Міністерство освіти та науки України

Національний університет «Львівська політехніка»

Кафедра систем автоматизованого проектування



**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до курсової роботи з навчальної дисципліни

«Розробка кросплатформенних додатків (Python)»

на тему «Вказати тему курсової роботи»

Виконав:

ст. гр. ПП-2x Іваненко Іван

Перевірив:

проф. каф. САП Щербовських С. В.

доцент каф. САП Оборська О. В.

доцент каф. САП Стефанович Т. О.

асистент каф. САП Черна Т. І.

(залишити ОДНЕ прізвище)

Львів – 2025

# Розділ 1. Аналіз проблеми та підготовка до вирішення [20 балів]

|  |
| --- |
| **Увага!**  Результатом курсової роботи є розроблення програмного забезпечення для вирішення проблемної ситуації, яку студент формулює самостійно. Програма має бути написана мовою Python, складатися з одного файлу і працювати на пристроях користувача без необхідності встановлення додаткових програм чи додатків.  Під час виконання курсової роботи обов’язково дотримуйтесь поданої структури.   * Текст та зображення вставляйте тільки у білі поля. * Текст у зафарбованих полях змінювати або видаляти заборонено.   Якщо ці вимоги не будуть виконані, то робота **не перевірятиметься**. |

## 1.1. Визначення проблемної ситуації [5 балів]

|  |
| --- |
| Описати проблемну ситуацію, розкриваючи такі складові:   * **Об’єкт:** Хто або що зазнає впливу проблемної ситуації. Це може бути людина, група осіб, організація, довкілля тощо. * **Суб’єкт:** Хто або що є причиною виникнення проблемної ситуації. Це може бути конкретна особа, організація, природний процес або технічний фактор. * **Об’єктивна сторона:** Описати небезпеку або наслідки проблемної ситуації для об’єкта. Як дії суб’єкта спричиняють ці наслідки. Також зазначити, де, коли і за яких обставин виникла проблемна ситуація. Пояснити, як саме відбувається вплив і що є безпосереднім носієм цього впливу (наприклад, дія певного механізму, рішення особи або природне явище). * **Суб’єктивна сторона:** Пояснити, чому суб’єкт винен у виникненні ситуації. Яка у нього мета, мотиви та емоційний стан (опишіть лише ті характеристики, які можна застосувати до суб’єкта в даній ситуації).   Вимога до обсягу тексту: **не менше 1000 символів.**  Опис має бути детальним з чіткім поясненням всіх елементів проблемної ситуації та їх взаємозв’язку. |
|  |

## 1.2. Потреби та цілі зацікавлених сторін [5 балів]

|  |  |
| --- | --- |
| Визначте **не менше п’яти зацікавлених сторін**. Це можуть бути особи, групи або організації, які мають інтерес у вирішенні проблеми або впливають на її результат.  Для кожної зацікавленої сторони:   * Описати цілі та потреби цієї сторони (що вона хоче досягти в результаті вирішення проблеми). * Визначити роль цієї сторони у проблемній ситуації (як вона пов’язана із проблемною ситуацією і який вплив має на її вирішення).   Зацікавлені сторони можуть бути активними учасниками проблемної ситуації або тими, хто лише стикається з її наслідками.  Кожна зацікавлена сторона має свої інтереси, цілі та мотивацію, які можуть бути різними.  Усі зацікавлені сторони повинні бути згадані у п. 1.1. | |
| **Назва зацікавленої сторони** | **Опис цілей та потреб зацікавленої сторони. Роль зацікавленої сторони у проблемній ситуації** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 1.3. Формулювання мети роботи [5 балів]

|  |
| --- |
| Сформулювати мету роботи, дотримуючись принципів SMART:   * **S (Specific)**: Мета повинна бути чіткою і зрозумілою. Вона повинна описувати, що саме потрібно досягти, без загальних або нечітких формулювань. * **M (Measurable)**: Мету слід формулювати так, щоб можна було оцінити її досягнення. Це може бути через кількісні (наприклад, кількість рядків коду) або якісні показники (наприклад, поліпшення ефективності програми). * **A (Achievable)**: Мета має бути реалістичною. Врахуйте доступні ресурси (час, знання, інструменти) і обмеження, щоб результат був досяжним. * **R (Relevant)**: Мета повинна мати практичну цінність і важливість для вирішення проблеми. Вона повинна мотивувати до виконання роботи і нести конкретну користь зацікавленим сторонам. * **T (Time-bound)**: Мета повинна містити чіткі часові обмеження. Це дозволить ефективно планувати виконання роботи і відслідковувати прогрес.   Обґрунтуйте, чому запропонована мета відповідає кожному з цих принципів. |
|  |

## 1.4. Аналіз існуючих рішень та огляд літератури [5 балів]

|  |  |  |
| --- | --- | --- |
| Виберіть **не менше п’яти літературних джерел**, які стосуються теми вашої курсової роботи. Це можуть бути наукові статті, книги, дослідження або інші ресурси, які допоможуть розкрити тему.  Для кожного джерела:   * Складіть короткий опис. В цьому описі потрібно проаналізувати які аналоги, прототипи, ідеї чи результати описані в джерелі. Після опису обов’язково додати посилання на джерело. * Визначте основні недоліки запропонованих рішень або невирішені питання, які стосуються вашої теми.   В результаті огляду потрібно зрозуміти існуючі підходи до вирішення вашої проблеми. Виявити їхні слабкі сторони та обґрунтувати, чому необхідно виконати саме ваше дослідження або розробку.  **Важливо!** Не використовуйте підручники та технічну документацію з мови програмування Python. Використовуйте лише наукові статті, книги або інші джерела, які **безпосередньо стосуються вашої теми**. | | |
| **№** | **Короткий опис аналогів, прототипів, ідей або результатів. Посилання на літературне джерело** | **Виявлені недоліки** |
| [1] |  |  |
| [2] |  |  |
| [3] |  |  |
| [4] |  |  |
| [5] |  |  |

# Розділ 2. Розробка програмного забезпечення [25 балів]

## 2.1. Програмний код мовою Python [5 балів]

|  |
| --- |
| Подати лістинг програми мовою Python.  Дотримуйтесь стандарту кодування PEP 8, зокрема, використовуйте відступи, правильні назви змінних, форматування і коментарі відповідно до стандарту.  Вимоги до структури програми:   * програма має містити не менше трьох класів, * сумарна кількість атрибутів і методів у всіх класах повинна бути не менше десяти, * код має бути зрозумілим і прокоментованим.   **Увага!** Подавайте тільки коректний та протестований код.  Див. PEP 8 «Style Guide for Python Code»: <https://www.python.org/dev/peps/pep-0008/> |
|  |

## 2.2. Архітектури програми [5 балів]

|  |
| --- |
| Опишіть архітектуру програми, пояснивши її структуру. Вкажіть, які класи та методи створено, чому було обрано саме таке розбиття, і як це допомагає досягти поставлених цілей. Зверніть увагу на логіку взаємодії між класами та методами, щоб показати, як кожен елемент програми підтримує її загальну функціональність.  Розробіть UML-діаграму класів для наочного представлення структури програми (відповідно до вимог п. 2.1). На діаграмі зазначте: назви класів, атрибути (з обов’язковим зазначенням типу, наприклад: `int`, `str`, `list` тощо) та методи кожного класу.  Діаграма має бути зрозумілою, точною та оформленою за стандартами UML. Це дозволить легко сприйняти логіку побудови програми та взаємодію між її компонентами. |
| C:\Users\Serhiy\AppData\Local\Microsoft\Windows\INetCache\Content.Word\pointAssocv3.png |

## 2.3. Опис класів, методів та атрибутів [5 балів]

|  |  |  |
| --- | --- | --- |
| Описати кожен клас, метод та атрибут, які ви подали на діаграмі класів у п. 2.2.  Для кожного класу поясніть його логіку (що саме робить цей клас, яку задачу вирішує).  Для кожного методу опишіть, що саме він виконує, які вхідні параметри та результат роботи.  Для кожного атрибута опишіть, що саме він зберігає та як використовується в класі.  Опис має бути чітким і зрозумілим, щоб можна було легко зрозуміти, як працюють класи і їхні елементи. | | |
| **Клас** | **Імена атрибутів та методів** | **Опис атрибутів та методів** |
| Point | x: int | Атрибут, який містить координату x заданої точки |
| y: int | Атрибут, який містить координату y заданої точки |
| distanceFromOrigin() | Метод, який визначає відстань між заданою точкою та початком координат. В основу методу покладено алгоритм… |
|  |  |  |

## 2.4. Перевірка та тестування програми [5 балів]

|  |
| --- |
| Перевірте всі методи програми, щоб переконатися, що вони повертають правильні результати за різних умов, включаючи крайні випадки. Оцініть повноту тестування, перевіривши всі методи на різні сценарії, зокрема стандартні, неочікувані та крайні. Проведіть тестування функціональності (чи виконуються завдання програми), адаптивності (чи працює програма з різними вхідними даними), зручності (чи легко користуватися програмою) та відповідності стандарту PEP 8 (чи дотримано правила форматування, таких як назви змінних, відступи та структура коду). Опишіть результати тестів, зазначивши, які методи працюють коректно, а які потребують виправлення, і поясніть зроблені виправлення. На основі тестування дайте оцінку якості програми, визначивши, чи відповідає вона заявленим вимогам і очікуванням. |
|  |

## 2.5. Інструкція користувача [5 балів]

|  |
| --- |
| Розробіть інструкцію користувача, дотримуючись такої структури:  **1. Вступ**  *1.1. Короткий опис програми та її мети.*  Опишіть, що робить програма та для чого вона створена. Наприклад: «Програма дозволяє розраховувати статистичні показники з набору даних»..  *1.2. Цільова аудиторія.*  Вкажіть, хто буде використовувати програму: студенти, професіонали, звичайні користувачі або інші категорії.  **2. Основні функції**  *2.1. Опис основних функцій програми.*  Перерахуйте та поясніть основні можливості програми. Наприклад: "Розрахунок середнього арифметичного, побудова графіків».  *2.2. Покрокові інструкції з використання кожної функції.*  Надайте зрозумілі інструкції, як використовувати кожну функцію програми, крок за кроком.  *2.3. Приклади використання програми.*  Наведіть реальні приклади застосування програми. Наприклад: «Як порахувати середнє арифметичне або створити графік із вхідних даних».  **3. Налаштування**  *3.1. Огляд доступних налаштувань програми.*  Опишіть усі доступні налаштування програми, наприклад, зміну мови, налаштування кольорів інтерфейсу або шрифтів.  *3.2. Як налаштувати програму.*  Дайте покрокову інструкцію з налаштування програми відповідно до потреб користувача.  Інструкція повинна бути написана простою мовою, включати чіткі пояснення та, за потреби, ілюстрації або скріншоти для кращого розуміння. |
|  |

# Розділ 3. Аналіз результатів роботи [15 балів]

## 3.1. Демонстрація виконання програми [5 балів]

|  |
| --- |
| Показати приклади реального використання вашої програми:   * подайте скріншоти екрана, де видно, як програма працює, * опишіть, які операції ви виконували під час демонстрації та які результати отримали, * поясніть, як саме програма виконувала кожен крок і що отримали в результаті.   Скріншоти мають бути чіткими, з видимим інтерфейсом програми та результатами її роботи.  Опис має бути зрозумілим і детальним, щоб показати, як програма працює на практиці. |
|  |

## 3.2. Оцінка досягнення поставлених цілей [5 балів]

|  |  |
| --- | --- |
| Пояснити, як результати вашої роботи відповідають цілям та потребам зацікавлених сторін, визначених у п. 1.2.  Подати якісну оцінку ступеня задоволення зацікавлених сторін. Це означає, що потрібно описати, чи були досягнуті поставлені цілі і чи задоволені вимоги, які ставили ці сторони. Оцінка має бути обґрунтованою, з конкретними прикладами того, як ваші результати відповідають вимогам.  Запропонуйте напрямки покращення програми. | |
| **Назва зацікавленої сторони** | **Опис та оцінка досягнення цілей та потреб зацікавленої сторони** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 3.3. SWOT-аналіз результатів розробки [5 балів]

|  |  |
| --- | --- |
| Виконайте SWOT-аналіз ваших результатів. Такий аналіз повинен оцінити:  **S (Strengths)** – сильні сторони (переваги вашої роботи).  **W (Weaknesses)** – слабкі сторони (недоліки або обмеження вашої роботи).  **O (Opportunities)** – можливості для покращення (як можна вдосконалити роботу або розширити її можливості).  **T (Threats)** – загрози (ризики або проблеми, які можуть виникнути в майбутньому).  Під час аналізу використати оцінку ступеня задоволеності зацікавлених сторін, яку ви описали в п. 3.2.  Сильні та слабкі сторони повинні бути конкретними і базуватися на реальних результатах.  Можливості та загрози мають вказувати на те, як можна покращити роботу чи які проблеми можуть виникнути в майбутньому. Оцінка повинна бути обґрунтованою та логічною. | |
| **Зацікавлені сторони, які задіяні у проб­лемній ситуації БЕЗПОСЕРЕДНЬО** | **Зацікавлені сторони, які задіяні у проб­лемній ситуації ОПОСЕРЕДКОВАНО** |
| **Сильні сторони** | **Можливості** |
|  |  |
| **Слабкі сторони** | **Загрози** |
|  |  |

# Додаток А. Самоаналіз оцінювання роботи [10 балів]

|  |  |  |
| --- | --- | --- |
| **Структурний елемент пояснювальної записки** | **Максимальна оцінка** | **Самооцінка** |
| 1.1. Визначення проблемної ситуації | 5 |  |
| 1.2. Потреби та цілі зацікавлених сторін | 5 |  |
| 1.3. Формулювання мети роботи | 5 |  |
| 1.4. Аналіз існуючих рішень та огляд літератури | 5 |  |
| 2.1. Програмний код мовою Python | 5 |  |
| 2.2. Архітектура програми | 5 |  |
| 2.3. Опис класів, методів та атрибутів | 5 |  |
| 2.4. Перевірка та тестування програми | 5 |  |
| 2.5. Інструкція користувача | 5 |  |
| 3.1. Демонстрація виконання програми | 5 |  |
| 3.2. Оцінка досягнення поставлених цілей | 5 |  |
| 3.3. SWOT-аналіз результатів розробки | 5 |  |
| Додаток А. Самоаналіз оцінювання роботи | 10 |  |
| **Всього:** | **70** |  |