МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра Комп’ютерні інформаційні технології

«ДО ЗАХИСТУ»

Завідувач кафедри

 \_\_\_\_\_\_\_\_ Вадим ГОРЯЧКІН

 (підпис) (ПІБ)

2022р.\_\_\_\_\_\_\_ «\_\_\_\_\_»

ДИПЛОМНА РОБОТА

на здобуття ОС «бакалавр»

Галузь знань 12 Інформаційні технології

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(повна назва)

Тема «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

Theme «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

(theme in English)

Керівник дипломної роботи \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада) (підпис) (ПІБ)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 (посада)       (підпис) (ПІБ)

Студент групи \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

   (номер групи) (підпис) (ПІБ)

Student \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Family name)

Дніпро 2022

**Український державний університет науки і технологій**

Факультет “Комп’ютерні технології і системи”

Кафедра “Комп’ютерні інформаційні технології”

Спеціальність “Інженерія програмного забезпечення”

«ДО ЗАХИСТУ»

Завідувач кафедри

\_\_\_\_\_\_\_\_ Вадим ГОРЯЧКІН

 (підпис) (ПІБ)

2022р.\_\_\_\_\_\_\_ «\_\_\_\_\_»

ЗАВДАННЯ

до дипломної роботи на на здобуття ОС «бакалавр»

студента групи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(номер групи) (ПІБ)

1 Тема дипломної роботи: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ затверджена наказом по університету від «\_\_\_» \_\_\_\_\_\_\_\_ 201\_\_ р. № \_\_.

2 Термін подання студентом закінченої роботи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

4 Зміст пояснювальної записки (перелік питань до розробки) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Перелік демонстраційного матеріалу \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Реферат**

Пояснювальна записка складається з 8 розділів:

– вступ – в даному розділі описується сутність розробки, її актуальність. Складається з 1 сторінки;

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

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

– тестування та налагодження – включає в себе вибір стратегії тестування, опис тестів методами «чорного» та «білого» ящика. Також аналіз помилок їх вплив на систему та вирішення проблеми. Складається з 36 сторінок;

– охорона праці. Складається з N сторінок;

– висновки. Складається з 1 сторінки;

– список літератури – включає в себе бібліографічний список використаної літератури. Складає 2 сторінки;

– додатки – містить технічне завдання і робочий проект. Кількість таблиць: N штук. Кількість рисунків: N штук.

Ключові слова: економіка, макроекономіка, ефективна економічна модель, моделювання, налог, зарплата, економічна діяльність.

ЗМІСТ

[Перелік умовних познак, символів, скорочень і термінів 5](#_Toc102657411)

[Вступ 6](#_Toc102657412)

[1 Збір та аналіз вимог 7](#_Toc102657413)

[1.1 Опис аналогів 9](#_Toc102657414)

[2 Зовнішнє проектування 12](#_Toc102657415)

[2.1 Функціональне призначення 12](#_Toc102657416)

[2.2 Експлуатаційне призначення 12](#_Toc102657417)

[2.3 Функціональні вимоги 12](#_Toc102657418)

[2.4 Загальні вихідні дані 12](#_Toc102657419)

[2.5 Суб’єкти економічної діяльності 13](#_Toc102657420)

[3 ВНУТРІШНЄ ПРОЕКТУВАННЯ 16](#_Toc102657421)

[3.1 Об’єктно-орієнтоване програмування 16](#_Toc102657422)

[3.2 Опис відповідальності класів 18](#_Toc102657423)

[3.3 Опис зв’язків класів 19](#_Toc102657424)

[3.4 Діаграма класів програми 19](#_Toc102657425)

[3.5 Опис специфікацій класів 19](#_Toc102657426)

[3.5.1 Опис специфікацій класу Program 19](#_Toc102657427)

[3.5.2 Опис специфікацій класу Form1 20](#_Toc102657428)

[3.5.3 Опис специфікацій класу Form2 20](#_Toc102657429)

[3.5.4 Опис специфікацій класу Form3 20](#_Toc102657430)

[3.5.5 Опис специфікацій класу Form4 20](#_Toc102657431)

[3.5.6 Опис специфікацій класу Form5 21](#_Toc102657432)

[3.5.7 Опис специфікацій класу Form6 21](#_Toc102657433)

[3.5.8 Опис специфікацій класу Form7 21](#_Toc102657434)

[3.5.9 Опис специфікацій класу Form8 21](#_Toc102657435)

[3.5.10 Опис специфікацій класу Form9 21](#_Toc102657436)

[3.5.11 Опис специфікацій класу InputData 22](#_Toc102657437)

[3.5.12 Опис специфікацій класу Сharacteristics 23](#_Toc102657438)

[3.5.13 Опис специфікацій класу Economy\_events 24](#_Toc102657439)

[3.6 Опис інтерфейсної частини класів 24](#_Toc102657440)

[3.7 Проектування інтерфейсу користувача 24](#_Toc102657441)

[3.8 Аналіз проекту 25](#_Toc102657442)

[3.9 Вибір мови програмування 26](#_Toc102657443)

# Перелік умовних познак, символів, скорочень і термінів

# Вступ

# 1 Збір та аналіз вимог

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

Методи збору вимог до програмного забезпечення При вивченні предметної області та збору вимог про майбутній об’єкт інформатизації використовується методологія системного аналізу. На цій стадії розробники повинні уточнити межі вивчення функціональних вимог до програми, вхідні та вихідні дані для функціонування програми та визначити коло користувачів майбутньої програми.

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

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

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

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

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

Бізнес процес для моделювання макроекономічної системи виконується наступним чином:

1. Користувач задає вхідні дані
2. Відбувається моделювання системи із виводом результатів на екран у вигляді діаграм
3. Запис усіх розрахунків до файлу

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

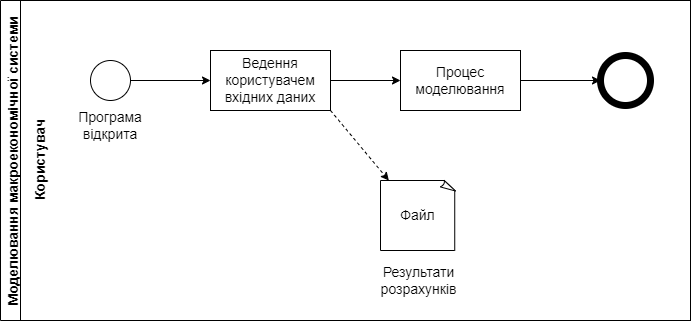


Рисунок 1. - Бізнес-процес моделювання макроекономічної системи

## 1.1 Опис аналогів

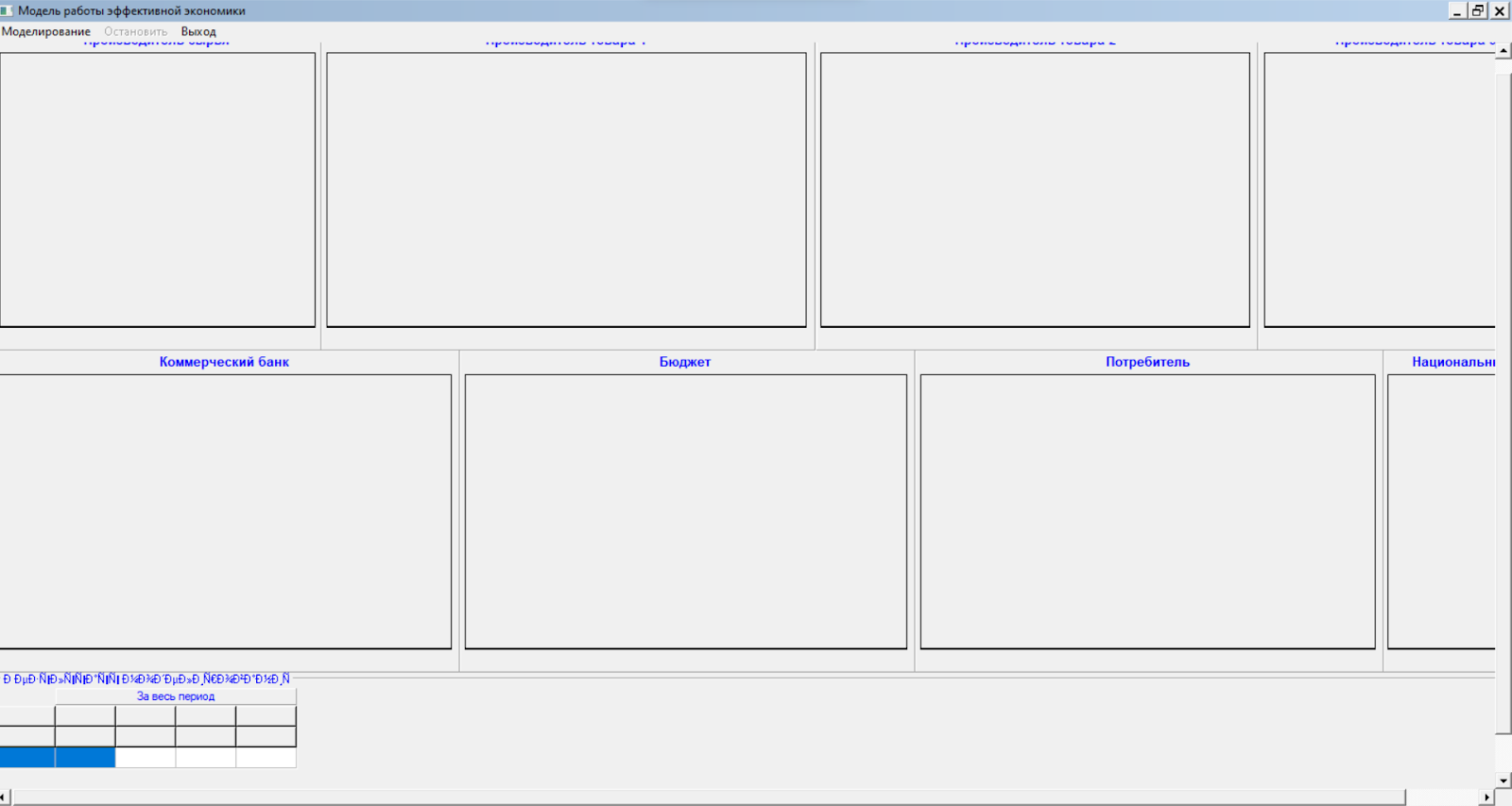
Існує тільки один аналог для моделювання макроекономічної моделі на базі безкризової економічної моделі та на базі класичної ринкової моделі - **SimpleModel**.

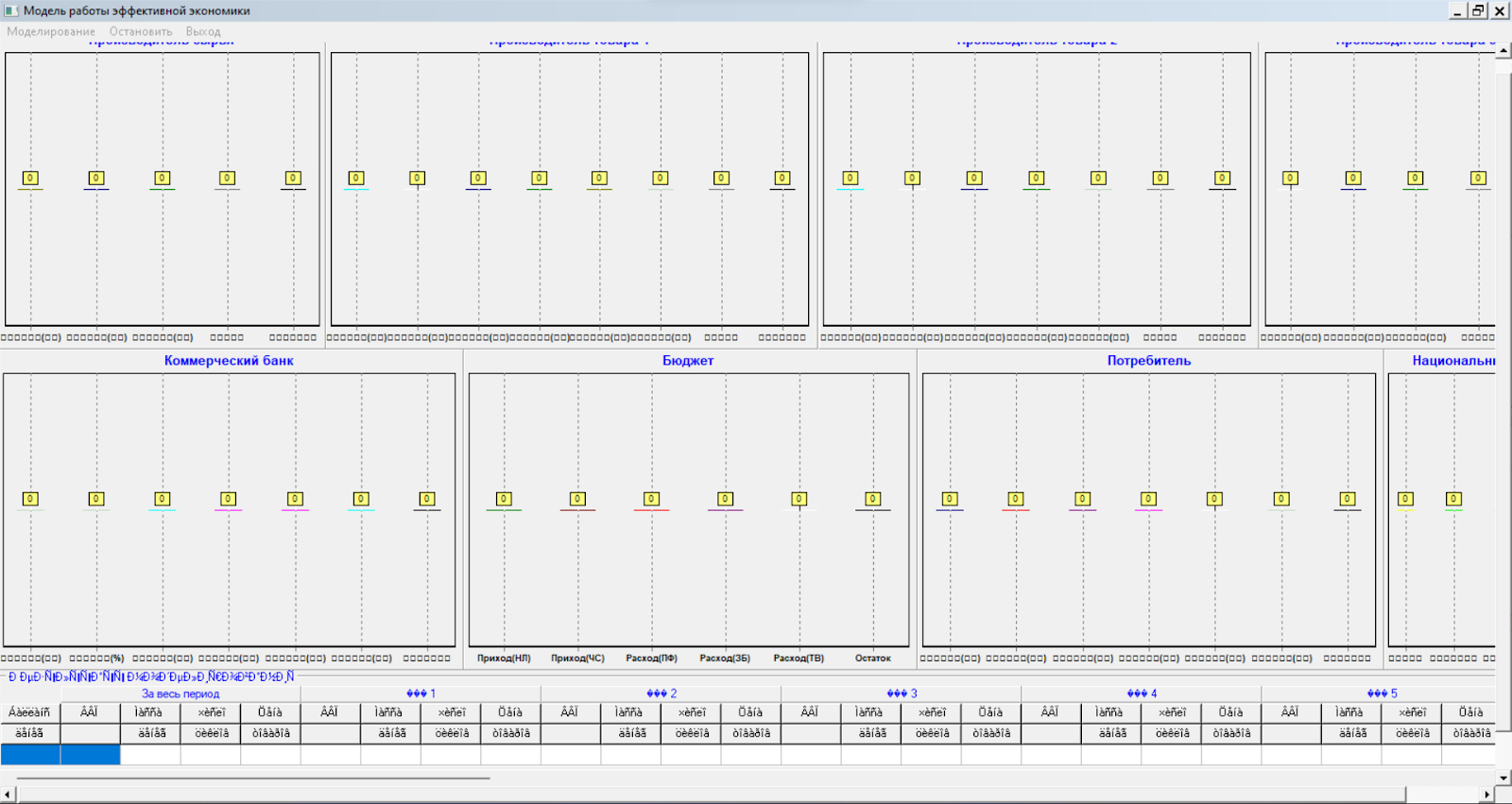
SimpleModel - перша версія додатку, яка була розроблена з метою моделювання макроекономічної системи.

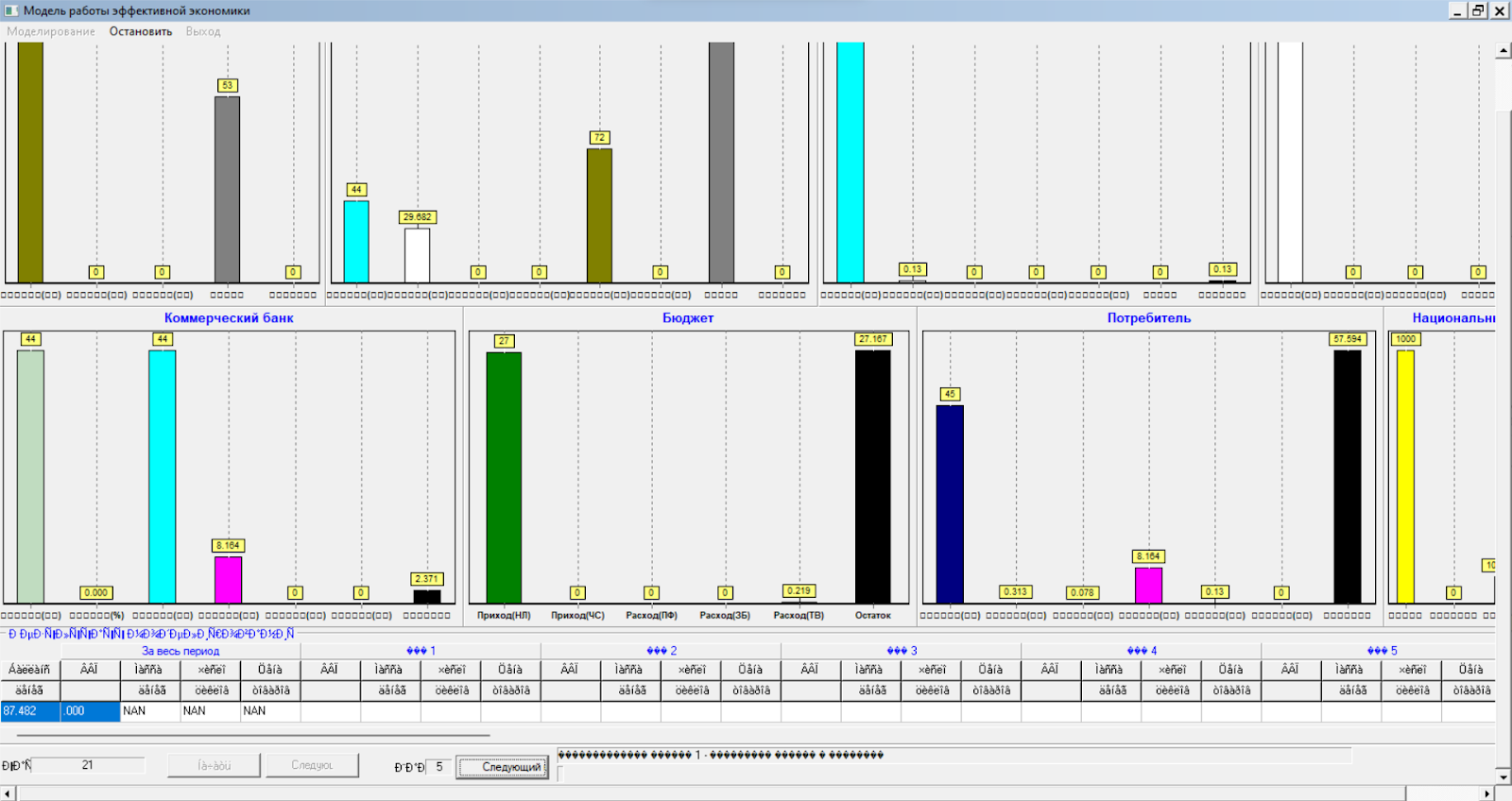
Головними перевагами цієї програми є те, що вона виконує свою головну функцію - виконує моделювання макроекономічної системи на базі ринкової моделі та на базі безкризової економічної моделі. Також вона має простий та зручний для користувача дизайн.

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

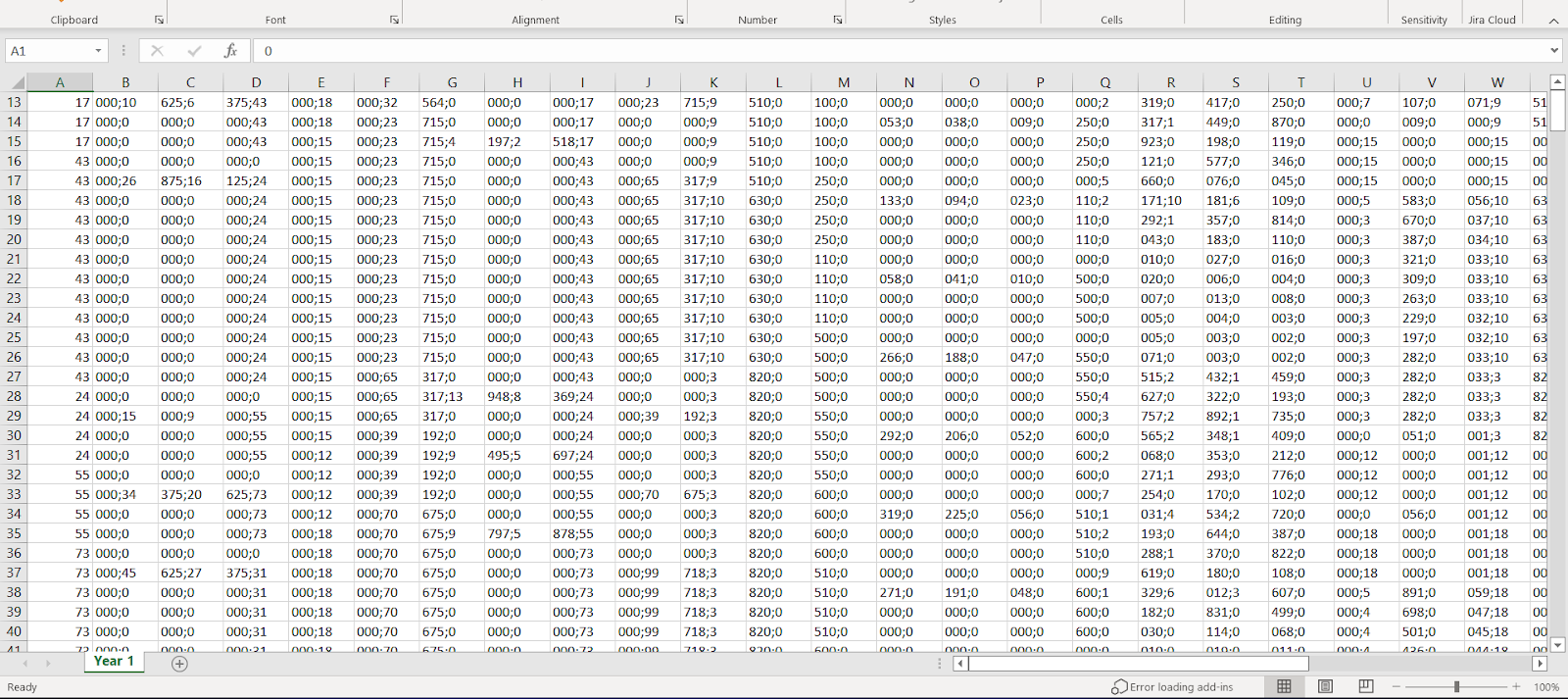
Приклади роботи програми:







Результатом моделювання є файл з розрахунками:



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

# 2 Зовнішнє проектування

## 2.1 Функціональне призначення

Програма моделює макроекономічну систему на основі базової ринкової моделі та на основі безкризової економічної моделі.

## 2.2 Експлуатаційне призначення

Дане програмне забезпечення допомагає розібратися як працює макроекономічна система на основі базової ринкової моделі та на основі безкризової моделі. Програма дає можливість порівнювати ці дві ринкові моделі та знаходити їх слабкі та сильні місця.

## 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).

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

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

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

## 2.4 Загальні вихідні дані

1. Задається відсоток грошей на сплату податку, %

2. Задається відсоток грошей, які спрямовуються на виплату заробітної плати, %

3. Задаються відсотки за кредитами та депозитами, %

4. Задається ставка Національного банку, %

5. Задається період повернення кредиту

6. Задається інтервал періоду моделювання на днях або вказується – 1 рік, 2 роки.

7. Задається період часу, через який виплачується

8. Зарплата та податок (або конкретна кількість днів, або можливий період). Може треба вказувати період, а якщо мова йде про конкретну кількість днів вказувати, наприклад, так 2-2 (два дні)

Після реалізації товару, наступного дня треба випадковим чином визначити кількість грошей, необхідні виготовлення наступного товару. Якщо після реалізації товару, грошей залишилося більше, то надлишок треба направити на виплату зарплати та податку. Якщо грошей недостатньо, треба необхідну суму дібрати рахунок кредиту.

## 2.5 Суб’єкти економічної діяльності

1. **Виробники сировини (ВС)**

Загальний лічильник фіксує 1-й день роботи.

Необхідно задати АСЕД, у яких ВС купує товари (в даному випадку 2, 4, 5) та яким продає (в даному випадку 2, 3, 5, 7) (табл. 1 та 2)

Визначається початкова кількість грошей в умовних одиницях (задається оператором). Це може бути або конкретне число, або вказано

можливий інтервал, з якого програма випадковим чином згідно

нормальному закону розподілу сама має обрати якесь число

(наприклад, 10-100). (мабуть другий варіант більш зручним). (наприклад, програма обрала 65) Це означає, що цю кількість грошей треба витратити на закупівлю необхідних товарів для випуску сировини.

Необхідно задати відсоток грошей, що спрямовуються на закупівлю товарів (наприклад, ВЗВ – 50, П-10, Т-40) (Задається Оператором).

Задається можливий період циклу технологічного процесу днями

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

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

Завдання вихідних даних по кожному АСЭД бачиться у вигляді з'являється форми або таблиці, куди вводяться потрібні числа.

1. **Виробники засобів виробництва (ВЗВ)**

Необхідно задати АСЕД у яких ВЗВ купує товари (в даному

випадку 1, 4, 5) та яким продає (в даному випадку 1, 3, 4, 5, 7) (табл. 1 та 2).

Необхідно задати відсоток грошей, що спрямовуються на закупівлю товарів (наприклад, ВС – 60, П-10, Т-30) (Задається Оператором).

Визначається початкова кількість грошей в умовних одиницях (задається оператором). Це може бути або конкретне число, або вказано

можливий інтервал, з якого програма випадковим чином згідно

нормальному закону розподілу сама має обрати якесь число (наприклад, 20-50). (мабуть другий варіант більш зручним). (наприклад,

програма вибрала 45) Це означає, що цю кількість грошей треба витратити на закупівлю необхідних товарів для випуску сировини

Задається можливий період циклу технологічного процесу днями

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

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

1. **Виробники споживчих товарів (ВСТ)**

Необхідно задати АСЕД у яких ВСТ купує товари (в даному

випадку 1, 2, 4, 5) і яким продає (у даному випадку 5) (табл. 1 та 2)

Необхідно задати відсоток грошей, що спрямовуються на закупівлю товарів (наприклад, ВС – 60, ВЗВ-20, П-5, Т-15) (Задається Оператором).

Визначається початкова кількість грошей в умовних одиницях (задається оператором). Це може бути або конкретне число, або вказано

можливий інтервал, з якого програма випадковим чином згідно

нормальному закону розподілу сама має обрати якесь число

(наприклад, 20-50). (мабуть другий варіант більш зручним). (наприклад,

програма обрала 30) Це означає, що цю кількість грошей треба витратити на закупівлю необхідних товарів для випуску сировини

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

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

1. **Послуги (П)**

Визначається початкова кількість грошей в умовних одиницях (задається оператором). Це може бути або конкретне число, або вказано

можливий інтервал, з якого програма випадковим чином згідно

нормальному закону розподілу сама має обрати якесь число

(наприклад, 20-50). (мабуть другий варіант більш зручним). (наприклад,

програма обрала 30) Це означає, що цю кількість грошей треба витратити на закупівлю необхідних товарів для випуску сировини

Необхідно поставити АСЕД у яких П купує товари (в даному випадку 2, 5) та яким надає послуги (в даному випадку 1, 2, 3, 5, 6, 7, 8) (табл. 1 та 2)

Задається процентне співвідношення купівлі товарів (ВЗВ та Т).

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

1. **Торгівля (Т)**

Визначається початкова кількість грошей в умовних одиницях (задається оператором). Це може бути або конкретне число, або вказано

можливий інтервал, з якого програма випадковим чином згідно

нормальному закону розподілу сама має обрати якесь число

(наприклад, 200-300). (мабуть другий варіант більш зручним). (наприклад, програма обрала 280). Це означає, що ця кількість грошей потрібна витратити на закупівлю необхідних товарів для подальшого продажу

Задається процентне співвідношення купівлі товарів (ВЗВ та Т).

1. **Покупці споживчих товарів (ПСТ)**

Задаються відсотки витрати грошей на «Послуги» та

«Торгівлю».

1. **Бюджет (Держава) (Б)**

Задаються відсотки витрати бюджетних грошей на

«Послуги», «Торгівлю» та оплату бюджетників та соціально захищених верств населення.

Задається ймовірність настання Надзвичайної

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

1. Комерційні банки (КБ)

Задається кількість грошей в умовних одиницях.

1. Нацбанк (НБ)

Задається загальна кількість грошей в умовних одиницях.

# 3 ВНУТРІШНЄ ПРОЕКТУВАННЯ

## 3.1 Об’єктно-орієнтоване програмування

Об'єктно-орієнтоване програмування або ООП − це методологія програмування, заснована на представленні програми у вигляді сукупності об'єктів кожен з яких являється реалізацією визначеного типу, що використовує механізм пересилки повідомлень і класи організовані в ієрархію наслідування.

Центральний елемент ООП − абстракція. В об'єктно-орієнтованому програмуванні абстракція − це додання об'єкту характеристик, які відрізняють його від всіх інших об'єктів, чітко визначаючи його концептуальні межі. Основна ідея полягає в тому, щоб відокремити спосіб використання складових об'єктів даних від деталей їх реалізації у вигляді більш простих об'єктів, подібно до того, як функціональна абстракція розділяє спосіб використання функції і деталей її реалізації в термінах більш примітивних функцій, таким чином, дані обробляються функцією високого рівня за допомогою виклику функцій низького рівня.

Такий підхід є основою об'єктно-орієнтованого програмування. Це дозволяє працювати з об'єктами, а не вдаючись в особливості їх реалізації. У кожному конкретному випадку застосовується той чи інший підхід: інкапсуляції, поліморфізм або успадкування. Наприклад, при необхідності звернутися до прихованих даними об'єкта, слід скористатися інкапсуляцією, створивши, так звану, функцію доступу або властивість.

Інкапсуляція − це властивість мови програмування, що дозволяє користувачеві не замислюватися про складність реалізації використовуваного програмного компонента, а взаємодіяти з ним за допомогою наданого інтерфейсу (публічних методів і членів), а також об'єднати і захистити життєво важливі для компонента дані. При цьому користувачеві надається тільки специфікація (інтерфейс) об'єкта.

Користувач може взаємодіяти з об'єктом тільки через цей інтерфейс. Користувач не може використовувати закриті дані і методи.

Інкапсуляція − це один з чотирьох найважливіших механізмів об'єктно-орієнтованого програмування (поряд з абстракцією, поліморфізмом і спадкуванням).

Приховування реалізації доцільно застосовувати в наступних випадках:

− гранична локалізація змін при необхідності таких змін;

− прогнозованість змін (які зміни в коді треба зробити для заданої зміни функціональності) і прогнозованість наслідків змін.

Спадкування − це один з чотирьох найважливіших механізмів об'єктно-орієнтованого програмування (поряд з інкапсуляцією, поліморфізмом і абстракцією), що дозволяє описати новий клас на основі вже існуючого (батьківського), при цьому властивості і функціональність батьківського класу запозичуються новим класом.

Іншими словами, клас-спадкоємець реалізує специфікацію вже існуючого класу. Це дозволяє звертатися к об'єктам класу-спадкоємця так само, як з об'єктами базового класу.

Клас, від якого відбулося спадкування, називається базовим або батьківським. Класи, які походять від базового, називаються нащадками, спадкоємцями або похідними класами.

Абстрактний клас − це клас, що містить хоча б один абстрактний метод, він описаний в програмі, має поля, методи і не може використовуватися для безпосереднього створення об'єкта. Тобто від абстрактного класу можна тільки наслідувати. Об'єкти створюються тільки на основі похідних класів, успадкованих від абстрактного.

При множинному спадкуванні у класі може бути більш одного предку. В цьому випадку клас успадковує методи всіх предків. Переваги такого підходу в більшій гнучкості.

Поліморфізм − це можливість об'єктів з однаковою специфікацією мати різну реалізацію.

Мова програмування підтримує поліморфізм, якщо класи з однаковою специфікацією можуть мати різну реалізацію − наприклад, реалізація класу може бути змінена в процесі успадкування.

Поліморфізм − це один з чотирьох найважливіших механізмів об'єктно-орієнтованого програмування (поряд з абстракцією, інкапсуляцією і спадкуванням).

Поліморфізм дозволяє писати більш абстрактні програми і підвищити коефіцієнт повторного використання коду. Загальні властивості об'єктів об'єднуються в систему, яку можуть називати по-різному − інтерфейс, клас. Спільність має зовнішнє і внутрішнє вираження:

− зовнішня спільність проявляється як однаковий набір методів з однаковими іменами та сигнатурами (ім'ям методів і типами аргументів і їх кількістю);

− внутрішня спільність − це однакова функціональність методів. Її можна описати інтуїтивно або висловити у вигляді строгих законів, правил, яким повинні підкорятися методи. Можливість приписувати різну функціональність одному методу (функції, операції) називається перевантаженням методу (перевантаженням функцій, перевантаженням операцій).

## 3.2 Опис відповідальності класів

Клас Program відповідає за запуск програми.

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

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

Клас Form3 відповідає за відображення інформації про саму програму.

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

Клас InputData відповідає за зберігання вхідних даних програми.

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

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

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

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

Клас Form7 відповідає за відображення інформації про вхідні дані виробників споживчих товарів програми для моделювання механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі.

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

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

## 3.3 Опис зв’язків класів

Описані вище класи мають наступні зв'язки.

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

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

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

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

## 3.4 Діаграма класів програми

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

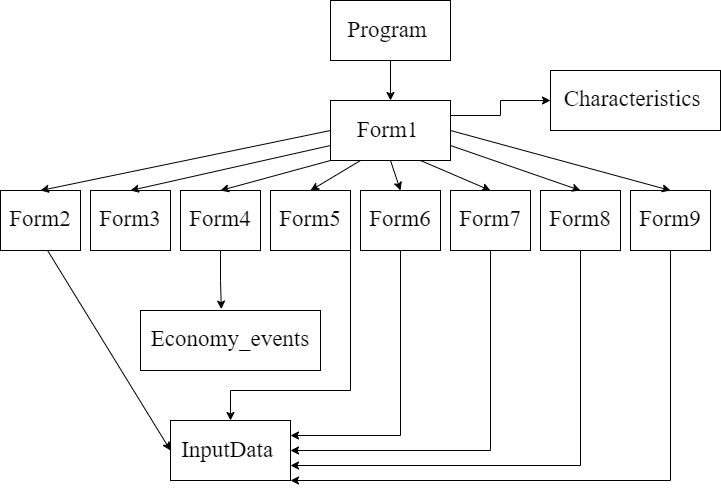


Рисунок 3.1 – Діаграма класів програми

## 3.5 Опис специфікацій класів

### 3.5.1 Опис специфікацій класу Program

Клас реалізовує наступні операції:

* запуск програми.

### 3.5.2 Опис специфікацій класу Form1

Клас реалізовує наступні операції:

* проведення розрахунків по моделюванню механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі та виведенні інформації на екран;
* моделюванню механізму функціонування макроекономічної системи держави на базі класичної ринкової моделі у часі;
* перехід на форму введення загальних вхідних даних;
* перехід на форму введення вхідних даних виробників сировини;
* перехід на форму введення вхідних даних виробників засобів виробництва;
* перехід на форму введення вхідних даних виробників споживчих товарів;
* перехід на форму введення вхідних даних торгівлі та послуг;
* перехід на форму введення інших вхідних даних;
* перехід на форму перегляду вікна подій;
* перехід на форму перегляду детальної інформації про програму.

### 3.5.3 Опис специфікацій класу Form2

Клас реалізовує наступні операції:

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

### 3.5.4 Опис специфікацій класу Form3

Клас реалізовує наступні операції:

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

### 3.5.5 Опис специфікацій класу Form4

Клас реалізовує наступні операції:

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

### 3.5.6 Опис специфікацій класу Form5

Клас реалізовує наступні операції:

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

### 3.5.7 Опис специфікацій класу Form6

Клас реалізовує наступні операції:

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

### 3.5.8 Опис специфікацій класу Form7

Клас реалізовує наступні операції:

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

### 3.5.9 Опис специфікацій класу Form8

Клас реалізовує наступні операції:

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

### 3.5.10 Опис специфікацій класу Form9

Клас реалізовує наступні операції:

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

### 3.5.11 Опис специфікацій класу InputData

Атрибути класу:

* змінна, що відповідає за швидкість покрокового моделювання;
* змінна, що відповідає за відсоток грошей на сплату податку;
* змінна, що відповідає за відсоток грошей на виплату заробітної плати;
* змінна, що відповідає за відсотки за кредитами та депозитами;
* змінна, що відповідає за ставку Національного банку;
* змінна, що відповідає за період часу повернення кредиту;
* змінна, що відповідає за інтервал періоду моделювання;
* змінна, що відповідає за первісну мінімальну кількість грошей виробників сировини;
* змінна, що відповідає за первісну максимальну кількість грошей виробників сировини;
* змінна, що відповідає за мінімальний період циклу технологічного процесу виробників сировини;
* змінна, що відповідає за максимальний період циклу технологічного процесу виробників сировини;
* змінна, що відповідає за мінімальний відсоток торгової націнки виробників сировини;
* змінна, що відповідає за максимальний відсоток торгової націнки виробників сировини;
* змінна, що відповідає за первісну мінімальну кількість грошей виробників засобів виробництва;
* змінна, що відповідає за первісну максимальну кількість грошей виробників засобів виробництва;
* змінна, що відповідає за мінімальний період циклу технологічного процесу виробників засобів виробництва;
* змінна, що відповідає за максимальний період циклу технологічного процесу виробників засобів виробництва;
* змінна, що відповідає за мінімальний відсоток торгової націнки виробників засобів виробництва;
* змінна, що відповідає за максимальний відсоток торгової націнки виробників засобів виробництва;
* змінна, що відповідає за первісну мінімальну кількість грошей виробників споживчих товарів;
* змінна, що відповідає за первісна максимальну кількість грошей виробників споживчих товарів;
* змінна, що відповідає за мінімальний період циклу технологічного процесу виробників споживчих товарів;
* змінна, що відповідає за максимальний період циклу технологічного процесу виробників споживчих товарів;
* змінна, що відповідає за мінімальний відсоток торгової націнки виробників споживчих товарів;
* змінна, що відповідає за максимальний відсоток торгової націнки виробників споживчих товарів;
* змінна, що відповідає за первісну мінімальну кількість грошей торгівлі та послуг;
* змінна, що відповідає за первісну максимальну кількість грошей торгівлі та послуг;
* змінна, що відповідає за відсоткове співвідношення купівлі товарів торгівлі та послуг;
* змінна, що відповідає за відсотки витрати грошей на «Послуги та торгівлю» ;
* змінна, що відповідає за відсотки витрачання бюджетних грошей на «Послуги та торгівлю» ;
* змінна, що відповідає за ймовірність настання Надзвичайної Ситуації;
* змінна, що відповідає за кількість грошей комерційного банку;
* змінна, що відповідає за загальну кількість грошей у Нацбанку.

### 3.5.12 Опис специфікацій класу Сharacteristics

Атрибути класу:

* змінна, що відповідає за прихід виробників сировини;
* змінна, що відповідає за витрати виробників сировини;
* змінна, що відповідає за сировина виробників сировини;
* змінна, що відповідає за остача виробників сировини;
* змінна, що відповідає за прихід виробників засобів виробництва;
* змінна, що відповідає за витрати виробників засобів виробництва;
* змінна, що відповідає за сировина виробників засобів виробництва;
* змінна, що відповідає за остача виробників засобів виробництва;
* змінна, що відповідає за прихід виробників споживчих товарів;
* змінна, що відповідає за витрати виробників споживчих товарів;
* змінна, що відповідає за сировина виробників споживчих товарів;
* змінна, що відповідає за остача виробників споживчих товарів;
* змінна, що відповідає за прихід торгівлі та послуг;
* змінна, що відповідає за витрати торгівлі та послуг;
* змінна, що відповідає за сировина торгівлі та послуг;
* змінна, що відповідає за остача торгівлі та послуг;
* змінна, що відповідає за прихід бюджету;
* змінна, що відповідає за витрати бюджету;
* змінна, що відповідає за остача бюджету;
* змінна, що відповідає за прихід комерційних банків;
* змінна, що відповідає за витрати комерційних банків;
* змінна, що відповідає за остача комерційних банків;
* змінна, що відповідає за прихід покупців споживчих товарів;
* змінна, що відповідає за витрати покупців споживчих товарів;
* змінна, що відповідає за остача покупців споживчих товарів;
* змінна, що відповідає за запас Нацбанку;
* змінна, що відповідає за витрати Нацбанку;
* змінна, що відповідає за емісія Нацбанку;
* змінна, що відповідає за остача Нацбанку.

### 3.5.13 Опис специфікацій класу Economy\_events

Атрибути класу:

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

## 3.6 Опис інтерфейсної частини класів

потом…

## 3.7 Проектування інтерфейсу користувача

Інтерфейс − це сукупність інформаційної моделі предметної областей, засобів і способів взаємодії з інформаційною моделлю, а також компонентів які забезпечує формування інформаційної моделі в процесі роботи програмних систем.

Основою принципів проектування інтерфейсів користувача є людські можливості. Принципи проектування:

* природність (інтуїтивність). Користувача не слід змушувати істотно змінювати звичні способи вирішення завдань, коли робота з системою не викликає у користувача необхідного пошуку елементів інтерфейсу для поставленого завдання. Доцільно зберегти позначення та термінологію даної предметної області;
* узгодженість (несуперечність). Забезпечення наступності знань і навичок, коли в процесі роботи з системою користувачем використовувалися деякі прийоми роботи з деякою частиною системи, то в іншій частині прийоми повинні бути ідентичні. Погоджений інтерфейс є впізнаваним і передбачуваним, а також відповідним нормам;
* ненадмірність. Користувач повинен вводити тільки мінімальну інформацію для роботи або управління системою. Слід позбутися повторного вводу інформації;
* дружність (безпосередній доступ до системи допомоги). Коли в процесі роботи система забезпечує користувача повідомленнями про помилки та інформацією що система робить. Повідомлення про помилку має бути корисним і зрозумілим користувачеві. Ефективний інтерфейс повинен запобігати ситуації, коли користувач може помилитися, і запобігати наслідок можливих помилок;
* гнучкість (адаптивність). Коли інтерфейс системи може обслуговувати користувачів з різним рівнем підготовки. Передбачається можливість зміни структури діалогу або вхідних даних. Для недосвідчених користувачів інтерфейс може бути організований, як ієрархічна структура меню, а для досвідчених, як комбінації клавіш;
* простота інтерфейсу. Забезпечення легкості в його вивченні і використанні, крім того він повинен надавати доступ до всього переліку можливостей, передбачених додатком (що суперечить простоті);
* принцип зворотного зв'язку. Програма повинна мати відгук на будь-яку дію користувача.

До візуальних атрибутів відображення інформації відносяться:

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

Колір є одним з найважливіших візуальних атрибутів інтерфейсу. Оскільки колір має здатність залучати погляд, його слід використовувати для ідентифікації тих елементів інтерфейсу, на які користувачеві слід звернути увагу. Колір також має асоціативний аспект, у багатьох випадках об'єкти одного кольору сприймаються людьми як взаємопов'язані. Кольорова гамма може також надавати певний емоційний або психологічний вплив. Недарма, наприклад, поділяють кольори на «холодні» і «теплі».

У роботах, присвячених дослідженню психологічного впливу різних кольорів, наводяться такі дані:

* блакитний (заспокоює);
* червоний (хвилює і стомлює);
* зелений (налаштовує на добродушний і безініціативний лад);
* жовтий (веселий, оптимістичний, викликає легковажний настрій);
* помаранчевий (розкріпачує фантазію);
* фіолетовий (згубний для очей, колір заздрості, тривоги, незадоволеності);
* коричневий (пригнічує розумову активність);
* чорний (похмурий, сприяє виникненню головного болю, але знижує число помилок).

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

## 3.8 Аналіз проекту

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

## 3.9 Вибір мови програмування

???