

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

НКЦ заочної освіти

Кафедра інформаційних технологій

**Бакалаврська кваліфікаційна робота**

на тему: Розробка програмного засобу автоматизованого планування багатоканальних етапів виробничого процесу

Виконала студентка 3 (інт.) курсу  
групи КН-3 (інт.)

Напрямок підготовки 6.050101

комп'ютерні науки

Новоженіна Ірина Анатоліївна

Керівник к. т. н., доцент

Великодний Станіслав Сергійович

Консультант \_\_\_\_\_

\_\_\_\_\_

Рецензент к. ф.-м. н., доцент

Буяджи Василь Володимирович

## ЗМІСТ

Перелік умовних позначок .....	6
Вступ.....	7
1 Аналітична частина.....	9
1.1 Аналіз предметної галузі роботи.....	9
1.2 Огляд найближчих аналогів та засобів їх представлення.....	12
1.2.1 Граф, як математичне поняття.....	12
1.2.2 Мови опису та програми побудови графів.....	13
1.2.3 Методологія PERT .....	14
2 Проектна частина .....	17
2.1 Проектування діаграми варіантів використання .....	17
2.2 Проектування діаграми послідовності.....	22
2.3 Проектування діаграми класів .....	28
2.4 Проектування діаграми компонентів .....	35
2.5 Висновки за розділом .....	38
3 Програмно-методичний комплекс проектування мережевих графіків .....	39
3.1 Призначення програмного засобу .....	39
3.2 Основні поняття та позначення, що прийняті у ПЗ.....	39
3.3 Порядок роботи з ПЗ.....	41
3.3.1 Створення нової задачі .....	41
3.3.2 Зміна параметрів задачі .....	43
3.3.3 Створення зв'язків між задачами .....	43
3.3.4 Робота з подіями.....	44
3.3.5 Генерування таблиць .....	47
3.4 Додаткові можливості ПЗ.....	49
3.4.1 Перегляд, друк та збереження проекту.....	49
3.4.2 Переміщення та зміни розміру структур.....	51
3.5 Додаткові файли .....	53
3.6 Висновки до розділу .....	55

Висновки .....	56
Перелік джерел посилання .....	58

## ПЕРЕЛІК УМОВНИХ ПОЗНАК

ВВ – варіант використання.

ВМС – військово-морські сили.

ДВВ – діаграма варіантів використання.

ДК – діаграми класів.

ДКМ – діаграма компонентів.

ДП – діаграма послідовності.

ЕОМ – електронно-обчислювальна машина.

Код – цифровий код задачі.

МГ – мережевий графік.

ПЗ – програмний засіб.

ПМК – програмно-методичний комплекс.

США – Сполучені Штати Америки.

$T_{п(i)}$  – пізній строк настання події.

$T_{пн}$  – пізній строк початку задачі.

$T_{по}$  – пізній строк закінчення задачі.

$T_{р(i)}$  – ранній строк настання події.

$T_{рн}$  – ранній строк початку задачі.

$T_{ро}$  – ранній строк закінчення задачі.

ERP – enterprise resources planning.

H – history.

LL – life line.

$L_m$  – критичний шлях.

PERT – program (project) evaluation and review technique.

R (*i*) – резерв часу настання події.

$R_{п}$  – повний резерв часу.

$R_c$  – вільний резерв часу.

t – тривалість задачі.

UML – unified modeling language.

## ВСТУП

Процес створення проекту, прототипу, прообразу майбутнього об'єкта, стану та способів його виготовлення називається проектуванням. У техніці під проектуванням розуміють розробку проектної, конструкторської та іншої технічної документації, призначеної для забезпечення створення нових видів та зразків.

У проектуванні застосовують системний підхід, який полягає у встановленні структури системи, типу зв'язків, визначені атрибутів, аналізуванні впливів зовнішнього середовища. В процесі проектування виконуються технічні та економічні розрахунки, схеми, графіки, пояснювальні записки, кошториси, калькуляції та описи.

Однією з головних складових проектування – є планування, що визначається як заздалегідь намічений порядок дій або оптимальний розподіл ресурсів, необхідних для досягнення поставленої мети.

Мережеве планування – це одна з форм графічного відображення змісту робіт і тривалості виконання стратегічних планів і довгострокових комплексів проектних, планових, організаційних та інших видів діяльності підприємства.

Поряд з лінійними графіками та табличними розрахунками мережеві методи планування знаходять широке застосування при розробці перспективних планів та моделей створення складних виробничих систем та інших об'єктів довгострокового використання. Мережеві плани робіт підприємств по створенню нової конкурентоздатної продукції містять не тільки загальну тривалість всього комплексу проектно-виробничої та фінансово-економічної діяльності, але й тривалість та послідовність здійснення окремих процесів чи етапів, а також потребу необхідних економічних ресурсів.

Вперше плани-графіки виконання виробничих процесів були розроблені в 1958 році консалтинговою фірмою «Буз, Ален і Гамільтон» спільно з корпорацією «Локхед» на замовлення підрозділу спеціальних проектів військово-

во-морських сил (ВМС) Сполучених Штатів Америки (США) у складі Міністерства Оборони США для проекту створення ракетної системи «Поларіс» (Polaris), що був відповіддю на кризу, що наступила після запуску Радянським Союзом першого космічного супутника.

Загальний зміст план-графіків такий: на лінійних або стрічкових графіках по горизонтальній осі в обраному масштабі часу відкладається тривалість робіт за всіма стадіями та етапами виробництва, зміст циклів робіт зображується за вертикальною віссю з необхідним ступенем їх розчленування на окремі частини або елементи.

На вітчизняних підприємствах, циклові або лінійні графіки, звичайно, застосовуються у процесі короткострокового чи оперативного планування виробничої діяльності. Основним недоліком таких планів-графіків – є відсутність можливості тісної взаємозв'язки окремих робіт в єдину виробничу систему або загальний процес досягнення запланованих кінцевих цілей підприємства (фірми). Мережеві графіки служать не тільки для планування різноманітних довгострокових робіт, але і для їх координації між керівниками та виконавцями проектів, також мережеві графіки необхідні для визначення необхідних виробничих ресурсів та їх раціонального використання.

Автоматизовані системи планування ресурсів виробництва (ERP), зазвичай включають комп'ютерні програми, які у тій, чи іншій мірі автоматизують деякі етапи складання та коригування мережевих графіків, проте подібні програми мають досить високу ліцензійну ціну та невід'ємні для вітчизняного виробника.

Таким чином, метою дипломної роботи буде створення програмного засобу (ПЗ) автоматизованого проектування мережевих графіків організації виробництва.

Об'єктом роботи стане мережевий графік виробничого процесу.

Предметом роботи – ПЗ засіб побудови мережевої динамічної моделі.

Пояснювальна записка до дипломної роботи складається з 60 стор. тексту, містить: 25 рисунків, 2 таблиці, перелік з 21 джерела посилання.

## 1 АНАЛІТИЧНА ЧАСТИНА

### 1.1 Аналіз предметної галузі роботи

Мережевий графік – це динамічна модель виробничого процесу, яка відображає технологічну залежність і послідовність виконання комплексу робіт, що погоджує їх звершення у часі з урахуванням витрат ресурсів і вартості робіт з виділенням при цьому вузьких (критичних) місць [1] <sup>1)</sup>.

Основні елементи мережевого графіка – робота та подія. Робота відображає трудовий процес, в якому беруть участь люди, машини, механізми, матеріальні ресурси (проекування споруди, поставки обладнання, кладка стін, рішення задач на електронно-обчислювальній машині (ЕОМ) та ін.) або процес очікування (твердіння бетону, сушка штукатурки тощо).

Кожна робота мережевого графіка має конкретний зміст. Робота як трудовий процес вимагає витрат часу і ресурсів, проте очікування – тільки часу. Для правильного і наочного відображення порядку передування робіт при побудові мережевого графіка, використовують зображувані штриховими лініями додаткові дуги, що звуться фіктивними роботами або зв'язками [2] <sup>2)</sup>. Вони не вимагають ні часу, ні ресурсів, а лише вказують, що початок однієї роботи залежить від закінчення іншої.

Подія виражає факт закінчення однієї або декількох робіт, що безпосередньо передують (входять в подію), необхідних для початку наступних робіт (що виходять з події). Подія, що стоїть на початку роботи, називається початковою, а в кінці –кінцевою. Початкова подія мережевого графіка називається вихідною, а кінцева – завершальною. Подія, яка не є ні вихідною, ні завершальною, називається проміжною. У вихідну подію мережевого графіка не входить, а з завершальної – не виходить жодної роботи. На відміну від робіт, події відбуваються миттєво без споживання ресурсів [3] <sup>3)</sup>.

<sup>1)</sup> [1] Мережеве планування. URL: <http://uk.wikipedia.org/wiki/> (дата звернення: 23.03.2019).

<sup>2)</sup> [2] Березина Л. Ю. Графы и их применение : Пособие для учителей. Москва: Высшая школа, 1989. 326 с.

<sup>3)</sup> [3] Михеева В. С. Математические методы в экономической географии. Ч. 2. Приложение теории графов : Курс лекций. Москва : Математика, 1983. 316 с.

На мережевих графіках попередні та наступні роботи мають свої позначення. Будь-яка послідовність робіт у мережевому графіку, при якому кінцева подія кожної роботи співпадає з початковою подією наступної, називається шляхом. Тривалість шляху визначається сумою тривалостей складових його робіт [4] <sup>1)</sup>. Шлях найбільшої довжини між вихідними і завершальними подіями називається критичним ( $L_m$ ). Якщо критичний час не відповідає заданому або нормативному, скорочення термінів виробничого процесу необхідно починати зі скорочення тривалості критичних робіт [5] <sup>2)</sup>.

Мережевий графік заснований на використанні математичної моделі – графа. Графами (застарілі синоніми: мережа, лабіринт, карта та ін.) математики називають "безліч вершин і набір упорядкованих або неупорядкованих пар вершин" [6] <sup>3)</sup>. Говорячи більш звичною для проектувальника (але менш точною) мовою, граф – це набір кіл (прямокутників, трикутників та ін.), що з'єднані спрямованими або ненаправленими відрізками. В цьому випадку самі кола (або інші використовувані фігури) за термінологією теорії графів будуть називатися "вершинами", а з'єднуючі їх ненаправлення відрізки – "ребрами", спрямовані (стрілки) – "дугами". Якщо всі відрізки є спрямованими – граф називається орієнтованим, якщо ненаправленими – неорієнтованим [7] <sup>4)</sup>.

Будь-яка послідовність робіт у мережевому графіку, в якому кінцева подія кожної роботи цієї послідовності, збігається з початковою подією наступної за нею роботою, називається шляхом.

Основні поняття, що використовуються у мережевому плануванні зведено до табл. 1.

---

<sup>1)</sup> [4] Голиков А. П., Трофимов А. М., Черванёв И. Г. Математические методы в географии. Харьков : Наука, 1986. 208 с.

<sup>2)</sup> [5] Глазжан І. М., Новиков В. Г. Основи мережевого планування та управління. Харків : Вид-во ХГУ, 1996. 198 с.

<sup>3)</sup> [6] Оре О. Теория графов. Москва : Наука, 1968. – 336 с.

<sup>4)</sup> [7] Уилсон Р. Введение в теорию графов : пер. с англ. Москва : Мир, 1977. 208 с.



Таблиця 1 – Поняття та визначення мережевого планування

Поняття	Визначення
Види робіт:	
Дійсна робота	Робота, що вимагає витрат праці, матеріальних ресурсів та часу (наприклад: підготовка траси змагань)
Очікування	Робота, що не вимагає витрат праці і матеріальних ресурсів, але займає деякий час
Фіктивна робота	Зв'язок між двома або більше подіями, який не вимагає витрат праці, матеріальних ресурсів і часу, але вказує, що можливість початку однієї операції безпосередньо залежить від виконання іншої. Тривалість такої роботи: 0
Види подій:	
Початкова подія	Початок виконання комплексу робіт
Завершальна подія	Кінцева подія, що означає досягнення кінцевої мети комплексу робіт
Проміжна подія	Результат однієї або декількох робіт, що представляють можливість почати одну або декілька безпосередньо наступних робіт. Тривалість проміжного події в часі завжди: 0
Види шляхів:	
Повний шлях	Початок якого збігається з вихідною подією мережі, а кінець – із завершальною
Шлях, що передую події	Шлях від вихідної події мережі до даної події
Шлях, наступний за подією	Шлях, що з'єднує подію із завершальною подією
Шлях між подіями $i$ та $j$	Шлях, що з'єднує будь-які дві події $i$ та $j$ , з яких жодна – не є вихідною або завершальною подією мережевого графіка
Критичний шлях	Шлях, який має найбільшу тривалість від вихідної події до завершальної

## 1.2 Огляд найближчих аналогів та засобів їх представлення

### 1.2.1 Граф, як математичне поняття

Мережевий графік – граф, вершини якого відображають стани деякого об'єкта (наприклад, будівництва), а дуги – роботи, що ведуться на цьому об'єкті. Кожній дузі зіставляється час, за який здійснюється робота і / або число робітників, які здійснюють роботу. Часто мережевий графік будується так, що розташування вершин по горизонталі відповідає часу досягнення стану, відповідного заданої вершини (популярна складова методології PERT).

Граф – це сукупність об'єктів із зв'язками між ними [8] <sup>1)</sup>. Об'єкти розглядаються як вершини, або вузли графу, а зв'язки – як дуги, або ребра (рис. 1).

Для різних областей використання види графів можуть відрізнятися орієнтовністю, обмеженнями на кількість зв'язків і додатковими даними про вершини або ребра. Велика кількість структур, які мають практичну цінність в математиці та інформатиці, можуть бути представлені графами [9] <sup>2)</sup>. Наприклад, будову Вікіпедії можна змоделювати за допомогою орієнтованого графу, в якому вершини – це статті, а дуги (орієнтовані ребра) – посилання на інші статті.

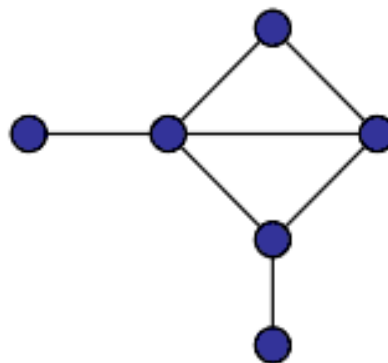


Рисунок 1– Приклад графа із 6-ма вершинами та 7-ма ребрами

<sup>1)</sup> [8] Bondy J.A. Graph Theory with Applications. North-Holland : Elsevier, 1986. 264 p.

<sup>2)</sup> [9] Емеличев В. А., Мельников О. И., Сарванов В. И. Лекции по теории графов. Изд. 2, испр. Москва: Наука, 2009. 392 с.

Графи за їх типом й характеристиками за ребрами та їх кратністю наведені у табл. 2.

Таблиця 1.2 – Типи графів

Тип графу	Ребра	Кратні ребра
Простий граф	Неорієнтовані	Ні
Мультиграф	Неорієнтовані	Так
Орієнтований граф	Орієнтовані	Ні
Орієнтований мультиграф	Орієнтовані	Так

### 1.2.2 Мови опису та програми побудови графів

Для опису графів в цілях, придатних для машинної обробки і, одночасно, зручному для людського сприйняття – використовується кілька стандартизованих мов, серед яких: DOT (мова), GraphML, Trivial Graph Format, GML, GXL, XGMML, DGML.

Відзначимо спеціалізовані програми для побудови графів [10]<sup>1)</sup>. До найбільш вдалих відносяться комерційні ILOG, GoView, Lassalle AddFlow, LEDA. З безкоштовних можна відзначити Boost Graph Library. Для візуалізації графів можна використовувати Graphviz (на думку експертів, вона добре працює для орграфов) або LION Graph Visualizer [11]<sup>2)</sup>. Особливо відзначимо програму Графоаналізатор – це російськомовна програма, з вельми простим для користувача інтерфейсом [12]<sup>3)</sup>.

<sup>1)</sup> [10] Коммерческие специализированные программы для построения графов. URL: <http://www.boost.org/> (дата звернення: 02.04.2019).

<sup>2)</sup> [11] LION Graph Visualizer. URL: <http://lion.disi.unitn.it/intelligent-optimization/visualizer.html> (дата звернення: 03.04.2019).

<sup>3)</sup> [12] Графоанализатор – среда визуализации графов. URL: <http://grafoanalizator.unick-soft.ru/> (дата звернення: 03.04.2019).

### 1.2.3 Методологія PERT

PERT – Program (Project) Evaluation and Review Technique (англ.) – техніка оцінки та аналізу програм (проектів), яка використовується при управлінні проектами [1]<sup>1)</sup>. PERT – це спосіб аналізу завдань, необхідних для виконання проекту, особливо, аналізу часу, який потрібен для виконання кожної окремої задачі, а також визначення мінімального необхідного часу для виконання всього проекту.

PERT був розроблений головним чином для спрощення планування на папері та створення графіків великих і складних проектів. PERT призначений для масштабних, унікальних, складних, нерутинних проектів. Метод передбачав наявність невизначеності, даючи можливість розробити робочий графік проекту без точного знання деталей і необхідного часу для всіх його складових.

Найпопулярнішою частиною PERT – є метод критичного шляху, що спирається на побудову мережевого графіку (мережевої діаграми PERT).

Метод критичного шляху – ефективний інструмент планування розкладу та управління термінами проекту. В основі методу лежить визначення найбільш тривалої послідовності завдань від початку проекту до його закінчення з урахуванням їх взаємозв'язку. Завдання, що лежать на критичному шляху (критичні завдання) мають нульовий резерв часу виконання та у разі зміни їх тривалості змінюються терміни всього проекту.

У зв'язку з цим при виконанні проекту критичні завдання вимагають більш ретельного контролю, зокрема, своєчасного виявлення проблем та ризиків, що впливають на терміни їх виконання і, отже, на строки виконання проекту в цілому. У процесі виконання проекту критичний шлях проекту може змінюватися тому, що при зміні тривалості задач деякі з них можуть опинитися на критичному шляху.

---

<sup>1)</sup> [1] Мережеве планування. URL: <http://uk.wikipedia.org/wiki/> (дата звернення: 04.04.2019).

Найвідоміша частина PERT – це діаграми взаємозв'язків робіт і подій (рис. 2). PERT пропонує використовувати діаграми-графи з роботами на вузлах, з роботами на стрілках (мережеві графіки), а також діаграми Ганта.

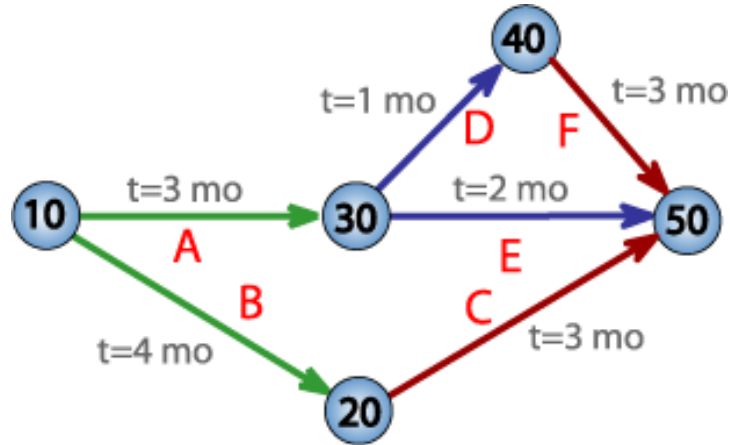


Рисунок 2 – Приклад мережевої діаграми PERT для проекту тривалістю у сім місяців із п'ятьма проміжними точками (від 10 до 50) і шістьма діяльностями (від А до F)

Діаграма PERT з роботами на стрілках являє собою множину точок-вершин (події) разом з їх з'єднуючими орієнтованими дугами (роботи). Будь якій дузі, що розглядається в якості якоїсь роботи з числа потрібних для здійснення проекту, приписуються визначені кількісні характеристики – це обсяги виділених на дану роботу ресурсів і, відповідно, її очікувана тривалість (довжина дуги). Будь яка вершина інтерпретується як подія завершення робіт, представлених дугами, які входять в неї, і одночасно початку робіт, що відображаються дугами, що виходять звідти.

Таким чином відображається той факт, що ні до однієї з робіт не можна приступити перш, ніж будуть виконані всі роботи, що передують їй згідно з технологією реалізації проекту. Початок цього процесу – вершина без вхідних, а закінчення – вершина без вихідних дуг. Решта вершин повинні мати і ті, й інші дуги.

Послідовність дуг, в якій кінець кожної попередньої збігається з початком наступної, трактується як шлях від відправної вершини до завершальної, а сума довжин таких дуг – як його тривалість.

Зазвичай, початок і кінець реалізації проекту – пов'язані безліччю шляхів, довжини яких розрізняються. Найбільша – визначає тривалість усього проекту, мінімально можливу при зафіксованих характеристиках дуг графа. Відповідний шлях – критичний, тобто саме від тривалості складових його робіт залежить загальна тривалість проекту, хоча при зміні тривалості будь-яких робіт проекту критичним може стати й інший шлях.

## 2 ПРОЕКТНА ЧАСТИНА

У проектній частині дипломної роботи розглядаються проекти рішення, що запропоновані розробником. Зміст проектної частини визначається, по-перше, специфікою теми дипломної роботи, по-друге, особливостями конкретних технічних пропозицій до роботи.

У поданій дипломній роботі проектується архітектура (проектний «каркас») програмного засобу, що розроблюється, у вигляді декількох діаграм різної природи, виконаних із дотриманням нотації UML [13] <sup>1)</sup>.

### 2.1 Проектування діаграми варіантів використання

Діаграми варіантів використання (ДВВ) використовуються для надання аналітику детальної уяви про галузь застосування програмного засобу (ПЗ), що розроблено [14] <sup>2)</sup>. З ДВВ стає зрозуміло для чого призначений ПЗ, які підсистеми та модулі він має, якими зв'язками поєднані елементи та сутності у ПЗ.

Центральним елементом ДВВ – є мережевий графік (МГ), який зображено у вигляді актору зі стереотипом «business actor» (рис. 3). Від актору «МГ» відходять різноманітні зв'язки, більшість з яких асоціації зі стереотипом «uses» (використання), проте також присутні залежності.

Розглянемо, спочатку, сутності, об'єднанні з МГ першим типом зв'язків. Таких на ДВВ міститься 9 – це:

- кооперація використання «Біологія» з асоціаціями «extended» (розширення) на: «Ланцюги харчування», «Екосистеми» та «Генетичні послідовності»;

---

<sup>1)</sup> [13] Ларман К. Применение UML 2.0 и шаблонов проектирования. 3-е изд. Москва : Вильямс, 2006. 736 с.

<sup>2)</sup> [14] Шмуллер Дж. Освой самостоятельно UML 2.0 за 24 часа. Практическое руководство. Москва : Вильямс, 2005. 416 с.

- сукупність кооперацій використання «Археологія» та «Геологія», об'єднаних межею «Порядок вивчення пластів»;
- кооперація використання «Генеалогічні дерева»;
- кооперація «Хімічні речовини» з асоціаціями «extended» (розширення) на: «Послідовність з'єднання» та «Будова молекул»;
- кооперація використання «Соціальні мережі» з асоціаціями «extended» (розширення) на: «Соціальні групи» та «Зв'язки»;
- кооперація використання «Карта шляхів» з асоціаціями «extended» (розширення) на: «Залізничні» та «Автомобільні»;
- кооперація використання «Файлова система комп'ютера» з асоціаціями «extended» (розширення) на: «Операційну систему» та «Порядок виконання файлів»;
- кооперація використання «Турнірні таблиці»;
- метаклас «Технологічні процеси», у структурі межі «Виробничий процес».

З цього переліку видно, що МГ використовуються у всіх галузях застосування, пов'язаних із наведеним переліком. Далі розглянемо кожен з перелічених сутностей детальніше із розкриттям сутностей другого рівня деталізації (все стосовно до рис. 3).

МГ використовуються у біології та екології (кооперація використання «Біологія») досить давно; прикладами використання можуть бути (асоціація «extended»): ланцюги харчування, екосистеми, генетичні послідовності (варіанти використання (ВВ) зі стереотипом «business use case»), генетичні карти, таксономічна ієрархія живих організмів тощо.

МГ знаходять своє використання у археології та геології (сукупність кооперацій використання «Археологія» та «Геологія») для аналізу та вивчення стратиграфії (на ДВВ – межа «Порядок вивчення пластів»).

МГ застосовуються у подіях, які виконуються протягом років – прикладом може стати послідовність складання генеалогічних дерев (кооперація використання «Генеалогічні дерева»).



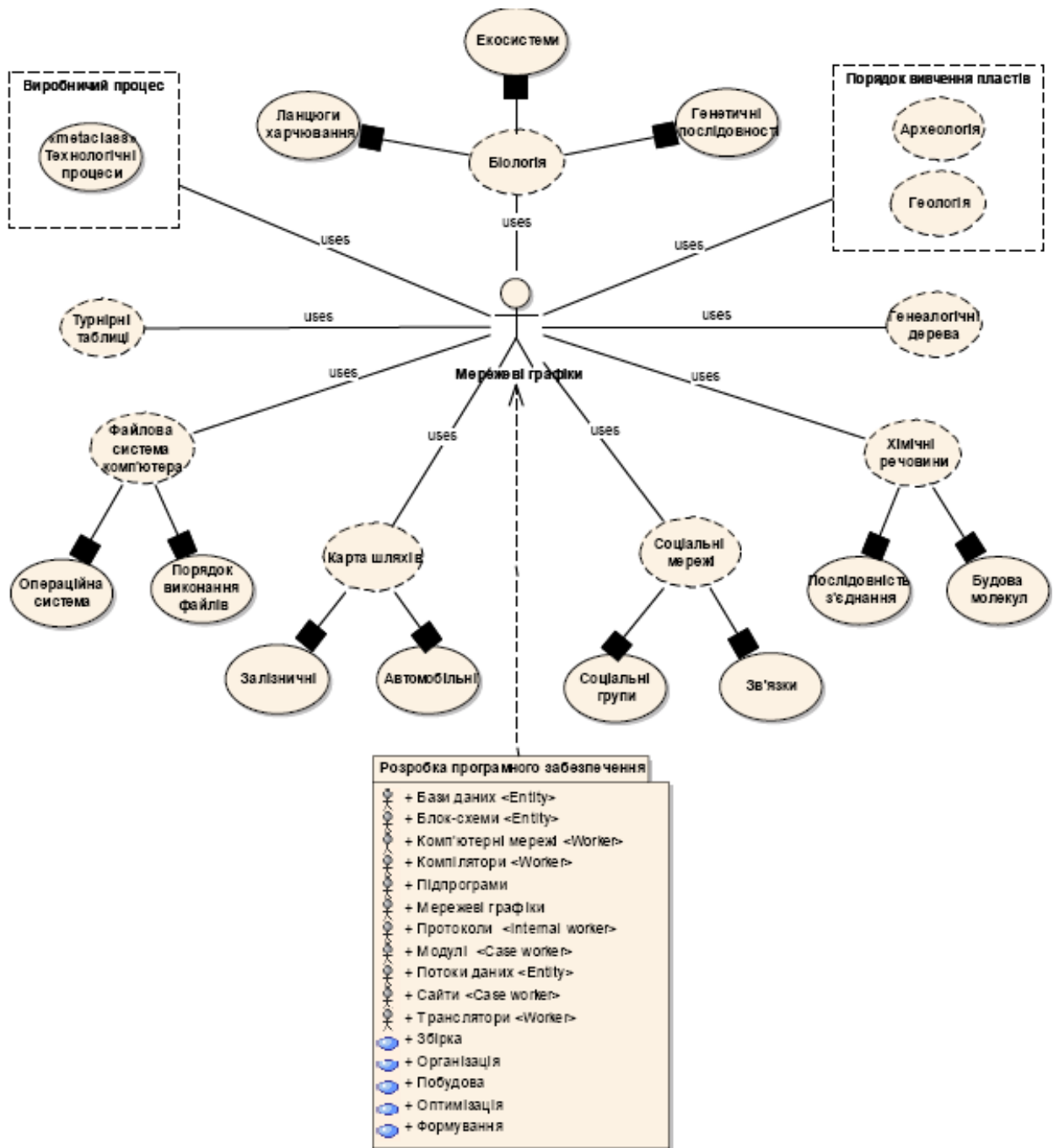


Рисунок 3 – ДВВ ПЗ проектування МГ

МГ знаходять своє застосування у моделюванні хімічних речовин (кооперація використання «Хімічні речовини»), за допомогою яких вивчаються (асоціації «extended») їх послідовності з'єднання та будова молекул (ВВ зі стереотипами «business use case»).

МГ використовуються у побудові соціальних мереж (кооперація використання «Соціальні мережі»), де кожне перебування людини чи соціальної

групи (ВВ зі стереотипом «business use case») – є вершиною, а зв'язки – ребрами (асоціації «extended»).

Карта проходження автомобільних чи будь-яких інших шляхів (кооперація використання «Карта шляхів») також є МГ, причому кожна дорога (залізнична чи автомобільна – ВВ зі стереотипами «business use case») може мати певне значення «ваги» (асоціації «extended»), наприклад: щільність транспортного потоку.

Файлова система комп'ютера (кооперація використання «Файлова система комп'ютера») може бути зображена у вигляді МГ, що показує порядок виконання файлів та тек у багатьох операційних системах (ВВ зі стереотипом «business use case»); такий МГ має вигляд дерева з асоціаціями «extended».

Турнірні таблиці спортивних чемпіонатів (кооперація використання «Турнірні таблиці») також можуть бути зображені у вигляді МГ.

Будь-який виробничий процес також може бути зображений за допомогою МГ (див. рис. 3, де усе існуюче розмаїття технологічних процесів зображено як метаклас («metaclass») «Технологічні процеси», у пакетній структурі-межі «Виробничий процес».

Перейдемо до розгляду сутності, що об'єднана з МГ типом зв'язку залежності «dependency». На ДВВ це пакет «Розробка програмного забезпечення», виконання якого дійсно базується (залежить) від МГ. До складу пакету входять багато сутностей, об'єднаних зв'язками, сукупність яких являє вкладену піддіаграму ДВВ (рис. 4).

Розробка програмного забезпечення та комп'ютерні науки взагалі є однією з тих галузей, де МГ застосовуються найчастіше, тому поданий пакет й винесено у вигляді окремої залежної структури (рис. 3). Складність та велика кількість модулів і протоколів у сучасних програмних продуктах сильно ускладнює розуміння їх роботи, керування нею та її оптимізацію. Тому дуже часто складаються МГ програм, причому найчастіше це робиться автоматично трансляторами чи компіляторами.

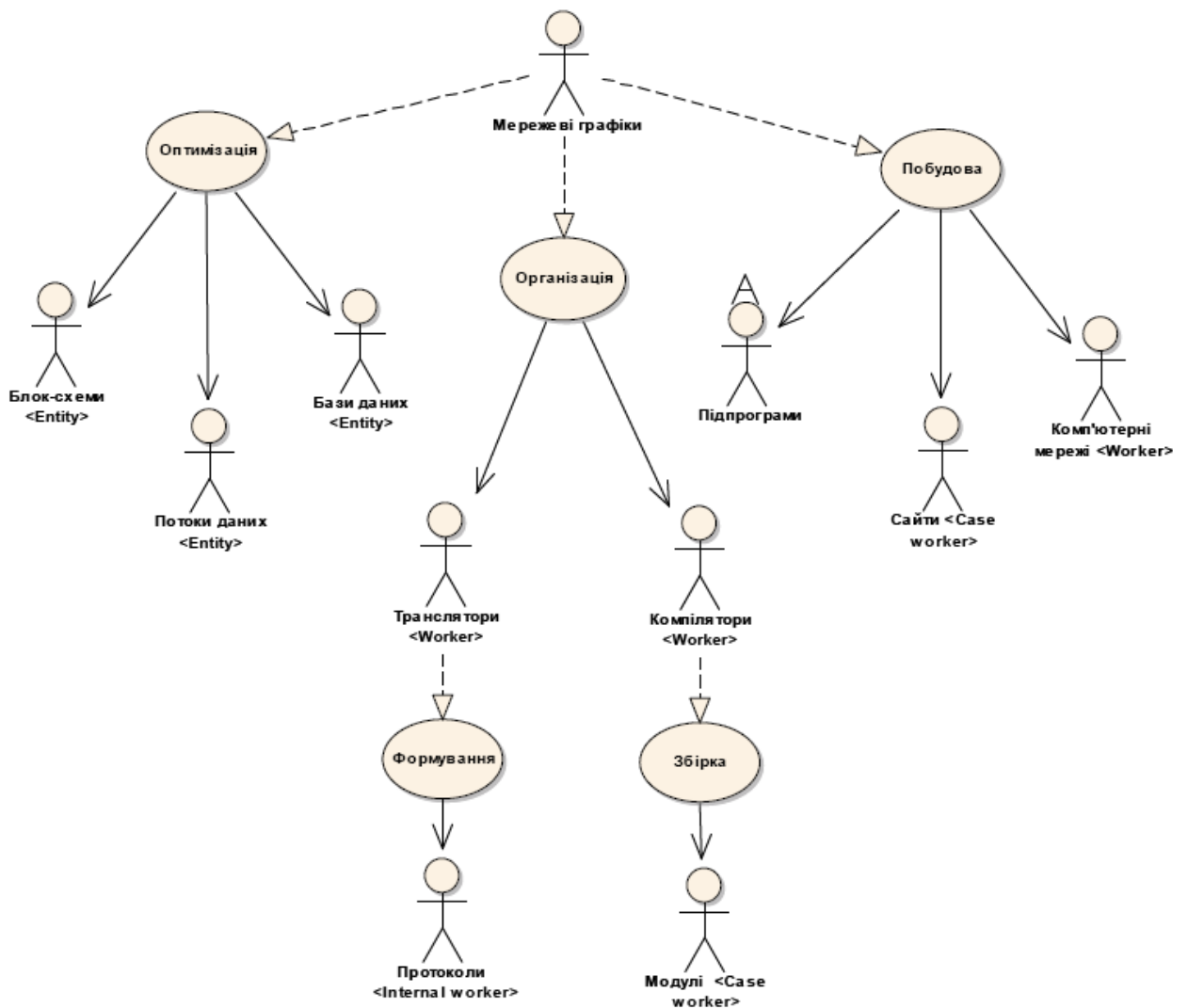


Рисунок 4 – Діаграма-пакет «Розробка програмного забезпечення»  
ДВВ ПЗ проектування МГ

МГ також є зручними для зображення структур даних, блок-схем, потоків даних, схем баз даних та баз знань, скінченних автоматів, схем комп'ютерних мереж та окремих сайтів, схем викликів підпрограм тощо. Також МГ широко використовуються у багатьох алгоритмах пошуку та сортування.

Крім того, одним з головних напрямків сучасних досліджень у галузі глобальних мереж – є задане консорціумом W3C – завдання побудови семантичної мережі (один з видів МГ) на базі існуючої мережі Інтернет.

В рамках виконання поданої дипломної роботи – була виконана спроба об'єднати усе вищезазначене у діаграму-пакет «Розробка програмного забезпечення», що наведена на рис. 4. Розглянемо детальніше спроектовану структуру.

МГ (актор зі стереотипом «business actor») реалізує (за допомогою зв'язків «realization») наступні ВВ (зі стереотипами «business use case»), стосовно до програмного забезпечення:

- оптимізація;
- організація;
- побудова.

В свою чергу, кожен з ВВ виконує дії, спрямовані на конкретні сутності, а саме відбувається:

- а) оптимізація (з асоціативними зв'язками «association») блок-схем, потоків даних, баз даних (актори зі стереотипами «entity»);
- б) організація (з асоціативними зв'язками «association») трансляторів та компіляторів (актори зі стереотипами «worker»), кожен з яких реалізує (за допомогою зв'язків «realization»):
  - 1) формування (ВВ зі стереотипом «business use case») протоколів за внутрішніми принципами (актор зі стереотипом «internal worker», об'єднаний асоціативним зв'язком («association») з ВВ «формування»);
  - 2) збірку (ВВ зі стереотипом «business use case») модулів за допомогою інструментальних засобів (актор зі стереотипом «case worker», об'єднаний асоціативним зв'язком («association») з ВВ «збірка»);
- в) побудова (з асоціативними зв'язками «association») підпрограм, як вбудованої структури (актор зі стереотипом «agent»), сайтів та комп'ютерних мереж (актори зі стереотипами «worker»).

## 2.2 Проектування діаграми послідовності

Діаграми послідовності (ДП) призначені для формування уяви програміста про порядок виконання дій при роботі з майбутнім ПЗ, які сформовані

системним архітектором [15]<sup>1)</sup>. Стандартно існує ДП двох основних типів: ДП для відображення дій програміста при розробці ПЗ та ДП для відображення дій користувача при подальшій роботі з майбутнім ПЗ. ДП саме другого типу використовуються для створення інструкцій користувача, які є складовою частиною програмно-методичних комплексів (ПМК).

Саме такий тип ДП спроектовано у поданій дипломній роботі. Перейдемо до детального опису сформованої структури (рис. 5).

Проектування ДП починається із визначення об'єктів – сутностей, які розташовані у верхній частині ДП. Критерієм для відбору сутностей є майбутні класи або компоненти із якими взаємодіє користувач (або розробник для першого типу ДП). Двокрапка у структурі назви об'єкту означає майбутній клас.

Кожен об'єкт має так звану «лінію життя» («life line») – пунктирна лінія, що простягається на усю довжину полотна ДП. У початковий момент «лінія життя» – є досить короткою, проте, з додаванням повідомлень до ДП – лінія автоматично збільшується.

Вздовж «лінії життя» ковіає індикатор активності – суцільний прямокутник, що показує довжину процесу активності кожного об'єкта. Саме до індикатора активності спрямовуються усі повідомлення. Цих індикаторів на лінії життя може бути безліч, проте, вони мають різну довжину, наприклад індикатор активності користувача (див. рис. 5) має найбільшу довжину, що означає його активність протягом дії повідомлень, що розглядаються.

На сформованій ДП присутні наступні об'єкти:

- користувач (актор);
- форма проекту (клас);
- робота (клас-сутність «entity»);
- розрахунок (обчислювальний клас «enumeration»);
- вид (метаклас «metaclass»);

---

<sup>1)</sup> [15] Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя. 2-е изд. Москва : ДМК-Пресс, 2004. 432 с.

- подія (клас-сутність «entity»);
- таблиця параметрів (табличний клас «table»)

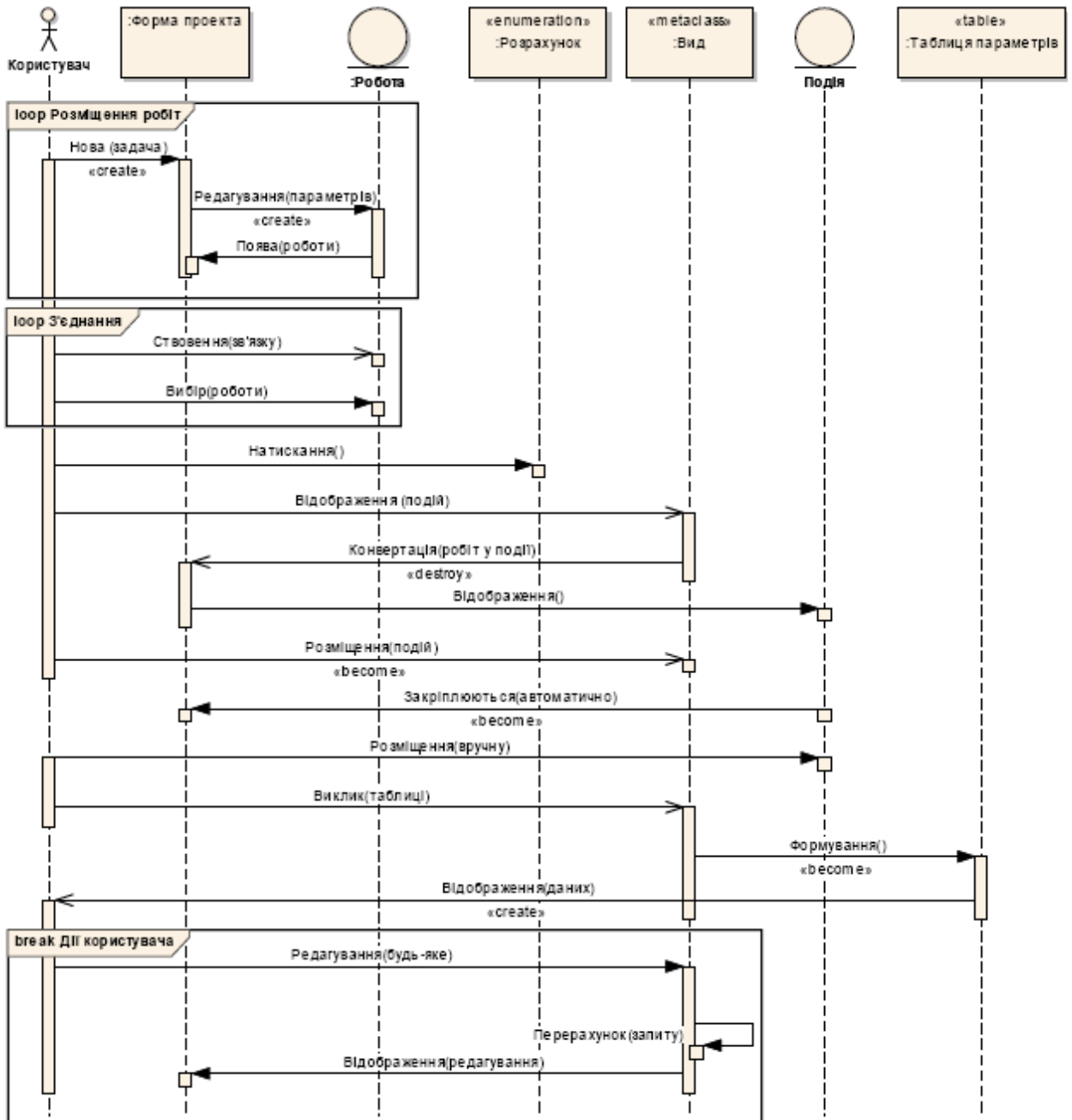


Рисунок 5 – ДП ПЗ проектування МГ

Перші повідомлення на ДП об'єднані у складову частину («combined fragment») «Розміщення робіт», що організована за замкненим циклом («loop»). До цього циклу включено повідомлення:

- нова (задача);
- редагування (параметрів);
- поява (роботи).

Розглянемо детальніше ці повідомлення на ДП.

Перше повідомлення «Нова (задача)» має назву «Нова» та аргумент «задача», що вказується у дужках (тут і далі усі аргументи повідомлень будуть відразу подаватися у дужках, без подальшої вказівки, що це є аргументом) зі стереотипом «create» – спрямовано від «Користувача» до «Форми проекту»; воно показує, що для створення нової задачі користувачеві необхідно звернутися до відповідного класу-форми, представленої інтерфейсом у вигляді робочої області МГ.

Друге повідомлення «Редагування (параметрів)» зі стереотипом «create» – спрямовано від «Форми проекту» до класу-сутності «Робота»; воно показує, що відразу після створення нової задачі на формі проекту – необхідно спочатку створити, а потім (за потреби) відредагувати параметри роботи.

Третє повідомлення «Поява (роботи)» з простим стереотипом – спрямовано у зворотному напрямку від «Роботи» до «Форми проекту»; воно показує, що відразу після внесення змін у параметри роботи – відповідна робота з'явиться на формі проекту та стане видимою для користувача, про що свідчить індикатор активності на лінії життя «Користувач», який не переривається.

На цьому повідомленні завершується складова частина («combined fragment») «Розміщення робіт» – розглянутий цикл буде виконано стільки разів, скільки знадобиться користувачеві, виходячи з необхідної кількості створення робіт або його власних задумів.

Наступні повідомлення на ДП об'єднані у наступну складову частину («combined fragment») «З'єднання», що також організована за замкненим циклом («loop»). Ця частина (за повтореннями) схожа з попередньою, також – цикл буде виконано стільки разів, скільки знадобиться користувачеві, тільки

вихідною кількістю циклів буде необхідна кількість зв'язків, що створюються, а не робіт. До цього циклу включено повідомлення:

- створення (зв'язку);
- вибір (роботи).

Перше повідомлення «Створення (зв'язку)» з асинхронним («asynchronous») типом керування потоку («Control Flow Type») – спрямовано від «Користувача» до класу-сутності «Робота»; воно показує, що користувач обирає команду «Створити зв'язок» – асинхронність показує відсутність відповіді з боку об'єкту «Робота».

Друге повідомлення «Вибір (роботи)» з простим стереотипом – спрямовано від «Користувача» до класу-сутності «Робота»; воно означає, що користувач створює зв'язок із тією роботою, яку необхідно включити до МГ – відповідна робота з'явиться на формі проекту та стане видною для користувача, про що свідчить синхронність повідомлення. Це повідомлення є останнім у межах циклу складової частини «З'єднання».

Два наступних повідомлення спрямовано від «Користувача» (рис. 5) Одне – «Натискання» – до обчислювального класу «Розрахунок» – вказує, що користувач може виконати необхідний розрахунок за допомогою виконання обчислення. Друге повідомлення «Відображення (подій)» – до «Метакласу «Вид»» – показує звертання до панелі меню «Вид», що планується з метою відображення топології МГ у вигляді подій. Саме це повідомлення виконано з асинхронним («asynchronous») типом керування потоку («Control Flow Type»).

Наступне повідомлення «Конвертація (робіт у події)» також є асинхронним. Спрямовано від «метакласу «Вид»» до «Форми проекту», має руйнуючий стереотип «destroy», що значить руйнування топології МГ з подальшою її перебудовою та за мету якого є виконання конвертації сутностей «Робота» у «Подію».

Слідом, від «Форми проекту» до «Події», йде службове повідомлення «Відображення», що як раз й виконує графічне представлення спроектованих робіт у формі нових сутностей у «Подія».



Помітимо, що протягом створення всіх цих повідомлень – індикатор активності «Користувача» – є безперервним (див. рис. 5), що свідчить про обов'язкову участь користувача у всій послідовності подій, що розглянуто.

Далі асинхронне повідомлення «Розміщення (подій)» зі стереотипом звертання («become»), що спрямовано від «Користувача» до «Метакласу «Вид»» – показує завдання на виконання команди розміщення подій, що віддається користувачем.

Відразу від класу-сутності «Подія» до «Форма проекту» йде синхронне повідомлення «Закріплюються (автоматично)» зі стереотипом звертання («become»), що означає автоматичне закріплення сформованих сутностей-подій за сіткою форми проекту.

У разі, якщо топологія виконаного закріплення не задовольняє користувача – він може розмістити події за формою вручну. На це вказує повідомлення «Розміщення (вручну)», спрямоване від «Користувача» до «Події».

Далі (за потребою) користувач може викликати побудову спеціальної таблиці, що містить усі розраховані параметри у числовій формі – асинхронне повідомлення «Виклик (таблиці)», що спрямоване від актора «Користувач» до «Метакласу «Вид»».

Слідом відбувається створення нового об'єкту «Таблиця параметрів», що являє табличний клас «table» – про це свідчить повідомлення «Формування» зі стереотипом звертання («become»), яке спрямовано від «Метакласу «Вид»» до «Таблиці параметрів».

Наступне асинхронне повідомлення «Відображення (даних)» зі стереотипом «create» – виконує відображення створених табличних чисельних даних й спрямовано від «Таблиці параметрів» до «Користувача» тому, що саме йому відображаються сформовані табличні структури.

На завершення створення ДП ПЗ проектування МГ залишається внести етап роботи користувача, пов'язаний з повним або частковим редагуванням відображення МГ (масштабування, переміщення тощо). При виконанні цього етапу, кожний раз йде перерахунок та перебудова відображення всієї тополо-

гії МГ, тому організація цього вбудованого фрагменту («combined fragment») буде мати тип «break» (переривання або розлом), а назву – «Дії користувача».

До складової частини («combined fragment») «Дії користувача» – включено наступні повідомлення:

- редагування (будь-яке);
- перерахунок (запиту);
- відображення (редагування).

Перше повідомлення фрагменту, що розглядається – «Редагування (будь-яке)» спрямовано від «Користувача» до «Метакласу «Вид»» – означає виконання користувачем будь-яких змін у редагуванні зображення МГ.

Друге повідомлення фрагменту – «Перерахунок (запиту)» є рефлексивним («self-message»), тобто спрямовано від «Метакласу «Вид»» само до себе – означає виконання перерахунку сформованого користувачем запиту всередині самого метакласу.

Третє повідомлення – «Відображення (редагування)» спрямовано від «Метакласу «Вид»» до «Форми проекту» – означає виконання відображення наслідків редагування МГ. Протягом створення всіх повідомлень складової частини «Дії користувача» – індикатор активності користувача – не переривається, що свідчить про фіксацію користувачем усієї послідовності подій складової частини та, у разі необхідності, багаторазового повторення редагування.

### **2.3 Проектування діаграми класів**

Діаграми класів (ДК) використовуються для проектування основного формового наповнення майбутнього ПЗ [16]<sup>1)</sup>. Класом зветься деяка сутність, що інкапсулює дані. Стосовно до нотації UML, класи містять атрибути

---

<sup>1)</sup> [16] Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. Пер. с англ. под общей редакцией О. С. Орлова. Санкт-Петербург : Питер, 2006. 736 с.

(власно інкапсульовані дані різного походження) та операції (дії на цими або іншими даними).

Кожні конкретні атрибути та операції класів, що спроектовані у поданій дипломній роботі буде розглянуто нижче. Крім класів на ДК також важливими для аналізу є зв'язки або відношення (це більш точне визначення стосовно до ДК). Взагалі, відношень, щодо методології ДК – існує багате розмаїття, проте зупинимося лише на тих, що присутні у спроектованій ДК проектування МГ (рис. 6).

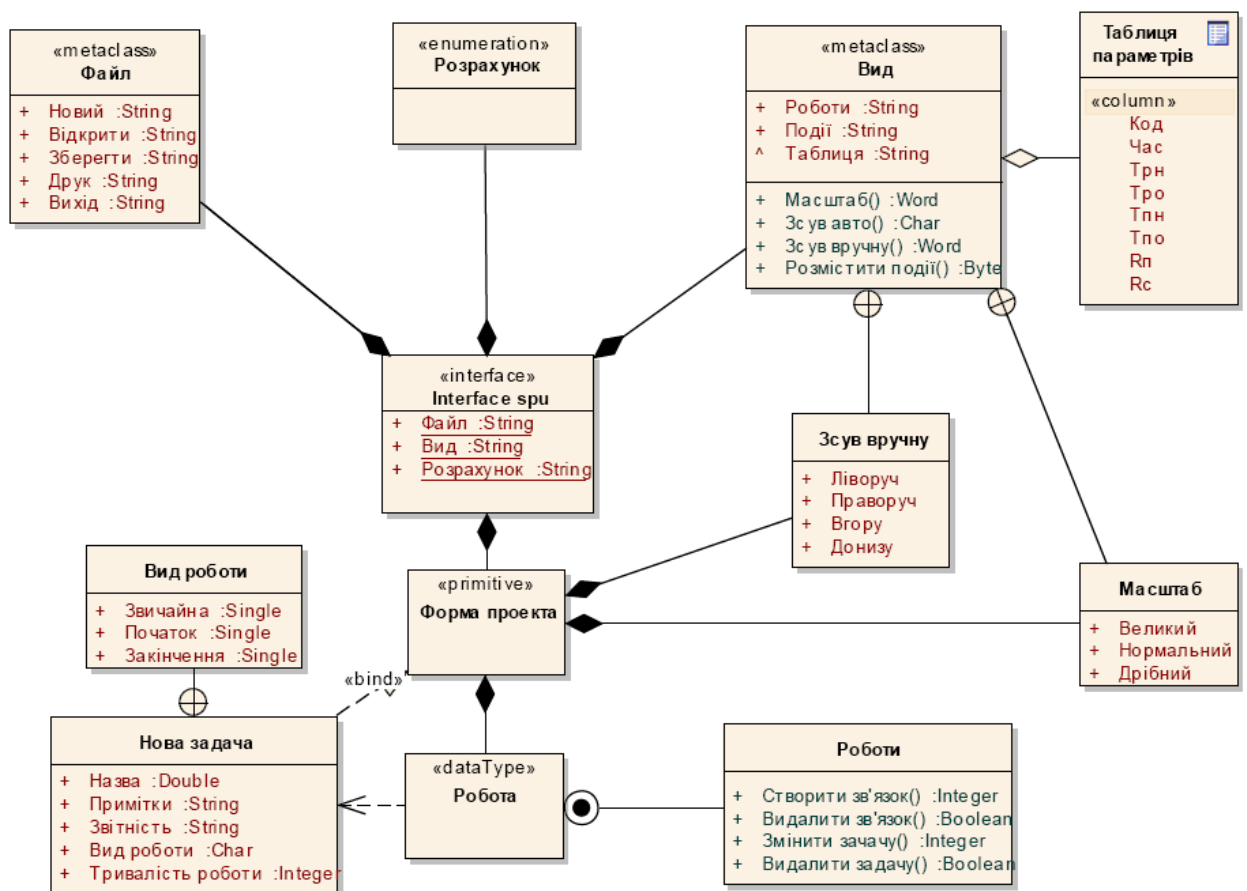


Рисунок 6 – ДК ПЗ проектування МГ

На діаграмі присутні наступні відношення:

- агрегація (aggregation) – це одна з форм асоціації, яка відрізняється деякою незалежністю складових частин так, що у випадку знищення

- цілого – часткове залишається, на ДК позначається у вигляді незафарбованого ромбу з боку цілого;
- композиція (composition) – також одна із форм асоціації, яка відрізняється від агрегації присутністю залежного зв'язку між цілим та частковим, при знищенні цілого – часткове також знищується, на ДК позначається стрілкою у вигляді зафарбованого ромбу з боку цілого;
  - вкладення (nesting) – особлива форма зв'язку (відношення), яке означає безпосереднє доповнення деяких функцій об'єкту-цілого окремими об'єктами-частками, що виконані, здебільшого, у вигляді вкладених переліків, на ДК позначається у вигляді знаку «+» у колі з боку цілого;
  - включення (contained) – форма зв'язку, що має на увазі особливу форму відношення об'єкту-частки, яка не наведена у загальному представленні об'єкта-цілого, а надається тільки після її окремого виклику, на ДК позначається у вигляді зафарбованого кола у ще одному колі з боку цілого;
  - залежність (dependency) – форма відношення, що показує безпосередню залежність одного класу від іншого (головного), зображується у вигляді пунктирної стрілки зі вказівником у бік головної частини;
  - прив'язка (bind) – форма відношення, яка визначає закріплення одного класу за іншим (головним), позначається пунктирною стрілкою, спрямованою у бік головного класу з відповідним написом «bind» вздовж стрілки.

Центральним класом на ДК – є клас-інтерфейс («interface») (рис. 6), що являє собою графічний інтерфейс користувача ПЗ проектування МГ. Цей клас першим постає перед користувачем у вигляді завантажувального примітивного класу (primitive) «Форма проекту», що має з класом «Інтерфейс» композиційну залежність та визначає первинні геометричні розміри вікна інтерфейсу. Також перед користувачем з'явиться панель меню, елементи якої – є атрибутами класу «Інтерфейс».

Отже, атрибутами класу «Інтерфейс» є:

- «Файл», що являє собою окремий метаклас (metaclass) та надає доступ для роботи з різними файлами проектів;
- «Вид», що також являє окремий метаклас (metaclass) та надає різноманітне представлення сутностей проекту;
- «Розрахунок» – являє клас підрахунків (enumeration) та виконує розрахунки параметрів МГ.

Всі атрибути класу-інтерфейсу мають рядковий тип представлення даних («String»).

Метаклас (metaclass) «Файл», пов'язаний композиційною залежністю з класом «Інтерфейс» містить у собі наступні атрибути:

- «Вихід» – надає вихід з ПЗ;
- «Відкрити» – здійснює відкривання збереженого проекту;
- «Друк» – відповідає за виклик налаштувань друку;
- «Зберегти» – виконує запит на шлях зберігання проекту;
- «Новий» – створює шаблон нового проекту.

Всі атрибути метакласу «Файл» мають рядковий тип представлення даних («String»).

Клас підрахунків (enumeration) «Розрахунок», пов'язаний композиційною залежністю з класом «Інтерфейс» (рис. 6), виконує реалізацію чисельних (табличні параметри представлення) та графічних (критичний шлях) параметрів МГ.

Метаклас (metaclass) «Вид», пов'язаний композиційною залежністю з класом «Інтерфейс» містить у собі наступні атрибути:

- «Роботи» – відображення задач у вигляді робіт;
- «Події» – відображення задач у вигляді подій;
- «Таблиця» – подання табличних даних.

Всі атрибути метакласу «Вид» мають рядковий тип представлення даних («String»). Крім атрибутів, метаклас «Вид» має у своєму складі операції:

- «Масштаб» – виконує масштабування МГ, має вкладений (nesting) клас «Масштаб» та word-type формат представлення даних;
- «Зсув авто» – виконує автоматичний зсув (на основі оптимального розміщення задач) МГ за формою, має символний формат (Char) представлення даних;
- «Зсув вручну» – виконує зсув МГ за визначеними користувачем розмірами зсуву, має вкладений (nesting) клас «Зсув вручну» та word-type формат представлення даних;
- «Розмістити події» – виконує автоматичне розміщення подій МГ за формою, має байтовий формат (Byte) представлення даних.

До метакласу «Вид» за допомогою агрегаційного відношення приєднано табличний клас (table) «Таблиця параметрів», що містить стовбцеві атрибути (column):

- Код – цифровий код задачі;
- Час – тривалість задачі;
- Трн – ранній строк початку задачі;
- Тро – ранній строк закінчення задачі;
- Тпн – пізній строк початку задачі;
- Тпо – пізній строк закінчення задачі;
- Rp – повний резерв часу;
- Rc – вільний резерв часу.

Вкладені класи (nesting) «Зсув вручну» та «Масштаб» – приєднані за допомогою відношення-вкладення до метакласу «Вид» та композиції – до примітивного класу (primitive) «Форма проекту», мають атрибути, що не потребують детальних пояснень, а говорять самі за себе. Так вкладений клас «Зсув вручну» має атрибути:

- «Ліворуч»;
- «Праворуч»;
- «Вгору»;

– «Донизу».

А вкладений клас «Масштаб»:

– «Великий»;

– «Нормальний»;

– «Дрібний».

До класу «Форма проекту» ще приєднано такі сутності (рис. 6): тип даних (DataType) «Робота» (відношення-композиція) та клас «Нова задача» (відношення-прив'язка). Також ці дві сутності поєднані між собою за допомогою залежності, спрямованої у бік класу «Нова задача».

Клас «Нова задача», що прив'язано (відношення «Bind») до «Форми проекту», призначено для створення нового та наступного редагування існуючого типу даних (DataType) «Робота», що являє собою графічне зображення сутності «Робота». Клас «Нова задача» містить у собі наступні атрибути:

- «Назва» – формує назву поточної роботи, має подвійний цифровий формат (double) представлення даних, що містить цифру попередньої та поточної роботи;
- «Примітки» – необхідний для створення додаткових пояснень, має рядковий тип («String») представлення даних;
- «Звітність» – формує визначену звітність по роботі, має рядковий тип («String») представлення даних;
- «Вид роботи» – відповідає за наявність або відсутність вхідних та вихідних зв'язків, має вкладений (nesting) клас «Вид роботи» та символічний формат (Char) представлення даних;
- «Тривалість роботи» – створює число, що означає тривалість роботи (в умовних часових одиницях), має цілочисельний (integer) формат представлення даних.

Вкладений (nesting) клас «Вид роботи» – вкладено до класу «Нова задача». Він формує види робіт, що відрізняються наявністю або відсутністю вхідних та вихідних зв'язків, містить у собі наступні атрибути виду:

- «Звичайна» – атрибут за замовченням, який формує вид роботи з можливістю додавання вхідних та вихідних зв'язків;
- «Початок» – атрибут, який формує вид роботи без можливості додавання вхідних зв'язків, але з можливістю – вихідних.
- «Закінчення» – атрибут, який формує вид роботи без можливості додавання вихідних зв'язків, але з можливістю – вхідних.

Усі атрибути вкладеного (nesting) класу «Вид роботи» мають одинично-розрядовий (Single) формат представлення даних.

Крім того, до типу даних (DataType) «Робота» включено (відношення «contained») операційний клас «Роботи», що викликається опційно (наприклад правим щигликом миші). Цей клас містить операції, що виконують визначені їхньою назвою дії над сутностями МГ та не потребують детального пояснення, проте їх перелік наступний:

- «Створити зв'язок» – має цілочисельний (Integer) формат операційного подання даних, що містить номер роботи від якої прокладається зв'язок;
- «Видалити зв'язок» – має формат операційного подання у вигляді логічного виразу (Boolean), ця операція видаляє або залишає виділений зв'язок;
- «Змінити задачу» – має цілочисельний (Integer) формат операційного подання даних, що вказує номер роботи над якою необхідно виконати редакційні виправлення;
- «Видалити задачу» – має формат операційного подання у вигляді логічного виразу (Boolean), ця операція видаляє або залишає виділену задачу.

Таким чином, спроектована ДК має 12 класів-сутностей та 14 відношень, з яких:

- 5 звичайних класів;
- 2 метакласи;
- 1 табличний клас;



- 1 клас-інтерфейс;
- 1 клас-примітив;
- 1 тип даних;
- 1 клас підрахунків;
- 7 композицій;
- 1 агрегація;
- 3 вкладення;
- 1 включення;
- 1 залежність;
- 1 прив'язка.

## 2.4 Проектування діаграми компонентів

Діаграма компонентів (ДКМ) призначена для вивчення складу компонентів майбутнього ПЗ та вказівки послідовності компіляції та збірки окремих модулів [16]<sup>1)</sup>.

Компонентом зветься фізичний модуль коду та фактично містить його у собі. Компонент є найменшою, неподільною складовою частиною ПЗ. Єдиний тип зв'язку, що присутній на ДКМ – це залежності, що спрямовані від компонента, що породжується у бік головного компонента.

Головною вимогою до ДКМ – стандартно висунуто – відсутність циклів, тобто послідовність компонентів повинна бути чіткою та прозорою. Користувач працює із компонентами, які можуть бути йому досяжні – у зворотному порядку.

Спроектвана ДКМ ПЗ проектування МГ приведена на рис. 7.

Розглянемо детальніше її склад.

Першим компонентом, із яким співпрацює користувач – є засіб ідентифікації (artifact) або ярлик виклику «Ярлик для `spi . Ink`», що прикріплено до

---

<sup>1)</sup> [16] Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. Пер. с англ. под общей редакцией О. С. Орлова. Санкт-Петербург : Питер, 2006. 736 с.

виконавчого файлу (executable) «spu2-2 . exe», що запускає новий проект, через відкритий компонент інтерфейсу (exposed interface) «Видимість інтерфейсу».

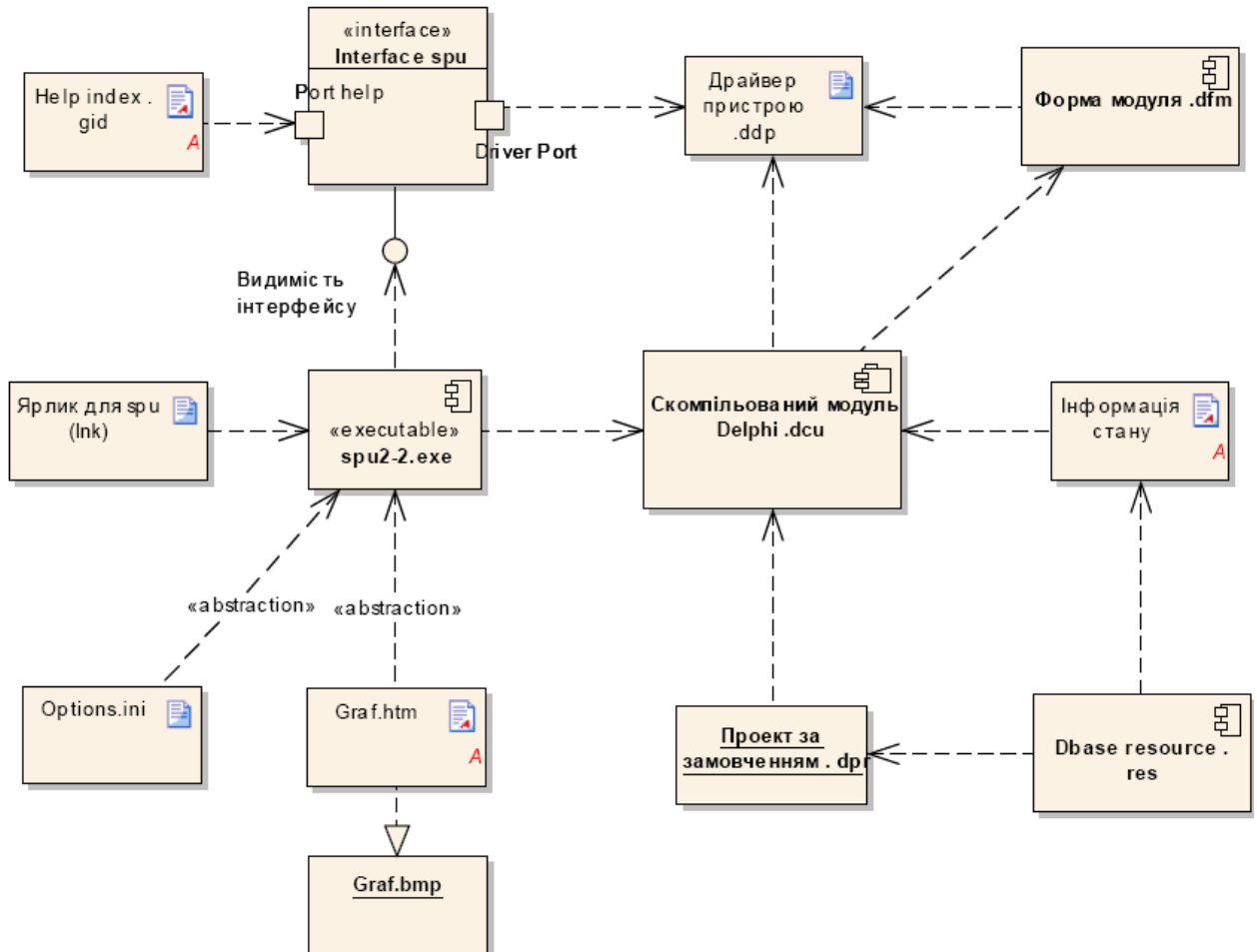


Рисунок 7 – ДКМ ПЗ проектування МГ

Цей компонент служить входом до самого інтерфейсу (interface) «Interface spu», що містить порти допомоги (Port help) та драйверу (Port Driver). До порту допомоги (Port help) підключено контекстну допомогу (document artifact) «Help index . gid», яка може бути опційно викликана користувачем при виниканні складної ситуації із проектом. Через порт драйвера (Port Driver) – підключено об'єкт (artifact), що містить спеціальні дані – драйвер пристрою (Device Driver Profile) «Драйвер пристрою . ddp». Від цього драй-

веру й залежить вигляд та підтримка розрішувальної здатності усього інтерфейсу ПЗ.

Також від «Драйвер пристрою . ddr» залежать компонент (component) форми Delphi-модуля «Форма модуля . dfm» та пакетний компонент (packaging component) скомпільованого модуля Delphi «Скомпільований модуль . dcu», що також залежить від «Форма модуля . dfm». Таким чином, пакетний компонент «Скомпільований модуль . dcu» має подвійну впорядкованість (рис. 7).

Пакетний компонент «Скомпільований модуль . dcu» виступає головним для цілої низки компонентів, а саме для:

- виконавчого файлу (executable) «sru2-2 . exe», який описано вище;
- службового протоколу (document artifact) «Інформація стану», що містить технічну інформацію щодо поточного та останнього стану проекту;
- об'єкту (object) «Проект за замовченням . dpr», який завантажується у вигляді шаблону при створенні нового проекту та який (при необхідності) може бути змінено та доповнено.

Компонент (component) накопичувальної бази даних «Dbase resource . res» має подвійну впорядкованість та компілюється кожного разу при додаванні або зміні інформації у службовому протоколі (document artifact) «Інформація стану» та об'єкту (object) «Проект за замовченням . dpr».

Виконавчий файл (executable) «sru2-2 . exe» є головним для:

- засобу ідентифікації (artifact) або ярлика виклику «Ярлик для sru . lnk», що описано вище;
- службового протоколу (document artifact) опцій та параметрів ПЗ «Options . ini», що містить технічну інформацію щодо останнього геометричного розміру форми проекту, встановленої користувачем;
- браузерної сторінки (Web-document artifact) «Graf . htm», яка відображує остаточну сформовану топологію розміщення задач на МГ; причому браузерна сторінка (Web-document artifact) «Graf . htm» формує

(реалізує) само графічне зображення (object) «Graf . bmp» у вигляді самостійного (автономного) bmp-файлу, що може бути як збережено (надруковано, відправлено тощо), так і відредаговано.

Причому, службовий протокол (document artifact) «Options . ini» та браузерна сторінка (Web-document artifact) «Graf . htm» впорядковані абстрактними (abstraction) залежностями (dependency), що означають створення них самих, як сутностей, тільки за потреби користувача та із неодмінною його участю (внесення змін розміру, команди друку чи Web-конвертації).

Таким чином, спроектована ДКМ ПЗ проектування МГ містить 16 компонентів, об'єднаних 15 зв'язками.

## **2.5 Висновки за розділом**

У даному розділі дипломної роботи змодельована архітектура (проектний «каркас») програмного засобу проектування мережеских графіків організації виробництва, у вигляді декількох діаграм різної природи, виконаних із дотриманням нотації UML 2.5, а саме спроектовано:

- діаграму варіантів використання із вкладеною діаграмою-пакетом;
- діаграму послідовності;
- діаграму класів;
- діаграму компонентів.

## **3 ПРОГРАМНО-МЕТОДИЧНИЙ КОМПЛЕКС ПРОЕКТУВАННЯ МЕРЕЖЕВИХ ГРАФІКІВ**

### **3.1 Призначення програмного засобу**

Програмний засіб (ПЗ), що створюється у поданій дипломній роботі, призначено для створення, розрахунку та оптимізації мережеских графіків. Для його створення став у нагоді матеріал з [17] – [21] <sup>1)</sup>. Для роботи з ним, досить базових знань в галузі мережевого планування. Сам процес створення мережевого графіка не складний, тому навіть його освоєння займе набагато часу.

Методика роботи з мережевими графіками заснована на редагуванні параметрів завдань (робіт). Будь-яка зміна параметрів проводиться правою кнопкою миші, після чого визначається: на якому елементі був зроблений щиклик і пропонується відповідне меню. Ліва кнопка миші призначена для переміщення зображень за формою. Події (віхи) змінюються автоматично при зміні завдань, також присутня можливість змінювати назву подій (права кнопка миші на події).

### **3.2 Основні поняття та позначення, що прийняті у ПЗ**

Для роботи із ПЗ, користувачеві необхідно спеціальною володіти термінологією, основні поняття якої наведено нижче.

---

<sup>1)</sup> [17] Пашеку Х. Программирование в Borland Delphi 2006 для профессионалов. Москва : Вильямс, 2006. 944 с.

[18] Рубенкинг Н. Дж. Язык программирования Delphi для «чайников». Введение в Borland Delphi 2006. Москва : Диалектика, 2007. 336 с.

[19] Культин Н. Основы программирования в Delphi XE. Санкт-Петербург : БХВ-Петербург, 2011. 416 с.

[20] Осипов Д. Базы данных и Delphi. Теория и практика. Санкт-Петербург : БХВ-Петербург, 2011. – 752 с.

[21] Вальвачев А. Н., Сурков К. А., Сурков Д. А. Программирование на языке Delphi: учеб. пос. – Киев : НТУ, 2005. 436 с.

Шлях – це будь-яка послідовність задач у мережевому графіку, у якій кінцева подія одної задачі збігається з початковою подією наступної за нею задачі.

Критичний шлях – максимальний (по тривалості) повний шлях.

Критична задача – будь-яка задача на критичному шляху. Критичні задачі на графіку виділяються червоними кольорами. Особливість критичних задач полягає в тому, що кожна з них повинна починатися точно в момент часу, коли закінчилася попередня та, крім того, тривати вона повинна не більше того часу, що відведено їй за планом. У протилежному випадку, критичний шлях збільшиться.

Отже, критичний шлях повинен бути завжди під контролем керівників робіт тому, що від виконання критичних задач цілком залежить виконання усього плану. Критичні задачі мають нульові резерви часу.

До часових параметрів подій відносяться:

$T_p(i)$  – ранній строк настання події  $i$  (час, необхідний для виконання всіх задач, що передують даній події);

$T_n(i)$  – пізній строк настання події  $i$  (час настання події  $i$ , перевищення якого викличе аналогічну затримку настання завершальної події мережі);

$R(i)$  – резерв часу настання події  $i$  (це такий проміжок часів, на який може бути відстрочене настання цієї події без порушення строків завершення розробки в цілому).

У ПЗ подія позначається колом, задачі – у вигляді стрілок, цифри над якими показують тривалість цих задач (рис. 8). Значення часових параметрів на графіку подій записуються у спосіб, що наведено на рис. 8.

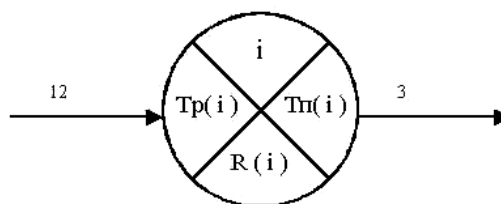


Рисунок 8 – Зображення подій із позначеннями у спроектованому ПЗ

Задачі на мережевих графіках у спроектованому ПЗ мають ідентичне позначення із роботами (рис. 9). Параметри работ записуються всередині квадрату, а стрілки – позначають зв'язки між роботами чи задачами (рис. 9).

Усі інші параметри можна побачити та проаналізувати їх, згенерувавши спеціальну таблицю (п. 3.3.5). Також, у відповідному пункті дипломної роботи, наведено принципи розрахунків цих параметрів.

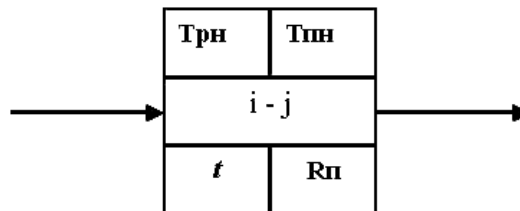


Рисунок 9 – Зображення робіт (задач) із позначеннями у спроектованому ПЗ

### 3.3 Порядок роботи з ПЗ

#### 3.3.1 Створення нової задачі

Для створення нової задачі, необхідно, виконати щиглик правою кнопкою миші по формі – після чого відкриється вікно редагування параметрів задачі, що зображено на рис. 10. Усі задачі, що створюються, поділяються на три види: початкові, кінцеві та звичайні, причому кількість початкових та кінцевих задач може бути більше одиниці. У початкових задачах немає попередників, у кінцевих немає послідовників.

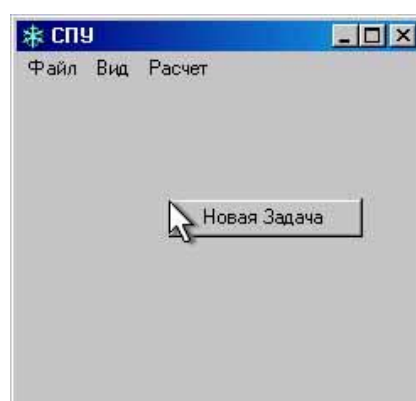


Рисунок 10 – Вікно створення нової задачі

Необхідно звернути увагу на те, що призначення виду роботи "Початок" або "Закінчення" – викликає автоматичне видалення всіх зв'язків задачі, що редагується, таким чином: їх необхідно призначати заново. Графа "Назва" використовується для швидкого пошуку потрібної задачі на мережевому графіку (при наведенні курсору на задачу – висвітлюється підказка). "Примітки" – просто більш докладні пояснення (не є необхідними для заповнення), а "Звітність" – назва події, якою завершується подана задача (ця фраза буде використана при автоматичному формуванні назви подій). Вікно редагування параметрів задач наведено на рис. 11.

Досвід побудови мережевих графіків показав, що графік зручніше будувати у два етапи. Спочатку створити всі задачі, заповнивши графи "Назва" та "Тривалість". Як назву зручно використовувати позначення задачі (1-2 або 1,2 за бажанням).

При розміщенні зв'язків слід використовувати спливаючу підказку, що з'являється при наведенні курсору на задачі. Коли графік вже побудований – варто виконати "Розрахунок". При необхідності ліквідувати помилки, цикли та ін. І тільки коли критичний шлях буде розрахований – можна приступати до заповнення задач додатковою інформацією (змінювати "Назву", додавати "Примітки" та "Звітність").

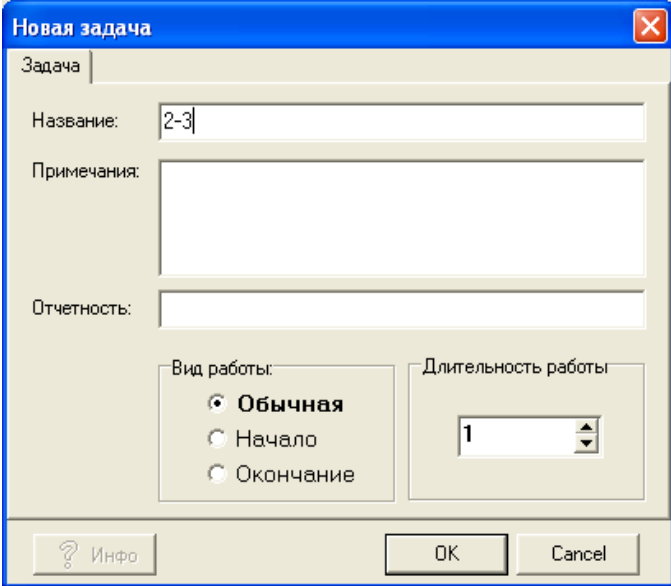


Рисунок 11 – Вікно редагування параметрів задачі



### 3.3.2 Зміна параметрів задачі

Виконавши щиглик правою кнопкою мишки по задачі, одержимо можливість виконати одну із чотирьох операцій:

- створити зв'язок (указати попередню задачу);
- видалити зв'язок (скасувати попередню задачу);
- змінити задачу (відкрити вікно редагування параметрів задачі);
- видалити задачу (разом із задачею віддаляються усі зв'язки, у яких вона бере участь).

### 3.3.3 Створення зв'язків між задачами

У ПЗ використовується тільки один вид зв'язку між задачами – "Попередник".

Порядок зв'язування задач наступний: спочатку клацаємо правою кнопкою по задачі, вибираємо "Створити Зв'язок", потім курсором (у вигляді руки) клацаємо по задачі, що є попередником. Між двома задачами рисується стрілка, спрямована від попередника до послідовника, яка означає, що наступна задача не може початися поки не закінчиться попередня.

Правила побудови мережевих графіків вимагають, щоб більш ранні події перебували на графіку ліворуч, а більш пізні – праворуч. Дотримання такого правила, дозволяє краще проаналізувати виникнення циклів на мережевому графіку. Якщо у конкретному випадку замість стрілки відображується пряма лінія – це значить, що висунута умова не виконується. Тоді, необхідно пересунути, за допомогою лівої кнопки миші, задачу або подію у потрібний бік.

У випадку, якщо ПЗ відмовляється з'єднувати дві задачі – це означає, що користувач намагається порушити встановлені правила складання мережевих графіків. Тоді слід уважно перевірити правильність операції, що проектується. Якщо вірність усіх операцій не викликає сумнівів – необхідно ство-

рити додаткові "Фіктивні" задачі із нульовою тривалістю та вставити їх між задачами у тому місці, де планувався зв'язок.

У деяких випадках, допомагає видалення декількох зв'язків (або задач) з наступним їхнім відновленням, проведеному в іншому порядку. Також ПЗ може запропонувати створити "Фіктивну" задачу автоматично. При цьому можна довірити цю процедуру ПЗ, натиснувши "ОК" або відмовитися та створити додаткову ("Фіктивну") задачу самостійно. В особо важких випадках, коли все з вище переліченого не допомогло – слід закрити ПЗ та почати все заново (із завантаження останнього варіанта збереженої моделі мережевого графіка).

Існує велика розмаїття зв'язків між задачами. Наприклад: наступна задача може починатися, коли попередня задача виконана на 50%. У цьому випадку попередню слід розбити на два прості задачі: першу (обсягом 50%) та другу (також 50%), після чого виконуємо зв'язок їх у звичайному порядку.

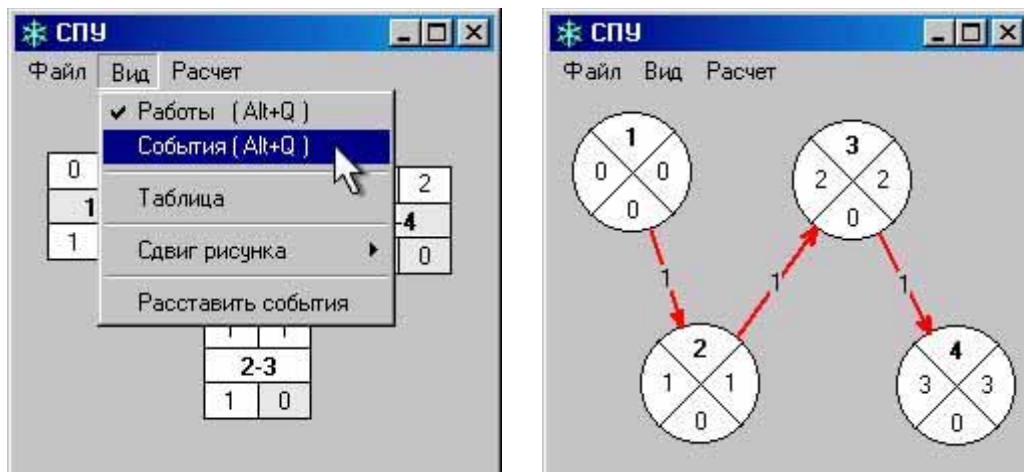
При встановленні одного зв'язку, ПЗ автоматично прораховує всі інші зв'язки так, що після однієї операції по зв'язуванню задач, раптом, може з'явитися ще кілька зв'язків – це одна з особливостей мережевого графіка, побудованого у вигляді "задачі-зв'язки". До речі, саме складність сприйняття користувачем цих численних зв'язків, змушує у звітах використовувати мережеві графіки у вигляді "подія-задача". Однак, процес створення та редагування зручніше виконувати на мережевих графіках, що виконані у вигляді "задачі-зв'язки".

### **3.3.4 Робота з подіями**

Виконати конвертацію мережевого графіка на графік, що побудовано з подій (цей графік не редагується) можна використавши пункт меню "Події" або одночасним натисканням клавіш "Alt + Q" на клавіатурі.

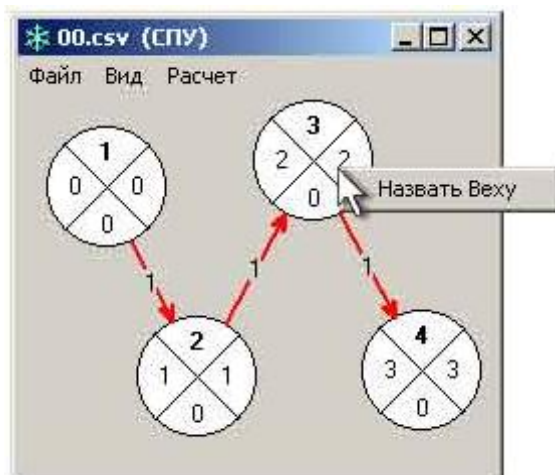
Для автоматичного розташування подій на формі, можна скористатися пунктом меню "Розставити події". Остаточне розміщення подій за формою

виконується вручну за допомогою лівої кнопки миші. Відповідно до встановлених правил, фіктивні задачі (задачі з нульовою тривалістю) на графіку позначаються пунктиром. Порядок конвертації робіт у події – наведено на рис. 12.



а)

б)



в)

Рисунок 12 – Порядок конвертації робіт у події:

а) обрання відображення; б) відображення подій; в) іменування подій

Екранні знімки спроектованих мережевих графіків виконання даної дипломної роботи наведено на рис. 13 (для випадку «вершини-роботи») та на рис. 14 (для випадку «вершини-події»).

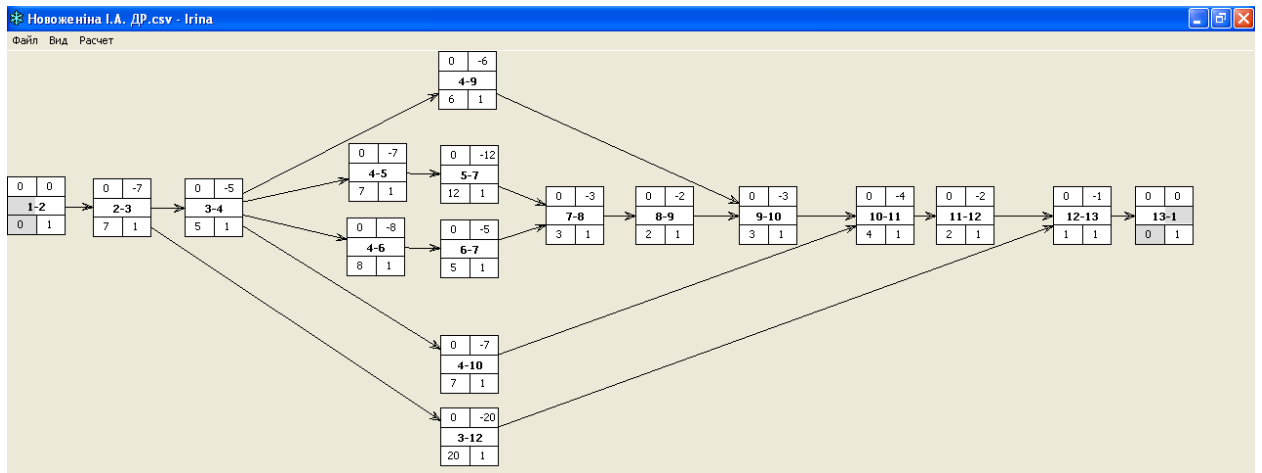


Рисунок 13 – Екранний знімок мережевого графіка виконання дипломної роботи у вигляді «вершини-роботи»

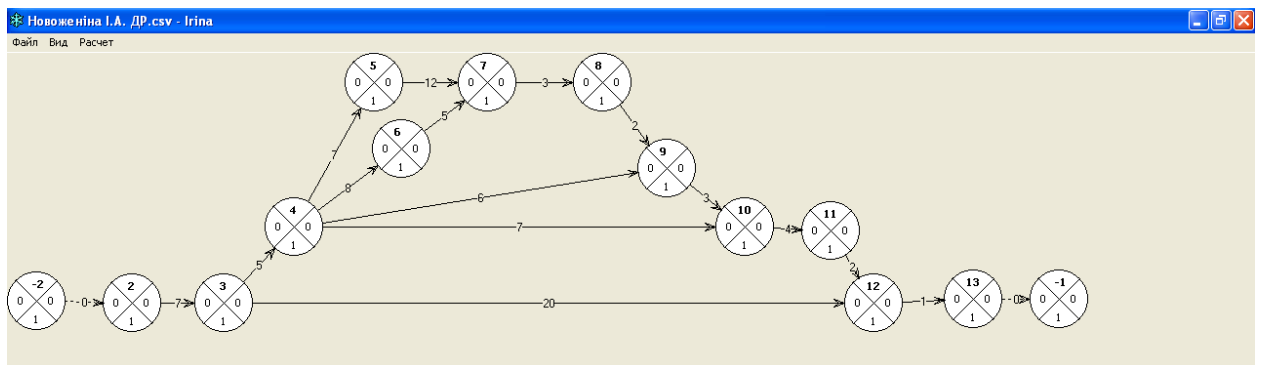


Рисунок 14 – Екранний знімок мережевого графіка виконання дипломної роботи у вигляді «вершини-події»

Слід зауважити, що поки не виконано розрахунок критичного шляху (натисканням пункту меню "Розрахунок"), вписані у задачах та подіях цифри не мають ніякого значення.

У свою чергу, розрахунок виконується тільки у тому випадку, якщо всі задачі пов'язані та присутні початкові й кінцеві задачі. У протилежному випадку, розрахунок буде зупинено, а задача, на якій припинено розрахунок, буде підсвічено жовтими кольором.

Екранний знімок розрахунку критичного шляху мережевого графіку виконання даної дипломної роботи наведено на рис. 15.

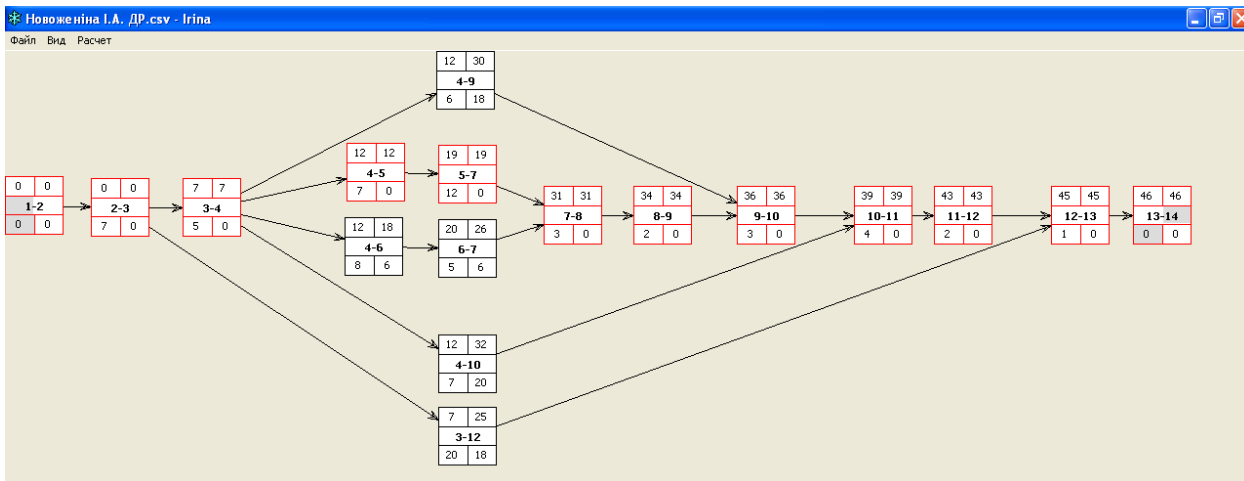


Рисунок 15 – Екранний знімок розрахованого критичного шляху мережевого графіка виконання дипломної роботи

### 3.3.5 Генерування таблиць

Для того, щоб виконати генерацію таблиці, що містить усі проектні розрахунки, необхідно обрати пункт головного меню «Вид», а далі натиснути «Таблиця» рис. 16.

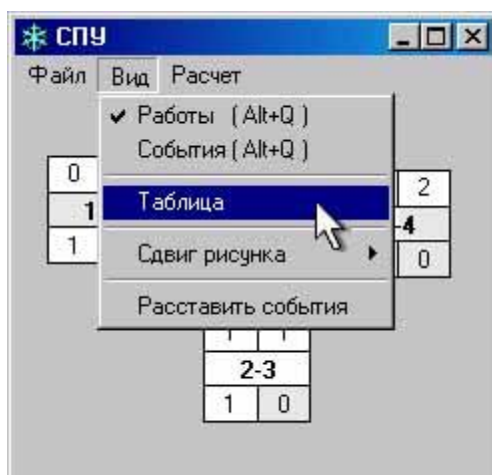


Рисунок 16 – Початкові дії для генерації таблиці

Згенеровану таблицю із розрахунками усіх параметрів мережевого графіку виконання даної дипломної роботи наведено на рис. 17.

Код	t	Трн	Тро	Тпн	Тпо	Рп	Рс
2-3	7	0	7	0	7	0	0
3-4	5	7	12	7	12	0	0
4-5	7	12	19	12	19	0	0
7-8	3	31	34	31	34	0	0
8-9	2	34	36	34	36	0	0
9-10	3	36	39	36	39	0	0
4-9	6	12	18	30	36	18	18
4-10	7	12	19	32	39	20	20
10-11	4	39	43	39	43	0	0
11-12	2	43	45	43	45	0	0
12-13	1	45	46	45	46	0	0
1-2	0	0	0	0	0	0	0
3-12	20	7	27	25	45	18	18
13-14	0	46	46	46	46	0	0
5-7	12	19	31	19	31	0	0
6-7	5	20	25	26	31	6	6
4-6	8	12	20	18	26	6	0

Рисунок 17 – Згенерована таблиця із розрахунками параметрів мережевого графіка виконання дипломної роботи

До основних позначень та їх скорочень (скорочення позначень у ПЗ виконано російською мовою), що зведено до даної таблиці відносять наведені нижче змінні.

$t$  – тривалість задачі (умовні одиниці часу – у нашому випадку – дні).

$T_{рн}$  – ранній строк початку задачі, що дорівнює:

$$T_{рн}(i, j) = T_p(i).$$

$T_{ро}$  – ранній строк закінчення задачі:

$$T_{ро}(i, j) = T_p(i) + t(i, j).$$

$T_{пн}$  – пізній строк початку задачі:

$$T_{пн}(i, j) = T_n(i) - t(i, j).$$

$T_{no}$  – пізній строк закінчення задачі:

$$T_{no}(i, j) = T_n(i).$$

$R_n$  – повний резерв часу:

$$R_n(i, j) = T_n(j) - Tp(i) - t(i, j).$$

$R_c$  – вільний резерв часу:

$$R_c(i, j) = T_p(j) - Tp(i) - t(i, j).$$

### 3.4 Додаткові можливості ПЗ

#### 3.4.1 Перегляд, друк та збереження проекту

При виборі пункту меню "Друк", у папці із проектом створюється файл "graf.htm", що автоматично проглядається за допомогою Internet-браузера, що встановлено на комп'ютері користувача (рис. 18).

Виконавши щиглик правою кнопкою миші по вільному від картинок полю браузера, можна викликати контекстне меню, у якому є пункт "Друк". Вибір цього пункту виводить мережевий графік на принтер. Слід зауважити, що друкується та картинка, яку видно у цей момент на екрані комп'ютера. При цьому сіре підсвічування особливих параметрів задач – відключається.

Також можливо (знову ж за допомогою правої кнопки миші) копіювання отриманих картинок і таблиць у буфер обміну із наступною вставкою до інших будь-яких редакторів документів.

Крім того, у тій саме папці, де зберігся файл "graf.htm" – автоматично створюється файл "graf.bmp", переглянути який можна не за допомогою Internet-браузера, а за допомогою будь-якої програми роботи із зображення-

ми. Цей файл можна переміщати в інші папки, вставляти у звітні документи, змінювати в графічних редакторах.

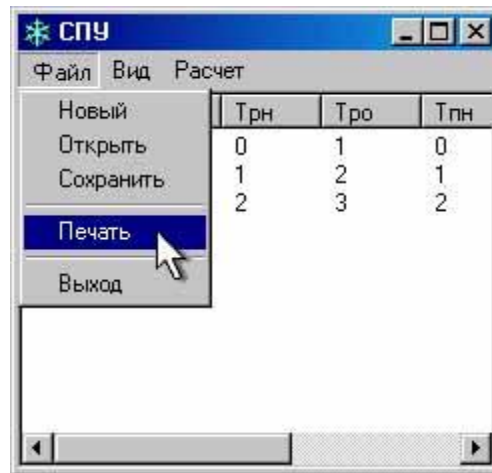


Рисунок 18 – Початкові дії для виконання друку

Екранний знімок мережевого графіку виконання даної дипломної роботи, збереженого у форматі «bmp» та відкритого за допомогою Microsoft Office Picture Manager, наведено на рис. 19.

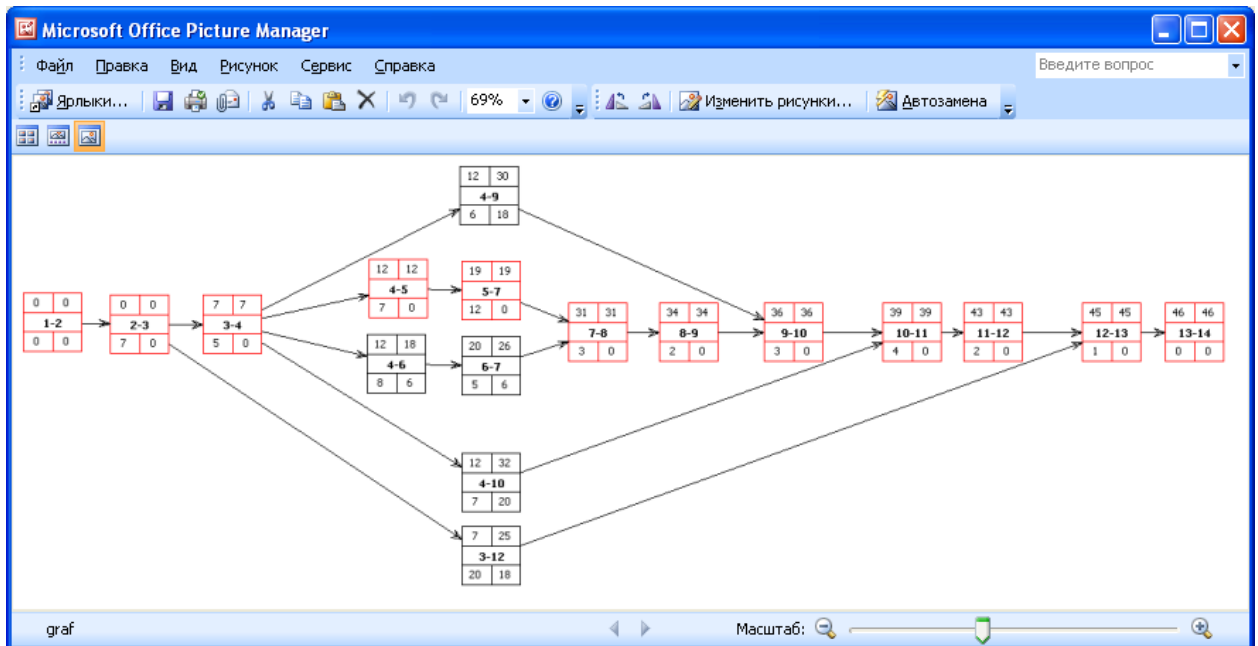


Рисунок 19 – Екранний знімок мережевого графіку виконання дипломної роботи, відкритого за допомогою Microsoft Office Picture Manager



### 3.4.2 Переміщення та зміни розміру структур

Для прискорення переміщень мережових графіків за формою, у спроектованому ПЗ передбачено кілька комбінацій "гарячих" клавіш. Всі вони зазначені у пункті "Зсув рисунку" та приведені на рис. 20.

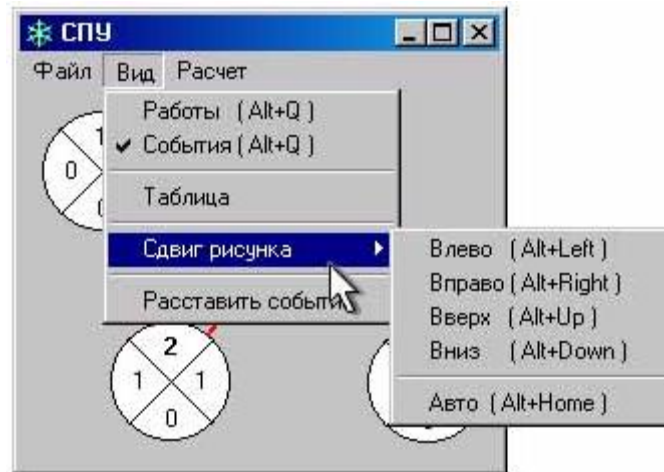


Рисунок 20 – Комбінації "гарячих" клавіш для переміщення сформованого мережевого графіку

Змінити розмір мережевого графіка можна двома способами (рис. 21):

- вибором відповідного пункту меню "Розмір задач";
- за допомогою комбінацій клавіш ("Alt + 1", "Alt + 2", "Alt + 3").

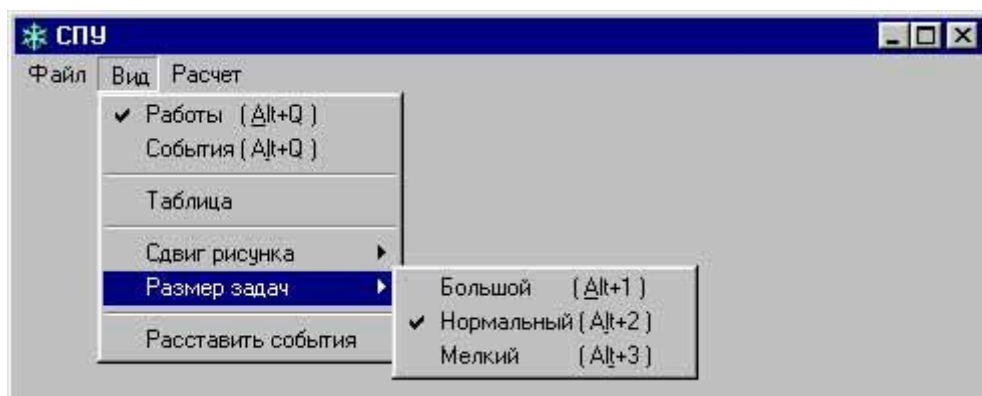
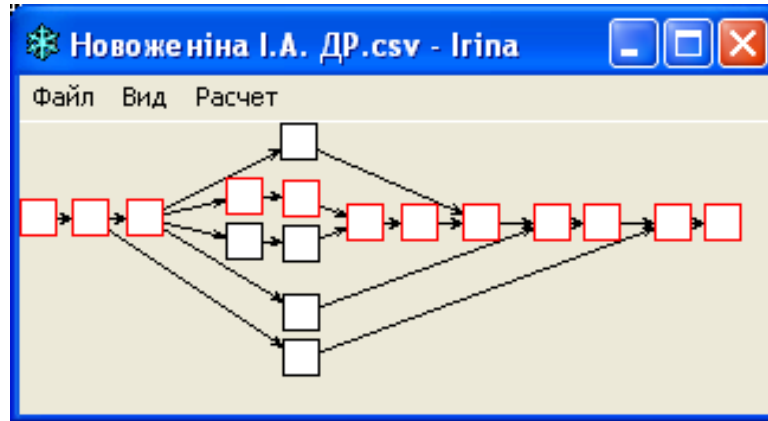
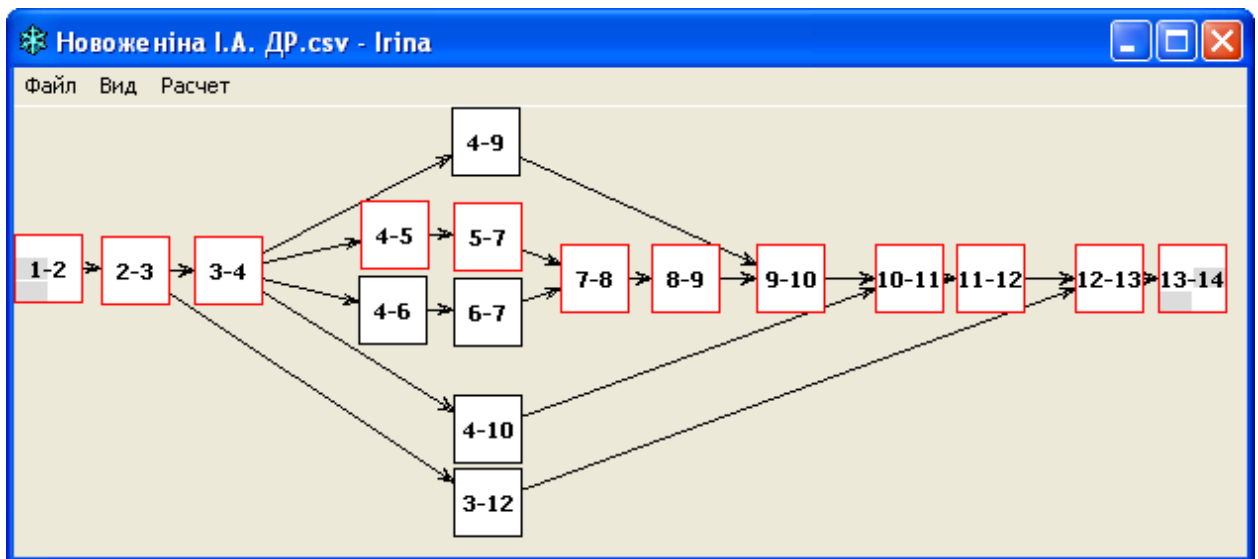


Рисунок 21 – Зміна розміру представлення мережевого графіка

Приклади зміни розміру подання мережеских графіків виконання поданої дипломної роботи для дрібного та середнього розмірів подання задач наведено на рис. 22, 23; для великого розміру – на рис. 13, 14.

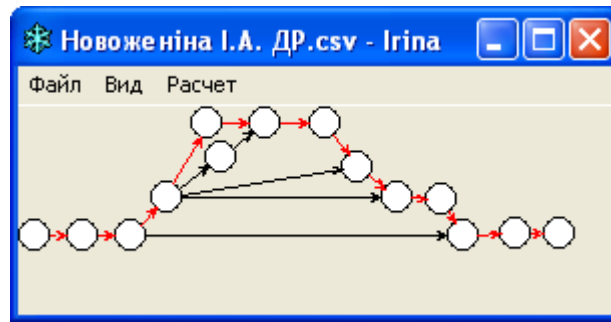


а)

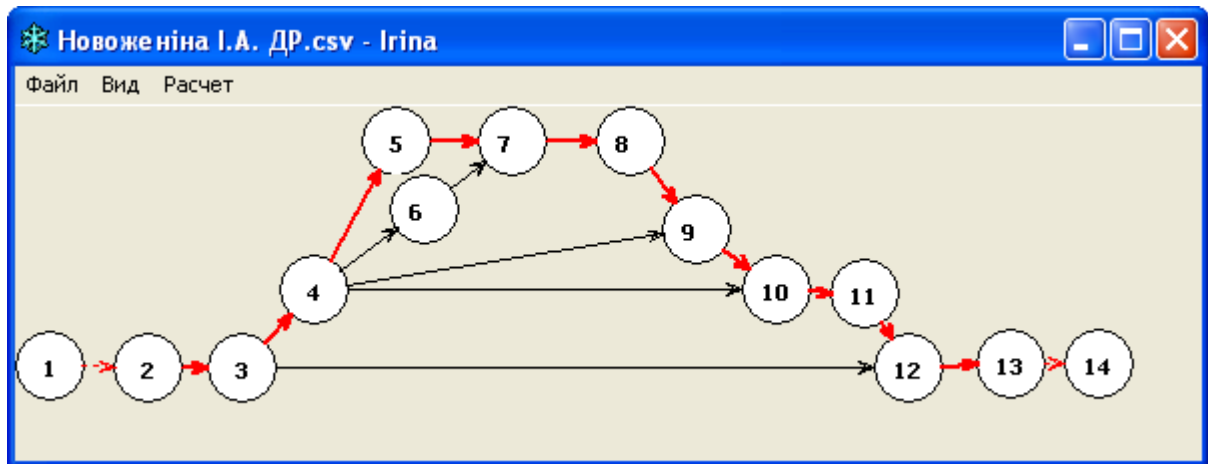


б)

Рисунок 22 – Екранні знімки подання мережеского графіка виконання дипломної роботи з: а) дрібним розміром представлення робіт; б) середнім розміром представлення робіт



а)



б)

Рисунок 23 – Екранні знімки подання мережевого графіка з: а) дрібним розміром представлення подій; б) середнім розміром представлення подій

### 3.5 Додаткові файли

Для повноцінної роботи готового ПЗ – досить одного файлу, що виконується (spu.exe), проте, в момент завершення роботи, у папці із проектом створюється службовий файл "Options.ini" у якому зберігаються останні обрані користувачем положення й розміри вікна. Приклад вмісту збереженого файлу "Options.ini" для останніх змін параметрів представлення мережевих графіків поданої дипломної роботи наведено на рис. 24.

Також папка із ПЗ містить усі збережені проекти мережевих графіків у вигляді файлів з розширенням "\*.csv", що складені за певними правилами. При цьому присутня можливість редагування проектів без входу до ПЗ, що

може знадобитись у деяких випадках (призначення завданням ресурсів та ін.) та хоча вся інформація про використані ресурси у файлі проекту зберігається, для практичного використання цих даних потрібен вже інший ПЗ.

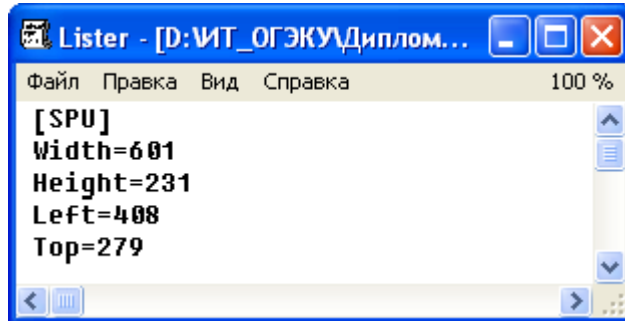


Рисунок 24 – Вміст збереженого файлу "Options.ini" з останніми змінами параметрів мережевих графіків дипломної роботи

Приклад вмісту збереженого файлу "Новоженіна І.А. ДР.csv" з останніми змінами параметрів задач мережевих графіків поданої дипломної роботи наведено на рис. 25.

	A	B	C	D	E	F	G
1	Irina						
2	Pred,Posl,Dlit,Fiktiv,X,Y,Text						
3	6,7,7,0,44,65,,1-2/Составление спецификации,						
4	7,8,5,0,91,65,,2-3/Высокоуровневое проектирование каркаса,						
5	8,9,7,0,175,47,,3-4/Детальное проектирование компонентов,						
6	11,12,3,0,276,69,,3-4/Генерация кода,						
7	12,13,2,0,322,69,,4-5/Тестирование элементов,						
8	13,14,3,0,375,69,,5-6/Комплексное тестирование,						
9	8,13,6,0,221,0,,2-7/Планирование сборки,						
10	8,14,7,0,222,145,,2-8/Планирование системного тестирования,						
11	14,15,4,0,435,69,,6-9/Системное тестирование,						
12	15,16,2,0,476,69,,9-10/Приёмочные испытания,						
13	16,17,1,0,536,69,,10-11/Установка,						
14	5,6,0,4,0,64,,0-1/Начало работ,						
15	7,16,20,0,222,182,,/Документирование,						
16	17,18,0,5,578,69,,/Конец,						
17	9,11,12,0,222,48,,4-5/Создание ветвей математического алгоритма,						
18	10,11,5,0,222,86,,4-6/Проектирование интерфейса пользователя,						
19	8,10,8,0,174,85,,3-4/Детальное проектирование компонентов,						
20	X,Y,N9,Hint						
21	0,112,2,Начало						
22	96,113,7,0-1						
23	132,74,8,2-3						
24	173,0,9,						
25	290,0,12,						
26	323,44,13,4-5/4-6						
27	363,74,14,						

Рисунок 25 – Приклад вмісту збереженого файлу «Новоженіна І.А. ДР.csv»

### 3.6 Висновки до розділу

В процесі виконання даного розділу дипломної роботи, було розроблено ПМК проектування мережевих графіків, створення якого складалося з наступних етапів:

- сфера призначення;
- виділення основних понять та позначень;
- визначення порядку роботи з ПМК, що складається зі створення нової задачі, зміни параметрів задачі, створення зв'язків між задачами, видалення зв'язків між задачами, роботи з подіями;
- генерування спеціальних таблиць;
- розкриття додаткових можливостей;
- реалізація перегляду, друку та збереження інформації, переміщення та зміни розміру структур;
- складання та підключення додаткових файлів.

## ВИСНОВКИ

В процесі виконання дипломної роботи, було досягнуто мету роботи, а саме – створення програмного засобу автоматизованого проектування мережевих графіків організації виробництва.

Об'єктом роботи став мережевий графік виробничого процесу.

Предметом роботи – програмний засіб побудови мережевої динамічної моделі.

Цей програмний засіб створено для кінцевого користувача, якому важливо знати лише послідовність робіт та тривалість кожної з них, і не має особливого значення, яким способом сформований графік, тобто якого він типу тощо.

Слід зазначити, що хоча у сучасних платних спеціалізованих пакетах комп'ютерних програм планування і оперативного управління, в основному, використовується тип графіків «вершини – роботи» – створений програмний засіб зможе працювати з усіма типами мережевих графіків із можливостями їх всебічної трансформації.

Під час дипломного проектування було виконано та, надалі, при складанні пояснювальної записки – детально описано наступні етапи:

- аналіз предметної галузі роботи;
- огляд найближчих аналогів та засобів їх представлення;
- принципи проектування мережевих графіків;
- проектування діаграми варіантів використання;
- проектування діаграми послідовності;
- проектування діаграми класів;
- проектування діаграми компонентів;
- призначення програмного засобу;
- основні поняття та позначення, що прийняті у ПЗ;
- порядок роботи з ПЗ;
- додаткові можливості ПЗ;

– додаткові файли.

У проектній частині дипломної роботи розглядаються проекти рішення, що запропоновані розробником. Зміст проектної частини визначається, по-перше, специфікою теми дипломної роботи, по-друге, особливостями конкретних технічних пропозицій до роботи.

У поданій дипломній роботі проектується архітектура (проектний «каркас») програмного засобу, що розроблюється, у вигляді декількох діаграм різної природи, виконаних із дотриманням нотації UML 2.5.

Для роботи зі створенням ПМК, досить базових знань в галузі мережевого планування. Сам процес створення мережевого графіка не складний, тому навіть його освоєння займе набагато часу.

У складі ПМК розроблено інструкцію користувача із застосування необхідного програмного забезпечення, що доповнені коментарями, відносно роботи спроектованого ПЗ.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Мережеве планування. URL: <http://uk.wikipedia.org/wiki/> (дата звернення: 04.04.2019).
2. Березина Л. Ю. Графы и их применение : Пособие для учителей. Москва: Высшая школа, 1989. 326 с.
3. Михеева В. С. Математические методы в экономической географии. Ч. 2. Приложение теории графов : Курс лекций. Москва : Математика, 1983. 316 с.
4. Голиков А. П., Трофимов А. М., Черванёв И. Г. Математические методы в географии. Харьков : Наука, 1986. 208 с.
5. Глазжан І. М., Новиков В. Г. Основи мережевого планування та управління. Харків : Вид-во ХГУ, 1996. 198 с.
6. Оре О. Теория графов. Москва : Наука, 1968. – 336 с.
7. Уилсон Р. Введение в теорию графов : пер. с англ. Москва : Мир, 1977. 208 с.
8. Bondy J.A. Graph Theory with Applications. North-Holland : Elsevier, 1986. 264 p.
9. Емеличев В. А., Мельников О. И., Сарванов В. И. Лекции по теории графов. Изд. 2, испр. Москва: Наука, 2009. 392 с.
10. Коммерческие специализированные программы для построения графов. URL: <http://www.boost.org/> (дата звернення: 02.04.2019).
11. LION Graph Visualizer. URL: <http://lion.disi.unitn.it/intelligent-optimization/visualizer.html> (дата звернення: 03.04.2019).
12. Графоанализатор – среда визуализации графов. URL: <http://grafoanalizator.unick-soft.ru/> (дата звернення: 03.04.2019).
13. Ларман К. Применение UML 2.0 и шаблонов проектирования. 3-е изд. Москва : Вильямс, 2006. 736 с.
14. Шмуллер Дж. Освой самостоятельно UML 2.0 за 24 часа. Практическое руководство. Москва : Вильямс, 2005. 416 с.



15. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя. 2-е изд. Москва : ДМК-Пресс, 2004. 432 с.
16. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. Пер. с англ. под общей редакцией О. С. Орлова. Санкт-Петербург : Питер, 2006. 736 с.
17. Пашеку Х. Программирование в Borland Delphi 2006 для профессионалов. Москва : Вильямс, 2006. 944 с.
18. Рубенкинг Н. Дж. Язык программирования Delphi для «чайников». Введение в Borland Delphi 2006. Москва : Диалектика, 2007. 336 с.
19. Культин Н. Основы программирования в Delphi XE. Санкт-Петербург: БХВ-Петербург, 2011. 416 с.
20. Осипов Д. Базы данных и Delphi. Теория и практика. Санкт-Петербург: БХВ-Петербург, 2011. – 752 с.
21. Вальвачев А. Н., Сурков К. А., Сурков Д. А. Программирование на языке Delphi: учеб. пос. – Киев : НТУ, 2005. 436 с.