**ПРОПОЗИЦІЇ ЩОДО СТРУКТУРИ КУРСОВОЇ РОБОТИ З ООП**

**Загальні зауваження.**

1. Зазвичай більшість людей дуже уважно читають реферат, вступ, загальні питання, решту переглядають. Тому реферат, вступ, загальні питання бажано написати **ідеально**. Зміст повинен показувати, що всі питання в Вашій роботі висвітлені. Саме тому пропонується структура курсової роботи як перелік розділів, підрозділів та додатків, які повинні бути висвітлені в роботі і зафіксовані в змісті.
2. Вимоги до оформлення (шрифт, відступи та інші речі щодо зовнішнього подання) повинні відповідати наданим у методичці, яка викладена на гіт-хаб.
3. Запускайте перевірку граматики, граматичних помилок не повинно бути.
4. Замість програма можна використовувати термін програмне забезпечення (далі – ПЗ) або застосунок. Уникайте жаргону, прийнятого в ІТ-сфері. Наприклад, функціонал – це функціональні можливості, движок – це в залежності від контексту або програмна платформа, або ПЗ, що реалізує основні функціональні можливості, або інше формулювання "людською" мовою.
5. Встановлюйте розрив сторінки, якщо потрібно наступний текст почати з нової сторінки (замість "підгону" виведенням пустих рядків). Прибирайте зайві пусті рядки, вони псують зовнішній вигляд тексту.
6. Перелік скорочень, на мою думку, так виглядає привабливіше, якщо його помістити в таблицю без накреслених границь.

|  |  |  |
| --- | --- | --- |
| ІТ | - | Інформаційна технологія |
| ООП | - | об‘єктно-орієнтоване програмування |
| ПЗ | - | програмне забезпечення |

1. Тема у більшості студентів " Розробка застосунку … з використанням принципів об’єктно-орієнтоване програмування ", тому всі акценти повинні бути на аспектах ООП, використаних у курсові. Кінцева мета курсової – опис процесу розроблення та працюючий застосунок.
2. **Звертаю увагу, що вам потрібно здати друкований варіант курсової. Доцільно скріншоти надавати на білому фоні. Це візуально більше вигідно, а також зменшує витрати чорної фарби.**

**СТРУКТУРА КУРСОВОЇ РОБОТИ**

Коментарі та пояснення надаються під назвою розділу/підрозділу курсивом. Назва розділу/підрозділу написана жирним шрифтом. Розділи/підрозділи, що не містять коментарів виконуються за методичкою.

**Титульний лист (**підписується в друкованому примірнику)

**Завдання на курсову** (сторінка підкладається в друкованому примірнику)

**Календарний план-графік** виконання курсової роботи з дисципліни «Вступ до об‘єктно – орієнтованого програмування» (сторінка підписується в друкованому примірнику)

**Зміст** (аркуш з широким колонтитулом)

*Формується автозбиранням*

**Реферат**

*Доцільно виклад побудувати за схемою:*

*Робота присвячена розробці застосунку <за змістом теми>. В курсовій роботі розглянуті питання розроблення застосунку методами об‘єктно – орієнтованого програмування на мові С++. Об’єктом розроблення є <за змістом теми, наприклад, системи обліку …,інформаційно-пошукової системи >. Мета роботи показати формування проектних рішень щодо <за змістом теми>. Використовувались сучасні методи та інструменти проектування та розроблення ПЗ, зокрема, … (наприклад, об’єктно –орієнтований підхід з використанням інструментарію Microsoft Visual Studio та засоби проектування Rational Unified Process або інша CASE-система….).*

*Далі виклад по методичці.*

**Перелік скорочень**

*Краще оформити таблицею без границь (3 графи – скорочення, "-", пояснення), буде акуратніше і привабливіше.*

**Вступ**………

*Обов’язкова частина:*

*Тема курсової роботи <назва теми за наказом>.*

*Мета курсової роботи: розробити застосунок <за змістом теми>.*

*Завдання курсової роботи: <за змістом теми> викласти в послідовності вивчити досліджувані проблеми на основі опрацьованих наукових та навчальних джерел → дослідити <об’єкт за змістом теми> → визначити проблеми → проаналізувати існуючі рішення → розробити. Відповідно, кожну позицію потрібно коротко розкрити. Формулювання у методичці.*

*Об’єкт розроблення: <за змістом теми, наприклад, система обліку на підприємстві, система оцінювання знань в навчальному закладі> для впровадження автоматизації <за змістом теми> за допомогою програмного забезпечення.*

*Запропоновані рішення пропонується застосовувати на <за змістом теми, наприклад, підприємствах, в навчальних закладах, агентствах нерухомості >. Далі розкриваєте переваги: наприклад, ПЗ дозволить впорядкувати інформаційну базу, що, в свою чергу, дозволить якісно підвищити рівень надання послуг користувачу …. Переваг доцільно дати дві-три. Щодо того, які переваги зорієнтуйтесь по своїй предметній області.*

*Надається перелік інструментів та засобів розроблення та оформлення проекту (компілятор, засоби моделювання тощо).*

*Коли виконується колективна розробка, то тут надається склад команди та розподіл робіт по виконанню курсового проекту.*

**РОЗДІЛ 1 ОПИС НАПРЯМУ РОЗРОБЛЕННЯ**

* 1. **Постановка задачі**

*Виконується за методичкою.*

**1.2 Опис предметної області об‘єкту**

*Викладається за методичкою. Показується, що являють собою інформаційні процеси прикладної предметної області, де буде застосовуватися ПЗ.*

* 1. **Огляд наукової літератури**

*Виконується за методичкою. Застосовуються такі підходи.*

Огляд наукової літератури зазвичай робиться за такою схемою.

У першій частині дається коротка характеристика теми, розкривається її значення, важливість. Після цього необхідно коротко охарактеризувати літературу, що розглядається, і зробити певні висновки. Потрібно розглянути три групи джерел: 1) звідки брались відомості про предметну область; 2) методологія об‘єктно – орієнтованого підходу та аспекти об‘єктно – орієнтованого програмування; 3) відомості про розроблення UML- діаграм.

Послідовність написання огляду літератури:

1. Прочитати зміст і поверхово переглянути роботу.

2. Виділити найважливіші частини тексту під час уважного прочитання джерела.

3. Скласти тезовий план прочитаного матеріалу, у пунктах якого зазначити найважливіші думки та ідеї.

4. Виписати з тексту повні та змістовні цитати з точними посиланнями на джерело, вказавши його вихідні дані.

5. Порівняти дану інформації з тією, що одержана з інших джерел.

6. Оформити всю одержану інформацію в єдиний текст, критично оцінивши прочитане.

Всі літературні джерела, на які є посилання в огляді, повинні бути включені в перелік посилань і пронумеровані. У курсовій роботі треба робити посилання, яке містить тільки номер публікації за списком. Воно береться у квадратні дужки. Наприклад: [12], [15], [1-5], [5,12, 15].

Наприкінці «Огляду літератури» доцільно зробити висновки. Вони необхідні для порівняння з результатами власного дослідження.

Приклад непоганого огляду з попередніх курсових робіт.

Джерельна база дослідження складається, переважно, з наукових праць вчених в галузі програмної інженерії, яка є сьогодні ключовою галуззю.

Дослідження процесу інформаційно-пошукової системи, її структури, складових частин знайшли відображення у працях І. Соммервіллема (автора книги «Інженерія програмного забезпечення) [14], В. Коцовського (розкрив поняття архітектури життєвого циклу програмного забезпечення, описав моделі життєвого циклу ПЗ та визначив особливості та необхідність супроводу ПЗ в книзі «Супровід програмних систем») [22] Г. В. Табунщика (визначив фази проектування ПЗ та необхідні процеси пов’язані з кожною фазою)[15].

Стан і потреби сучасної бібліотечної справи детально розглядаються в статті Анастасії Куркіної «Бібліотека – це тисячі книжок, журналів і бібліотек»[19]. З погляду на надані відомості розглядалися можливості подальшого розвитку ПЗ і, відповідно, з цією перспективою проводилась розробка.

Отже, наведені праці допоможуть краще розібратися з процесом проектування інформаційно-пошукової системи «Електронна бібліотека».

**1.4 Огляд існуючих програмних засобів**…

*На початку потрібно написати, що на ринку існує ПЗ для розроблення програмного застосунку мовою С++, надаєте перелік через кому. Надається оцінка та визначаються причини з яких обирається засіб розроблення. Аналогічно розглядаються CASE-засоби, якими розроблялися діаграми курсової.*

**РОЗДІЛ 2. ОПИС ПРОЕКТНОГО РІШЕННЯ**

**2.1 Алгоритм рішення**

*Викладається відповідно до постановки задачі. Надається блок-схема алгоритму з детальним описом роботи всіх блоків алгоритму та надається технологічна схема (див. методичку).*

**2.2 Модель програмного рішення**

*Надається UML - діаграма класів з відповідним текстовим описом (призначення класу, місце в застосунку тощо) та поясненням щодо зв‘язку між класами.*

*Під діаграмою надати пояснення щодо призначення класів у вигляді*

Таблиця … - Призначення класів застосунку

|  |  |
| --- | --- |
| Назва класу | Призначення класу |
|  |  |
|  |  |

*Призначення методів та членів класів повинно надаватися в коментарях інтерфейсу класу в тексті коду. Доцільно надати посилання на додаток та програмну одиницю, де знаходиться інтерфейс класу.*

**2.3 Ієрархічно-функціональна схема застосунку**

*Формується за прикладом наданим в методичці.*

**2.4 Методи та механізми ООП, використані в застосунку**

*В цьому підрозділі обгрунтовується використання тих механізмів та методів, які використовуються в застосунку. Зокрема, у всіх буде проведене розділення інтерфейсу та реалізації, будуть використані певні типи конструкторів, використані аксессори - гетери та сетери, зв‘язки між класами. Окрім того кожен буде використовувати інші методи та механізми. Тут потрібно надати короткий опис методу / механізму та причину його використання у застосунку.*

**2.5 Опис програмного інтерфейсу**

Виконується за методичкою, тут надається схема меню та відповідні пояснення.

**2.6 Тестування програмного застосунку і результати її виконання**

Виконується за методичкою. Доцільно надати скріншот прикладу тестування.

**2.7 Керівництво для користувача програми**

Виконується за методичкою. Детальний опис надається в Додатку 3.

**2.8 Організація розроблення програмного застосунку**

Виконується за методичкою. План надається у вигляді таблиці.

|  |  |  |  |
| --- | --- | --- | --- |
| Номер  етапу | Дії, які виконується | Кількість  днів | Період  виконання |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Висновки**

Виконується за методичкою.

**Перелік використаних джерел**

Виконується за методичкою.

**Додаток 1 Тексти програми**

Виконується за методичкою. При командній роботі можна власний текст виділити жирним шрифтом, надавши відповідне пояснення.

**Додаток 2 Результати виконання програмного застосунку**

Надаються скріншоти виконання застосунку з відповідними поясненнями.

**Додаток 3 Керівництво для користувача програми**

Коротко викладаються такі відомості.

1. Призначення застосунку
2. Умови застосування

Тут – вимоги до апаратно-програмної платформи, операційної системи.

1. Установка застосунку у користувача
2. Реалізація технологічної схеми

Описується стандартний сценарій роботи.

1. Оброблення нештатних ситуацій

Надаються повідомлення програми у вигляді таблиці

|  |  |  |  |
| --- | --- | --- | --- |
| Номер | Текст повідомлення | Причина повідомлення | Дії користувача |
|  |  |  |  |
|  |  |  |  |

Нижче надаються загальновживані вимоги до Керівництва користувача. Можете використати ці настанови для власного проекту.

**Стандартні вимоги до Керівництва користувача**

Керівництво користувача складають на підставі наступних регламентуючих документів: ГОСТ 34.201-89, РД 50-34.698-90, IEEE 1163-2001

Керівництво користувача - один з основних програмних документів, основне завдання якого полягає в забезпеченні користувачам можливістю самостійно вирішувати основні завдання, на які націлена програма.

Керівництво користувача містить повний опис програми з точки зору цільового застосування останньої, а саме:

- Призначення програми;

- Основні завдання і можливості;

- Спосіб відображення предметної області в програмі;

- Призначений для користувача інтерфейс програми;

- Порядок вирішення основних користувальницьких завдань;

- Всі функції програми і порядок їх застосування;

- Призначена для користувача настройка програми;

- Проблеми при використанні і способи їх вирішення.

При документуванні невеликих програм в керівництво користувача часто включають інструкції з установки, настройки, адміністрування, оновлення та іншого обслуговування програми.

Залежно від особливостей програми і цільової аудиторії керівництво користувача за способом викладу матеріалу може наближатися до підручника або до довідника. Порядок викладу матеріалу в керівництві користувача визначається користувальницької перспективою програми.

Якщо програма являє собою інструмент, що дозволяє вирішувати практичні завдання з деякого кінцевого набору, в керівництві призводять типові процедури вирішення кожного з них.

Наприклад, користувачеві поштового клієнта необхідно знати, як написати і відправити повідомлення, як завантажити нові повідомлення з сервера, як відповісти на повідомлення і т.д. Кожне з цих дій можна розкласти на послідовні елементарні кроки. У велику програму подібних призначених для користувача завдань може бути багато, але не нескінченно. Керівництво користувача, побудоване за принципом призначених для користувача завдань, нагадує підручник, хоча, як правило, позбавлене властивого підручниками методичного апарату: перевірочних завдань, питань, вправ.

Якщо програма являє собою середовище, в межах якої користувач може вирішувати завдання, поставлені їм самостійно, керівництво користувача повинно бути ближче до довідника. У ньому послідовно і систематично повинні бути описані всі функції програми і порядок їх застосування. Так, в керівництві користувача з графічного редактору ми знайдемо опис всіх графічних примітивів, інструментів, фільтрів, проте, там не буде безпосередньо сказано, як зобразити той чи інший предмет.

Можливі й інші призначені для користувача перспективи. Так, в програмах, за допомогою яких користувач контролює стан того чи іншого об'єкта (нехай промислової установки) керівництво користувача будується за принципом таблиці: повідомлення програми - реакція або можливі реакції користувача.

Якщо користувач застосовує програму для вирішення завдань в нетривіальних предметних областях, в керівництво рекомендується включити концептуальний розділ. У ньому повинен бути описаний реалізований в програмі спосіб представлення об'єктів реального світу, щоб користувач добре розумів, з якими з них і на якому рівні абстракції він може працювати.

Незважаючи на те, що в кожному конкретному випадку структура керівництва користувача визначається особливостями описуваної програми, зазвичай вона має вигляд:

1. Загальні відомості.

2. Встановлення та первісна настройка.

3. Основні поняття та визначення.

4. Інтерфейс користувача.

5. Робота з програмою.

6. Користувацький.

7. Повідомлення про помилки.