettings.TradeAndServicesRemainder -=

```

```

    ModelingSettings.MaterialProducerRaw *
    InputData.VsMarkupPercentage / 300;

    ModelingSettings.ManufacturerConsumption +=
    ModelingSettings.MaterialProducerRaw *
    InputData.VsMarkupPercentage / 300; //делаем расход
    купившим товар
    ModelingSettings.ConsumerConsumption +=
    ModelingSettings.MaterialProducerRaw *
    InputData.VsMarkupPercentage / 300;
    ModelingSettings.TradeAndServicesConsumption
    += ModelingSettings.MaterialProducerRaw *
    InputData.VsMarkupPercentage / 300;

    ModelingSettings.MaterialProducerRaw = 0;

    EconomyEvents.ResultEventsString +=
    EconomyEvents.Material +
    EconomyEvents.SellGoodsEvent;
    EconomyEvents.ResultEventsString +=
    EconomyEvents.Manufacturer +
    EconomyEvents.BuyGoodsEvent;
    EconomyEvents.ResultEventsString +=
    EconomyEvents.Consumer +
    EconomyEvents.BuyGoodsEvent;
    EconomyEvents.ResultEventsString +=
    EconomyEvents.Trade + EconomyEvents.BuyGoodsEvent;
}
else
{
    if (_materialCounter ==
    InputData.VsProcessCycleTime)
    {
        ModelingSettings.MaterialProducerConsumption +=
        ModelingSettings.MaterialProducerRemainder / 10;
        ModelingSettings.MaterialProducerRemainder -
        = ModelingSettings.MaterialProducerRemainder / 10;
        ModelingSettings.MaterialProducerRaw +=
        (ModelingSettings.MaterialProducerRemainder +
        ModelingSettings.MaterialProducerRemainder / 10);
        _materialCounter = 0;

        EconomyEvents.ResultEventsString +=
        EconomyEvents.Material +
        EconomyEvents.ProduceGoodsEvent;
    }
    else
    {
        _materialCounter++;
    }
}

if (ModelingSettings.ManufacturerRaw > 0)
{
    ModelingSettings.ManufacturerComing +=
    ModelingSettings.ManufacturerRaw *
    InputData.VzvMarkupPercentage / 100;

    ModelingSettings.MaterialProducerRemainder -=
    ModelingSettings.ManufacturerRaw *
    InputData.VzvMarkupPercentage / 300;
    ModelingSettings.ConsumerRemainder -=
    ModelingSettings.ManufacturerRaw *
    InputData.VzvMarkupPercentage / 300;
    ModelingSettings.TradeAndServicesRemainder -=
    ModelingSettings.ManufacturerRaw *
    InputData.VzvMarkupPercentage / 300;

    ModelingSettings.MaterialProducerConsumption
    += ModelingSettings.ManufacturerRaw *
    InputData.VzvMarkupPercentage / 300;
    ModelingSettings.ConsumerConsumption +=
    ModelingSettings.ManufacturerRaw *
    InputData.VzvMarkupPercentage / 300;

```

```

        ModelingSettings.TradeAndServicesConsumption
+= ModelingSettings.ManufacturerRaw *
InputData.VzvMarkupPercentage / 300;

        ModelingSettings.ManufacturerRaw = 0;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Manufacturer +
EconomyEvents.SellGoodsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Material +
EconomyEvents.BuyGoodsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Consumer +
EconomyEvents.BuyGoodsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Trade + EconomyEvents.BuyGoodsEvent;
    }
    else
    {
        if (_manufacturerCounter ==
InputData.VzvProcessCycleTime)
        {
            ModelingSettings.ManufacturerConsumption +=
ModelingSettings.ManufacturerRemainder / 10;
            ModelingSettings.ManufacturerRemainder -=
ModelingSettings.ManufacturerRemainder / 10;
            ModelingSettings.ManufacturerRaw +=
(ModelingSettings.ManufacturerRemainder +
ModelingSettings.ManufacturerRemainder / 10);

            ModelingSettings.ManufacturerComing = 0;

            EconomyEvents.ResultEventsString +=
EconomyEvents.Manufacturer +
EconomyEvents.ProduceGoodsEvent;
        }
        else
        {
            _manufacturerCounter++;
        }

        if (ModelingSettings.ConsumerRaw > 0)
        {
            ModelingSettings.ConsumerComing +=
ModelingSettings.ConsumerRaw *
InputData.VstMarkupPercentage / 100;

            ModelingSettings.TradeAndServicesRemainder -=
ModelingSettings.ConsumerRaw *
InputData.VstMarkupPercentage / 100;

            ModelingSettings.TradeAndServicesConsumption
+= ModelingSettings.ConsumerRaw *
InputData.VstMarkupPercentage / 100;

            ModelingSettings.ConsumerRaw = 0;

            EconomyEvents.ResultEventsString +=
EconomyEvents.Consumer +
EconomyEvents.SellGoodsEvent;
            EconomyEvents.ResultEventsString +=
EconomyEvents.Trade + EconomyEvents.BuyGoodsEvent;
        }
        else
        {
            if (_consumerCounter ==
InputData.VstProcessCycleTime)
            {
                ModelingSettings.ConsumerConsumption +=
ModelingSettings.ConsumerRemainder / 10;
                ModelingSettings.ConsumerRemainder -=
ModelingSettings.ConsumerRemainder / 10;
                ModelingSettings.ConsumerRaw +=
(ModelingSettings.ConsumerRemainder +
ModelingSettings.ConsumerRemainder / 10);

```

```

        _consumerCounter = 0;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Consumer +
EconomyEvents.ProduceGoodsEvent;
    }
    else
    {
        _consumerCounter++;
    }

    if (ModelingSettings.TradeAndServicesRaw > 0)//2
    {
        ModelingSettings.TradeAndServicesComing +=
ModelingSettings.TradeAndServicesRaw / 10;

        ModelingSettings.MaterialProducerConsumption
+= ModelingSettings.TradeAndServicesRaw / 70;
        ModelingSettings.ConsumerConsumption +=
ModelingSettings.TradeAndServicesRaw / 70;
        ModelingSettings.ManufacturerConsumption +=
ModelingSettings.TradeAndServicesRaw / 70;
        ModelingSettings.BudgetConsumption +=
ModelingSettings.TradeAndServicesRaw / 70;
        ModelingSettings.CommercialBankConsumption
+= ModelingSettings.TradeAndServicesRaw / 70;
        ModelingSettings.BuyersConsumption +=
ModelingSettings.BuyersRemainder *
InputData.MoneySpendingPercentage / 100;
        ModelingSettings.NationalBankConsumption +=
ModelingSettings.TradeAndServicesRaw / 70;

        ModelingSettings.MaterialProducerRemainder -=
ModelingSettings.TradeAndServicesRaw / 70;
        ModelingSettings.ConsumerRemainder -=
ModelingSettings.TradeAndServicesRaw / 70;
        ModelingSettings.ManufacturerRemainder -=
ModelingSettings.TradeAndServicesRaw / 70;
        ModelingSettings.BudgetRemainder -=
ModelingSettings.TradeAndServicesRaw / 70;
        ModelingSettings.CommercialBankRemainder -=
ModelingSettings.TradeAndServicesRaw / 70;
        ModelingSettings.BuyersRemainder -=
ModelingSettings.BuyersRemainder *
InputData.MoneySpendingPercentage / 100;
        ModelingSettings.NationalBankRemainder -=
ModelingSettings.TradeAndServicesRaw / 70;

        ModelingSettings.TradeAndServicesRaw = 0;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Trade + EconomyEvents.SellGoodsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Material +
EconomyEvents.BuyGoodsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Consumer +
EconomyEvents.BuyGoodsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Manufacturer +
EconomyEvents.BuyGoodsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Budget + EconomyEvents.BuyGoodsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Commercial +
EconomyEvents.BuyGoodsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Buyers + EconomyEvents.BuyGoodsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Bank + EconomyEvents.BuyGoodsEvent;
    }
    else
    {
        if (_tradeCounter == 1)
        {

```

```

ModelingSettings.TradeAndServicesConsumption +=
ModelingSettings.TradeAndServicesRemainder / 10;
    ModelingSettings.TradeAndServicesRemainder
-= ModelingSettings.TradeAndServicesRemainder / 10;
    ModelingSettings.TradeAndServicesRaw +=
(ModelingSettings.TradeAndServicesRemainder +
ModelingSettings.TradeAndServicesRemainder / 10);
    _tradeCounter = 0;

    EconomyEvents.ResultEventsString +=
EconomyEvents.Trade +
EconomyEvents.ProduceGoodsEvent;
    }
    else
        _tradeCounter++;
    }

    if (ModelingSettings.MaterialProducerComing > 0)
    {
        ModelingSettings.MaterialProducerRemainder +=
ModelingSettings.MaterialProducerComing -
        (ModelingSettings.MaterialProducerComing *
InputData.TaxMoneyPercentage / 100
        + ModelingSettings.MaterialProducerComing *
InputData.PayWagesPercentage / 100);

        ModelingSettings.BudgetComing +=
ModelingSettings.MaterialProducerComing -
ModelingSettings.MaterialProducerComing *
InputData.TaxMoneyPercentage / 100;
        ModelingSettings.BudgetRemainder +=
ModelingSettings.MaterialProducerComing -
ModelingSettings.MaterialProducerComing *
InputData.TaxMoneyPercentage / 100;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Material +
EconomyEvents.CalculateSalariesEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Material +
EconomyEvents.ProcessPaymentsEvent;
    }

    if (ModelingSettings.ManufacturerComing > 0) // 3 5
    {
        ModelingSettings.ManufacturerRemainder +=
ModelingSettings.ManufacturerComing -
        (ModelingSettings.ManufacturerComing *
InputData.TaxMoneyPercentage / 100
        + ModelingSettings.ManufacturerComing *
InputData.PayWagesPercentage / 100);

        ModelingSettings.BudgetComing +=
ModelingSettings.ManufacturerComing -
ModelingSettings.ManufacturerComing *
InputData.TaxMoneyPercentage / 100;
        ModelingSettings.BudgetRemainder +=
ModelingSettings.ManufacturerComing -
ModelingSettings.ManufacturerComing *
InputData.TaxMoneyPercentage / 100;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Manufacturer +
EconomyEvents.CalculateSalariesEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Manufacturer +
EconomyEvents.ProcessPaymentsEvent;
    }

    if (ModelingSettings.ConsumerComing > 0) // 3
    {
        ModelingSettings.ConsumerRemainder +=
ModelingSettings.ConsumerComing -
        (ModelingSettings.ConsumerComing *
InputData.TaxMoneyPercentage / 100

```

```

        + ModelingSettings.ConsumerComing *
InputData.PayWagesPercentage / 100);

        ModelingSettings.BudgetComing +=
ModelingSettings.ConsumerComing -
ModelingSettings.ConsumerComing *
InputData.TaxMoneyPercentage / 100;
        ModelingSettings.BudgetRemainder +=
ModelingSettings.ConsumerComing -
ModelingSettings.ConsumerComing *
InputData.TaxMoneyPercentage / 100;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Consumer +
EconomyEvents.CalculateSalariesEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Consumer +
EconomyEvents.ProcessPaymentsEvent;
    }

    if (ModelingSettings.TradeAndServicesComing >
0) // 3
    {
        ModelingSettings.TradeAndServicesRemainder +=
ModelingSettings.TradeAndServicesComing -
        (ModelingSettings.TradeAndServicesComing *
InputData.TaxMoneyPercentage / 100
        + ModelingSettings.TradeAndServicesComing *
InputData.PayWagesPercentage / 100);

        ModelingSettings.BudgetComing +=
ModelingSettings.TradeAndServicesComing -
ModelingSettings.TradeAndServicesComing *
InputData.TaxMoneyPercentage / 100;
        ModelingSettings.BudgetRemainder +=
ModelingSettings.TradeAndServicesComing -
ModelingSettings.TradeAndServicesComing *
InputData.TaxMoneyPercentage / 100;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Trade +
EconomyEvents.CalculateSalariesEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Trade +
EconomyEvents.ProcessPaymentsEvent;
    }

    if (ModelingSettings.MaterialProducerRemainder <
(InputData.VsInitialAmountOfMoney / 10))
    {
        ModelingSettings.MaterialProducerRemainder +=
InputData.CommercialBankAmount *
InputData.InterestOnLoans / 70;
        ModelingSettings.CommercialBankRemainder -=
ModelingSettings.CommercialBankRemainder / 10;

        ModelingSettings.CommercialBankConsumption
+= ModelingSettings.CommercialBankRemainder / 10;
        ModelingSettings.MaterialProducerComing +=
InputData.CommercialBankAmount *
InputData.InterestOnLoans / 70;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Material +
EconomyEvents.TakeCreditEvent;
    }

    if (ModelingSettings.ManufacturerRemainder <
(InputData.VzvInitialAmountOfMoney / 10))
    {
        ModelingSettings.ManufacturerRemainder +=
InputData.CommercialBankAmount *
InputData.InterestOnLoans / 70;
        ModelingSettings.CommercialBankRemainder -=
ModelingSettings.CommercialBankRemainder / 10;

```

```

        ModelingSettings.CommercialBankConsumption
+= ModelingSettings.CommercialBankRemainder / 10;
        ModelingSettings.ManufacturerComing +=
InputData.CommercialBankAmount *
InputData.InterestOnLoans / 70;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Manufacturer +
EconomyEvents.TakeCreditEvent;
    }

    if (ModelingSettings.ConsumerRemainder <
(InputData.VstInitialAmountOfMoney / 10))
    {
        ModelingSettings.ConsumerRemainder +=
InputData.CommercialBankAmount *
InputData.InterestOnLoans / 70;
        ModelingSettings.CommercialBankRemainder -=
ModelingSettings.CommercialBankRemainder / 10;

        ModelingSettings.CommercialBankConsumption
+= ModelingSettings.CommercialBankRemainder / 10;
        ModelingSettings.ConsumerComing +=
InputData.CommercialBankAmount *
InputData.InterestOnLoans / 70;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Consumer +
EconomyEvents.TakeCreditEvent;
    }

    if (ModelingSettings.TradeAndServicesRemainder <
(InputData.TtpInitialAmountOfMoney / 10))
    {
        ModelingSettings.TradeAndServicesRemainder +=
InputData.CommercialBankAmount *
InputData.InterestOnLoans / 70;
        ModelingSettings.CommercialBankRemainder -=
ModelingSettings.CommercialBankRemainder / 10;

        ModelingSettings.CommercialBankConsumption
+= ModelingSettings.CommercialBankRemainder / 10;
        ModelingSettings.TradeAndServicesComing +=
InputData.CommercialBankAmount *
InputData.InterestOnLoans / 70;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Trade + EconomyEvents.TakeCreditEvent;
    }

    if (ModelingSettings.CommercialBankRemainder <
(InputData.CommercialBankAmount / 10))
    {
        ModelingSettings.CommercialBankRemainder +=
InputData.NationalBankTotalAmount *
InputData.NationalBankRate / 70;
        ModelingSettings.NationalBankRemainder -=
ModelingSettings.NationalBankRemainder / 7;

        ModelingSettings.NationalBankConsumption +=
ModelingSettings.NationalBankRemainder / 7;
        ModelingSettings.CommercialBankComing +=
InputData.NationalBankTotalAmount *
InputData.NationalBankRate / 70;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Commercial +
EconomyEvents.TakeCreditEvent;
    }

    if (ModelingSettings.BudgetRemainder > 300)
    {
        ModelingSettings.BuyersComing +=
ModelingSettings.BudgetRemainder / 2;

```

```

        ModelingSettings.BuyersRemainder +=
ModelingSettings.BudgetRemainder / 2;

        ModelingSettings.BudgetConsumption +=
ModelingSettings.BudgetRemainder;
        ModelingSettings.BudgetRemainder = 0;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Budget +
EconomyEvents.CalculatePaymentsEvent;
        EconomyEvents.ResultEventsString +=
EconomyEvents.Budget +
EconomyEvents.PaySalariesEvent;
    }

    if (rand.Next(1, 100) <
InputData.LikelihoodOfAnEmergency)
    {
        ModelingSettings.BudgetConsumption +=
ModelingSettings.BudgetRemainder / 2;
        ModelingSettings.BudgetRemainder -=
ModelingSettings.BudgetRemainder / 2;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Budget +
EconomyEvents.SolveProblemEvent;
    }

    if (ModelingSettings.NationalBankRemainder <
(InputData.NationalBankTotalAmount / 10))
    {
        ModelingSettings.NationalBankStock += 100;
        ModelingSettings.NationalBankEmission += 100;
        ModelingSettings.NationalBankRemainder +=
100;

        EconomyEvents.ResultEventsString +=
EconomyEvents.Bank +
EconomyEvents.IncreaseEmissionEvent;
    }

    chart1.Series["Data"].Points.DataBindXY(new[] { "",
"", "", "" }, new[] {
ModelingSettings.MaterialProducerComing,
ModelingSettings.MaterialProducerConsumption,
ModelingSettings.MaterialProducerRaw,
ModelingSettings.MaterialProducerRemainder });
    chart1.Series["Series2"].Points.DataBindXY(new[] {
"", "", "", "" }, new[] {
ModelingSettings.ManufacturerComing,
ModelingSettings.ManufacturerConsumption,
ModelingSettings.ManufacturerRaw,
ModelingSettings.ManufacturerRemainder });
    chart1.Series["Series3"].Points.DataBindXY(new[] {
"", "", "", "" }, new[] { ModelingSettings.ConsumerComing,
ModelingSettings.ConsumerConsumption,
ModelingSettings.ConsumerRaw,
ModelingSettings.ConsumerRemainder });
    chart1.Series["Series4"].Points.DataBindXY(new[] {
"", "", "", "" }, new[] {
ModelingSettings.TradeAndServicesComing,
ModelingSettings.TradeAndServicesConsumption,
ModelingSettings.TradeAndServicesRaw,
ModelingSettings.TradeAndServicesRemainder });
    chart1.Series["Series5"].Points.DataBindXY(new[] {
"", "", "" }, new[] { ModelingSettings.BudgetComing,
ModelingSettings.BudgetConsumption,
ModelingSettings.BudgetRemainder });
    chart1.Series["Series6"].Points.DataBindXY(new[] {
"", "", "" }, new[] {
ModelingSettings.CommercialBankComing,
ModelingSettings.CommercialBankConsumption,
ModelingSettings.CommercialBankRemainder });
    chart1.Series["Series7"].Points.DataBindXY(new[] {
"", "", "" }, new[] { ModelingSettings.BuyersComing,

```

```

ModelingSettings.BuyersConsumption,
ModelingSettings.BuyersRemainder });
    chart1.Series["Series8"].Points.DataBindXY(new[] {
    "", "", "", "" }, new[] {
ModelingSettings.NationalBankStock,
ModelingSettings.NationalBankConsumption,
ModelingSettings.NationalBankEmission,
ModelingSettings.NationalBankRemainder });

    if (ModelingSettings.NationalBankEmission ==
1000)
    {
        ModelingSettings.NationalBankEmission -= 500;
    }

}

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

private void
переглянутиToolStripMenuItem_Click(object sender,
EventArgs e)
{
    var newForm = new EventsForm();
    newForm.Show();
}

private void
startStepByStepModelingToolStripMenuItem_Click(object
sender, EventArgs e)
{
    timer1.Start();

startStepByStepModelingToolStripMenuItem.Enabled =
false;
    stopByStepModelingToolStripMenuItem.Enabled =
true;
}

private void
stopStepByStepModelingToolStripMenuItem_Click(object
sender, EventArgs e)
{
    timer1.Stop();

startStepByStepModelingToolStripMenuItem.Enabled =
true;
    stopByStepModelingToolStripMenuItem.Enabled =
false;
}

private void
modelOneStepToolStripMenuItem_Click(object sender,
EventArgs e)
{
    RefreshCharts();
}

private void
updateInputDataToolStripMenuItem_Click(object sender,
EventArgs e)
{
    if (timer1.Enabled)
    {
        timer1.Stop();
        var result = MessageBox.Show("Зупинити
покрокове моделювання та змінити вхідні дані?",
"Увага", MessageBoxButtons.YesNo);
        if (result == DialogResult.Yes)
        {
            var newForm = new InputDataForm();
            newForm.Show();

```

```

        Hide();
    }
    else
        timer1.Start();
}
else
{
    var newForm = new InputDataForm();
    newForm.Show();
    Hide();
}

}

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

private void Form1_Load(object sender, EventArgs e)
{
    chart1.Series["Data"].Points.AddXY("",
ModelingSettings.MaterialProducerComing);
    chart1.Series["Data"].Points.AddXY("",
ModelingSettings.MaterialProducerConsumption);
    chart1.Series["Data"].Points.AddXY("",
ModelingSettings.MaterialProducerRaw);
    chart1.Series["Data"].Points.AddXY("",
ModelingSettings.MaterialProducerRemainder);
    chart1.ChartAreas[0].AxisY.Maximum = 500;
    chart1.Series["Data"].Palette =
ChartColorPalette.Excel;
    chart1.Legends[0].CustomItems[0].Color =
Color.Orchid;
    chart1.Legends[0].CustomItems[1].Color =
Color.Brown;
    chart1.Legends[0].CustomItems[2].Color =
Color.Beige;
    chart1.Legends[0].CustomItems[3].Color =
Color.Aqua;

    chart1.Series["Series2"].Points.AddXY("",
ModelingSettings.ManufacturerComing);
    chart1.Series["Series2"].Points.AddXY("",
ModelingSettings.ManufacturerConsumption);
    chart1.Series["Series2"].Points.AddXY("",
ModelingSettings.ManufacturerRaw);
    chart1.Series["Series2"].Points.AddXY("",
ModelingSettings.ManufacturerRemainder);
    chart1.ChartAreas[1].AxisY.Maximum = 500;
    chart1.Series["Series2"].Palette =
ChartColorPalette.Excel;
    chart1.Legends[1].CustomItems[0].Color =
Color.Orchid;
    chart1.Legends[1].CustomItems[1].Color =
Color.Brown;
    chart1.Legends[1].CustomItems[2].Color =
Color.Beige;
    chart1.Legends[1].CustomItems[3].Color =
Color.Aqua;

    chart1.Series["Series3"].Points.AddXY("",
ModelingSettings.ConsumerComing);
    chart1.Series["Series3"].Points.AddXY("",
ModelingSettings.ConsumerConsumption);
    chart1.Series["Series3"].Points.AddXY("",
ModelingSettings.ConsumerRaw);
    chart1.Series["Series3"].Points.AddXY("",
ModelingSettings.ConsumerRemainder);
    chart1.ChartAreas[2].AxisY.Maximum = 500;
    chart1.Series["Series3"].Palette =
ChartColorPalette.Excel;
    chart1.Legends[2].CustomItems[0].Color =
Color.Orchid;

```



```

        chart1.Legends[2].CustomItems[1].Color =
Color.Brown;
        chart1.Legends[2].CustomItems[2].Color =
Color.Beige;
        chart1.Legends[2].CustomItems[3].Color =
Color.Aqua;

        chart1.Series["Series4"].Points.AddXY("",
ModelingSettings.TradeAndServicesComing);
        chart1.Series["Series4"].Points.AddXY("",
ModelingSettings.TradeAndServicesConsumption);
        chart1.Series["Series4"].Points.AddXY("",
ModelingSettings.TradeAndServicesRaw);
        chart1.Series["Series4"].Points.AddXY("",
ModelingSettings.TradeAndServicesRemainder);
        chart1.ChartAreas[3].AxisY.Maximum = 500;
        chart1.Series["Series4"].Palette =
ChartColorPalette.Excel;
        chart1.Legends[3].CustomItems[0].Color =
Color.Orchid;
        chart1.Legends[3].CustomItems[1].Color =
Color.Brown;
        chart1.Legends[3].CustomItems[2].Color =
Color.Beige;
        chart1.Legends[3].CustomItems[3].Color =
Color.Aqua;

        chart1.Series["Series5"].Points.AddXY("",
ModelingSettings.BuyersComing);
        chart1.Series["Series5"].Points.AddXY("",
ModelingSettings.BuyersConsumption);
        chart1.Series["Series5"].Points.AddXY("",
ModelingSettings.BuyersRemainder);
        chart1.ChartAreas[4].AxisY.Maximum = 500;
        chart1.Series["Series5"].Palette =
ChartColorPalette.Excel;
        chart1.Legends[4].CustomItems[0].Color =
Color.Orchid;
        chart1.Legends[4].CustomItems[1].Color =
Color.Brown;
        chart1.Legends[4].CustomItems[2].Color =
Color.Beige;

        chart1.Series["Series6"].Points.AddXY("",
ModelingSettings.BudgetComing);
        chart1.Series["Series6"].Points.AddXY("",
ModelingSettings.BudgetConsumption);
        chart1.Series["Series6"].Points.AddXY("",
ModelingSettings.BudgetRemainder);
        chart1.ChartAreas[5].AxisY.Maximum = 500;
        chart1.Series["Series6"].Palette =
ChartColorPalette.Excel;
        chart1.Legends[5].CustomItems[0].Color =
Color.Orchid;
        chart1.Legends[5].CustomItems[1].Color =
Color.Brown;
        chart1.Legends[5].CustomItems[2].Color =
Color.Beige;

        chart1.Series["Series7"].Points.AddXY("",
ModelingSettings.CommercialBankComing);
        chart1.Series["Series7"].Points.AddXY("",
ModelingSettings.CommercialBankConsumption);
        chart1.Series["Series7"].Points.AddXY("",
ModelingSettings.CommercialBankRemainder);
        chart1.ChartAreas[6].AxisY.Maximum = 500;
        chart1.Series["Series7"].Palette =
ChartColorPalette.Excel;
        chart1.Legends[6].CustomItems[0].Color =
Color.Orchid;
        chart1.Legends[6].CustomItems[1].Color =
Color.Brown;
        chart1.Legends[6].CustomItems[2].Color =
Color.Beige;

```

```

        chart1.Series["Series8"].Points.AddXY("",
ModelingSettings.NationalBankStock);
        chart1.Series["Series8"].Points.AddXY("",
ModelingSettings.NationalBankConsumption);
        chart1.Series["Series8"].Points.AddXY("",
ModelingSettings.NationalBankEmission);
        chart1.Series["Series8"].Points.AddXY("",
ModelingSettings.NationalBankRemainder);
        chart1.ChartAreas[7].AxisY.Maximum = 500;
        chart1.Series["Series8"].Palette =
ChartColorPalette.Excel;
        chart1.Legends[7].CustomItems[0].Color =
Color.Orchid;
        chart1.Legends[7].CustomItems[1].Color =
Color.Brown;
        chart1.Legends[7].CustomItems[2].Color =
Color.Beige;
        chart1.Legends[7].CustomItems[3].Color =
Color.Aqua;

```

```

startStepByStepModelingToolStripMenuItem.Enabled =
true;
stopByStepModelingToolStripMenuItem.Enabled =
false;

```

```

        timer1.Interval = 1000 / InputData.Speed;
    }

```

```

private void chart1_Click(object sender, EventArgs e)
{
}

```

```

private void
showMoreInfoAboutProgramToolStripMenuItem_Click(object sender, EventArgs e)
{
    var newForm = new ProgramInfoForm();
    newForm.Show();
    Hide();
}

```

```

private void
rawProducersToolStripMenuItem_Click(object sender, EventArgs e)
{
    var newForm = new RawProducerInputDataForm();
    newForm.Show();
    Hide();
}

```

```

private void
producerMeansProducerToolStripMenuItem_Click(object sender, EventArgs e)
{
    var newForm = new
CapitalGoodsProducerInputDataForm();
    newForm.Show();
    Hide();
}

```

```

private void
goodProducerToolStripMenuItem_Click(object sender, EventArgs e)
{
    var newForm = new
ConsumerGoodsProducerInputDataForm();
    newForm.Show();
    Hide();
}

```

```

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

```

```

        var newForm = new
TradeAndServicesInputDataForm();
        newForm.Show();
        Hide();
    }

    private void otherToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        var newForm = new GeneralInputDataForm();
        newForm.Show();
        Hide();
    }
}

```

ModelingSettings.cs

namespace EffectiveEconomy

```

{
    public static class ModelingSettings
    {
        public static int MaterialProducerComing = 0;
        public static int MaterialProducerConsumption = 0;
        public static int MaterialProducerRaw = 0;
        public static int MaterialProducerRemainder = 0;

        public static int ManufacturerComing = 0;
        public static int ManufacturerConsumption = 0;
        public static int ManufacturerRaw = 0;
        public static int ManufacturerRemainder = 0;

        public static int ConsumerComing = 0;
        public static int ConsumerConsumption = 0;
        public static int ConsumerRaw = 0;
        public static int ConsumerRemainder = 0;

        public static int TradeAndServicesComing = 0;
        public static int TradeAndServicesConsumption = 0;
        public static int TradeAndServicesRaw = 0;
        public static int TradeAndServicesRemainder = 0;

        public static int BudgetComing = 0;
        public static int BudgetConsumption = 0;
        public static int BudgetRemainder = 0;

        public static int CommercialBankComing = 0;
        public static int CommercialBankConsumption = 0;
        public static int CommercialBankRemainder = 0;

        public static int BuyersComing = 0;
        public static int BuyersConsumption = 0;
        public static int BuyersRemainder = 0;

        public static int NationalBankStock = 0;
        public static int NationalBankConsumption = 0;
        public static int NationalBankEmission = 0;
        public static int NationalBankRemainder = 0;
    }
}

Program.cs
using System;
using System.Windows.Forms;

namespace EffectiveEconomy
{
    static class Program

```

```

{
    /// <summary>
    /// Главная точка входа для приложения.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();

        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new ModelingForm());
    }
}

ProgramInfoForm.cs
using System;
using System.Drawing;
using System.Windows.Forms;

namespace EffectiveEconomy
{
    public partial class ProgramInfoForm : Form
    {
        public ProgramInfoForm()
        {
            InitializeComponent();

            private void Form3_Load(object sender, EventArgs e)
            {
                const int newFontSize = 14;
                richTextBox1.Select(0, 37);
                richTextBox1.SelectionAlignment =
HorizontalAlignment.Center;
                richTextBox1.SelectionFont = new
Font(richTextBox1.Font.FontFamily, newFontSize,
FontStyle.Bold);
                richTextBox1.Select(0, 0);
            }
        }
    }

RawProducerInputDataForm.cs
using System;
using System.Windows.Forms;

namespace EffectiveEconomy
{
    public partial class RawProducerInputDataForm : Form
    {
        public RawProducerInputDataForm()
        {
            InitializeComponent();

            private void label1_Click(object sender, EventArgs e)
            {

            }

            private void button1_Click(object sender, EventArgs e)
            {
                InputData.VsInitialAmountOfMoneyMin =
int.Parse(textBox1.Text);
                InputData.VsInitialAmountOfMoneyMax =
int.Parse(textBox2.Text);
                InputData.VsProcessCycleTimeMin =

```

```

int.Parse(textBox4.Text);
    InputData.VsProcessCycleTimeMax =
int.Parse(textBox5.Text);
    InputData.VsMarkupPercentageMin =
int.Parse(textBox6.Text);
    InputData.VsMarkupPercentageMax =
int.Parse(textBox7.Text);

    var rand = new Random();
    InputData.VsProcessCycleTime =
rand.Next(InputData.VsProcessCycleTimeMin,
InputData.VsProcessCycleTimeMax + 1);
    InputData.VsMarkupPercentage =
rand.Next(InputData.VsMarkupPercentageMin,
InputData.VsMarkupPercentageMax + 1);

    var newForm = new ModelingForm();
    newForm.Show();
    Hide();
}

private void Form5_Load(object sender, EventArgs e)
{
    textBox1.Text =
InputData.VsInitialAmountOfMoneyMin.ToString();
    textBox2.Text =
InputData.VsInitialAmountOfMoneyMax.ToString();
    textBox4.Text =
InputData.VsProcessCycleTimeMin.ToString();
    textBox5.Text =
InputData.VsProcessCycleTimeMax.ToString();
    textBox6.Text =
InputData.VsMarkupPercentageMin.ToString();
    textBox7.Text =
InputData.VsMarkupPercentageMax.ToString();
}
}
TradeAndServicesInputDataForm.cs
using System;
using System.Windows.Forms;

namespace EffectiveEconomy
{
    public partial class TradeAndServicesInputDataForm :
Form
    {
        public TradeAndServicesInputDataForm()
        {
            InitializeComponent();
        }

        private void textBox3_TextChanged(object sender,
EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            InputData.TtpInitialAmountOfMoneyMin =
int.Parse(textBox3.Text);
            InputData.TtpInitialAmountOfMoneyMax =
int.Parse(textBox4.Text);
            InputData.TtpPercentageOfPurchasesGoods =
int.Parse(textBox9.Text);

            var newForm = new ModelingForm();
            newForm.Show();
            Hide();
        }

        private void Form8_Load(object sender, EventArgs e)
        {
            textBox3.Text =
InputData.TtpInitialAmountOfMoneyMin.ToString();
            textBox4.Text =
InputData.TtpInitialAmountOfMoneyMax.ToString();
            textBox9.Text =
InputData.TtpPercentageOfPurchasesGoods.ToString();
        }
    }
}

```

```

var newForm = new ModelingForm();
newForm.Show();
Hide();
}

private void Form8_Load(object sender, EventArgs e)
{
    textBox3.Text =
InputData.TtpInitialAmountOfMoneyMin.ToString();
    textBox4.Text =
InputData.TtpInitialAmountOfMoneyMax.ToString();
    textBox9.Text =
InputData.TtpPercentageOfPurchasesGoods.ToString();
}
}
RefreshChartsUnitTests.cs
using System;
using System.Windows.Forms;

namespace EffectiveEconomy
{
    public partial class TradeAndServicesInputDataForm :
Form
    {
        public TradeAndServicesInputDataForm()
        {
            InitializeComponent();
        }

        private void textBox3_TextChanged(object sender,
EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            InputData.TtpInitialAmountOfMoneyMin =
int.Parse(textBox3.Text);
            InputData.TtpInitialAmountOfMoneyMax =
int.Parse(textBox4.Text);
            InputData.TtpPercentageOfPurchasesGoods =
int.Parse(textBox9.Text);

            var newForm = new ModelingForm();
            newForm.Show();
            Hide();
        }

        private void Form8_Load(object sender, EventArgs e)
        {
            textBox3.Text =
InputData.TtpInitialAmountOfMoneyMin.ToString();
            textBox4.Text =
InputData.TtpInitialAmountOfMoneyMax.ToString();
            textBox9.Text =
InputData.TtpPercentageOfPurchasesGoods.ToString();
        }
    }
}

```