

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Я.І. Кінах, І.В. Бойко

**МЕТОДОЛОГІЯ ТА ТЕХНОЛОГІЯ СТВОРЕННЯ
СКЛАДНИХ ПРОГРАМНИХ
СИСТЕМ**

Навчально-методичний посібник

Тернопіль
2017

УДК 681.3
М54

Укладач:

Кінах Я.І., канд. техн. наук, доцент,
Бойко І.В., канд. техн. наук, доцент.

Рецензенти:

І.З. Якименко, канд. техн. наук, доцент,
Є.О. Марценюк, канд. техн. наук, доцент.

Навчально-методичний посібник розглянуто й затверджено на засіданні факультету комп'ютерно-інформаційних систем і програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя.
Протокол № 2 від 27 вересня 2017 р.

Методологія та технологія створення складальних програмних
М54 систем : навчально-методичний посібник / Укладачі : Кінах Я.І.,
Бойко І.В. – Тернопіль : Тернопільський національний технічний
університет імені Івана Пулюя, 2017 – 40 с.

УДК 681.3

Даний посібник написано згідно програми дисципліни “Методологія та технологія створення складних програмних систем” що читається на факультеті комп'ютерно-інформаційних систем і програмної інженерії. Посібник також містить методичні вказівки призначені для виконання курсових робіт студентами, що навчаються за напрямом підготовки 121 “Інженерія програмного забезпечення” та інформацію про структуру та зміст курсової роботи

Для студентів спеціальності 121 – “Інженерія програмного забезпечення”, аспірантів та викладачів вищих навчальних закладів.

© Кінах Я.І., Бойко І.В., 2017
© Тернопільський національний технічний
університет імені Івана Пулюя, 2017

ВСТУП

Сьогодні програмна індустрія пропонує сучасному інженеру низку різноманітних засобів моделювання складних об'єктів, що дозволяють не тільки моделювати складні динамічні системи, але і проводити з ними експерименти. Найбільш повне дослідження загальносистемних проблем виходить у результаті моделювання об'єктів за допомогою сучасних технологій, реалізованих у спеціалізованих обчислювальних пакетах візуального моделювання.

Існує багато пакетів візуального моделювання. В них користувачеві надається можливість описувати моделюючу систему переважно у візуальній формі, наприклад, графічно представляючи як структуру системи, так і її поведінку (наприклад, за допомогою карти станів). Такий підхід дозволяє користувачеві не слідкувати за програмною реалізацією моделі, що значно спрощує процес моделювання. Результати експерименту в пакетах візуального моделювання надаються в більш наочній формі: у вигляді графіків, гістограм, схем, із застосуванням анімації і т.д. Також підтримується технологія об'єктно-орієнтованого моделювання, що дозволяє повторно використовувати екземпляри моделей з можливістю внесення в них тих чи інших змін.

З безлічі існуючих на сьогоднішній день пакетів візуального моделювання складних об'єктів особливий інтерес викликають універсальні пакети, не орієнтовані на визначену вузькоспеціальну область, а що дозволяють моделювати структурно-складні гібридні системи у різних прикладних задачах.

Сучасна практика програмування вимагає вдосконалення прийомів розробки програмного забезпечення, прискорення та полегшення процесу його проектування. При цьому програмні засоби повинні бути орієнтовані на тривалу роботу з користувачем. Це ставить перед розробниками завдання, що спрямовані не тільки на виконання технологічних вимог до програмної системи, а й застосування сучасних методів програмної інженерії для розробки програмного забезпечення.

Особливі можливості в цьому контексті надає об'єктно-орієнтована методологія і відповідне середовище проектування та програмування, що дозволяє швидко конструювати програмний засіб, використовуючи вже існуючі компоненти та моделюючи майбутню програмну систему.

Проектування – це діяльність, що спрямована на досягнення визначеного результату при заданих обмеженнях по ресурсах і термінах, а також вимогах до якості і допустимого рівня ризику. В результаті проектування предметні сутності проблемного середовища представляються системою взаємопов'язаних логічних об'єктів[8].

ЗМІСТ

ВСТУП.....	3
ЗМІСТ	5
1. ОСНОВНИЙ ЗМІСТ	6
1.1. Тематика курсової роботи.....	6
1.2. Завдання на курсову роботу.....	6
1.3. Порядок роботи над курсовим проектом	7
1.4. Загальні вимоги до обсягу курсової роботи (КР).....	9
1.5. Технічне завдання	9
1.6. Вимоги до змісту й оформлення курсових робіт та проектів	10
1.6.1.Титульний лист.....	10
1.6.2. Завдання на курсове проектування.....	10
1.6.3.Реферат	10
1.6.4.Технічне завдання.....	11
1.6.5.Пояснювальна записка (ПЗ).	11
2. ЗАВДАННЯ НА КУРСОВУ РОБОТУ	13
2.1. Розробка складної системи мовою блокового моделювання.....	13
2.2. Розробка складної системи мовою фізичного моделювання	13
2.3. Розробка складної системи на схемах гібридного автомата	14
2.4. Опис системи MatLab	15
2.5. Опис системи Wolfram Mathematica	20
3. ЗАГАЛЬНІ ВИМОГИ ДО ОФОРМЛЕННЯ ТЕКСТОВИХ ДОКУМЕНТІВ... 26	
4. ЗАГАЛЬНІ ВИМОГИ ДО ГРАФІЧНОЇ ДОКУМЕНТАЦІЇ.....	30
4.1. Вимоги до змісту графічної документації.....	30
4.2. Вимоги до оформлення графічної документації	30
4.3. Вимоги до виконання схем виробів	30
4.4. Умовні графічні позначення, що застосовуються на схемах	32
Список використаних джерел	33
Додаток А	34
Додаток Б.....	35
Додаток В	37
Додаток Г.....	38

1. ОСНОВНИЙ ЗМІСТ

1.1. Тематика курсової роботи

Тематика курсового проектування за даним курсом охоплює методологію та технологію створення складних програмних систем.

Теми курсових робіт розробляються і затверджуються кафедрою, що веде цей курс. Об'єктом розробки в курсовій роботі можуть бути програмні модулі і мережні засоби аналогової та цифрової техніки.

Тематика курсового проектування будується на основі фактичного матеріалу промислових підприємств, на матеріалах практики студентів, на базі наукових праць викладачів і студентських навчально-дослідницьких робіт і інших розробок.

Загальні вимоги до об'єкта проектування полягають у наступному. Об'єктом проектування може бути технічна система, що складається з різнотипних модулів, з'єднаних між собою певним чином і виконують спільну функцію.

1.2. Завдання на курсову роботу

Відповідно до теми курсової роботи студент отримує вхідні дані.

Вимоги до проектування, що оформляються на спеціальному бланку (Додаток Б) у вигляді завдання, що містить назву теми, вихідні дані, обсяг роботи, термін представлення проекту і т.д.

На підставі вихідних даних до роботи студент розробляє технічне завдання (ТЗ) на проектування програмного засобу. Обсяг і зміст курсової в кожному конкретному випадку визначається темою роботи і завданням на проектування.

Студенту, що не з'явився за одержанням завдання у встановлений термін, завдання видається керівником тільки з дозволу деканату. Курсова робота

повинна бути виконана не пізніше ніж за два тижні до початку екзаменаційної сесії.

Керівник курсової роботи видає завдання, допомагає студенту скласти календарний графік його виконання і проводити систематичні консультації. У вступній лекції роз'яснює значення проектування для даної дисципліни, вимоги, пропоновані до роботи, креслення до пояснювальної записки, а так само сутність виданих завдань, вузлові питання найбільш типового завдання і дає загальні вказівки по виконанню проекту. Рекомендує студентам основну літературу і довідкові матеріали по темі, призначає в календарному графіку термін виконання окремих розділів курсової роботи і періодичних звітів про хід виконання роботи.

Контроль керівника ні в якій мірі не звільняє студента від відповідальності за правильність виконання роботи і прийняття рішень. Основною задачею керівника курсової роботи, крім технічного керівництва, є надання навичок самостійної і систематичної роботи студента з дотриманням установленого плану. Обов'язок консультанта – застерегти студента від грубих помилок.

У виборі тих чи інших рішень ініціатива надається студенту. При цьому керівник може рекомендувати відповідну літературу, журнальні статті і т.п., переслідуючи основну мету – поглиблене самостійне вивчення студентом даного питання. Тільки коли студент розглянув кілька варіантів, керівник може підказати й інші рішення

Студент цілком відповідає за прийняті науково технічні рішення, правильність виконання розрахунків, якість виконання й оформлення курсової роботи, а так само за своєчасне завершення роботи над курсовою роботою.

1.3. Порядок роботи над курсовим проектом

Курсове проектування починається з отримання від керівника індивідуального завдання. У тижневий термін студент повинен розробити і затвердити в керівника календарний графік виконання проекту. У процесі проектування він консультиється з керівником у міру потреби й у зв'язку з

виникаючими питаннями. Незалежно від цього студент зобов'язаний регулярно (не рідше раз у тиждень) інформувати керівника про хід виконання проекту.

Своєчасне одержання завдання, складання календарного графіка, систематичні консультації і контроль виконання роботи є необхідними умовами своєчасного і якісного виконання курсової роботи.

Виконання курсової роботи вимагає уміння організувати свою працю, правильно розподілити час і сили на окремі етапи роботи. Робота над курсовою повинна проводитися щодня. При виконанні кожного розділу роботи необхідно в чорнових записах фіксувати свої розуміння, ідеї, розрахунки.

Курсова робота повинна виконуватися в строгій відповідності із завданням. Відступ від завдання дозволяється тільки за узгодженням з керівником. За прийняте остаточне рішення і правильність усіх проведених розрахунків відповідає студент як автор проекту.

Виконання курсової роботи необхідно починати з ознайомлення з технічним завданням, опрацювання його з урахуванням області застосування розроблювального програмного засобу, місця його інсталяції, реальних умов експлуатації й основних технічних характеристик. Результатом виконання цього етапу роботи є складання календарного графіка. Потім необхідно приступити до вивчення літературних джерел, аналізу відомих технічних рішень по аналогічних засобах. При цьому варто критично підходити до вивчення джерел – рекомендується відображати найбільш актуальні, нові дані і використовувати фахові джерела.

Усі роботи з проекту повинні бути закінчені не пізніше чим за п'ять днів до захисту і за два тижні до початку екзаменаційної сесії. Закінчується проект оформленням пояснювальної записки і графічного матеріалу.

Готову роботу студент передає керівнику для остаточної перевірки. Керівник перевіряє роботу і дає рецензію про виправлення чи доповнення (додаток Г), що студенту варто розглянути і внести в роботу, після чого підписує пояснювальну записку і креслення.

На цьому курсова робота вважається закінченою і може бути подана до захисту.

1.4. Загальні вимоги до обсягу курсової роботи (КР)

Курсова робота містить текстову і графічну документацію.

Текстова документація включає наступні документи: при обсязі (в аркушах):

- титульний лист (додаток А) – 1;
- рецензія (додаток В) – 1;
- завдання на проектування (додаток Б) – 1(двостороння);
- технічне завдання (ТЗ) – 2;
- пояснювальну записку – 25 – 30.

Отже, обсяг текстової документації повинний складати 30 – 35 сторінок рукописного тексту формату А4.

Графічна частина (форматів А4 або А3) включає наступні документи:

- структурну або функціональну схеми – 1 лист,
- схему об'єкта проектування – 1 лист,
- модель розробки та результати досліджень – 1 лист.

1.5. Технічне завдання

ТЗ розробляється на основі вимог ДСТУ ISO/IEC 14598-1 «Інформаційні технології. Оцінювання програмного продукту. Ч.1. Загальний огляд». Порядок виконання, складання й оформлення ТЗ приведені в доповненні до ГОСТ 15.001-73.

Студент за домовленістю з керівником обмежується розробкою тільки окремо узятих розділів ТЗ. ТЗ розробляють на основі аналізу існуючої проблеми і її складової частини – технічної задачі, на розв'язок якої орієнтується курсова робота. Обов'язковими розділами ТЗ повинні бути:

1. Назва й область застосування програмної системи.
2. Підстава для проведення розробки.
3. Мета і призначення розробки.
4. Джерело розробки.

5. Технічні вимоги і норми, що визначають основні показники якості й експлуатаційні показники системи з урахуванням діючих стандартів і норм:

- склад програмного засобу і вимоги до його конструкції;
- назва, кількість і призначення складових частин;
- конструктивні вимоги до виробу і складових частин;
- вимоги до стійкості і зовнішніх впливів;
- основні технічні параметри, що визначають мету і застосування, припустимі відхилення від заданих параметрів;
- вимоги до надійності, безвідмовності;
- вимоги до технологічності й уніфікації;
- естетичні й ергономічні вимоги;
- вимоги до даних, апаратних платформ, обмеження по застосуванню;
- вимоги до маркування упакування, транспортування і зберігання.

6. Стадії й етапи розробки.

7. Порядок контролю. Таким чином, розробка ТЗ – важливий етап у виконанні курсової роботи. Він чітко установлює вимоги до розробки, що відповідають дійсності. Термін виконання етапів курсової роботи встановлюється керівником під час видачі завдання. Графік роботи над курсовою роботою розробляється і видається одночасно з завданням на курсове проектування.

1.6. Вимоги до змісту й оформлення курсових робіт та проектів

При написанні окремих текстових документів необхідно керуватися приведеними далі положеннями.

1.6.1. Титульний лист. Оформляється відповідно до вимог ЄСКД. Зразок титульного листа приведений у додатку А.

1.6.2. Завдання на курсове проектування (див. додаток Б).

1.6.3. Реферат. Реферат повинен бути оформлений відповідно до ГОСТ 3008-95.

1.6.4. Технічне завдання. На підставі завдання на курсове проектування, виданого керівником, студент розробляє технічне завдання (ТЗ). ТЗ є вихідним документом для розробки проектного пристрою і відповідної технічної документації. ТЗ розробляється на основі вимог ДСТУ ISO/IEC 14598-1 «Інформаційні технології. Оцінювання програмного продукту. Ч.1. Загальний огляд».

Технічне завдання в загальному випадку складається з наступних розділів:

- найменування й область застосування (використання);
- підстава для розробки;
- мета і призначення розробки;
- джерело розробки;
- технічні вимоги;
- економічні показники;
- стадії й етапи розробки;
- порядок контролю і приймання;
- додаток.

1.6.5. Пояснювальна записка (ПЗ).

ПЗ до курсової роботи в загальному випадку повинна включати наступні основні розділи:

- перелік скорочень;
- вступ;
- огляд існуючих рішень і обґрунтування вибору прийнятого принципу проектного програмного засобу;
- розробка й опис структурної і функціональної схем проектного засобу;
- розробка й опис схеми об'єкта проектування; обґрунтування вибору й опис конструкції проектного програмного засобу; вибір і обґрунтування окремих модулів, блоків і елементів;
- розробка моделі проектного складного програмного об'єкту та представлення результатів моделювання;
- висновок;

- список використаних джерел;

- додатки.

У залежності від змісту пояснювальної записки за узгодженням з керівником допускається об'єднання чи виключення деяких з перерахованих розділів.

2. ЗАВДАННЯ НА КУРСОВУ РОБОТУ

Завдання на курсову роботу видається кожному студенту викладачем згідно підпунктів 2.1. – 2.4. Студентам, що працюють над науково-дослідними та лабораторними розробками, видаються індивідуальні завдання. Метою даної роботи є набуття навиків попереднього (ескізного) та остаточного розрахунків технічних параметрів програмного продукту.

2.1. Розробка складної системи мовою блокового моделювання

Використовують графічну мову ієрархічних блок-схем. Блок вищого рівня ієрархії збирається з деякого набору стандартних блоків (створених раніше розроблювачами пакета, або написаних самим користувачем), що з'єднуються односпрямованими функціональними зв'язками. Зібрану функціональну схему можна використовувати як блок на наступному рівні ієрархії і можна запам'ятати в бібліотеці блоків.

До переваг цього підходу варто віднести простоту створення складних моделей для початківців. Проте доводиться будувати досить громіздкі багаторівневі блок-схеми, що не відбивають природної структури системи, яка моделюється. Це ускладнює процес моделювання.

Найбільш відомими представниками мов блокового моделювання є:

підсистема Simulink пакета MATLAB [8-12];

пакет Wolfram Mathematica [13-16];

пакет MathCAD;

пакет EASY5 [5];

підсистема SystemBuild пакета MATRIXx [6];

VisSim [16].

2.2. Розробка складної системи мовою фізичного моделювання

При створенні моделі використовують неорієнтовані і потокові зв'язки. Користувач може сам визначати нові класи блоків. Аналогова складова

елементарного блоку задається системою алгебро-диференціальних рівнянь і формул. Дискретна складова задається описом дискретних подій, при виникненні яких можуть виконуватися миттєві присвоювання змінним нових значень. Дискретні події можуть поширюватися спеціальними каналами. Зміна структури рівнянь можлива тільки побічно через коефіцієнти в правих частинах, це обумовлено необхідністю символічних перетворень при переході до еквівалентної системи. Підхід дуже зручний і природний для опису типових блоків фізичних систем. Недоліками є необхідність символічних перетворень, що різко звужує можливості опису гібридного поведіння, а також необхідність чисельного розв'язку великого числа алгебраїчних рівнянь, що значно ускладнює задачу автоматичного отримання достовірного рішення.

Під час практичного моделювання можна скористатися програмними пакетами:

Wofram Mathematica;

MATLAB;

Maple;

MathCAD;

Dymola [8];

Omola і OmSim [9];

OmSim – діалогове середовище інтегрованих інструментальних засобів, що підтримує розробку моделі і моделювання. OmSim заснований на базі об'єктно-орієнтованої мови моделювання Omola.

Smile [10];

Modelica [11].

2.3. Розробка складної системи на схемах гібридного автомата

Третя група містить у собі пакети, засновані на використанні схеми гібридного автомата. Використання карти станів при описі переключень станів, а також безпосередній опис неперервної поведінки системи алгебро-диференціальних рівнянь надає великі можливості в описі гібридного

поводження зі складною логікою переключень.

До недоліків варто також віднести надмірність опису при моделюванні безупинних систем.

До цієї групи відносяться:

- пакет Shift [12];

Shift – мова програмування для опису динамічних мереж гібридних автоматів.

- пакет Model Vision Studium [13].

2.4. Опис системи MatLab

Сучасна програмна інженерія пропонує цілий набір інтегрованих програмних систем і пакетів програм для автоматизації моделювання: Eureka, Gauss, TK Solver!, Derive, Mathcad, Mathematica, Maple V і ін. Одна з найбільш популярних систем серед інженерів це MATLAB.

MATLAB – одна з найстарших, ретельно пророблених й перевічених часом систем автоматизації математичних розрахунків, яка побудована на розширеному представленні й застосуванні матричних операцій. Це знайшло відображення в назві системи – MATrix LABoratory – матрична лабораторія. Однак синтаксис мови програмування системи продуманий настільки ретельно, що ця орієнтація майже не відчувається тими користувачами, яких не цікавлять безпосередньо матричні обчислення.

Матриці широко застосовуються в складних математичних розрахунках, наприклад, при рішенні задач лінійної алгебри і математичного моделювання статичних і динамічних систем і об'єктів. Вони є основою автоматичного складання й рішення рівнянь стану динамічних об'єктів і систем. Прикладом може служити розширення MATLAB – Simulink. Це істотно підвищує інтерес до системи MATLAB, що увібрала в себе кращі досягнення в області швидкого рішення матричних задач.

Однак сьогодні MATLAB далеко вийшла за межі спеціалізованої матричної системи і стала однією з найбільш могутніх універсальних

інтегрованих обчислювальних систем. Слово «інтегрована» вказує на те, що в цій системі об'єднані зручна оболонка, редактор виразів і текстових коментарів, обчислювач і графічний програмний процесор. У новій версії використовуються типи даних, як багатомірні масиви, масиви осередків, масиви структур, масиви Java і розріджені матриці, що відкриває можливості застосування системи при створенні й налагодженні нових алгоритмів матричних і заснованих на них рівнобіжних обчислень і великих баз даних.

У цілому MATLAB – це унікальна колекція реалізацій сучасних чисельних методів комп'ютерної математики, створених за останні три десятиліття років. Вона увібрала в себе і досвід, правила і методи математичних обчислень, які були накопичені за тисячі років розвитку математики. Це сполучається з могутніми засобами графічної візуалізації і анімаційної графіки. Систему з прикладеною до неї великою документацією цілком можна розглядати як фундаментальний багатотомний електронний довідник по математичному забезпеченню ЕОМ – від масових персональних комп'ютерів до супер-ЕОМ. Можливості MATLAB дуже великі, а по швидкості виконання задач система нерідко перевершує своїх конкурентів. Вона застосовна для розрахунків практично в будь-якій області науки й техніки. Наприклад, дуже широко використовується при математичному моделюванні механічних пристроїв і систем, зокрема, в динаміці, гідродинаміці, аеродинаміці, акустиці, енергетиці і т.д. Цьому сприяє не тільки розширений набір матричних і інших операцій і функцій, але і наявність пакета розширення (toolbox) Simulink, спеціально призначеного для рішення задач блокового моделювання динамічних систем і пристроїв, а також десятків інших пакетів розширень.

У великому й постійно поповнюваному комплексі команд, функцій і прикладних програм системи MATLAB містяться спеціальні засоби для електротехнічних і радіотехнічних розрахунків (операції з комплексними числами, матрицями, векторами й поліномами, обробка даних, аналіз сигналів і цифрова фільтрація), обробки зображень, реалізації нейронних мереж, а також засоби, що відносяться до інших нових напрямків науки й техніки. Вони

ілюструються безліччю практично корисних прикладів.

До розробок розширень для системи MATLAB притягнуто багато наукових шкіл світу, якими керують великі вчені й педагоги університетів. Важливими достоїнствами системи є її відкритість і розширюваність. Більшість команд і функцій системи реалізовані у виді текстових m-файлів (з розширенням .m) і файлів мовою C, крім того усі файли доступні для модифікації. Користувачеві надана можливість створювати не тільки окремі файли, але і бібліотеки файлів для реалізації специфічних задач.

Разюча легкість модифікації системи і можливість її адаптації до рішення специфічних задач науки і техніки привели до створення десятків пакетів прикладних програм (toolbox), які набагато розширили сфери застосування системи.

Однією з проблем сучасної науки є розробка і впровадження в практику методів дослідження функціонування складних систем. До класу складних систем відносять технологічні, виробничі, енергетичні комплекси, системи автоматизації керування й інших об'єктів. Моделювання - один із найбільш розповсюджених засобів вивчення різних процесів і явищ. Моделлю вихідного об'єкта називається представлення об'єкта в деякій формі, відмінної від форми його реального існування [1]. В інженерній практиці модель, звичайно, створюється для:

- проведення на моделі експериментів, які неможливо або складно провести на реальному об'єкті (що надає можливість одержання нових знань про об'єкт);
- прискорення, здешевлення, спрощення і будь-якого іншого удосконалення процесу проектування, що досягається за рахунок роботи з більш простим об'єктом, чим вихідний, тобто з моделлю.

Сьогодні відомі і широко використовуються в наукових дослідженнях і інженерній практиці різні типи моделей і численні методи моделювання. Якщо взяти за основу ступінь абстрактності (ступінь відмінності від реального об'єкта), то можна визначити наступні типи моделей:

- фізичні (натурні) моделі (відтворюють досліджуваний процес із збереженням його фізичної природи і є інструментом фізичного моделювання);
- аналогові моделі (заміняють один об'єкт на іншій зі схожими властивостями);
- математичні моделі (абстрактні моделі, існують у формі спеціальних математичних конструкцій).

Під математичним моделюванням розуміють засіб дослідження різних процесів шляхом вивчення явищ, що мають різний фізичний зміст, але описуваних однаковими математичними співвідношеннями [2].

Серед великого числа пакетів візуального моделювання пакет Matlab займає особливе місце. Спочатку орієнтований на дослідницькі проекти, пакет в останні роки став робочим інструментом інженерів, студентів, керівників, фізиків, зв'язківців. Однією з основних причин широкого використання пакета Matlab є великий спектр засобів, що надає користувачеві для рішення різноманітних задач у різних областях людської діяльності. Серед цих засобів особливе місце займає підсистема Simulink.

Simulink – це інтерактивне середовище для моделювання й аналізу широкого класу динамічних систем за допомогою блок-діаграм.

Основні властивості підсистеми Simulink:

- містить у собі велику бібліотеку блоків (безупинні елементи, дискретні елементи, математичні функції, нелінійні елементи, джерела сигналів, засоби відображення, додаткові блоки), які можна використовувати для графічного збирання систем;
- надає можливість моделювання лінійних, нелінійних, безупинних, дискретних і гібридних систем;
- блок-діаграми можуть бути об'єднані в складені блоки, що дозволяє використовувати ієрархічне представлення структури моделі, тим самим забезпечуючи спрощений погляд на компоненти і підсистеми;
- містить засоби для створення користувальницьких блоків і бібліотек

блоків;

- підтримує підсистеми, що працюють за умовами, тригерів.

Simulink забезпечують інтерактивне середовище для моделювання, при цьому поведіння моделі і результати її функціонування відображаються в процесі роботи, і існує можливість змінювати параметри моделі у той момент, коли вона виконується. Simulink дозволяє створювати власні блоки і бібліотеки блоків з доступом із програм на Matlab, Fortran чи C, зв'язувати блоки з розробленими раніше програмами на Fortran і C, що містять вже перевірені моделі.

Починаючи з версії 3.0 у Simulink з'явилися спеціалізовані додатки, що значно збільшили міць даного середовища моделювання:

Stateflow – графічний інструментарій для проектування складних систем керування. Stateflow дає можливість моделювати поведінку складних подійно-керувальних систем, базуючись на теорії кінцевих автоматів. Це дозволяє користувачам Simulink додавати подійно-керувальне поведінку до їхніх моделей.

Stateflow Coder – генерація C коду для діаграм Stateflow . Використовуючи Stateflow і Stateflow Coder, користувач може генерувати код винятково для Stateflow-частин моделі Simulink.

Real-Time Workshop – доповнює Simulink і Stateflow Coder, забезпечуючи автоматичну генерацію коду C з моделей Simulink. За допомогою Real-Time Workshop можна легко генерувати код для дискретних, безупинних і гібридних систем, включаючи системи, що містять підсистеми працюючі при виконанні визначених умов.

DSP Blockset – бібліотеки блоків Simulink для створення, моделювання і макетування цифрових систем обробки сигналів.

Nonlinear Control Design Blockset – інтерактивний підхід до автоматизованого проектування систем керування.

Fixed-Point Blockset – бібліотеки блоків Simulink для моделювання поведінки систем керування і динамічних фільтрів з фіксованою крапкою.

Simulink Report Generator – дозволяє створювати і будувати звіти з моделей Simulink і Stateflow у різних форматах, серед яких HTML, RTF, XML і SGML.

2.5. Опис системи Wolfram Mathematica

Система Mathematica завжди була лідером серед систем комп'ютерної алгебри. Версії Mathematica 9.0, 10.0, як і 5.0, 6.0 десять років тому, сьогодні є найбільш відомими і використовуваними серед користувачів системами цього класу, є спрямованими на збереження цього лідерства та розширення сфери застосування програмних продуктів даного типу.

Сучасні версії Mathematica 9.0, 10.0 – це вже не просто лідер. Це дійсно революційний програмний продукт з величезним числом новацій, які висувають систему в особливе становище, і що надає їй можливості, які, мають значні переваги порівняно з іншими системами. Це стосується новацій інтерфейсу документів системи, високій швидкості обчислень і величезного набору корисних функцій, а також, що дуже корисно, інтерфейсу користувача.

Величезне число вбудованих засобів обчислень дозволяє розв'язувати переважну більшість завдань з побудови складних програмних систем на мові Mathematica, сильно наближеної до звичайної мови математичних обчислень та мови C++ – базової мови сучасного програміста. Тому користувач зможе програмувати в середовищі Mathematica, фактично, починаючи з етапів побудови моделі розв'язку поставленої задачі. Крім того, є всі підстави уважати, що Mathematica дійсно виконує обчислення швидше своїх конкурентів, зокрема, завдяки вперше реалізованій підтримці можливостей сучасних надшвидкісних мікропроцесорів – в тому числі багатоядерних та за рахунок багатопотокових обчислень.

Інсталяція системи Mathematica нічим не відрізняється від інсталяції будь-якої програми в операційній системі Windows. Після інсталяції на робочому столі з'являється ярлик, активізація якого призводить до завантаження системи в пам'ять комп'ютера та появи вікон системи (рис. 1).

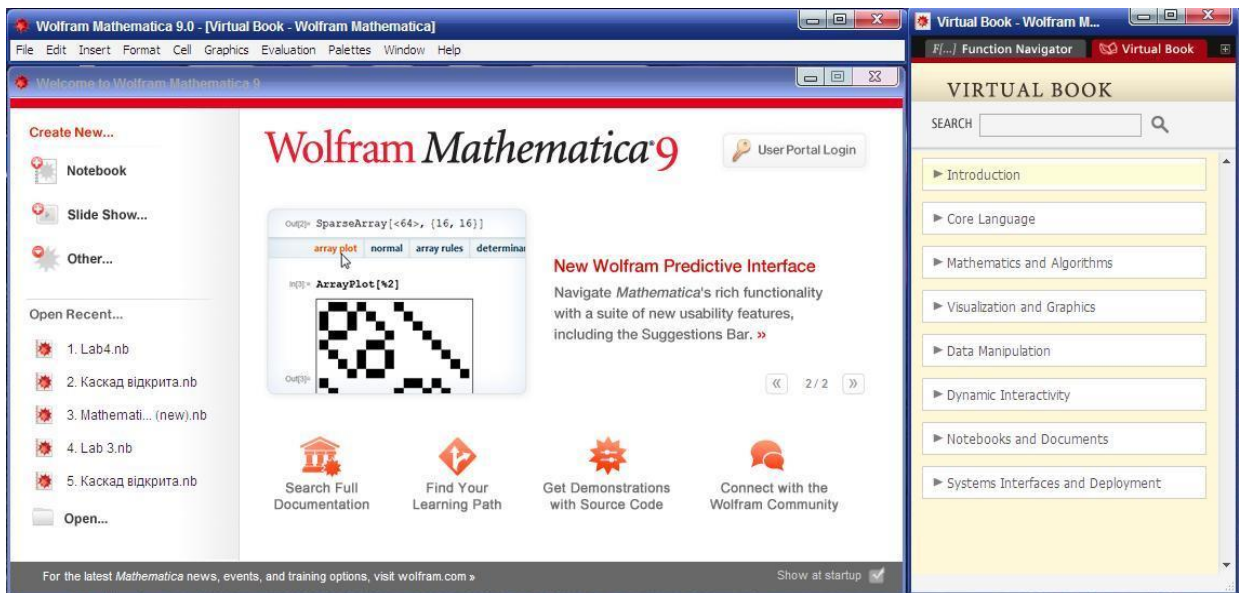


Рис. 1. Огляд початкового інтерфейсу системи Mathematica 9

Спочатку після запуску Mathematica з'являються панель меню і велике вікно робочого ноутбука. На рис. 1 показано також вікно з віртуальною книгою по роботі з системою Mathematica та навігатор вбудованих функцій ядра (Kernel). Усі вікна закриваються активізацією кнопки зі знаком «х».

Панель головного меню має всього два рядки:

- з назвами системи і завантаженого файлу;
- позиціями головного меню.

Праворуч і знизу вікна редагування містяться лінійки прокрутки з характерними повзунками, що можуть бути керованими мишею. У самому низу на початку лінійки прокрутки є так звана статусний рядок з інформацією про поточний режим роботи (Status bar).

Головне меню системи (рис. 1) містить такі вкладки:

- File – робота з файлами: створення нового файлу, вибір існуючого файлу з каталогу, закриття файлу, збереження поточного файлу, збереження файлу зі зміною імені, друк документа і вихід в Windows;
- Edit – виконання основних операцій редагування (відміна операції, копіювання виділених частин документа в буфер з їх подальшим видаленням і без нього, перенесення виділених частин, їх стирання);

- Insert – завдання елементів введення (графіків, матриць, гіперпосилань, додавання у робочий ноутбук елементів файлів, вибір кольору робочої комірки та нумерація комірок);
- Format – установка формату для документів;
- Cell – робота з функціональними комірками (об'єднання і роз'єднання комірок, установка статусу комірки, відкриття і закриття);
- Evaluation – управління ядром системи та його конфігурацією;
- Palettes – робота з палітрами математичних операторів та функцій, засоби введення математичних символів та їх опції;
- Window – операції з вікнами і їх розташуванням;
- Help – управління засобами довідкової системи.

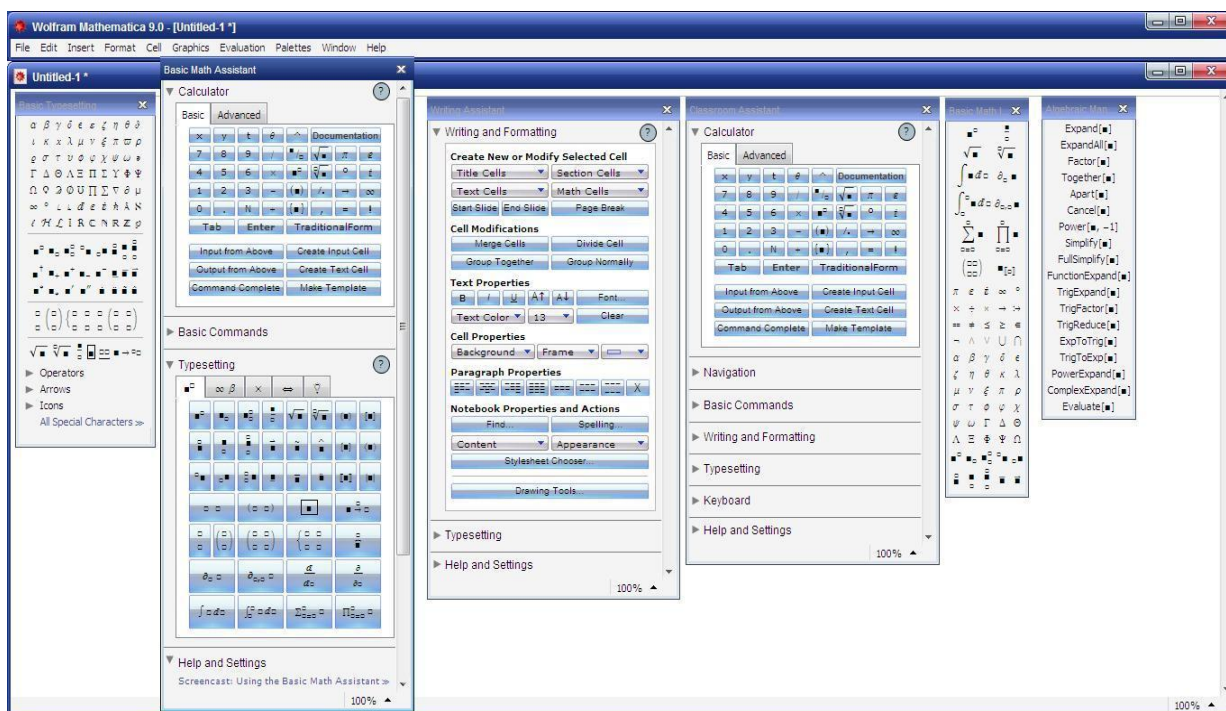


Рис. 2. Вікно системи Mathematica 9 з відкритими палітрами та засобами введення алгебраїчних виразів і дій з ними

Основні властивості пакету Wolfram Mathematica:

- кількість спеціальних математичних знаків (грецьких і латинських букв, операторів, функцій і команд), що вводяться за допомогою палітр, більше 700. Багато операцій та символів мають альтернативні варіанти їх введення з застосуванням комбінацій клавіш – їх можна знайти в

- довідковій базі даних системи;
- надає можливість динамічного моделювання (функція *Manipulate*) довільних складних програмних систем та наочності кожного кроку блочного програмування;
 - наявність засобів управління, редагуванням графікою;
 - можливість створення елементів та засобів функцій та процедур користувача. Система має ряд пакетів розширення (в оригіналі доповнення – *AddOn*) системи, розташованих в каталозі *ADDON*;
 - можливість багато потокового здійснення обчислень на ЕОМ малих потужностей з використанням вбудованого ядра системи, орієнтованого на широкий спектр обчислень.

Робота з документами зводиться до набору в комірках виразів (наприклад, математичних) і їх виконання. Для виконання виразів досить натиснути комбінацію клавіш **Shift** + **Enter** одночасно (оскільки клавіша **Enter** використовується тільки для падання перекладу рядка всередині поточного рядка введення). Наприклад, щоб обчислити суму $5 + 6$, необхідно спочатку ввести цей вислів в рядок введення. У *Mathematica* рядок введення формується по мірі введення об'єктів для виразів 5 і 6 . Після натискання клавіш **Shift** і **Enter** отримаємо:

```
In[1]: = 5 + 6
```

```
Out [1] = 11.
```

Слід зауважити, що при першому обчисленні *Mathematica* виконує його з відчутною затримкою. Це пов'язано із завантаженням ядра системи (*Wolfram Mathematica Kernel*). Подальші обчислення відбуваються практично миттєво.

Документ системи *Mathematica* своїми рядками (комірками) введення і виведення, текстовими коментарями, малюнками дуже нагадує сторінку блокнота вченого або інженера (рис. 5.1). Саме тому він і називається ноутбуком (від англійського *notebook*). Ноутбуки зберігаються з розширенням ***назва файлу*.nb**.

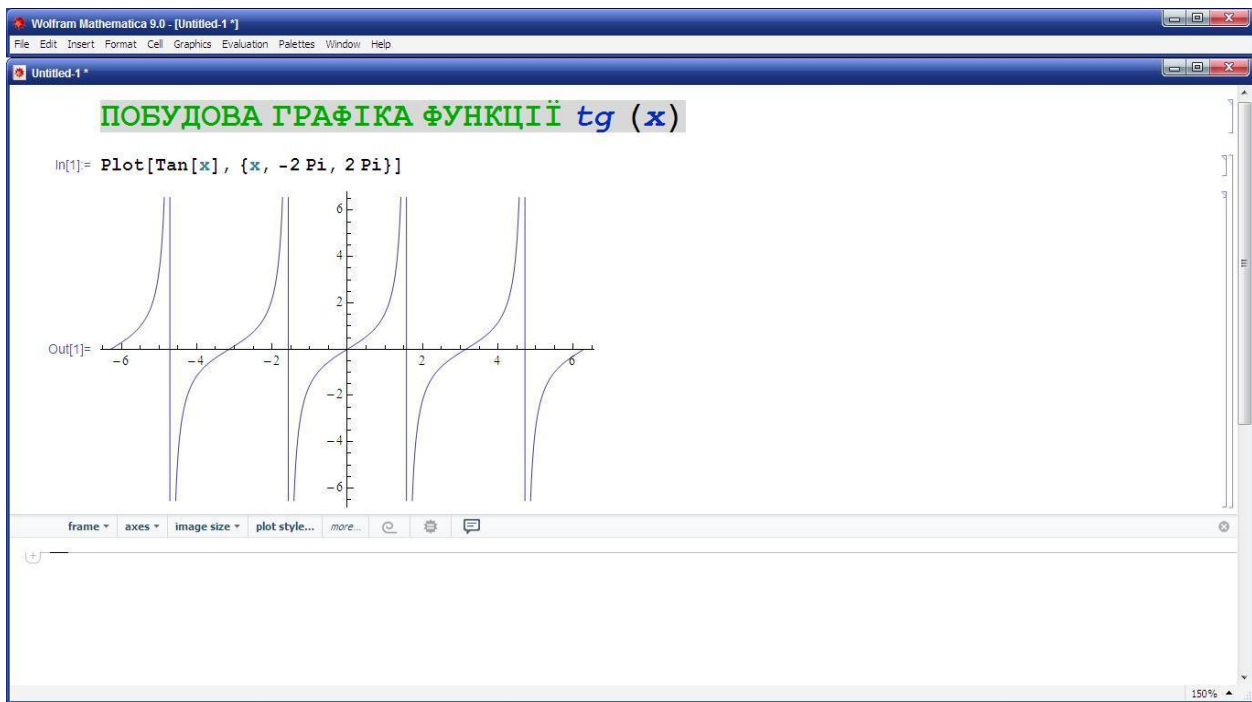


Рис. 3. Вікно системи Mathematica 9 відкритим ноутбуком та прикладом побудови графіка

Окремі комірки з математичними виразами і результатами їх обчислень позначаються в правій частині головного вікна редагування характерними тонкими квадратними дужками синього кольору. Це дозволяє відстежувати те, до чого відносяться математичні вирази – до вихідних даних або отриманих результатів. Крім того, комірки обчислень можуть мати різний статус, який позначається відповідними значками над квадратними дужками.

Для того, щоб документ мав наочний вид блокнота (ноутбука), необхідно зробити певні операції з форматування документа і надання йому потрібного вигляду. Перш за все, кожен крок обчислень слід супроводжувати пояснювальними написами. Вони можуть міститися в рядках введення, але потім їх необхідно відформатувати згідно текстового формату відповідного стилю. Для цього виділяється рядок введення з текстовим написом. Простір усередині дужки при затемнюється. Потім виконується команда **Format-Style-Text** (Alt + 7). Вона задає текстовий формат напису, який є не виконуваним.

За допомогою інших команд опції **Format** головного меню, можна задати напис різним шрифтом, різним кольором з виділенням фону. Як уже

зазначалося, для введення математичного виразу згідно шаблону і для представлення його в природній математичній формі використовується стандартний формат StandardForm комірок введення.

У робочих блокнотах бажано, щоб форма подання математичних виразів хоча б частково нагадувала загальноприйняту. Mathematica дозволяє задавати форми подання документів, прийняті в таких потужних мовах програмування, як Fortran, C і навіть TeX (мова для програмування засобів типографічного набору складних наукових текстів, що активно використовується в наукових журналах).

Кожен напис, математичний вираз або графік займають окрему комірку – Cell. Комірка може займати один або кілька рядків і завжди є виділеним своєю квадратною дужкою. Важливою властивістю комірок усіх систем Mathematica є можливість їх еволюції по всьому робочому документу. Таким способом здійснюється динамічний обмін даними в ході символічних перетворень.

3. ЗАГАЛЬНІ ВИМОГИ ДО ОФОРМЛЕННЯ ТЕКСТОВИХ ДОКУМЕНТІВ

Оформлення пояснювальної записки (ПЗ) повинні відповідати вимогам ГОСТ 3008-95.

Текст розрахунково-пояснювальної записки може бути написаний як від руки, так і надрукований на через півтора інтервали на одній сторінці стандартного листа білого паперу формату А4 (297 x 210мм.) із полями по всіх чотирьох сторонах листа.

Розмір лівого поля – 25мм., правого – не менше 10мм., верхнього і нижнього полів не менше 20мм.

Текст ПЗ пишеться розбірливо без помарок і виправлень. Помилки, описки і графічні неточності необхідно акуратно виправити, але не більше трьох виправлень на сторінку. Ушкодження аркушів, помарки не допускаються. Виклад змісту проекту в ПЗ повинен бути коротким, чітким, що виключає можливість суб'єктивного тлумачення, і вестися від першої особи множини, наприклад: «приймаємо», «вибираємо» і т.д. Такі вираження як «я беру», «вважаю», «мені задане» і т.ін. уживати не слід.

Мова викладу повинна бути технічно грамотною. Варто скористатися єдиною і відповідно встановленою стандартом термінологією і визначеннями, а при їхній відсутності – загальноприйнятими в науково-технічній літературі. Скорочення слів у тексті і підписах під ілюстраціями, як правило, не допускається.

Літерним позначенням різних величин при першому їхньому використанні варто давати тлумачення. Значення символів і числових коефіцієнтів, що входять у формулу, необхідно приводити безпосередньо під формулою. Розмірність того самого параметра в межах пояснювальної записки повинна бути постійною.

У записі обчислень по формулах приводиться тільки вихідна формула, вирази з підставленими цифрами й остаточний результат. Проміжні викладення виключаються.

Весь текст пояснювальної записки розбивають на розділи і підрозділи. Розділи нумеруються арабськими цифрами в межах усієї пояснювальної записки. Підрозділи варто нумерувати арабськими цифрами в межах кожного розділу. Номер підрозділу повинен складатися з номерів розділу і підрозділу, розділених крапкою.

Розділи і підрозділи пояснювальної записки забезпечуються змістовними заголовками (назвами).

Назви розділів пишуться прописними буквами, назви підрозділів – рядковими (крім першої прописної). Якщо назва складається з двох чи більш речень, їх розділяють крапкою. Наприкінці назви крапку не ставлять. Підкреслювати назви і переносити слова в них не допускається.

Кожен розділ пояснювальної записки рекомендується починати з нового листа. Кожен пункт тексту записується з абзацу.

Нумерація сторінок пояснювальної записки повинна бути наскрізною. Нумерацію сторінок проставляють арабськими цифрами в правому нижньому кутку сторінки. На титульному листі число 1 не ставиться.

Текст пояснювальної записки ілюструється схемами, характеристиками програмних модулів, графіками, таблицями. Кількість ілюстрацій повинна бути не більш 7-10 і визначатися змістом курсової роботи (проекту).

Всі ілюстрації іменуються рисунками, що нумеруються послідовно в межах розділу арабськими цифрами.

Кожен рисунок повинен мати змістовий підпис. Підписи під рисунками та над таблицями повинні бути короткими і пояснювати основний зміст.

На всі ілюстрації повинна бути посилання в тексті. Посилання на раніше згадані ілюстрації і таблиці дають зі слів, наприклад: «див. рисунок 1», «див. таблицю 1».

Графіки необхідно супроводжувати координатною сіткою, що відповідає масштабності шкали (рівномірної чи логарифмічної) по осях абсцис і ординат. При цьому осі координат кресляться суцільними лініями товщиною приблизно в два рази більше, ніж товщина ліній координатної сітки, без стрілок на кінцях. Числові значення масштабу шкал осей координат пишуть лівіше осі ординат і нижче осі абсцис. Багатозначні числові значення по осі координат рекомендується по можливості приводити у вигляді добутку цілих чисел на деякий постійний множник, що вказується при літерному позначенні величини, що відкладається, чи зводиться в розмірність цієї величини.

Цифровий матеріал, покладений у пояснювальну записку, рекомендується оформляти у вигляді таблиць.

Таблицю варто поміщати після першого згадування про неї в тексті, так, щоб її можна було читати без повороту записки чи з поворотом по годинниковій стрілці. При переносі таблиці на наступну сторінку шапку таблиці повторюють і над нею поміщають слова «Продовження таблиці» із вказуванням номера.

Таблиці повинні нумеруватися в межах розділу арабськими цифрами. Номер таблиці складається з номера розділу і порядкового номера таблиці, розділених крапкою, наприклад:

«Таблиця 2.4» – четверта таблиця другого розділу. При посиланні на таблицю вказують її повний номер, наприклад, «Таблиця 2.4».

Формули нумерують у межах розділу арабськими цифрами. Номер формули складається з номера розділу і порядкового номера формули, розділених крапкою, записується в круглій дужки (наприклад: (3.4) – четверта формула третього розділу) і міститься в правому краї рядка тексту на рівні рядка формули, до якої він відноситься.

При посиланні в тексті на формулу необхідно вказувати її повний номер, наприклад: «у формулі (3.4)».

Розмірність того самого параметра в межах записки повинна бути постійною.

У список використаних джерел включають усі використані джерела в порядку появи посилань на них у тексті пояснювальної записки або в алфавітному порядку.

При посиланні на джерела в тексті пояснювальної записки варто приводити його порядковий номер за списком використаної літератури у квадратних дужках.

Додатки оформляються як продовження пояснювальної записки. Кожен додаток починають з нової сторінки. У центрі пишуть слово «Додаток» та його позначення. Кожен додаток повинен мати змістовний заголовок. Два і більш додатки нумерують послідовно буквами українського алфавіту (наприклад: «Додаток А», «Додаток Б» і т.д.).

Рисунки, таблиці і формули, що містяться в додатку, нумеруються арабськими цифрами в межах кожного додатка (наприклад: «рис. В.2» – другий рисунок додатку В).

4. ЗАГАЛЬНІ ВИМОГИ ДО ГРАФІЧНОЇ ДОКУМЕНТАЦІЇ

4.1. Вимоги до змісту графічної документації

Зміст і обсяг графічної документації до курсової роботи (проекту) залежить від характеру і тематики виконуваної роботи.

На першому кресленні поміщають структурну схему програмного засобу, розроблену автором роботи.

Допускається структурну схему приводити в пояснювальній записці у вигляді рисунків. Друге креслення повинно відтворювати функційну схему проєктованого програмного засобу.

У тих випадках, коли студент пропонує до захисту виготовлений ним чи за його участі програмний засіб, обсяг роботи може бути, за узгодженням з керівником, скорочений.

4.2. Вимоги до оформлення графічної документації

Усі схеми і креслення курсового проєкту виконуються в повній відповідності з вимогами ДОСТУ, ГОСТ, ЄСКД.

Схеми і креслення виконуються, з використанням рекомендованого програмного забезпечення.

4.3. Вимоги до виконання схем виробів

Загальні вимоги, пред'явлені до виконання схем виробів, приведені в ГОСТ 2.701-76 (СТСЭВ 651-77) «Схеми, види, типи. Загальні вимоги до виконання». У залежності від основного призначення схеми поділяються на наступні типи: структурні, функціональні, схеми підключення, розташування, загальні. Студенти спеціальності «Програмна інженерія» часто виконують структурні, функціональні і загальні схеми.

Структурна схема – це схема, що визначає основні функціональні частини програми, їхнє призначення і взаємозв'язок.

Функціональні частини зображують, як правило, у вигляді прямокутників. Прямокутники модулів, що мають самостійну програму, виконують суцільною лінією, прямокутники функціональних груп – штрих пунктирною лінією. Окремі елементи структурної схеми допускається зображати в їх умовних графічних позначеннях. При зображенні елементів схеми у вигляді прямокутників найменування, позначення (номер) чи типи (шифри) елементів і пристроїв вписують усередину прямокутників.

У випадку позначення функціональних частин схеми номерами або цифрами останні повинні бути розшифровані на полі схеми в таблиці довільної форми.

Побудова структурної схеми, розташування функціональних частин і ліній зв'язку повинні давати уявлення про структуру виробу, взаємодії його складових частин, а також про проходження даних у напрямку зліва направо. На лініях мережі (виконуються товщиною 0,2 – 1 мм.) напрямок проходження даних позначають стрілками.

На структурній схемі можна поміщати написи, що пояснюють, математичні моделі, тимчасові діаграми, що дають уявлення про послідовність проходження й обробки даних. Такі пояснення вказують найчастіше зверху або праворуч від графічних позначень функціональних частин структурної схеми.

Функціональна схема роз'ясняє окремі процеси, що протікають у програмних модулях або у системі в цілому.

На функціональних схемах рекомендується вказувати технічні характеристики функціональних частин поруч з їх графічними позначеннями. Написи, що пояснюють, діаграми або таблиці, що визначають послідовність процесів у часі, розміщують над основним написом.

Допускається вказувати необхідні параметри в характерних точках, наприклад, значення міток, процедур, функцій, математичні залежності та ін.

4.4. Умовні графічні позначення, що застосовуються на схемах

Елементи на схемах зображують у вигляді умовних графічних позначень, що регламентуються у відповідних стандартах.

Умовне графічне позначення пропонують на схемах у розмірах, встановлених у відповідних стандартах.

Лінії зв'язку й умовні графічні позначення елементів рекомендується виконувати товщиною 0,2-1 мм. Відстань між рівнобіжними лініями зв'язку повинне бути не менше 5 мм.

Список використаних джерел

1. Гради Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд. / Буч Г. – М. : "Бином", 1999. – 560 с.
2. Иан Соммервилл. Инженерия программного обеспечения. 6-е изд. / Соммервилл И. – М.: "Вильямс", 2006. – 624 с.
3. Леоненков А. Самоучитель UML. Эффективный инструмент моделирования информационных систем. / Леоненков А. – ВHV-Санкт-Петербург, 2001. – 304 с.
4. Фаулер М. UML в кратком изложении. Применение стандартного языка объектного моделирования. / Фаулер М., Скотт К. – М.: Мир, 1999. – 192 с.
5. Орлов С.А. Технологии разработки программного обеспечения: Учебник. / Орлов С. – СПб.: "Питер", 2002. – 464 с.
6. Бесекерский В.А., Власов В.Ф., Гомзин В.Н. и др. Руководство по проектированию систем автоматического управления. – М.: Высш.шк., 1983.
7. Романычева Э.Т., Иванова А.К. и др. Разработка и оформление конструкторской документации РЭА. – М.: Радио и связь, 1984.
8. С.П. Иглин. Математические расчеты на базе Matlab. Издательство "ВHV-Санкт-Петербург" 2005г. 640 стр.
9. В.П. Дьяконов. MATLAB 6.0/6.1/6.5/6.5 + SP1 + Simulink 4/5. Обработка сигналов и изображений. М.: СОЛОН-Пресс, 2004. – 592 с.
10. В.П. Дьяконов. Matlab 6.5 SP1/7 + Simulink 5/6 в математике и моделировании. М.: СОЛОН-Пресс, 2005. – 576 с.
11. Оленев Н.Н., Печенкин Р.В., Чернецов А.М. Параллельное программирование в MATLAB и Simulink с приложениями к моделированию экономики. – М.: ВЦ РАН, 2015. – 123 с.
12. Дьяконов В.П. MATLAB и SIMULINK для радиоинженеров. – М.: «ДМК-Пресс», 2011. – 976 с. – ISBN 978-5-94074-492-4.
13. Глушко В.П., Глушко А.В. Курс уравнений математической физики с использованием пакета Mathematica. – СПб.: «Лань», 2010. – С. 320.
14. Дьяконов В.П. Mathematica 5/6/7. Полное руководство. – М.: «ДМК Пресс», 2009. – С. 624. – ISBN 978-5-94074-553-2.
15. Дьяконов В.П. Mathematica 5.1/5.2/6. Программирование и математические вычисления. – М.: «ДМК-Пресс», 2008. – С. 576.
16. Акименко А.М. Періодичні розв'язки сингулярно збуреної виродженої системи диференціальних рівнянь [Текст] : автореф. дис... канд. фіз.-мат. наук: 01.01.02 / Акименко Андрій Миколайович ; Київський національний ун-т ім. Тараса Шевченка. – К., 2008. – 14 с.

Додаток А

Міністерство освіти і науки України
Тернопільський національний технічний університет ім. Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра програмної інженерії

КУРСОВА РОБОТА

з дисципліни

«Методологія та технологія створення складних програмних систем»

на тему: _____

Дата готовності роботи _____

Студента (ки) гр. СП- 4 _____ (_____)
П.І.П підпис
Керівник роботи Бойко Ігор Володимирович (_____)
підпис

Захищена “ _____ ” _____ 2017 р. з оцінкою _____

Члени комісії: _____ (_____)
прізвище, ім'я, по-батькові підпис
_____ (_____)
прізвище, ім'я, по-батькові підпис
_____ (_____)
прізвище, ім'я, по-батькові підпис

Тернопіль, 2017

Додаток Б

Тернопільський національний технічний університет ім. Івана Пулюя

Факультет інформаційно-комп'ютерних систем і програмної інженерії

Кафедра програмної інженерії

ЗАВДАННЯ

на курсову роботу з дисципліни «Методологія та технологія створення програмних систем моделювання складних об'єктів»

студента групи СП -4_____

(прізвище, ім'я, по-батькові)

1. Тема роботи _____

2. Термін здачі студентом закінченої роботи _____

3. Вхідні дані по роботі

4. Зміст розрахунково-пояснювальної частини (перелік питань, котрі підлягають розробці)

5. Перелік графічного матеріалу (з точним вказуванням обов'язкових креслень)

Продовження додатку Б.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання курсової роботи	Термін виконання етапів курсової роботи	Примітка
1			
2			
3			
4			
5			
6			
7			
8			

Дата видачі завдання “ ___ ” _____ 2017 р.

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

**Додаток Г
РЕЦЕНЗІЯ**

На курсову роботу ст. гр. СП-4 _____ П.І.П.

1. Відповідність роботи завданню _____
(Так/Ні)

2. Відповідність вимогам ЄСКД _____
(Так/Ні)

3. Питання, що підлягають доопрацюванню або переробці:

Керівник роботи _____ (_____)
Підпис П.І.П.

Повторна рецензія

Керівник роботи _____ (_____)
Підпис П.І.П.

Навчально-методична література

Кінах Я.І., Бойко І.В.

**МЕТОДОЛОГІЯ ТА ТЕХНОЛОГІЯ СТВОРЕННЯ
СКЛАДНИХ ПРОГРАМНИХ
СИСТЕМ**

Навчально-методичний посібник

Комп'ютерне макетування та верстка *А.П. Катрич*

Формат 60x90/16. Обл. вид. арк. 1,4. Тираж 30 прим. Зам. № 2959.

Тернопільський національний технічний університет імені Івана Пулюя.

46001, м. Тернопіль, вул. Руська, 56.

Свідоцтво суб'єкта видавничої справи ДК № 4226 від 08.12.11.