**Контрольні питання**

**1. Опишіть структуру C# програми.**

Типова програма на C# складається з директив using, які імпортують необхідні простори імен (наприклад, System), простору імен (namespace), що служить для організації коду та запобігання конфліктам імен, та одного або кількох класів (class), що є основними будівельними блоками, які містять дані (поля) та поведінку (методи). Обов'язковою точкою входу для виконання програми є статичний метод Main(), з якого починається робота всієї програми.

**2. Із якого рядка починає працювати програмний код.**

Програмний код у C# починає виконуватися з першого рядка методу Main(). Цей метод є спеціальною точкою входу, яку компілятор C# шукає для ініціалізації та запуску програми.

**3. Опишіть структуру типів даних. Чим відрізняються типи значень від типів посилань?**

У C# типи даних поділяються на типи значень та типи посилань. Типи значень (наприклад, int, bool, struct) зберігають свої дані безпосередньо у пам'яті, де оголошена змінна (зазвичай у стеку), і при присвоєнні або передачі створюється повна копія цих даних. Натомість, типи посилань (наприклад, class, string, array) зберігають у змінній лише адресу (посилання) на об'єкт, який розташований у купі (heap), а при копіюванні чи передачі копіюється лише це посилання, що дозволяє кільком змінним посилатися на один і той самий об'єкт, і зміни, внесені через одну змінну, відображаються через інші.

**4. Опишіть структуру умовного оператора. Чим відрізняється повна та скорочена форма умовного оператора? У якому випадку використовується оператор вибору?**

Умовний оператор if виконує блок коду, якщо вказана умова істинна, може бути доповнений else if для перевірки додаткових умов і необов'язковим else для виконання коду, якщо жодна з попередніх умов не є істинною. Повна форма if-else надає два шляхи виконання (для істинної та хибної умови), тоді як скорочена форма if без else виконує код лише за умови істинності, пропускаючи блок, якщо умова хибна. Оператор вибору (switch) використовується, коли потрібно порівняти одну змінну з кількома фіксованими значеннями, пропонуючи чистішу та ефективнішу альтернативу довгим ланцюжкам if-else if для багатьох дискретних випадків.

**5. Чи може оператор else бути самостійним? У якому випадку використовується тернарна операція?**

Ні, оператор else не може бути самостійним; він завжди повинен слідувати за оператором if або else if, вказуючи блок коду для виконання, якщо попередня умова виявилась хибною. Тернарна операція (умовний оператор ?:) використовується для компактного запису простого умовного виразу, що повертає одне з двох значень залежно від істинності умови, і найчастіше застосовується для присвоєння змінній значення або безпосереднього використання значення у виразі на основі певної умови.