Тема 1. Основні поняття і принципи об’єктно-орієнтованого аналізу та проектування

1. **Абстрагування.**
2. **Наслідування.**
3. **Інкапсуляція.**
4. **Поліморфізм.**

Рекомендована література:

Базова

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е изд.: Пер. с англ. / Г. Буч, Р. А. Максимчук, М. У. Энг и др. – М.: Вильямс, 2008. – 720 с.
2. Брила А. Ю. Основи об’єктно-орієнтованого програмування у C#. Методичні вказівки до лабораторних робіт для студентів І-го курсу математичного факультету спеціальності "Прикладна математика" / [А. Ю. Брила, П. П. Антосяк, М. І. Глебена та ін.]. – Ужгород, 2014. – 73 с.
3. Васильев А. C#. Объектно-ориентированное программирование. Учебный курс / Алексей Васильев. – СПб.: Питер, 2012. – 320 с.
4. Стиллмен Э. Изучаем C#. 3-е изд / Эндрю Стиллмен, Дженнифер Грин. – СПб.: Питер, 2014. – 816 с.
5. Троелсен Э. Язык программирования C# 6.0 и платформа .NET 4.6. 7-е изд. / Эндрю Троелсен, Филипп Джепикс. – СПб.: Диалектика-Вильямс, 2018. – 1440 с.
6. Албахари Д. C# 6.0. Справочник. Полное описание языка: Пер. с англ. / Джозеф Албахари, Бен Албахари. – М.: Вильямс, 2018. – 1040 с.
7. C#. Спецификация языка. Версия 5.0 / Microsoft Corporation, 2012. – 577 с.
8. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#. Мастер-класс: Пер. с англ. / Дж. Рихтер. – СПб.: Питер, 2007. – 656 с.

Допоміжна

9. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 2-е изд.: Пер. с англ. / Г. Буч. – М.: "Издательство Бином" 1998. – 560 с.
10. Кнут Д. Е. Искусство программирования для ЭВМ. Т. 3: Сортировка и поиск / Д. Е. Кнут. – 2-е изд. – М.: Вильямс, 2008. – 824 с.
11. Жарков В. А. Самоучитель Жаркова по Visual Studio .Net: Visul Basic .NET, Visual C# .NET, Visual C++ .NET, Visual J# .NET / В. А. Жарков. – М.: Жарков Пресс, 2002. – 592 с.
12. Климов Л. П. C#. Советы программистам / Л. П. Климов. – СПб.: БХВ-Петербург, 2008. – 544 с.
13. Задачи по программированию / С. А. Абрамов,, Г. Г. Гнездилова, Е. Н. Капустина и др.. – М.: Наука, 2000. – 596 с.

1. В своєму розвитку мови програмування поступово еволюціонують від алгоритмічних до об'єктно-орієнтованих. Застосування останнього підходу прискорює розробку програм, зменшує кількість потенційних помилок, хоча й може збільшити кількість використаної пам'яті. Переваги об'єктно-орієнтованого підходу базуються на зменшені кількості повторюваних описів.

Абстрагування – це здатність описувати об'єкти оточуючого світу засобами мови програмування в рамках поставленої задачі.

В процесі програмування описують тип об'єкта, а під час завантаження програми створюються об'єкти обраного типу (наприклад, в wordi описаний тип об'єкта – документ, а в процесі в експлуатації користувач може обробляти декілька файлів-документів).

В C# типи об'єктів називають класами, а самі об'єкти екземплярами класу. Кожен об'єкт оточуючого світу здатний породжувати різні об'єкти в процесі програмування в рамках поставленої задачі. Наприклад, в банку для клієнта не зберігають дані про те, чи любить він оперу. Тобто в кожному проекті зберігаються лише ті дані, які необхідні для розв'язку задачі. Тому описувати об'єкти певного типу означає вказати, які дані про нього необхідно зберігати і які дії можливі над цими даними . Опис таких дій називається методом об'єкта. Тому фактично об'єкт – це сукупність даних та методів їх обробки. Для кожного об'єкта зберігають лише його дані, а методи зберігаються для типу об'єкта.

2. Наслідування – це здатність одного типу об'єкта бути породженим від іншого типу об'єкта.

В C# всі об'єкти породженні від загального класу **Tobject.** Ми будемо створювати власні форми, породженні від вже розробленого класу **Tform.** В кожній мові програмування використовується своя ієрархія об'єктів. Піднімаючись по ієрархії збільшується рівень абстрагування, а спускаючись рівень деталізації.

Наслідування прискорює розробку програм за рахунок зменшення кількості повторюваних описів та виправлення виявлених помилок.

Терміни:

- клас від якого породжений клас називається *батьківський клас (базовий, предок)*;
- породжений клас називається *нащадок (потомок)*;

В процесі наслідування спостерігається нарощування полів і методів.

Типи наслідування:

Найчастіше використовуються *одиночне наслідування*, тобто клас може породжуватися лише від одного класу. При *множинному наслідуванні* клас може наслідуватися від кількох класів, що прискорює розробку, але вимагає відслідковування сумісності (якщо два батьківські класи мають два методи з однаковою назвою, то для нащадка необхідно зазначити, який з уих двох класів буде використаний).

3. Інкапсуляція об'єктів передбачає:

- Здатність поєднання полів, властивостей і методів їх обробки в описі самого класу;
- Незалежність об'єкта (всі потрібні поля і методи мають бути в об'єкті описані, а всі зайняті ресурси мають бути звільнені при знищенні об'єкта);
- Здатність класу приховувати від зовнішнього світу внутрішні деталі реалізації та відображати поля і методи, які можуть бути використані для подальшої розробки;

4. Поліморфізм (poli – багато; morfa – форма) – здатність різних об'єктів мати методи з однаковою назвою і по-різному реагувати в процесі їх виклику. Наприклад, різні

компоненти форми (мітки, поля) мають процедури для перемальовування з однаковою назвою, але відтворюють потрібні елементи керування.

Поліморфізм дозволяє викликати методи об’єктів циклічно. В об’єктно-орієнтованому програмуванні одиночне зберігання методів для кожного класу об’єктів і множинне зберігання даних для кожного об’єкта.

Питання для самоконтролю

1. Базові принципи ООП.
2. Абстрагування в ООП.
3. Опис об'єктів в ООП.
4. Класи та об'єкти в ООП.
5. Наслідування в ООП.
6. Ієрархія об'єктів в ООП.
7. Інкапсуляція, як один з базових принципів ООП.
8. Використання поліморфізму в ООП.