

1. Досліджено механізм явища специфічного волокноутворення при наявності у бінарній суміші полімерів компатибілізатора.
2. Вивчено підходи до планування для сумішевих систем.
3. Розрахували робочий план експерименту.
4. Реалізована можливість вибору необхідних вихідних параметрів, по яким в подальшому буде здійснена оптимізація.

Отже, розроблено програмне забезпечення для інтерактивного планування експерименту для явища специфічного волокноутворення .

Ключові слова: програмне забезпечення, план експерименту, суміш полімерів, волокноутворення.

Література

1. Rezanova N.M., Rezanova V.G., Plavan V.P., Viltzaniuk O.O. The influence of nano-additives on the formation of matrix-fibrillar structure in the polymer mixture melts and on the properties of complex threads // *Vlákna a textil* (Bratislava, Slovak Republic) - №2, 2017. - p. 37-42
2. Архангельский, А. Я. Программирование в C++Builder 6 / А. Я. Архангельский. М.: ЗАО «Издательство БИНОМ», 2003.
3. Глушаков, С. В. Программирование в среде Borland C++Builder 6 / С. В. Глушаков, В. Н. Зорянский, С. Н. Хоменко. Харьков: Фолио, 2003.
4. Шамис, В. А. Borland C++Builder 6. Для профессионалов / В. А. Шамис. СПб.: Питер, 2003.
5. Stroustrup B. Programming: Principles and Practice Using C++ (2nd Edition). Addison-Wesley Professional, 2014. – 1312 p.
6. Мейерс С. Эффективный и современный C++. М.: Вильямс, 2016. - 304 с.

РЕЗАНОВА В.Г., МАРЧЕНКО В.І.

АРХІТЕКТУРНІ ПАТЕРНИ ПРИ РОЗРОБЦІ КЛІЄНТ-СЕРВЕРНИХ ДОДАТКІВ

REZANOVA V.G., MARCHENKO V.I.

ARCHITECTURAL PATTERNS IN THE DEVELOPMENT OF CLIENT-SERVER APPLICATION

Purpose and tasks. Explore the most popular architecture patterns, which used at the developing client application.

Object and subject of research to learn about each of the design patterns. The task is to investigate three of the most popular architectural patterns in this area. It is the MVP (Model-View-Presenter), MVC (Model-View-Controller), MVVM (Model-View-ViewModel).

Consider the difference between each of the patterns, the complexity of the implementation of each and the interaction between the components. Find the pros and cons of each.

The subject of the study is to find an architecture pattern which is best to use in a given situation based on research data.

Вступ

При розробці клієнт-серверного додатку, який претендує на довготривале існування дуже важливо вибрати архітектурний патерн, який повністю задовільнить всі потреби команди та самого продукту.

Шаблони проектування програмного забезпечення (software design patterns) — ефектні способи вирішення задач проектування програмного забезпечення. Шаблон не є закінченим зразком, який можна безпосередньо транслювати в програмний код. Об'єктно-орієнтований шаблон найчастіше є зразком вирішення проблеми і відображає відношення між класами та об'єктами, без вказівки на те, як буде зрештою реалізоване це відношення.

Постановка завдання

При розробці клієнтної частини програмного забезпечення найчастіше використовують такі патерни: MVP (Model - View - Presenter), MVC (Model - View - Controller), MVVM (Model - View - ViewModel).

Кожен з патернів відрізняється роллю компонентів та способом взаємодії з іншими. З огляду на це, доцільним є дослідження всіх популярних рішень та способів реалізації.

Основна частина

Архітектурні патерни являються набором універсальних правил, які не залежать від жодної з платформ та мов. Для наглядності будемо використовувати мову Swift та платформу iOS. Розглянемо основні патерни. Проаналізуємо всі плюси та мінуси кожного та оберемо найбільш підходящий для нашої абстрактної задачі.

Найпростіший паттерн це MVC. Компанія Apple рекомендує використовувати його для архітектурно простих додатків. MVC - це схема розробки програмного забезпечення, що складається з трьох основних об'єктів:

Model - це місце, де перебувають ваші дані. Там живуть такі речі, як об'єкти моделей, парсери, менеджери та мережевий код.

View - обличчя вашої програми. Її класи часто використовуються для багаторазового використання, оскільки не містять певної логіки. Наприклад, UILabel - це представлення, яке представляє текст на екрані, і його можна використовувати повторно та розширювати.

Controller - опосередковує погляд і модель за допомогою шаблону делегування. В ідеальному сценарії суб'єкт контролера не знатиме конкретного погляду, з яким він має справу. Натомість він буде спілкуватися з абстракцією за допомогою протоколу. Класичний приклад -

спосіб спілкування UITableView зі своїм джерелом даних через протокол UITableViewDataSource.

MVP - схема розробки, що складається з таких основних компонентів Model, View, Presenter. Model, View уже знайомі компоненти. Presenter відповідає за взаємодію між Model та View. При реалізації цього патерну View взагалі немає ніякої логіки і відповідає лише за відображення. Різниця Presenter буде відповідати за всю логіку, включаючи реагування на дії користувача та оновлення інтерфейсу користувача (через делегата). і найголовніше - наш ведучий не буде залежати від View. це означає, що добре ізольований, тому легко засвідчений;)

Важливо зазначити, що MVP використовує пасивний режим перегляду. це означає, що всі дії будуть передані Presenter. Що запустить оновлення інтерфейсу користувача за допомогою делегатів. тож перегляд буде лише проходити дії та слухати оновлення ведучого.

З розвитком реактивного програмування все більше набувають популярності патерни, в основу яких лежить патерн Спостерігач. Одним з таких є MVVM. Суть цього патерна полягає в тому, що компоненти не взаємодіють між собою на пряму, а реагують на зміну ViewModel. Отже, MVVM має знайомі нам компоненти (Model, View) та новий (ViewModel) (Модель уявлення).

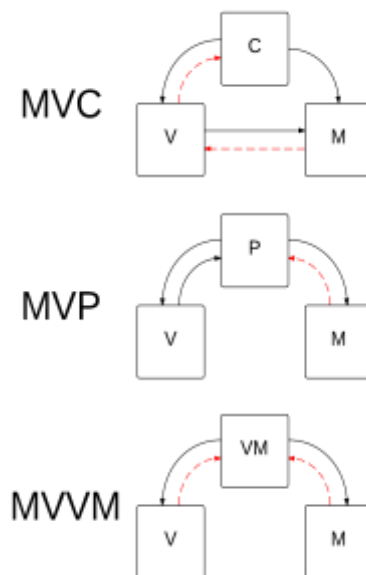


Рисунок 1 -Зображення архітектурних патернів програмування

Модель уявлення пов'язує модель і уявлення через механізм прив'язки даних. Якщо в моделі змінюються значення властивостей, при реалізації моделлю інтерфейсу автоматично йде зміна відображуваних даних в

інших компонентах. Всі компоненти незалежні між собою і обмінюються інформацією лиш через спостереження за ViewModel.

Висновки

Всі патерни використовуються при розробці реальних продуктів. Кожен з них краще використовувати в тій чи іншій ситуації. Немає єдиного вірного паттерна.

Важливо від складності додатку чітко розбивати класи на компоненти залежно від їхньої ролі. В такому випадку розробка додатку в команді буде займати набагато менше часу й містити меншу кількість помилок.

Ключові слова: програмне забезпечення, архітектурні патерни проектування, клієнт- серверні додатки.

Література

1. Joshua Greene, Jay Strawn. Design Patterns by Tutorials (Third Edition): Learning Design Patterns in Swift // Razeware LLC., -2019 -р. 43

2. Ерік Фрімен, Елізабет Фрімен, Берт Бейтс. Head First. Патерни програмування // O`Reilly., -2018 -153 с.

ЩЕРБАНЬ В.Ю., КОЛИСКО О.З., ЩЕРБАНЬ Ю.Ю., ШОЛУДЬКО М.І., МЕЛЬНИК Г.В., КАЛАШНИК В.Ю.

ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ РЕКУРСІЇ ПРИ ВИКОРИСТАННІ ЗВОРОТНІХ ПОЛЬСКИХ ЗАПИСІВ

V.Yu. SHCHERBAN, A.K.PETKO, O.Z.KOLISKO, Y.Yu. SHCHERBAN, M.I. SHOLUDKO, G.V.MELNIK, V.YU.KALASHNIK

PROGRAMMATIC REALIZATION OF ALGORITHM OF RECURSION IS AT THE USE OF REVERSE POLISH OF RECORDS

The paper describes the structure of a computer program for the implementation of the recursion algorithm to determine the technological effort. A block diagram of a program for determining the technological effort of textile, knit and sewing machines for a wide range of yarns and yarns is presented. The computer program provides the ability to change the speed of the filament, the length of the line of the elastic filling system, the intersection of the filament, its input tension, the material of the guide surfaces, their shape. As structural elements, it is possible to use high and low curvature guides, thread tensioners and tension compensators. The basic procedures and functions are described.

The normal flow of technological processes is possible in the case when the thread in front of the zone will have the minimum required tension. If you increase the tension in the feed system, there will be breaks, which will stop the process equipment and reduce its productivity. In addition, the development of new strands of the filing system requires an operational assessment of the tension value before the working area. To do this, use specially designed programs to perform a computational experiment.

A characteristic feature of the structural schemes of thread filling on the process equipment is that the output thread tension at the current zone will be input to the next. This