

Архітектура та проектування програмного забезпечення

Software Design and Architecture

Lab.1 C++ templates

http://www.java2s.com/Tutorial/Cpp/0260_template/Catalog0260_template.htm

(тут щось по C++, узагалі я спочатку все на C# зроблю, а потім перепишу на C++ й Java (спочатку її для захисту лаб), паралельно вчитиму окремо C++, бо щось поки не дуже)

ЗАДАЧА – практичне застосування шаблонів до даних власної предметної області (я ж кажу тут треба курсову проаналізувати)

Lab_2 C++ templates visualizations (use diagrams or graphs to show the relationships between different types and template parameters. For example, you could create a class hierarchy diagram that shows how a template class is related to its base class and derived classes, or you could create a graph that shows the dependencies between different template parameters. **(Шакета раптом згадав, що знає англійську, там типу треба будувати діаграми класів, щоб зрозуміти, що ми взагалі робимо, а перша типу чисто описова, мабуть)**

Для побудови графічної діаграми класу в Visual C++, можна скористатися інструментами, що постачаються разом з програмою. Ось кроки, які

Потрібно виконати: (щось таке розписує... Коротше робимо десь на сторонньому ресурсі)

1. Відкрийте Visual Studio та створіть новий проект.
2. В меню "Project", виберіть "Add New Item".
3. Виберіть "Class Diagram" зі списку шаблонів та дайте ім'я діаграмі.
4. Натисніть на кнопку "Class" у панелі інструментів та намалюйте клас на діаграмі.
5. Переіменуйте клас і додайте поля та методи.
6. Додайте атрибути до полів і параметри до методів.
7. Збережіть діаграму та продовжуйте розробку проекту.

Також можна використати сторонні інструменти **(а я що казав)** для побудови діаграм класів, такі як Visual Paradigm, StarUML та інші. У цих програмах можуть бути доступні більш розширені можливості для побудови діаграм класів та інших UML-діаграм.

Visual Paradigm - Online Productivity Suite (visual-paradigm.com)

Lab_3 Creational Patterns in C++

ЗАДАЧА – практичне застосування шаблонів до даних власної предметної області (**чим це відрізняється від першої... Мабуть треба код писати**)

<https://refactoring.guru/design-patterns/cpp> (прикольний сайт, я там багато патернів вивчив)

Lab_4 Structural Patterns in C++

ЗАДАЧА – практичне застосування шаблонів до даних власної предметної області (**коротше навалюємо далі, на це й раз структурні патерни, тоді були творчі**)

<https://refactoring.guru/design-patterns/cpp> (реально, почитайте)

Lab_5 Behavioral Patterns in C++

ЗАДАЧА – практичне застосування шаблонів до даних власної предметної області (**а от і поведінкові, вони найприкольніші**)

<https://refactoring.guru/design-patterns/cpp> (ну справді, там багато цікавого)

Lab_6 C++ STL (STL – стандартна бібліотека шаблонів, Василь Іванович натякає не винаходити велосипед, та хороший він таки викладач, просто дивний)

ЗАДАЧА – практичне застосування шаблонів до даних власної предметної області (**пора би вже й доробляти, захист курсової не за горами**)

http://www.java2s.com/Tutorial/Cpp/0280_STL-Introduction/Catalog0280_STL-Introduction.htm (там якраз про STL)