**Теоретичне завдання**

1. **Що таке рефлексія?**

Рефлексія − це механізм, що дозволяє отримувати інформацію про типи даних використовуючи внутрішні властивості цих типів. Рефлексія забезпечує взаємодію зі збіркою на етапі виконання програми. Рефлексія дає можливість отримувати доступ до метаданих типів, таких як поля, властивості, методи, конструктори, події, інтерфейси та інше, і змінювати їх властивості і поведінку під час виконання програми.

1. **Для чого використовується Reflection?**

Reflection дає можливість отримувати доступ до метаданих типів. Отримання інформації про типи виконання: Reflection дає змогу дізнатися про типи об'єктів виконання, їх атрибути, властивості, методи, події і т.д.

Reflection дозволяє створювати об'єкти типу виконання за допомогою конструкторів і викликати методи цих об'єктів в режимі виконання, дозволяє динамічно завантажувати і взаємодіяти з компонентами, що визначаються в режимі виконання, дозволяє визначити типи об'єктів виконання в режимі виконання і вирішувати, який метод чи властивість слід використовувати в залежності від зовнішніх умов та дозволяє виводити інформацію про структуру типів даних виконання, що може бути корисним в процесі відлагодження програм.

1. **Чи можна працювати за допомогою Reflection з компілюваними збірками?**

Reflection дозволяє працювати з компільованими збірками, включаючи збірки, які написані на інших мовах програмування і зібрані для виконання на .NET Framework або .NET Core.

1. **Який namespace у Reflection?**

Класи, пов'язані з Reflection, містяться в просторі імен System.Reflection.

1. **Якщо збірка містить поле/метод з модифікатором доступу private, чи маємо ми можливість працювати з полем/методом за допомогою Reflection?**

За допомогою Reflection можна працювати з полями та методами, незалежно від їхнього модифікатора доступу, включаючи private. Для того щоб отримати доступ до private поля або методу, потрібно використовувати параметр BindingFlags.NonPublic під час виклику методів класу Type.