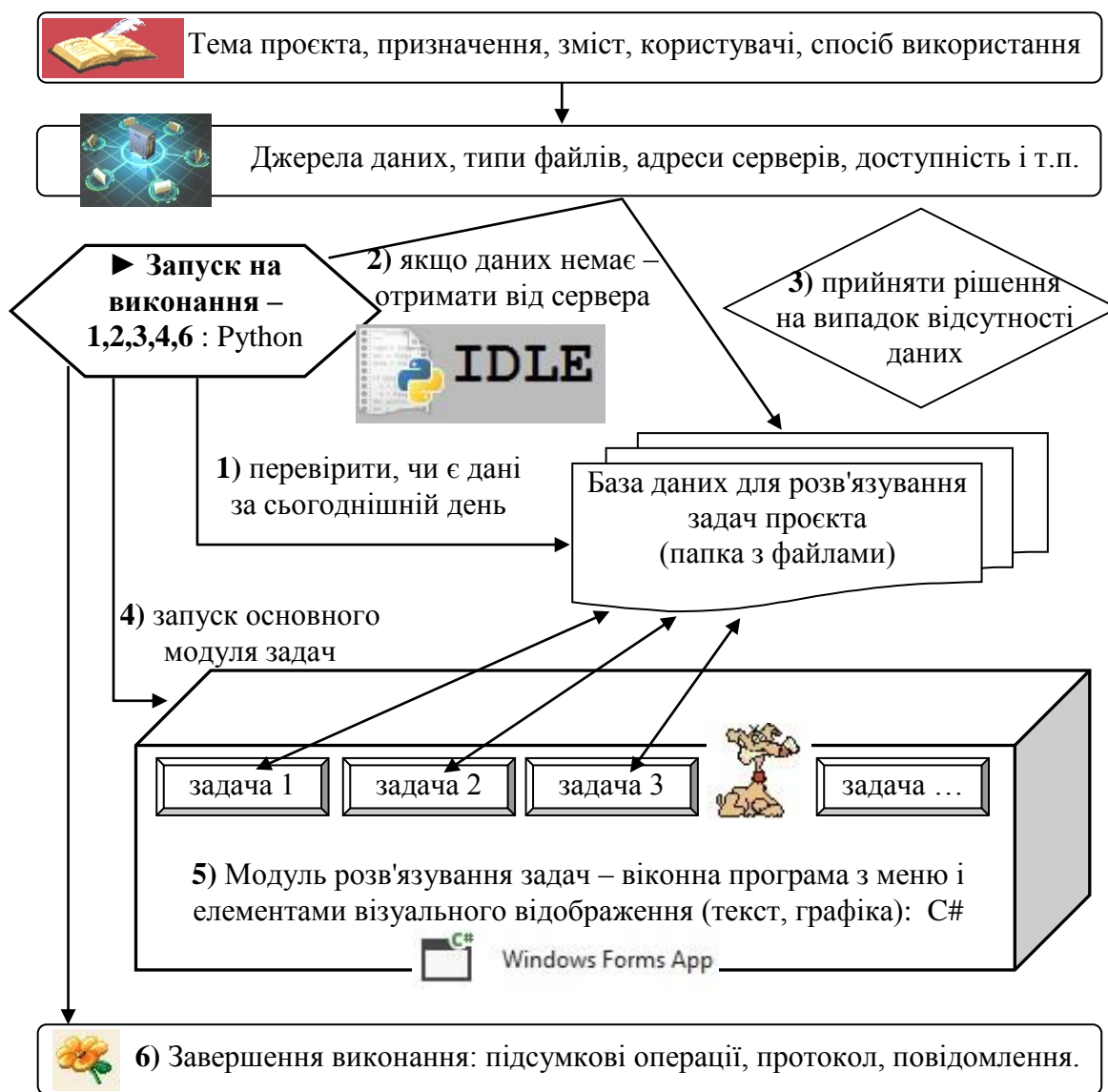


Комплексна модель програмного проєкта

Це завдання є підсумковим на основі декількох останніх вивчених тем. Метою завдання є напрацювання навиків комплексної розробки окремих складових частин програмних проєктів певного цільового призначення.

За основу свого проєкта можна початково прийняти архітектуру, описану в файлі матеріалів "Модель проєкта за даними JSON.pdf" в розділі "Постановка задачі". Зразу підкреслимо, що можна будувати проєкт на основі файлів й інших типів, наприклад, xml. Приклад архітектури проєкта в графічному зображенні показано на малюнку.



Зміст завдання:

- обрати тему, визначити загальний зміст і перелік задач, зафіксувати умови застосування (операційна система, необхідна пам'ять, сторонні програмні засоби);
- визначити джерела постачання даних для виконання задач проєкта; зафіксувати адреси серверів чи інші, звідки отримувати дані;
- загальний сценарій проєкта будувати мовою Python; елементи сценарія показані в файлі "Модель проєкта за даними JSON.pdf";
- архітектура проєкта не зобов'язана бути такою як на малюнку; можна запропонувати свою архітектуру, ініціатива заохочується;
- реалізувати просту схему виконання проєкта: накопичувати в базі за кожен день отримані дані від сервера (папка з файлами), перевіряючи, чи сьогодні вже отримали їх; після цього можна побудувати мовою Python розв'язки 5-6 наперед фіксованих задач і зберегти в файлах, або перебудувати файли з даними до іншого формату для розв'язування мовою C#; передати керування модулю розв'язування задач, який просто читає з файлів розв'язки і показує їх в текстовій і/чи графічній формі, або додатково сам розв'язує задачі; (наш проєкт є лише моделлю, а не цілим реальним проєктом); після закінчення роботи модуль розв'язування задач повертає керування основному сценарію Python, який завершує виконання;
- запропонований варіант архітектури проєкта передбачає реалізацію різних частин мовами Python і C#; *модуль розв'язування задач* має показати результати в текстовій і графічній формах, для цього потрібні віконні елементи відображення, які реалізовані в класах C#; функціонально аналогічні елементи є так само в мові Python в модулі tkinter (Interface to Tcl/Tk for graphical user interfaces), але обсяг нашого курсу не надає можливості для окремого вивчення, це можна зробити самостійно;
- якщо прикладна програма на C# вважається остаточно налагодженою (в нашому випадку це модуль розв'язування задач), будують фінальний випуск програми Release:
Project→Properties→Property Pages→Configuration→Release;
після повернення до головного вікна на панелі інструментів обираємо Solution Configuration→Release;
повторно компілюємо Build→Compile і компонуємо програму Build→Build Solution;
має з'явитись папка Release з EXE-файлом та іншими файлами, цю папку зазвичай передають користувачам нашої програми, в нашому випадку це буде просто частина проєкта.

Надіслати у відповідь:

- 1) знайдені в інтернет веб-адреси даних для формату json або xml, використані для виконання завдання;
- 2) файл головного сценарію мовою Python – кроки 1,2,3,4,6;
- 3) папку Release, побудовану мовою C#, з exe-файлом модуля розв'язування задач і решти файлів (dll, інші) – крок 5; файлів з текстами програм не треба;
- 4) коротку інструкцію до використання – звичайний текстовий файл;
- 5) 1-2 скрін-шоти виконання на своєму комп'ютері.

Це заключна робота нашого курсу. Термін виконання проєкта продовжений. Вона велика за обсягом, тому можна її виконати командою в складі 2 або 3 студенти. Команди можна створити ті самі, що й були на одному з попередніх проєктів. Повідомити через чат склад команди – прізвище, ім'я, академічна група. В надісланому звіті повідомити про розподіл роботи між членами команди. Незалежно від розподілу всі члени команди отримують однакову оцінку.