

Міністерство освіти і науки України

Київський національний університет імені Тараса Шевченка



Кафедра програмних систем і технологій

**Методичні вказівки до виконання випускної кваліфікаційної
роботи здобувача вищої освіти першого (бакалаврського) рівня
вищої освіти за спеціальністю
121 “Інженерія програмного забезпечення”**

Київ-2020

УДК 004.412 (076.5)

ББК В 192.145

Е 605

Укладачі: *О. С. Бичков, доктор технічних наук, доцент*

В. Л. Шевченко, доктор технічних наук, професор

Д. С. Берестов, кандидат технічних наук

Т. В. Ковалюк, кандидат технічних наук, доцент

Рецензент: д.т.н., с.н.с., професор кафедри радіоприймання та оброблення сигналів Національного технічного університету України “Київський політехнічний інститут ім. Ігоря Сікорського”

Степанов Михайло Миколайович

Подано основні вимоги щодо тематики, організації, змісту, оформлення та захисту випускної кваліфікаційної роботи здобувачами вищої освіти першого (бакалаврського) рівня вищої освіти за спеціальністю 121 “Інженерія програмного забезпечення”

ЗМІСТ

1	Загальні положення	4
2	Мета виконання випускної кваліфікаційної бакалаврської роботи	5
3	Організація виконання випускної кваліфікаційної бакалаврської роботи	6
4	Структура випускної кваліфікаційної бакалаврської роботи	8
5	Вимоги до змісту випускної кваліфікаційної бакалаврської роботи	8
6	Правила оформлення випускної кваліфікаційної бакалаврської роботи	12
7	Графічна частина випускної кваліфікаційної бакалаврської роботи	20
8	Критерії оцінювання бакалаврської роботи	20
9	Список використаних джерел	21
10	Додатки	22
	Додаток А Приклад форми обкладинки пояснювальної записки	22
	Додаток Б Позначення документів бакалаврської роботи	23
	Додаток В Форма титульного аркуша	24
	Додаток Г Форма завдання на випускної кваліфікаційної бакалаврської роботу	25
	Додаток Д Приклад подання формул та посилань на літературні джерела	26
	Додаток Е Приклад подання таблиць	27
	Додаток Ж Приклад подання графіків та рисунків	29
	Додаток З Приклад подання переліку посилань на джерела	30
	Додаток К Приклади виконання основних сторінок пояснювальної записки	31
	Додаток Л Приклад документування архітектури розробленого програмного забезпечення у відповідності до стандарту <i>Software Architecture Document</i>	48

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Випускна кваліфікаційна бакалаврська робота виконується студентом під керівництвом наукового керівника за темою, затвердженою ректором університету. Робота повинна містити обґрунтування актуальності теми, зв'язок роботи з науковим напрямком науково-дослідної роботи кафедри, огляд літератури за темою і вибір напрямку досліджень, виклад методики досліджень, проведені теоретичні і (або) експериментальні дослідження, проведені студентом, аналіз і узагальнення результатів досліджень, виявлення наукової новизни одержаних результатів, ступеню новизни (вперше одержано, удосконалено, дістало подальший розвиток), висновки, список використаних джерел.

Теми випускних кваліфікаційних бакалаврських робіт, запропоновані викладачами кафедри, які ведуть бакалаврську підготовку, та кандидатури керівників ухвалюються наказом ректора одночасно з наказом на дипломне проектування.

Студент має право змінити назву роботи, узгодивши цю зміну з науковим керівником і завідувачем кафедри, не пізніше ніж за три місяці визначеного терміну подання завершеної бакалаврської роботи.

Випускна кваліфікаційна бакалаврська робота оформляється в одному примірнику обсягом до 70 сторінок за правилами, прийнятими для науково-технічних звітів. Після захисту цей примірник роботи передається для зберігання. Кафедра сприяє студенту в оформленні матеріалів роботи. Кожна робота направляється на рецензію. Рецензентів визначає завідувач кафедри.

Студент зобов'язаний не пізніше, ніж за тиждень перед захистом роботи, подати її в готовому вигляді (переплетену та підписану ним, науковим керівником та нормоконтролером) завідувачу кафедри для направлення на рецензію.

За керівництво випускною кваліфікаційною бакалаврською роботою науковому керівнику зараховується 14 год., рецензенту-опоненту – 2 години навчального навантаження.

Захист роботи відбувається на відкритому засіданні державної екзаменаційної комісії (ДЕК), склад якої визначається та затверджується у порядку, встановленому чинним законодавством.

До захисту студент повинен подати:

- 1) примірник випускної кваліфікаційної бакалаврської роботи;
- 2) відгук наукового керівника про роботу студента над випускною кваліфікаційною бакалаврською роботою;
- 3) рецензію опонента;
- 4) ілюстративний матеріал до випускної кваліфікаційної бакалаврської роботи (електронний демонстративний матеріал, плакати, макети тощо).

Процедура захисту випускної кваліфікаційної бакалаврської роботи містить:

- представлення студента та поданих документів;
- виступ студента з тезами роботи;

відповідь студента на запитання членів ДЕК;
виступ опонента або представлення рецензії;
відповіді студента на зауваження опонента;
виступ керівника або представлення його відгуку;
обговорення роботи та захисту її студентом і прийняття рішення ДЕК;
оголошення рішення ДЕК.

У випадку, коли ДЕК за результатами голосування визнає роботу не захищеною, студент підлягає відрахуванню як такий, що не виконав теоретичний курс навчання на бакалаврському рівні за відповідним напрямком підготовки з виданням йому академічної довідки встановленого зразку.

В цьому випадку ДЕК може рекомендувати допрацювати роботу і подати її до повторного захисту через рік. Це фіксується у протоколі засідання ДЕК і студенту видається витяг з цього протоколу.

ДЕК може висловити свою позицію щодо підготовки бакалавра, наукового рівня керівника випускної кваліфікаційної бакалаврської роботи або якості відгуку рецензента-опонента, а також свої пропозиції або рекомендації щодо підготовки бакалаврів відповідного напрямку. Свою позицію ДЕК формує у вигляді протоколу, який подає до вченої ради факультету інформаційних технологій Київського національного університету імені Тараса Шевченка для наступного розгляду та відповідного реагування.

2. МЕТА ВИКОНАННЯ ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ БАКАЛАВРСЬКОЇ РОБОТИ

Випускна кваліфікаційна бакалаврська робота є самостійно виконаною і відповідно оформленою творчою роботою студента на першому етапі навчання у вищому навчальному закладі з вирішення конкретної науково-технічної або виробничої проблеми на основі набутих у процесі навчання знань та практичних навиків на підтвердження кваліфікації та готовності студента до самостійної роботи в умовах сучасного стану науки і техніки, нових форм організації виробництва.

Виконання випускної кваліфікаційної бакалаврської роботи, як завершальний етап навчального процесу в бакалавратурі, ставить за мету:

розвиток вміння проводити бібліографічну роботу із залученням традиційних та сучасних інформаційних технологій;

оцінити вміння формулювати кінцеву мету та основні напрямки проведення досліджень;

систематизацію, закріплення та набуття досвіду застосування теоретичних знань та практичних навиків при самостійному розв'язуванні конкретних науково-дослідних задач;

розвиток навиків ведення самостійної роботи, оволодіння методикою дослідження та експериментування при вирішенні розроблюваних у випускній кваліфікаційній бакалаврській роботі проблем і задач;

засвоєння сучасних методик проектування нового технічного, математичного, алгоритмічного та програмного забезпечення, а також розробки

нових технологій (способів) із застосуванням сучасних математичних методів та програмних технологій;

оцінити вміння подавати підсумки виконаної роботи у вигляді звітів, рефератів, наукових статей, доповідей і заявок на винаходи тощо, які оформлені згідно з установленими вимогами із залученням сучасних засобів редагування і друку.

3. ОРГАНІЗАЦІЯ ВИКОНАННЯ ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ БАКАЛАВРСЬКОЇ РОБОТИ

Випускна кваліфікаційна бакалаврська робота за змістом і обсягом виконаної роботи повинна відповідати певній стадії проведеного науково-технічного, науково-економічного, науково-методичного тощо дослідження і бути придатним для реалізації окремо або в складі комплексної розробки.

Вихідним документом для виконання роботи є завдання, яке видається студентові на початку навчання на останньому курсі бакалаврського рівні. Завдання на роботу, на відповідному бланку (додаток Г), видається керівником і затверджується завідувачем кафедри.

Назва роботи повинна бути, по можливості, короткою, відповідати обраній спеціальності та суті вирішеної наукової задачі, вказувати на мету дослідження і його завершеність.

У назві не бажано використовувати ускладнену термінологію псевдонаукового характеру. Треба уникати назв, що починаються зі слів "Дослідження питання...", "Дослідження деяких шляхів...", "Деякі питання...", "Матеріали до вивчення...", "До питання..." і т. ін., в яких не відбито в достатній мірі суть проблеми.

При написанні роботи студент повинен обов'язково посилатися на авторів і джерела, з яких запозичив матеріали або окремі результати.

Використовуючи в роботі ідеї або розробки, що належать також і співавторам, разом з якими були написані наукові праці, студент повинен відзначити цей факт у своїй роботі.

В разі використання запозиченого матеріалу без посилання на автора та джерело робота знімається з розгляду незалежно від стадії проходження без права її повторного захисту.

В роботі необхідно стисло, логічно і аргументовано викладати зміст і результати досліджень, уникати загальних слів, бездоказових тверджень, тавтології.

Випускна кваліфікаційна бакалаврська роботу на здобуття кваліфікації фахівець з інженерії програмного забезпечення подають у вигляді спеціально підготовленого рукопису в твердому переплетенні.

Керівник роботи надає студенту консультаційну допомогу в розробці календарного графіка роботи на весь період виконання роботи із зазначенням термінів і черговості розробки окремих розділів.

Контроль за ходом виконання роботи забезпечує керівник не рідше одного разу на два тиждні.

Графік консультацій керівника роботи бакалавра затверджується на засіданні кафедри, яка видала завдання. Графіки роботи консультантів з окремих розділів роботи затверджуються на засіданні відповідних кафедр. Графіки консультацій доводять до відома студентів не пізніше, ніж за тиждень від початку запланованого терміну початку виконання роботи.

Студент розробляє і подає керівникові (консультантові) проекту підготоване рішення питань поставленої задачі. Керівник перевіряє проведену роботу і надає свої зауваження по суті проведених досліджень, запропонованих рішень, фіксує ступінь готовності та відповідного обсягу проекту і при необхідності інформує завідувача кафедри.

До розробки наступного розділу бакалаврської роботи студент приступає тільки після перевірки, погодження і візування керівником проекту виконаної роботи в попередньому розділі.

За правильність поданих у проекті даних та прийнятих рішень відповідає студент – автор бакалаврської роботи.

Питання допуску виконаних випускних кваліфікаційних бакалаврських робіт до захисту розглядається на засіданні випускної кафедри.

Готова і підписана студентом та консультантами робота подається керівникові за **14 робочих днів** до призначеного дня захисту.

Після перевірки відповідності випускної кваліфікаційної бакалаврської роботи до завдання і погодження роботи, керівник її підписує і оформляє письмовий відгук, який повинен містити характеристику виконаної роботи за усіма розділами проекту.

Разом з письмовим відгуком керівника за **10 робочих днів** до захисту студент подає свою роботу на нормоконтроль у відповідності з графіком, затвердженим завідуючим кафедрою. Після перевірки роботи нормоконтролером і одержанням його підпису, випускної кваліфікаційної бакалаврської робота подається завідувачу кафедри не пізніше ніж за **тиждень** до дня захисту.

Допуск випускної кваліфікаційної бакалаврської роботи до захисту затверджується на засіданні кафедри у присутності керівника. Підписана завідувачем кафедри робота направляється на рецензію. В рецензії надається короткий опис роботи, характеристика виконання кожного розділу, позитивні та негативні особливості роботи і на основі цього рецензент виставляє свою оцінку випускної кваліфікаційної бакалаврської роботи. Після одержання рецензії пояснювальна записка випускної кваліфікаційної бакалаврської роботи разом зі всіма необхідними документами подається студентом секретарю Екзаменаційної Комісії (ЕК) за **день** до захисту.

Секретар екзаменаційної комісії перевіряє якість оформлення документів і подає випускну кваліфікаційну бакалаврську роботу в ДЕК.

Після захисту один примірник випускної кваліфікаційної бакалаврської роботи передається в установленому порядку на зберігання у бібліотеку навчального закладу, другий примірник зберігається на випускній кафедрі.

4. СТРУКТУРА ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ БАКАЛАВРСЬКОЇ РОБОТИ

Випускна кваліфікаційна бакалаврська робота повинна містити:
титульний аркуш,
завдання на роботу, календарний план,
зміст,
перелік умовних позначень,
анотації українською, англійською (німецькою) мовами,
вступ,
основну частину,
висновки,
список використаних джерел,
додатки (за необхідності).

5. ВИМОГИ ДО ЗМІСТУ ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ БАКАЛАВРСЬКОЇ РОБОТИ

Пояснювальна записка (далі ПЗ) випускної кваліфікаційної бакалаврської роботи – це документ, в якому приводяться результати проведених досліджень, необхідні розрахунки та обґрунтування прийнятих при виконанні роботи структурних, системних, конструкторських, техніко-економічних, програмних та інших рішень. В загальному випадку ПЗ повинна складатися з таких частин:

	Обсяг в аркушах
1 Титульний аркуш	1
2 Завдання на роботу, календарний план	1
3 Анотація українською мовою	0,5
5 Анотація іноземною (англ., нім.) мовою	0,5
6 Зміст	1 - 2
7 Перелік основних позначень, символів, скорочень	1
8 Вступ	3
9 Основна частина	35 - 45
10 Висновки	1 - 2
11 Список використаних джерел	1 - 3
12 Додатки	0 - 25

Загальний обсяг пояснювальної записки не повинен перевищувати **60-70** сторінок.

Оформлення роботи повинно відповідати вимогам стандарту ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання» введеного в дію наказом національного органу стандартизації України ДП «УкрНДНЦ» від 22.06.2015 р. №61.

5.1 Титульний аркуш випускної кваліфікаційної бакалаврської роботи

Титульний аркуш випускної кваліфікаційної бакалаврської роботи містить: найменування вищого навчального закладу, де виконана робота; прізвище, ім'я, по батькові автора; індекс УДК; назву роботи; шифр і найменування спеціальності; науковий ступінь, вчене звання, прізвище, ім'я, по батькові наукового керівника і консультантів; місто і рік.

На титульному аркуші роботи обов'язково зазначається: "На правах рукопису".

5.2 Завдання на виконання випускної кваліфікаційної бакалаврської роботи та календарний план

Завдання на випускну кваліфікаційну бакалаврську роботу оформляють на відповідному бланку, виконаному друкарським способом, додаток Г.

5.3 Анотація українською мовою

Анотація призначена для ознайомлення з основним напрямком, ідеями та результатами випускної кваліфікаційної бакалаврської роботи і повинна містити стисло характеристику виконаної роботи. В анотації мають бути коротко описані наступні пункти:

Випускна кваліфікаційна бакалаврська робота: кількість сторінок, рисунків, таблиць, додатків та джерел використаних в роботі;

Тема;

Об'єкт дослідження;

Мета роботи;

Предмет дослідження;

Результати дослідження;

Висновок.

Після кожної анотації наводять ключові слова. Ключовим словом називають слово або стійке словосполучення із тексту анотації, яке з погляду інформаційного пошуку несе смислове навантаження. Сукупність ключових слів повинна відображати поза контекстом основний зміст роботи бакалавра.

5.4 Анотація іноземною (англ., нім., франц. тощо) мовою

Анотація іноземною мовою за змістом повинна відповідати українському варіанту (змістовний переклад).

5.5 Зміст

Зміст подають на початку роботи. Він містить найменування та номери початкових сторінок усіх розділів, підрозділів та пунктів (якщо вони мають заголовок), зокрема вступу, висновків до розділів, загальних висновків, додатків, списку використаної літератури та ін.

5.6 Перелік умовних позначень, символів, одиниць, скорочень і термінів (за необхідності)

Якщо в випускній кваліфікаційній бакалаврській роботі вжита специфічна термінологія, а також використано маловідомі скорочення, нові символи, позначення і таке інше, то їх перелік може бути поданий в роботі у вигляді окремого списку, який розміщують перед вступом.

Перелік потрібно друкувати двома колонками, в яких зліва за абеткою наводять, наприклад, скорочення, справа - їх детальну розшифровку.

Якщо в випускній кваліфікаційній бакалаврській роботі спеціальні терміни, скорочення, символи, позначення і таке інше повторюються менше трьох разів, перелік не складають, а їх розшифровку наводять у тексті при першому згадуванні.

5.7 Вступ

Розкриває сутність і стан проблеми (задачі) та її значущість, підстави і вихідні дані для розробки теми, обґрунтування необхідності проведення дослідження.

Далі подають загальну характеристику роботи в наступній послідовності.

Актуальність теми. Шляхом критичного аналізу та порівняння з відомими розв'язаннями проблеми (наукової задачі) обґрунтовують актуальність та доцільність роботи для розвитку відповідної галузі науки чи виробництва, особливо на користь України. Висвітлення актуальності не повинно бути багатослівним. Досить кількома реченнями висловити головне – сутність проблеми або наукового завдання.

Зв'язок роботи з науковими програмами, планами, темами. Коротко викладають зв'язок вибраного напрямку досліджень з планами кафедри і університету, а також з галузевими та (або) державними планами та програмами.

Мета і задачі дослідження. Формулюють мету роботи і задачі, які необхідно вирішити для досягнення поставленої мети. Не слід формулювати мету як "Дослідження...", "Вивчення...", тому що ці слова вказують на засіб досягнення мети, а не на саму мету.

Об'єкт дослідження – це процес або явище, що породжує проблемну ситуацію й обране для вивчення.

Предмет дослідження – міститься в межах об'єкта.

Методи дослідження. Подають перелік використаних методів дослідження для досягнення поставленої в роботі мети. Перераховувати їх треба не відірвано від змісту роботи, а коротко і змістовно визначаючи, що саме досліджувалось чим ти іншим методом.

Новизна одержаних результатів. Подають коротку анотацію нових положень (рішень), запропонованих студентом особисто. Необхідно показати відмінність одержаних результатів від відомих раніше, описати ступінь новизни (вперше одержано, удосконалено, дістало подальший розвиток).

Практичне значення одержаних результатів. В випускній кваліфікаційній бакалаврській роботі, що має теоретичне значення, треба подати відомості використання результатів досліджень або рекомендації щодо їх

використання, а в роботі, що має прикладне значення, відомості про практичне застосування одержаних результатів або рекомендації щодо їх використання. Відзначаючи практичну цінність одержаних результатів, необхідно подати інформацію щодо ступеня готовності до використання або масштабів використання. Необхідно дати короткі відомості щодо впровадження результатів досліджень із зазначенням назв організацій, в яких здійснена реалізація, форм реалізації та реквізитів відповідних документів.

Особистий внесок студента. У випадку використання в випускній кваліфікаційній бакалаврській роботі ідей або розробок, що належать співавторам, разом з якими були опубліковані наукові праці, студент повинен відзначити цей факт у бакалаврській роботі з обов'язковим зазначенням конкретного особистого внеску в ці праці або розробки.

Апробація результатів випускної кваліфікаційної бакалаврської роботи. Вказується, на яких наукових з'їздах, конференціях, симпозіумах, нарадах оприлюднено результати досліджень, що включені до роботи.

Публікації. Вказують, у скількох статтях у наукових журналах, збірниках наукових праць, матеріалах і тезах конференцій, авторських свідоцтвах опубліковані результати роботи.

5.8 Основна частина

Основна частина випускної кваліфікаційної бакалаврської роботи складається з розділів, підрозділів, пунктів, підпунктів. Кожний розділ починають з нової сторінки. Основному тексту кожного розділу може передувати передмова з коротким описом вибраного напрямку та обґрунтуванням застосованих методів досліджень. В кінці кожного розділу формують висновки із стислим викладенням наведених у розділі наукових і практичних результатів, що дає змогу вивільнити загальні висновки від другорядних подробиць.

В розділах основної частини подають:

- огляд літератури за темою і вибір напрямків досліджень;
- виклад загальної методики і основних методів досліджень;
- експериментальну частину і методику досліджень;
- проведені теоретичні і (або) експериментальні дослідження;
- аналіз і узагальнення результатів досліджень.

В огляді літератури студент окреслює основні етапи розвитку наукової думки за своєю проблемою. Стисло, критично висвітлюючи роботи попередників, студент повинен назвати ті питання, що залишились невирішеними і, отже, визначити своє місце у розв'язанні проблеми. Бажано закінчити цей розділ коротким резюме стосовно необхідності проведення досліджень у даній галузі. Загальний обсяг огляду літератури не повинен перевищувати 5% обсягу основної частини роботи.

В другому розділі, як правило, обґрунтовують вибір напрямку досліджень, наводять методи вирішення задач і їх порівняльні оцінки, розробляють загальну методику проведення досліджень. В теоретичних роботах розкривають методи розрахунків, гіпотези, що розглядають, в прикладних -

принципи дії і характеристики розробленого програмного забезпечення.

В наступних розділах з вичерпною повнотою викладають результати власних досліджень автора з висвітленням того нового, що він вносить у розробку проблеми. Студент повинен давати оцінку повноти вирішення поставлених задач, оцінку достовірності одержаних результатів (характеристик, параметрів), їх порівняння з аналогічними результатами вітчизняних і зарубіжних праць, обґрунтування потреби додаткових досліджень, негативні результати, які обумовлюють необхідність припинення подальших досліджень.

Виклад матеріалу підпорядковують одній провідній ідеї, чітко визначеній автором.

5.9 Висновки

Викладають найбільш важливі наукові та практичні результати, одержані в випускній кваліфікаційній бакалаврській роботі, які повинні містити формулювання розв'язаної проблеми (задачі), її значення для науки і практики. Далі формулюють висновки та рекомендації щодо наукового та практичного використання здобутих результатів. В першому пункті висновків оцінюють стан питання. Далі у висновках розкривають методи вирішення поставленої в випускній кваліфікаційній бакалаврській роботі наукової проблеми (задачі), їх практичний аналіз, порівняння з відомими розв'язаннями.

У висновках необхідно наголосити на якісних та кількісних показниках здобутих результатів, обґрунтувати достовірність результатів, викласти рекомендації щодо їх використання.

5.10 Список використаних джерел

Список використаних джерел слід розміщувати в порядку згадування джерел у тексті за їх наскрізною нумерацією.

5.11 Додатки

За необхідності до додатків доцільно включати допоміжний матеріал, необхідний для повноти сприйняття бакалаврської роботи:

- проміжні математичні доведення, формули і розрахунки;
- таблиці допоміжних цифрових даних;
- протоколи і акти випробувань, впровадження;
- інструкції і методики, опис алгоритмів і програм вирішення задач на ЕОМ, які розроблені в процесі виконання випускної кваліфікаційної бакалаврської роботи;
- допоміжні ілюстрації.

6. ПРАВИЛА ОФОРМЛЕННЯ ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ БАКАЛАВРСЬКОЇ РОБОТИ

6.1. Загальні вимоги

Випускню кваліфікаційну бакалаврську роботу друкують за допомогою

комп'ютера на одній стороні аркуша білого паперу формату А4 (210 x 297 мм) з міжрядковим інтервалом 1,5 до тридцяти рядків на сторінці. Мінімальна висота шрифту 1,8 мм. Можна також подати таблиці та ілюстрації на аркушах формату А3.

Обсяг роботи на здобуття кваліфікації фахівця з інженерії програмного забезпечення повинен становити 60-70 сторінок з списком літератури і додатками.

Текст випускної кваліфікаційної бакалаврської роботи необхідно друкувати, залишаючи поля таких розмірів: лівий – не менше 20 мм, правий – не менше 10 мм, верхній – не менше 20 мм, нижній – не менше 20 мм.

Шрифт друку повинен бути чітким, стрічка – чорного кольору середньої жирності. Щільність тексту роботи повинна бути однаковою.

Вписувати в текст випускної кваліфікаційної бакалаврської роботи окремі іншомовні слова, формули, умовні знаки можна чорнилом, тушшю, пастою тільки чорного кольору, при цьому щільність вписаного тексту повинна бути наближеною до щільності основного тексту.

Друкарські помилки, описки і графічні неточності, які виявилися в процесі написання роботи, можна виправляти підчищенням або зафарбуванням білою фарбою і нанесенням на тому ж місці або між рядками виправленого тексту (фрагменту малюнка).

Роздруковані на ЕОМ програмні документи повинні відповідати формату А4 (мають бути розрізаними). Їх включають до загальної нумерації сторінок роботи і розміщують, як правило, в додатках.

Текст основної частини випускної кваліфікаційної бакалаврської роботи поділяють на розділи, підрозділи, пункти та підпункти.

Заголовки структурних частин бакалаврської роботи "ЗМІСТ", "ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ", "ВСТУП", "РОЗДІЛ", "ВИСНОВКИ", "СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ", "ДОДАТКИ" друкують великими літерами симетрично до тексту. Заголовки підрозділів друкують маленькими літерами (крім першої великої) з абзацного відступу. Крапку в кінці заголовка не ставлять. Якщо заголовок складається з двох або більше речень, їх розділяють крапкою. Заголовки пунктів друкують маленькими літерами (крім першої великої) з абзацного відступу в розрядці в підбір до тексту. В кінці заголовка, надрукованого в підбір до тексту, ставиться крапка.

Відстань між заголовком (за виключенням заголовка пункту) та текстом повинна дорівнювати 3-4 інтервалам.

Кожну структурну частину випускної кваліфікаційної бакалаврської роботи потрібно починати з нової сторінки.

6.2. Нумерація

Нумерацію сторінок, розділів, підрозділів, пунктів, підпунктів, малюнків, таблиць, формул подають арабськими цифрами без знака №.

Першою сторінкою випускної кваліфікаційної бакалаврської роботи є титульний аркуш, який включають до загальної нумерації сторінок роботи. На титульному аркуші номер сторінки не ставлять, на наступних сторінках номер

проставляють у правому верхньому куті сторінки без крапки в кінці.

Зміст, перелік умовних позначень, вступ, висновки, список використаних джерел не нумерують. Номер розділу ставлять після слова "РОЗДІЛ", після номера крапку не ставлять, потім з нового рядка друкують заголовок розділу.

Підрозділи нумерують у межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, між якими ставлять крапку. В кінці номера підрозділу повинна стояти крапка, наприклад: "2.3." (третій підрозділ другого розділу). Потім у тому ж рядку йде заголовок підрозділу.

Пункти нумерують у межах кожного підрозділу. Номер пункту складається з порядкових номерів розділу, підрозділу, пункту, між якими ставлять крапку. В кінці номера повинна стояти крапка, наприклад: "1.3.2." (другий пункт третього підрозділу першого розділу). Потім у тому ж рядку йде заголовок пункту. Пункт може не мати заголовка.

Підпункти нумерують у межах кожного пункту за такими ж правилами, як пункти.

Ілюстрації (фотографії, креслення, схеми, графіки, карти) і таблиці необхідно подавати в випускній кваліфікаційній бакалаврській роботі безпосередньо після тексту, де вони згадані вперше, або на наступній сторінці. Ілюстрації і таблиці, які розміщені на окремих сторінках роботи, включають до загальної нумерації сторінок. Таблицю, малюнок або креслення, розміри якого більше формату А4, враховують як одну сторінку і розміщують у відповідних місцях після згадування в тексті або у додатках.

Ілюстрації позначають словом "Рис." і нумерують послідовно в межах розділу, за виключенням ілюстрацій, поданих у додатках.

Номер ілюстрації повинен складатися з номера розділу і порядкового номера ілюстрації, між якими ставиться крапка.

Наприклад:

Рис.1.2 (другий рисунок першого розділу). Номер ілюстрації, її назва і пояснювальні підписи розміщують послідовно під ілюстрацією. Якщо в роботі подано одну ілюстрацію, то її нумерують за загальними правилами.

Таблиці нумерують послідовно (за винятком таблиць, поданих у додатках) в межах розділу. В правому верхньому куті над відповідним заголовком таблиці розміщують на зміщують напис "Таблиця" із зазначенням її номера. Номер таблиці повинен складатися з номера розділу і порядкового номера таблиці, між якими ставиться крапка, наприклад: "Таблиця 1.2" (друга таблиця першого розділу).

Якщо в випускній кваліфікаційній бакалаврській роботі одна таблиця, її нумерують за загальними правилами.

При переносі частини таблиці на інший аркуш (сторінку) слово "Таблиця" і номер її вказують один раз справа над першою частиною таблиці, над іншими частинами пишуть слова "Продовження табл." і вказують номер таблиці, наприклад: "Продовження табл.1.2".

Формули в випускній кваліфікаційній бакалаврській роботі (якщо їх більше одної) нумерують у межах розділу. Номер формули складається з номера

розділу і порядкового номера формули в розділі, між якими ставлять крапку. Нумери формул пишуть біля правого берега аркуша на рівні відповідної формули в круглих дужках, наприклад: (3.1) (перша формула третього розділу).

Примітки до тексту і таблиць, в яких вказують довідкові і пояснювальні дані, нумерують послідовно в межах одної сторінки. Якщо приміток на одному аркуші декілька, то після слова "Примітки" ставлять двокрапку, наприклад:

Примітки:

1. ...

2. ...

Якщо є одна примітка, то її не нумерують і після слова "Примітка" ставлять крапку.

6.3. Ілюстрації

Якість ілюстрацій повинна забезпечувати їх чітке відтворення (електрографічне копіювання, мікрофільмування). Ілюстрації повинні бути виконані на комп'ютері або тушшю, пастою чорного кольору на білому непрозорому папері.

В випускній кваліфікаційній бакалаврській роботі слід застосовувати лише штрихові ілюстрації і оригінали фотознімків.

Фотознімки розміром меншим за формат А4 повинні бути наклеєні на стандартні аркуші білого паперу формату А4.

Ілюстрації повинні мати назву, яку розміщують після номера ілюстрації. При необхідності ілюстрації доповнюють пояснювальними даними (підрисунковий текст).

Для захисту випускної кваліфікаційної бакалаврської роботи застосовуються слайди формату А4 в кількості 6-10 шт.

6.4. Таблиці

Цифровий матеріал, як правило, повинен оформлятися у вигляді таблиць.

Приклад побудови таблиці

		Назва таблиці				Таблиця (номер)
Головка						Заголовки граф
						Підзаголовки граф
Рядки						
Боковик (заголовки рядків)		Графи (колонки)				

Кожна таблиця повинна мати назву, яку розміщують над таблицею і друкують симетрично до тексту. Назву і слово "Таблиця" починають з великої літери. Назву наводять жирним шрифтом.

Заголовки граф повинні починатися з великих літер, підзаголовки з маленьких, якщо вони складають одне речення із заголовком, і з великих, якщо вони є самостійними. Висота рядків повинна бути не меншою 8 мм. Графу з порядковими номерами рядків до таблиці включати не треба.

Таблицю розміщують після першого згадування про неї в тексті, таким чином, щоб її можна було читати без повороту переплетеного блоку роботи або з поворотом за годинниковою стрілкою. Таблицю з великою кількістю рядків можна переносити на інший аркуш. При перенесенні таблиці на інший аркуш (сторінку) назву вміщують тільки над її першою частиною. Таблицю з великою кількістю граф можна ділити на частини і розміщувати одну частину під іншою в межах одної сторінки. Якщо рядки або графи таблиці виходять за формат сторінки, то в першому випадку в кожній частині таблиці повторюють її головку, в другому випадку - боковик.

Якщо текст, який повторюється в графі таблиці, складається з одного слова, його можна замінити лапками; якщо з двох або більше слів, то при першому повторенні його замінюють словами "Те ж", а далі лапками. Ставити лапки замість цифр, марок, знаків, математичних і хімічних символів, які повторюються, не слід. Якщо цифрові або інші дані в якому-небудь рядку таблиці не подають, то в ньому ставлять прочерк. Приклад подання таблиці наведено в додатку Е.

6.5. Формули

Пояснення значень символів і числових коефіцієнтів треба подавати безпосередньо під формулою в тій послідовності, в якій вони дані у формулі. Значення кожного символу і числового коефіцієнта треба подавати з нового рядка. Перший рядок пояснення починають зі слова "де" без двокрапки.

Рівняння і формули треба виділяти з тексту вільними рядками. Вище і нижче кожної формули потрібно залишити не менше одного вільного рядка. Якщо рівняння не вміщується в один рядок, його слід перенести після знака рівності (=) або після знаків плюс (+), мінус (-), множення (x) і ділення (:).

Формули подають у форматі Equation 3...6, вирівнювання – по центру і нумерують в круглих дужках з правого краю. Шрифт - звичайний – 14 пт, великий індекс – 10 пт, маленький індекс – 8 пт, великий символ – 18 пт, маленький символ – 12 пт. Приклад подання формул наведено в додатку Д.

6.6. Посилання

При написанні випускної кваліфікаційної бакалаврської роботи студент повинен давати посилання на джерела, матеріали або окремі результати з яких наводяться в роботі, або на ідеях і висновках яких розроблюються проблеми, задачі, питання, вивченню яких присвячена випускна кваліфікаційна бакалаврська робота. Такі посилання дають змогу відшукати документи і перевірити достовірність відомостей про цитування документа, дають необхідну інформацію щодо нього, допомагають з'ясувати його зміст, мову тексту, обсяг. Посилатися слід на останні видання публікацій. На більш ранні видання можна посилатися лише в тих випадках, коли в них наявний матеріал, який не включено до останнього видання.

Якщо використовують відомості, матеріали з монографій, оглядових статей, інших джерел з великою кількістю сторінок, тоді в посиланні необхідно точно вказати номери сторінок, ілюстрацій, таблиць, формул з джерела, на яке дано посилання в бакалаврській роботі.

Посилання в тексті роботи на джерела слід зазначати порядковим номером за переліком посилань, виділеним двома квадратними дужками, наприклад, "... у працях [1-7]...".

Допускається наводити посилання на джерела у виносках, при цьому оформлення посилання має відповідати його бібліографічному опису за переліком посилань із зазначенням номера.

Приклад:

Цитата в тексті: "... щорічно в Україні утворюється 1,7 мільярдів тон різноманітних твердих промислових відходів... [6] 1)".

Відповідний опис у переліку посилань:

6. Бент О.Й. Про розробку концепції ресурсозбереження в мінерально-сировинному комплексі України// Мінеральні ресурси України.-1995.- № 2.- С.20-21.

Посилання на джерела країни-агресора є НЕПРИПУСТИМИМ у роботі !

Відповідне подання виноски:

1) [6] Бент О.Й. Про розробку концепції ресурсозбереження в мінерально-сировинному комплексі України // Мінеральні ресурси України.- 1995.- № 2.- С.20-21.

Рекомендується в основному тексті або у заключних абзацах розділів давати посилання на особисті наукові праці студента.

Посилання на ілюстрації випускної кваліфікаційної бакалаврської роботи вказують порядковим номером ілюстрації, наприклад, “рис.1.2”.

Посилання на формули випускної кваліфікаційної бакалаврської роботи вказують порядковим номером формули в дужках, наприклад “... у формулі (2.1)”.

На всі таблиці випускної кваліфікаційної бакалаврської роботи повинні бути посилання в тексті, при цьому слово "таблиця" в тексті пишуть скорочено, наприклад: “... в табл.1.2”.

У повторних посиланнях на таблиці та ілюстрації треба вказувати скорочено слово “дивись”, наприклад: “див. табл. 1.3”.

6.7. Список використаних джерел

Джерела можна розміщувати в списку одним із таких способів: в порядку появи посилань у тексті (найбільш зручний для користування), в алфавітному порядку прізвищ перших авторів або заголовків, в хронологічному порядку.

Відомості про джерела, які включені до списку, необхідно давати згідно з вимогами державного стандарту з обов'язковим наведенням назв праць. Приклад оформлення відомостей подано у додатку 3.

Посилання на джерела країни-агресора є НЕПРИПУСТИМИМ у роботі !

6.8. Додатки

Додатки оформлюють як продовження випускної кваліфікаційної бакалаврської роботи на наступних її сторінках або у вигляді окремої частини (книги), розміщуючи їх у порядку появи посилань у тексті роботи.

Якщо додатки оформлюють на наступних сторінках роботи, кожний такий додаток повинен починатися з нової сторінки. Додаток повинен мати заголовок, надрукований угорі малими літерами з першої великої симетрично відносно тексту сторінки. Посередині рядка над заголовком малими літерами з першої великої друкується слово “Додаток ____” і велика літера, що позначає додаток.

Додатки слід позначати послідовно великими літерами української абетки, за винятком літер Г, Є, І, Ї, Й, О, Ч, Ь, наприклад, додаток А, додаток Б і т.д. Один додаток позначається як додаток А.

Текст кожного додатка за необхідності може бути поділений на розділи й підрозділи, які нумерують у межах кожного додатка. У цьому разі перед кожним номером ставлять позначення додатку (літеру) і крапку, наприклад, А.2 - другий розділ додатка А; В.3.1 - підрозділ 3.1 додатка В.

Ілюстрації, таблиці і формули, які розміщені в додатках, нумерують у межах кожного додатка, наприклад: рис. Д.1.2 - другий рисунок першого розділу

додатка Д; формула (А.1) - перша формула додатка А.

6.9. Анотації

На перших сторінках випускної кваліфікаційної бакалаврської роботи мають бути розміщені короткі (до одної тисячі друкованих знаків кожна) анотації українською та англійською мовами. Анотації повинні містити стислу інформацію про основні ідеї та висновки роботи. Всі три анотації мають бути ідентичними за змістом.

Анотації складаються за формою, яка має такий зміст:

прізвище та ініціали студента;

назва випускної кваліфікаційної бакалаврської роботи;

спеціальність (шифр і назва);

установа, де відбудеться захист;

місто, рік;

основні ідеї, результати та висновки випускної кваліфікаційної бакалаврської роботи.

Викладення матеріалу а анотації повинно бути стислим і точним. Належить використовувати синтаксичні конструкції, притаманні мові ділових документів, уникати складних граматичних зворотів. Необхідно використовувати стандартизовану термінологію, уникати маловідомих термінів і символів.

Після кожної анотації наводять ключові слова відповідною мовою. Ключовим словом називається слово або стійке словосполучення із тексту анотації, яке з точки зору інформаційного пошуку несе смислове навантаження. Сукупність ключових слів повинна відображувати поза контекстом основний зміст наукової праці. Загальна кількість ключових слів повинна бути не меншою трьох і не більшою десяти.

Ключові слова подають у називному відмінку, друкують в рядок, через кому.

Змінання аркушів пояснювальної записки (ПЗ), помарки та інші технічні пошкодження не допускаються.

ПЗ випускної кваліфікаційної бакалаврської роботи повинна мати тверду палітурку (з паперу, більш щільного, ніж аркуші ПЗ). ПЗ роботи необхідно прошити і проклеїти.

На кольорову обкладинку ПЗ потрібно наклеїти етикетку з білого паперу розміром 120x80мм, на якій чорним кольором вказують назву документу, його позначення, шифр групи, ім'я та прізвище студента, рік виконання роботи.

На білу обкладинку ПЗ вище згадані дані наносять безпосередньо в рамці, що відповідає розмірам етикетки (додаток А).

7. ГРАФІЧНА ЧАСТИНА ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ БАКАЛАВРСЬКОЇ РОБОТИ

Графічний матеріал при потребі повинен лаконічно відображати суть проведеної роботи в випускній кваліфікаційній бакалаврській роботі. Рекомендований обсяг графічного матеріалу – шість аркушів формату А1.

Перелік графічного матеріалу визначається керівником випускної кваліфікаційної бакалаврської роботи.

Графічні матеріали (*UML*-діаграми, структурні схеми інформаційних потоків, блок-схеми, візуалізації інтерфейсів та їх компонент тощо) повинні відповідати вимогам стандартів Єдиної системи конструкторської документації (ЄСКД).

Графічну частину виконують за допомогою спеціалізованих комплексів інженерної графіки (*P-CAD, AutoCAD, Visio, Corel Draw* та інші) з подальшим формуванням презентаційного відео із застосуванням комплексів ділової комп'ютерної графіки (*FoxGraph, Microsoft PowerPoint, та інші*) або друкують на паперових носіях.

Зображення повинні бути наочними і займати весь аркуш вибраного формату.

8. КРИТЕРІЇ ОЦІНЮВАННЯ БАКАЛАВРСЬКОЇ РОБОТИ

Загальне оцінювання бакалаврської роботи здійснюється з урахуванням якості виконаної письмової роботи та її відповідності встановленим вимогам, додержання студентом затвердженого завдання на виконання бакалаврської роботи, узгодженого з науковим керівником календарного графіка та встановленого регламенту її виконання, а також результатів прилюдного усного захисту на відкритому засіданні атестаційної екзаменаційної комісії. Під час визначення оцінки бакалаврської роботи береться до уваги рівень теоретичної та практичної підготовки студента.

Таблиця 8.1 – Оцінювані параметри і кількість балів

№	Оцінюваний параметр	Кількість балів
1.	Якість оформлення роботи у відповідності до вимог (методичні рекомендації)	5
2.	Теоретична частина:	30
	- повнота опрацювання теоретичних питань роботи	10
	- наявність публікацій/апробацій	10
	- участь у Всеукраїнських конкурсах студентських робіт	10
3.	Документування архітектури розробленого програмного забезпечення у відповідності до стандарту <i>Software Architecture Document</i> (додаток Л)	10
4.	Реалізація програмного забезпечення *:	30
	- рівень інженерної та технологічної реалізації програмного забезпечення;	20
	- наявність промислового впровадження/продуктивної експлуатації	10
5.	Захист роботи **::	25
	- якість доповіді та презентації;	10
	- відповіді на питання.	15
Загалом сумарний бал за роботу		100

* Процент використання існуючих модулів (бібліотек) повинен складати $\leq 30\%$

** Без демонстрації програмного забезпечення робота приймається лише за наявності витягу із засідання кафедри про її фундаментальний теоретичний напрям.

9. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008-2015: Державний стандарт України «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення». – Київ:ДП «УкрНПНЦ», 2016. – 25 с.

2. ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. Госстандарт СССР, 1990.

3. Положення про організацію освітнього процесу у Київському національному університеті ім. Тараса Шевченка. – К.: 2008. – 124 с.

4. Бюлетень вищої атестаційної комісії України – Київ, - №6, 2007 р.

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

УДК XXXX

На правах рукопису

ВИПУСКНА КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Тема: “Назва теми”

Спеціальність: 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

Студент

ІПЗ-XX_____ / Ім’я ПРІЗВИЩЕ /

Науковий керівник

Вчений ступінь, вчене звання /_____ / Ім’я ПРІЗВИЩЕ /

Завідувач кафедри

д.т.н., проф. _____ /Олексій БИЧКОВ/

Київ – 2023

Форма завдання на випускню кваліфікаційну бакалаврську роботу

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра програмних систем і технологій
Спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖУЮ:

Завідувач кафедри програмних
систем і технологій

_____ (Олексій БИЧКОВ)
„___” _____ 20__р.

**ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ
СТУДЕНТУ**

_____ (прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної бакалаврської роботи _____
затверджена наказом вищого навчального закладу від „___” _____ 20__р. № _____

2. Строк здачі студентом закінченої роботи _____

3. Вихідні дані до роботи _____

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)

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

Форма завдання на випускню кваліфікаційну бакалаврську роботу (на звороті першого аркуша)

6. Консультанти з роботи із зазначенням розділів роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник _____

(підпис) (розшифровка підпису)

Завдання прийняв до виконання _____

(підпис) (розшифровка підпису)

КАЛЕНДАРНИЙ ПЛАН

Номер і назва етапів бакалаврської роботи	Термін виконання етапів роботи	Примітка

Студент – бакалавр _____

(підпис) (розшифровка підпису)

Керівник роботи _____

(підпис) (розшифровка підпису)

Приклад подання формул та посилань на літературні джерела

Використавши джерела [2,4] можна легко отримати два наслідки:

1) в одній М-послідовності збільшення на 1 ступені М-послідовності збільшує кількість серій однакової довжини в p разів:

$$\frac{K(S+1)}{K(S)} = \frac{1}{p}; n, p = const . \quad (1.1)$$

2) для однієї і тієї ж основи p збільшення на 1 ступені М-послідовності збільшує кількість серій однакової довжини в p разів:

$$\frac{K(n+1)}{K(n)} = p; S, p = const . \quad (1.2)$$

Приклад подання таблиць

На основі аналізу можна запропонувати залежності $K = f(n, S)$ кількості K серій довжини S від основи p M -послідовності та її ступеня n , які зведені в таблицю 2.4.

Таблиця 2.4

Залежність кількості серій від основи p

S	3	4	5
5	--	--	p^{-1}
4	--	p^{-1}	$(p-n+3)*p+1$
3	p^{-1}	$(p-n+2)*p+1$	$((p-n+3)*p+1)*p$
2	$(p-n+1)*p+1$	$((p-n+2)*p+1)*p$	$((p-n+3)*p+1)*p^2$
1	$((p-n+1)*p+1)*p$	$((p-n+2)*p+1)*p^2$	$((p-n+3)*p+1)*p^3$

Приклад подання графіків та рисунків

Для дослідження структурних властивостей багаторівневих М-послідовностей доцільно провести аналіз їх функцій автокореляції (рис. 1.9)



Рис. 1.9. Графік автокореляційної функції $p=5$

Введені морфізми (рис. 2.1) дозволяють виконати трансляцію однієї множини специфікації в іншу зі збереженням введених теорем.

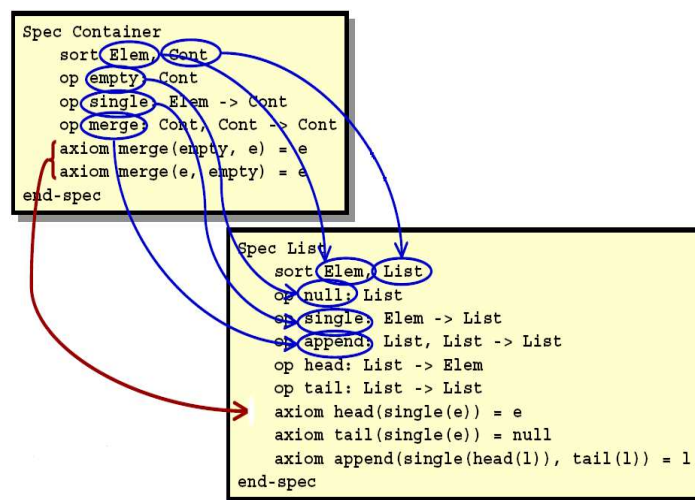


Рис. 2.1. Імплементація специфікацій морфізмів

Приклад подання списку використаних джерел

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Приклад – статті з журналу

1. Юрчишин В.М., Методика формування баз знань експертних систем для аналізу режимів роботи нафтових родовищ. - Івано-Франківськ. // Нафтогазова енергетика, 2007. - №1. – 22 с.

Приклад – книги

2. Корнеев С.Л. Программирование на языке Java. Справочник. Л.: Информационные технологии. Киев, 2008. – 701 с.

Приклад – патентні документи

3. Патент UA 85387 А / Україна. Спосіб вимірювання покладів геотермальних вод на території Прикарпатського регіону / Юрчишин В.М. Опубл. 25.03.2007, Бюл. №35.

Приклад – стандарти

4. ДСТУ 2293-93. Система стандартів безпеки праці. Терміни та визначення.

Приклад – дисертація

5. Самохіна Н.Ф. Фонд наукової бібліотеки: модель обігу документів, засоби рівневої організації: дис.: канд. техн. наук 05.13.06 – захищена 24.06.2007.

Приклад – інтернет-посилання

6. Програмне забезпечення - Wikipedia, the free encyclopedia. – [Електронний ресурс]. – Режим доступу: [http://en.wikipedia.org/wiki/ Програмне_забезпечення](http://en.wikipedia.org/wiki/Програмне_забезпечення)

Приклади виконання основних сторінок пояснювальної записки
випускної кваліфікаційної бакалаврської роботи для студентів спеціальності
121 “Інженерія програмного забезпечення”

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

УДК XXXX

На правах рукопису

ВИПУСКНА КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Тема: “Назва теми”

Спеціальність: 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

Студент

ІПЗ-XX_____ / Ім’я ПРІЗВИЩЕ /

Науковий керівник

Вчений ступінь, вчене звання /_____ / Ім’я ПРІЗВИЩЕ /

Завідувач кафедри

д.т.н., проф. _____ /Олексій БИЧКОВ/

Київ – 2023

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра програмних систем і технологій
Освітньо-кваліфікаційний рівень бакалавр
Спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖЕНО

Зав. кафедри програмних систем і технологій

_____ (Олексій БИЧКОВ)
(підпис) (прізвище та ініціали)

ЗАВДАННЯ

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Іванову Петру Петровичу

(прізвище, ім'я, по-батькові)

1. Тема бакалаврської роботи “Розробка програмних методів та алгоритмічних моделей категоризації інформаційних інтелектуальних систем”

керівник проекту (роботи) Шекета Василь Іванович, к.т.н., доцент _____

затверджені наказом вищого навчального закладу від “ 13 ” жовтня 2017 р. № 736/7

2. Строк подання студентом роботи 18 лютого 2017 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних та програмних технологій певного класу

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз концепцій абстрактної алгебри

2. Аналіз застосування концепцій абстрактної алгебри в комп'ютерних технологіях

3. Вивчення категорійних концепцій в алгоритмах категоризації даних

4. Розробка алгоритмічної моделі та програмної реалізації методів категоризації

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

1. Зображення об'єктів та морфізмів(рис. 1.1, ст. 19)

2. Властивість асоціативності (рис. 1.2, ст. 21)

3. Множини та функції між ними (рис. 1.3, ст. 22)

4. Морфізми ідентифікації (рис. 1.4, ст. 25)

5. Інверсний морфізм (рис. 1.5, ст. 29)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 13 жовтня 2019 р.

Керівник _____ (підпис)

Завдання прийняв до виконання _____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури	25.10.2019	виконано
2	Аналіз концепцій та алгоритмів	10.11.2019	виконано
3	Вивчення категоризаційних концепцій в алгоритмах кластеризації даних	30.11.2019	виконано
4	Розробка алгоритмічної моделі	15.12.2019	виконано
5	Опис розробленого алгоритму	25.12.2019	виконано
6	Програмна реалізація методу категоризації	15.01.2020	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	25.01.2020	виконано

Студент – бакалавр _____ (підпис)

Керівник роботи _____ (підпис)

АНОТАЦІЯ *(на трьох мовах)*

Випускна кваліфікаційна бакалаврська робота: 68 с., 22 рис., 8 табл., 3 додат., 33 джерела.

Тема: Розробка програмних методів та алгоритмічних моделей категоризації інформаційних інтелектуальних систем

Об'єкт дослідження: дані нафтогазової предметної області, представлені у вигляді реляційних структур даних.

Мета роботи: розробка формальних та програмних методів категоризації масивів даних .

Предмет дослідження: технологія категоризації даних, досліджуються методи категоризації даних на основі алгоритму C4.5.

Результати дослідження:

Досліджено можливості застосування концепцій абстрактної алгебри та теорії категорій до виділених задач програмної та комп'ютерної інженерії. Запропоновано формально-логічний підхід до виконання категоризації даних нафтогазової предметної області..

Висновок

В результаті досліджень було отримано алгоритм, що застосовується для розбиття на кластери будь-яких масивів категорійних даних, що дозволяє спростити їх подальшу обробку і прийняти рішення, шляхом застосування до кожного кластера відповідного методу аналізу

АЛГОРИТМ, КЛАСТЕР, КАТЕГОРІЙНІ МАСИВИ ДАНИХ,
ІНТЕЛЕКТУАЛЬНА СИСТЕМА, НАФТОГАЗОВА ПРЕДМЕТНА
ОБЛАСТЬ.

ЗМІСТ

Стр.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І

ТЕРМІНІВ.....6

ВСТУП.....8

РОЗДІЛ 1

АНАЛІЗ КОНЦЕПЦІЇ АБСТРАКТНОЇ АЛГЕБРИ ТА ТЕОРІЇ КАТЕГОРІЙ І
ЇХ ЗАСТОСУВАНЬ

1.1 Аналіз концепції абстрактної алгебри11

1.2 Застосування теоретичних концепцій абстрактної алгебри і теорій
категорій в комп'ютерних технологіях.....17

1.3

1.4

1.5 Висновки до розділу.....22

РОЗДІЛ 2

ІМПЛЕМЕНТАЦІЯ КАТЕГОРІЙНИХ КОНЦЕПЦІЙ В АЛГОРИТМАХ
КАТЕГОРИЗАЦІЇ ДАНИХ

2.1 Складнощі і проблеми, які можуть виникнути при застосуванні
кластерного аналізу.....25

2.2 Огляд неієрархічних алгоритмів кластерного аналізу.....28

2.3

2.4

2.5 Висновки до розділу.....30

РОЗДІЛ 3

РОЗРОБКА АЛГОРИТМІЧНОЇ МОДЕЛІ ТА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ
МЕТОДІВ КАТЕГОРИЗАЦІЇ ІНФОРМАЦІЙНИХ СИСТЕМ

3.1 Суть алгоритму C4.5.....	37
3.2 Опис алгоритму C4.5.....	40
3.3	
3.4 Опис програми.....	46
3.5 Висновки до розділу.....	57
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66
ДОДАТКИ.....	65

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД	-	база даних
ЕОМ	-	електронно-обчислювальна машина
ІС	-	інформаційна система
ІТ	-	інформаційні технології
НДІ	-	науково-дослідницький інститут
ООП	-	об'єктно-орієнтоване програмування
ОС	-	операційна система
ПЗ	-	програмне забезпечення
СУБД	-	система управління базами даних
ГК	-	гамма каротаж
НГК	-	нейтронний гамма каротаж
АК	-	акустичний каротаж
МВП	-	метод власних потенціалів

ВСТУП

Актуальність роботи

На сучасному етапі першим кроком при аналізі даних являється кластеризація. Кластеризація (об'єднання в групи схожих об'єктів) – є однією з фундаментальних задач у області аналізу даних і технології Data Mining [1]. Список прикладних областей, де вона застосовується, широкий: сегментація зображень, маркетинг, боротьба з шахрайством, прогнозування, аналіз текстів і багато інших. На сучасному етапі кластеризація часто виступає першим кроком при аналізі даних. Після виділення схожих груп застосовуються інші методи, для кожної групи будується окрема модель.

.....

Тому в останнє десятиліття ведуться активні дослідження у області розробки алгоритмів кластеризації категорійних і транзакційних даних, що масштабуються (scalable).

Порівняння роботи з відомими розв'язаннями проблеми

Вирішення проблем кластеризації даних являється ключовим завданням для багатьох наукових центрів Європи, Азії та Північної Америки.

Концепції, що використовуються в Birch дозволили запропонувати два алгоритми кластеризації, що масштабуються, для даних в просторі з довільною метрикою [3].

Пол Бредлі і його колеги розробляють цілий клас інтерактивних алгоритмів кластеризації, що масштабуються. Стартуючи з деякого первинного розбиття безлічі даних, в цих ітераційних алгоритмах кластеризації точки багато разів переносяться з одного кластера в інший, поки не буде знайдений розподіл, відповідний оптимальному значенню деякої критерійної функції.

.....

У даній роботі був проведений аналіз багатьох існуючих методів кластеризації та вибраний і реалізований найбільш підходящий алгоритм для кластеризації категорійних даних, що показує досить високу ефективність порівняно з іншими подібними алгоритмами, оскільки забезпечує порівняно високу чистоту розбиття при невеликій кількості кластерів.

Мета і задачі дослідження

Метою бакалаврської роботи є розробка формальних та програмних методів кластеризації категорійних масивів даних на прикладі даних нафтогазової предметної області.

Досліджувана модель повинна розбивати на кластери будь-які категорійні масиви, працюючи з ними як з транзакційними (під терміном транзакція розуміється деякий довільний набір об'єктів, наприклад геологічні дані про породу, список ключових слів статті, симптоми пацієнта). Особливо бажаними вимогами являються мінімально можлива кількість сканувань таблиці та робота в обмеженому об'ємі оперативної пам'яті, а також можливість переривання роботи алгоритму із збереженням проміжних результатів.

Досягнення мети включало розв'язання таких **задач**:

- 1) огляд існуючих концепцій абстрактної алгебри;
- 2) вивчення можливостей застосування принципів теорій категорій у програмній інженерії;
- 3) аналіз існуючих алгоритмів кластеризації даних;
- 4) вибір релевантного алгоритму та обґрунтування доцільності його використання;
- 5) реалізація алгоритма C4.5.

Об'єктом дослідження є дані нафтогазової предметної області, представлені у вигляді реляційних структур даних.

Предметом дослідження є технологія кластеризації даних, досліджуються методи кластеризації категорійних даних на основі алгоритму C4.5.

Методи дослідження

Для виділення на предметній області зв'язків між ними застосовуються методи категоризаційного моделювання. Для побудови формальної логічної моделі процесу кластеризації даних використовується апарат абстрактної алгебри, теорії категорій, а також формально-логічних конструкцій на їх основі.

Наукова новизна отриманих результатів

Досліджено можливості застосування концепцій абстрактної алгебри та теорії категорій до виділених задач програмної та комп'ютерної інженерії. Запропоновано формально-логічний підхід до виконання категоризації даних нафто-газової предметної області. Виконана програмна реалізація алгоритму C4.5 для кластеризації категорійних даних.

Практичне значення одержаних результатів

Одержана реалізація алгоритма C4.5, що дозволяє виконувати розбиття на кластери будь-яких масивів категорійних даних. А це дозволяє спростити подальшу обробку даних і ухвалення рішень, застосовуючи до кожного кластера свій метод аналізу.

.....

Виділено ієрархічні та неієрархічні методи кластеризації. Показано, що класичні неієрархічні алгоритми ефективні при опрацюванні числових даних та малопридатними для опрацювання категорійних даних.

Особистий внесок студента

Основним результатом є:

1. запропонований автором підхід до категоризації даних нафто-газової предметної області;
2. приведена реалізація алгоритму C4.5, що виконує кластеризацію категорійних даних, одержаних як результат прогнозування нафтогазоносності по виділених родовищах.

Апробація результатів випускної кваліфікаційної бакалаврської роботи

Результати бакалаврського дослідження були представлені та схвалені на науковій конференції факультету автоматизації та комп'ютерних наук ІФНТУНГ, а також міжнародній науково-технічній фаховій конференції Інституту програмних систем НАН України “Укрпрограмування 2016”, що відбулася в м. Києві 27-29 листопада 2016 р.

Публікації

За результатами наукових досліджень, проведених у бакалаврській роботі, підготовлено доповідь для участі в міжнародній науково-технічній фаховій конференції Інституту програмних систем НАН України “Укрпрограмування 2016”, секція Експертні та інтелектуальні інформаційні системи, яка відбулася 27-29 листопада 2016 р. В м. Києві, а також підготовлено до друку фахову статтю по напрямку Експертні та інтелектуальні інформаційні системи, що прийнята до друку в міжнародному фаховому журналі “Проблеми програмування” Інституту програмних систем НАН України.

Структура та обсяг роботи

Робота викладена на 68 сторінках друкованого тексту, який складається із вступу, трьох розділів, висновків, списку використаних джерел (23 найменування). Робота містить 12 таблиць, 32 рисунки та 3 додатки, обсягом 12 стор.

РОЗДІЛ 1

АНАЛІЗ КОНЦЕПЦІЇ АБСТРАКТНОЇ АЛГЕБРИ ТА ТЕОРІЇ КАТЕГОРІЙ І ЇХ ЗАСТОСУВАНЬ

1.1 Аналіз концепції абстрактної алгебри

1.1.1 Концепція груп, напівгруп, моноїдів та квазігруп

Абстрактна алгебра або вища алгебра — розділ математики, який вивчає алгебраїчні системи (також іноді звані структурами алгебри), такі як групи, кільця, поля, частково впорядковані множини, решітки, а також відображення між такими структурами [3-5]. Історично структури алгебри виникали спочатку в інших областях математики. Після абстрагування від непотрібних деталей і виділення аксіоматичних визначень вони ставали предметом вивчення абстрактної алгебри.

Саме тому абстрактна алгебра знаходить численні застосування в більшості інших областей математики. Прикладами структур алгебри з бінарною операцією є півгрупи, моноїди, групи та квазігрупи.

Як видно з таблиці 1.1. в М-послідовності немає тільки n - значної комбінації, що складається з одних 0. У будь-якому випадку, коли період схеми не максимальний, послідовність містить не всі комбінації, і значить, вид послідовності визначається первинним станом схеми.

Таблиця 1.1

Фази М-послідовності

Комбінація	Розряди	Комбінація	Розряди
0001	1-4	1010	9
0011	2-5	0100	10
0111	3-6	1100	11
1111	4-7	1001	12

1110	5-8	0010	13
1101	6-9	0100	14

Продовження таблиці 1.1.

1010	7-10	1000	15
0101	8-11	—	—

Векторні простори і лінійні відображення між ними вивчаються в розділі лінійної алгебри. Рівняння алгебри вищих порядків від однієї змінної, а також, більш у загальних рисах, властивості груп автоморфізмів різних систем алгебри є предметом теорії Галуа [6].

.....

В результаті, ці алгоритми не в змозі адекватно розбити на кластери неопуклі множини (рис. 1.1), тим більше вкладені структури.

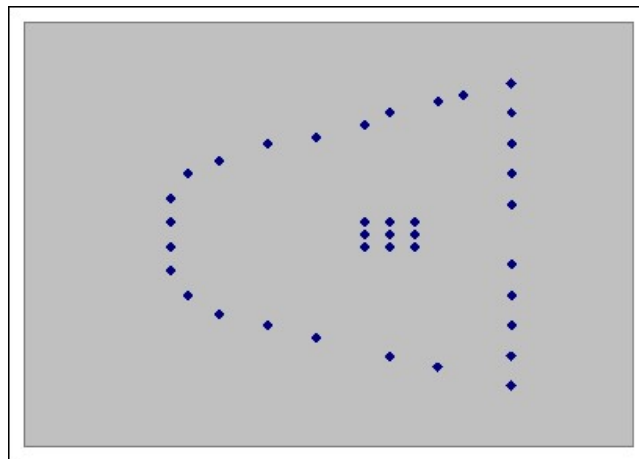


Рис. 1.1. Приклад неопуклих множин

Таким чином, формула стандартизації показників відносно нормативного показника матиме такий вигляд:

$$X_{ij} = \frac{M_{ij}}{NormM_{ij}} K_v, \text{-----} (1.1)$$

де K_v — ваговий коефіцієнт, який визначається експертним шляхом і характеризує числову значимість того чи іншого показника відносно інших показників .

.....

ВИСНОВКИ

Застосування теорій категорій дозволяє одержати теоретичний фундамент для введення класів об'єктів і класів стрілок (морфізмів). В якості основних властивостей категорійних конструкцій розглядаються властивості ідентифікації, композиції та асоціативності.

Виконаний аналіз показує перспективність застосування теорій категорій в області програмної інженерії в таких напрямках:

- виконання моделювання абстрактних та прикладних програмних конструкцій, а також моделювати логічний висновок на їх основі;
- одержання точних означення для принципів модульності та композиційності в програмних застосуваннях;

.....

Виконаний аналіз показує, що не існує єдиного універсального алгоритму кластеризації, тому при використанні будь-якого з них важливо враховувати переваги та недоліки, природу даних, з якими даний алгоритм дає найкращий результат .

Пропонована реалізація алгоритму застосовується для розбиття на кластери будь-яких масивів категорійних даних, що дозволяє спростити їх подальшу обробку і прийняти рішення, шляхом застосування до кожного кластера відповідного методу аналізу.

Одержані в даному дослідженні результати обговорені та схвалені на науковій конференції інституту інформаційних технологій ІФНТУНГ, а також міжнародній науково-технічній фаховій конференції Інституту програмних систем НАН України “Укрпрограмування 2019”, що відбулася в м. Києві 27-29 листопада 2019 р.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Соловьева Ф.И. Введение в теорию кодирования. Учебное пособие / Новосибирский ун-т. Новосибирск, 2006. – 127 с.
2. Іляш Ю.Ю., Горелов В.О. Аналіз адаптивних систем зменшення надлишковості інформаційних потоків / Електроніка та системи управління. – 2009. – № 3 (21). – С. 133 – 137.
3. Гринчишин Т.М. Безнадлишковий метод кодування на основі моделювання процесорів формування бісигнальної оптичної послідовності / Штучний інтелект. – 2009. – № 1. – С. 250 – 253.
4. Пилипенко А.І., Доценко Р.В. Дослідження рекурентних методів стиснення інформації / Вісник Хмельницького національного університету. Т. 1. – 2007. – № 2. – С. 80 – 82.
5. Петришин Л.Б. Теоретичні основи перетворення форми та цифрової обробки інформації в базисі Галуа: Навч. посібник. – Київ: ІЗіМН МОУ – 1997. – 237 с.
6. Салкин В.У., Чеголин П.М., Шварко В.П. Методы и средства обработки сигналов в дискретных базисах. М.: Наука и техника. – 1987. – 296 с.

<Insert OrganizationName>

**<Insert Project Name>
Software Architecture Document (SAD)**

CONTENT OWNER: <Insert Name>

DOCUMENT NUMBER:

-
-
-
-
-
-

RELEASE/REVISION:

-
-
-
-
-
-

RELEASE/REVISION DATE:

-
-
-
-
-
-

BACKGROUND

This template is based on the Software Engineering Institute's "View and Beyond" method for documenting software architectures, as described in Clements, et al., [Documenting Software Architecture: Views and Beyond](#) (Addison Wesley, 2002). The current version is available for [free download](#) from the SEI's architecture web site.

TIPS FOR USING THIS TEMPLATE

To create an instance of this document:

- Insert relevant information on cover sheet and in placeholders throughout.
- Insert relevant information in page header: Move to a page of the body of the report, select *View > Header and Footer* from the main menu, and then replace relevant information in the header box at the top of the page.

To update the contents and page numbers in the Table of Contents, List of Figures, and List of Tables:

- Position the cursor anywhere in the table to be updated.
- Click the *F9* function key.
- Answer "Update entire table".

To insert a figure or table caption:

- From the main menu, choose *Insert > Reference > Caption* and then either *Figure* or *Table* as needed.
- Click the OK button.
- Add a colon and a tab stop after the figure number in the caption itself.
- The caption should use the *Caption* style.
- Add a colon and a tab stop after the table/figure number in the caption itself.

TIPS FOR MAKING YOUR DOCUMENT MORE READABLE

- A gray box containing *CONTENTS OF THIS SECTION* is provided at the beginning of most sections and subsections. After determining what specific information will be included in your document, you can remove this gray box or leave it to serve as a quick-reference section overview for your readers. In the case that text has been provided in the template, inspect it for relevance and revised as necessary.
- Consider hyperlinking key words used in the document with their entries in the [Glossary](#) or other location in which they are defined. Choose *Insert > Hyperlink*.
- Don't leave blank sections in the document. Mark them "To be determined" (ideally with a promise of a date or release number by which the information will be provided) or "Not applicable."
- Consider packaging your SAD as a multi-volume set of documentation. It is often helpful to break your documentation into more than one volume so that the document does not become unwieldy. There are many ways that this can be accomplished. The structuring of the document must support the needs of the intended audience and must be determined in the context of the project. Each document that you produce should include the date of issue and status; draft, baseline, version number, name of issuing organization; change history; and a summary. A few decomposition options are:
 - *A 2-Volume approach*: Separate the documentation into two volumes; one that contains the views of the software architecture and one that contains everything else. A common variant of this approach has one volume per view, and one volume for everything else.
 - *A 3-Volume approach*: Document organizational policies, procedures, and the directory in one volume, system specific overview material in a second, and view documentation in a third.
 - *A 4-Volume approach*: Create one volume for each viewtype [Clements 2002] (module, component-and-connector, allocation) that contains the documentation for the relevant [views](#). Include all of the other information in the fourth volume.
 - Software interfaces are often documented in a separate volume.

In any case, the information should be arranged so that readers begin with the volume containing the Documentation Roadmap (Section 1 in this template).

Table of Contents

1	Documentation Roadmap	iii
1.1	Document Management and Configuration Control Information.....	iii
1.2	Purpose and Scope of the SAD	iv
1.3	Viewpoint Definitions	v
1.3.1	<Insert name of viewpoint> Viewpoint Definition	vi
1.3.1.1	Abstract	vii
1.3.1.2	Stakeholders and Their Concerns Addressed.....	vii
1.3.1.3	Elements, Relations, Properties, and Constraints	vii
1.3.1.4	Language(s) to Model/Represent Conforming Views	vii
2	Architecture Background	viii
2.1	Problem Background	viii
2.1.1	System Overview	viii
2.1.2	Goals and Context	viii
2.1.3	Significant Driving Requirements.....	viii
2.2	Solution Background	viii
2.2.1	Architectural Approaches	ix
2.2.2	Analysis Results	ix
2.2.3	Requirements Coverage	ix
3	Referenced Materials	x
4	Directory	xi
4.1	Index.....	xi
4.2	Glossary.....	xi
4.3	Acronym List	xii
5	Sample Figures & Tables	xiii

List of Figures

Figure 1: Sample Figure.....	xiii
------------------------------	------

List of Tables

Table 1:	Stakeholders and Relevant Viewpoints	vi
Table 2:	Sample Table.....	xiii

1. Documentation Roadmap

The Documentation Roadmap should be the first place a new reader of the SAD begins. But for new and returning readers, it is intended to describe how the SAD is organized so that a reader with specific interests who does not wish to read the SAD cover-to-cover can find desired information quickly and directly.

Sub-sections of Section 1 include the following.

- Section 1.1 (“Document Management and Configuration Control Information”) explains revision history. This tells you if you’re looking at the correct version of the SAD.
- Section 1.2 (“Purpose and Scope of the SAD”) explains the purpose and scope of the SAD, and indicates what information is and is not included. This tells you if the information you’re seeking is likely to be in this document.
- Section **Ошибка! Источник ссылки не найден.** (“**Ошибка! Источник ссылки не найден.**”) explains the information that is found in each section of the SAD. This tells you what section(s) in this SAD are most likely to contain the information you seek.
- Section **Ошибка! Источник ссылки не найден.** (“**Ошибка! Источник ссылки не найден.**”) explains the stakeholders for which the SAD has been particularly aimed. This tells you how you might use the SAD to do your job.
- Section 1.3 (“Viewpoint Definitions”) explains the *viewpoints* (as defined by IEEE Standard 1471-2000) used in this SAD. For each viewpoint defined in Section 1.3, there is a corresponding view defined in Section **Ошибка! Источник ссылки не найден.** (“**Ошибка! Источник ссылки не найден.**”). This tells you how the architectural information has been partitioned, and what views are most likely to contain the information you seek.
- Section **Ошибка! Источник ссылки не найден.** (“**Ошибка! Источник ссылки не найден.**”) explains the standard organization used to document architectural views in this SAD. This tells you what section within a view you should read in order to find the information you seek.

1.1. Document Management and Configuration Control Information

CONTENTS OF THIS SECTION: This section identifies the version, release date, and other relevant management and configuration control information associated with the current version of the document. Optional items for this section include: change history and an overview of significant changes from version to version.

- Revision Number: << >>
- Revision Release Date: << >>
- Purpose of Revision: << >>

-
- Scope of Revision: <<list sections or page numbers that have been revised; provide a summary overview of the differences between this release and the previous one.>>

1.2. Purpose and Scope of the SAD

CONTENTS OF THIS SECTION: This section explains the SAD's overall purpose and scope, the criteria for deciding which design decisions are architectural (and therefore documented in the SAD), and which design decisions are non-architectural (and therefore documented elsewhere).

This SAD specifies the software architecture for <insert scope of SAD>. All information regarding the software architecture may be found in this document, although much information is incorporated by reference to other documents.

What is software architecture? The software architecture for a system¹ is the structure or structures of that system, which comprise software elements, the externally-visible properties of those elements, and the relationships among them [Bass 2003]. "Externally visible" properties refers to those assumptions other elements can make of an element, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on. This definition provides the basic litmus test for what information is included in this SAD, and what information is relegated to downstream documentation.

Elements and relationships. The software architecture first and foremost embodies information about how the elements relate to each other. This means that architecture specifically omits certain information about elements that does not pertain to their interaction. Thus, a software architecture is an *abstraction* of a system that suppresses details of elements that do not affect how they use, are used by, relate to, or interact with other elements. Elements interact with each other by means of interfaces that partition details about an element into public and private parts. Software architecture is concerned with the public side of this division, and that will be documented in this SAD accordingly. On the other hand, private details of elements—details having to do solely with internal implementation—are not architectural and will not be documented in a SAD.

Multiple structures. The definition of software architecture makes it clear that systems can and do comprise more than one structure and that no one structure holds the irrefutable claim to being the architecture. The neurologist, the orthopedist, the hematologist, and the dermatologist all take a different perspective on the structure of a human body. Ophthalmologists, cardiologists, and podiatrists concentrate on subsystems. And the kinesiologist and psychiatrist are concerned with different aspects of the entire arrangement's behavior. Although these perspectives are pictured differently and have very different properties, all are inherently related; together they describe the architecture of the human body. So it is with software. Modern systems are more than complex enough to make it difficult to grasp them all at once. Instead, we restrict our attention at any one moment to one (or a small number) of the software system's structures. To communicate meaningfully about an architecture, we must make clear which structure or structures we are discussing at the moment—which *view* we are taking of the architecture. Thus, this SAD follows the principle that documenting a software architecture is a matter of documenting the relevant views and then documenting information that applies to more than one view.

¹ Here, a system may refer to a system of systems.

For example, all non-trivial software systems are partitioned into implementation units; these units are given specific responsibilities, and are the basis of work assignments for programming teams. This kind of element will comprise programs and data that software in other implementation units can call or access, and programs and data that are private. In large projects, the elements will almost certainly be subdivided for assignment to sub-teams. This is one kind of structure often used to describe a system. It is a very static structure, in that it focuses on the way the system's functionality is divided up and assigned to implementation teams.

Other structures are much more focused on the way the elements interact with each other at runtime to carry out the system's function. Suppose the system is to be built as a set of parallel processes. The set of processes that will exist at runtime, the programs in the various implementation units described previously that are strung together sequentially to form each process, and the synchronization relations among the processes form another kind of structure often used to describe a system.

None of these structures alone is *the* architecture, although they all convey architectural information. The architecture consists of these structures as well as many others. This example shows that since architecture can comprise more than one kind of structure, there is more than one kind of element (e.g., implementation unit and processes), more than one kind of interaction among elements (e.g., subdivision and synchronization), and even more than one context (e.g., development time versus runtime). By intention, the definition does not specify what the architectural elements and relationships are. Is a software element an object? A process? A library? A database? A commercial product? It can be any of these things and more.

These structures will be represented in the views of the software architecture that are provided in Section **Ошибка! Источник ссылки не найден..**

Behavior. Although software architecture tends to focus on structural information, *behavior of each element is part of the software architecture* insofar as that behavior can be observed or discerned from the point of view of another element. This behavior is what allows elements to interact with each other, which is clearly part of the software architecture and will be documented in the SAD as such. Behavior is documented in the element catalog of each view.

1.3. Viewpoint Definitions

CONTENTS OF THIS SECTION: This section provides a short textual definition of a viewpoint and how the concept is used in this SAD. The section describes viewpoints that may be used in the SAD. The specific viewpoints will be tailored by the organization.

The SAD employs a stakeholder-focused, multiple view approach to architecture documentation, as required by ANSI/IEEE 1471-2000, the recommended best practice for documenting the architecture of software-intensive systems [IEEE 1471].

As described in Section 1.2, a software architecture comprises more than one software structure, each of which provides an engineering handle on different system qualities. A *view* is the specification of one or more of these structures, and documenting a software architecture, then, is a matter of documenting the relevant views and then documenting information that applies to more than one view [Clements 2002].

ANSI/IEEE 1471-2000 provides guidance for choosing the best set of views to document, by bringing stakeholder interests to bear. It prescribes defining a set of viewpoints to satisfy the stakeholder community. A viewpoint identifies the set of concerns to be addressed, and identifies the modeling techniques, evaluation techniques, consistency checking techniques, etc., used by any conforming view. A view, then, is a viewpoint applied to a system. It is a representation of a set of software elements, their properties, and the relationships among them that conform to a defining viewpoint. Together, the chosen set of views show the entire architecture and all of its relevant properties. A SAD contains the viewpoints, relevant views, and information that applies to more than one view to give a holistic description of the system.

The remainder of Section 1.5 defines the viewpoints used in this SAD. The following table summarizes the stakeholders in this project and the viewpoints that have been included to address their concerns.

Table 1: Stakeholders and Relevant Viewpoints

Stakeholder	Viewpoint(s) that apply to that class of stakeholder's concerns

1.3.1. <Insert name of viewpoint> Viewpoint Definition

There will be one of these subsections for each viewpoint defined. The subsections are as follows:

- **Abstract:** A brief overview of the viewpoint
- **Stakeholders and their concerns addressed:** This section describes the stakeholders and their concerns that this viewpoint is intended to address. Listed are questions that can be answered by consulting views that conform to this viewpoint. Optionally, the section includes significant questions that cannot be answered by consulting views conforming to this viewpoint.
- **Elements, relations, properties, and constraints:** This section defines the types of elements, the relations among them, the significant properties they exhibit, and the constraints they obey for views conforming to this viewpoint.
- **Language(s) to model/represent conforming views:** This section lists the language or languages that will be used to model or represent views conforming to this viewpoint, and cite a definition document for each.
- **Applicable evaluation/analysis techniques and consistency/completeness criteria:** This section describes rules for consistency and completeness that apply to views in this viewpoint, as well as any analysis of evaluation techniques that apply to the view that can be used to predict qualities of the system whose architecture is being specified.
- **Viewpoint source:** This section provides a citation for the source of this viewpoint definition, if any.

Following is an example of a viewpoint definition.

1.5.1 Module decomposition viewpoint definition

1.5.1.1 Abstract. Views conforming to the module decomposition viewpoint partition the system into a unique non-overlapping set of hierarchically decomposable implementation units (*modules*).

1.5.1.2 Stakeholders and Their Concerns Addressed. Stakeholders and their concerns addressed by this viewpoint include

- project managers, who must define work assignments, form teams, and formulate project plans and budgets and schedules;
- COTS specialists, who need to have software elements defined as units of functionality, so they can search the marketplace and perform trade studies to find suitable COTS candidates;
- testers and integrators who use the modules as their unit of work;
- configuration management specialists who are in charge of maintaining current and past versions of the elements;
- system build engineers who use the elements to produce a running version of the system;

-
- maintainers, who are tasked with modifying the software elements;
 - implementers, who are required to implement the elements;
 - software architects for those software elements sufficiently large or complex enough to warrant their own software architectures;
 - the customer, who is concerned that projected changes to the system over its lifetime can be made economically by confining the effects of each change to a small number of elements.

1.5.1.3 Elements, Relations, Properties, and Constraints. Elements of the module decomposition viewpoint are modules, which are units of implementation that provide defined functionality. Modules are hierarchically decomposable; hence, the relation is “is-part-of.” Properties of elements include their names, the functionality assigned to them (including a statement of the quality attributes associated with that functionality), and their software-to-software interfaces. The module properties may include requirements allocation, supporting requirements traceability.

1.5.1.4 Language(s) to Model/Represent Conforming Views. Views conforming to the module decomposition viewpoint may be represented by (a) plain text using indentation or outline form [Clements 2002]; (b) UML, using subsystems or classes to represent elements and “is part of” or nesting to represent the decomposition relation.

1.3.1.1. Abstract

1.3.1.2. Stakeholders and Their Concerns Addressed

1.3.1.3. Elements, Relations, Properties, and Constraints

1.3.1.4. Language(s) to Model/Represent Conforming Views

2. Architecture Background

2.1. Problem Background

CONTENTS OF THIS SECTION: The sub-parts of Section 2.1 explain the constraints that provided the significant influence over the architecture.

2.1.1. System Overview

CONTENTS OF THIS SECTION: This section describes the general function and purpose for the system or subsystem whose architecture is described in this SAD.

2.1.2. Goals and Context

CONTENTS OF THIS SECTION: This section describes the goals and major contextual factors for the software architecture. The section includes a description of the role software architecture plays in the life cycle, the relationship to system engineering results and artifacts, and any other relevant factors.

2.1.3. Significant Driving Requirements

CONTENTS OF THIS SECTION: This section describes behavioral and quality attribute requirements (original or derived) that shaped the software architecture. Included are any scenarios that express driving behavioral and quality attribute goals, such as those crafted during a Quality Attribute Workshop (QAW) [Barbacci 2003] or software architecture evaluation using the Architecture Tradeoff Analysis MethodSM (ATAMSM) [Bass 2003].

2.2. Solution Background

CONTENTS OF THIS SECTION: The sub-parts of Section 2.2 provide a description of why the architecture is the way that it is, and a convincing argument that the architecture is the right one to satisfy the behavioral and quality attribute goals levied upon it.

SM Quality Attribute Workshop and QAW and Architecture Tradeoff Analysis Method and ATAM are service marks of Carnegie Mellon University.

2.2.1. Architectural Approaches

CONTENTS OF THIS SECTION: This section provides a rationale for the major design decisions embodied by the software architecture. It describes any design approaches applied to the software architecture, including the use of architectural styles or design patterns, when the scope of those approaches transcends any single architectural view. The section also provides a rationale for the selection of those approaches. It also describes any significant alternatives that were seriously considered and why they were ultimately rejected. The section describes any relevant COTS issues, including any associated trade studies.

2.2.2. Analysis Results

CONTENTS OF THIS SECTION: This section describes the results of any quantitative or qualitative analyses that have been performed that provide evidence that the software architecture is fit for purpose. If an Architecture Tradeoff Analysis Method evaluation has been performed, it is included in the analysis sections of its final report. This section refers to the results of any other relevant trade studies, quantitative modeling, or other analysis results.

2.2.3. Requirements Coverage

CONTENTS OF THIS SECTION: This section describes the requirements (original or derived) addressed by the software architecture, with a short statement about where in the architecture each requirement is addressed.

3. Referenced Materials

CONTENTS OF THIS SECTION: This section provides citations for each reference document. Provide enough information so that a reader of the SAD can be reasonably expected to locate the document.

Barbacci 2003	Barbacci, M.; Ellison, R.; Lattanze, A.; Stafford, J.; Weinstock, C.; & Wood, W. <i>Quality Attribute Workshops (QAWs)</i> , Third Edition (CMU/SEI-2003-TR-016). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. < http://www.sei.cmu.edu/publications/documents/03.reports/03tr016.html >.
Bass 2003	Bass, Clements, Kazman, <i>Software Architecture in Practice</i> , second edition, Addison Wesley Longman, 2003.
Clements 2001	Clements, Kazman, Klein, <i>Evaluating Software Architectures: Methods and Case Studies</i> , Addison Wesley Longman, 2001.
Clements 2002	Clements, Bachmann, Bass, Garlan, Ivers, Little, Nord, Stafford, <i>Documenting Software Architectures: Views and Beyond</i> , Addison Wesley Longman, 2002.
IEEE 1471	ANSI/IEEE-1471-2000, <i>IEEE Recommended Practice for Architectural Description of Software-Intensive Systems</i> , 21 September 2000.

4. Directory

4.1. Index

CONTENTS OF THIS SECTION: This section provides an index of all element names, relation names, and property names. For each entry, the following are identified:

- the location in the SAD where it was defined
- each place it was used

Ideally, each entry will be a hyperlink so a reader can instantly navigate to the indicated location.

4.2. Glossary

CONTENTS OF THIS SECTION: This section provides a list of definitions of special terms and acronyms used in the SAD. If terms are used in the SAD that are also used in a parent SAD and the definition is different, this section explains why.

Term	Definition
software architecture	The structure or structures of that system, which comprise software elements, the externally visible properties of those elements, and the relationships among them [Bass 2003]. "Externally visible" properties refer to those assumptions other elements can make of an element, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on.
view	A representation of a whole system from the perspective of a related set of concerns [IEEE 1471]. A representation of a particular type of software architectural elements that occur in a system, their properties, and the relations among them. A view conforms to a defining viewpoint.
view packet	The smallest package of architectural documentation that could usefully be given to a stakeholder. The documentation of a view is composed of one or more view packets.

viewpoint	A specification of the conventions for constructing and using a view; a pattern or template from which to develop individual views by establishing the purposes and audience for a view, and the techniques for its creation and analysis [IEEE 1471]. Identifies the set of concerns to be addressed, and identifies the modeling techniques, evaluation techniques, consistency checking techniques, etc., used by any conforming view.
-----------	---

4.3. Acronym List

API	Application Programming Interface; Application Program Interface; Application Programmer Interface
ATAM	Architecture Tradeoff Analysis Method
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CORBA	Common object request broker architecture
COTS	Commercial-Off-The-Shelf
EPIC	Evolutionary Process for Integrating COTS-Based Systems
IEEE	Institute of Electrical and Electronics Engineers
KPA	Key Process Area
OO	Object Oriented
ORB	Object Request Broker
OS	Operating System
QAW	Quality Attribute Workshop
RUP	Rational Unified Process
SAD	Software Architecture Document
SDE	Software Development Environment
SEE	Software Engineering Environment
SEI	Software Engineering Institute Systems Engineering & Integration Software End Item
SEPG	Software Engineering Process Group
SLOC	Source Lines of Code
SW-CMM	Capability Maturity Model for Software
CMMI-SW	Capability Maturity Model Integrated - includes Software Engineering
UML	Unified Modeling Language

5. Sample Figures & Tables

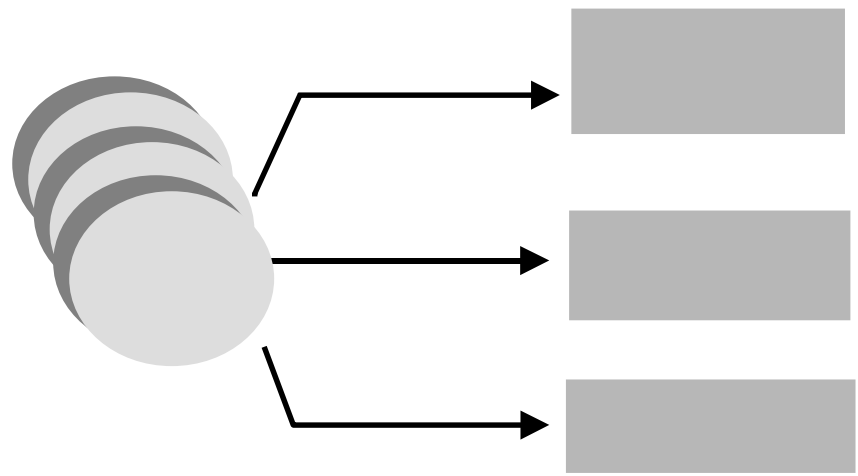


Figure 1: Sample Figure

Table 2: Sample Table

Table Heading	Table Heading	Table Heading	Table Heading
Table Body	Table Body	Table Body	Table Body
Table Body	Table Body	Table Body	Table Body
Table Body	Table Body	Table Body	Table Body
Table Body	Table Body	Table Body	Table Body

Appendix A Appendices

CONTENTS OF THIS SECTION: Appendices may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data, API specification). As applicable, each appendix is referenced in the main body of the document where the data would normally have been provided. Appendices may be bound as separate documents for ease in handling. If your SAD has no appendices, delete this page.

A.1 Heading 2 - Appendix

A.2 Heading 2 - Appendix