

DOI: <https://doi.org/10.32836/2521-6643-2021.2-62.3>
УДК 004.75:519.86

Зінченко А. Ю., кандидат технічних наук,
завідувач кафедри комп'ютерних наук
Київського міжнародного університету

ВИКОРИСТАННЯ ІН'ЄКЦІЇ ЗАЛЕЖНОСТЕЙ У РОЗРОБЦІ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДОСЛІДЖЕННЯ ПАРАМЕТРИЧНИХ МЕТОДІВ VaR

Розглянуто використання технології слабозв'язаних програмних модулів автоматизованої інформаційної системи на основі ін'єкцій залежностей для оцінки і прогнозу ринкових ризиків. На основі реалізованої автоматизованої системи для основних параметричних методів VaR виконані розрахунки та проведено моделювання, зокрема для експоненціальної згладженої моделі, варіаційно-коваріаційної (методом для багатокomпонентного портфелю), моделі з врахуванням волатильності та евристичної моделі. На основі теорії інтервального оцінювання запропоновано використовувати евристичну модель, ідея якої полягає в тому, що коефіцієнти моделі прогнозу також прогножуються. Тобто модель працює не на коефіцієнтах, адаптованих до попередньої статистики, а на коефіцієнтах, які прогножуються для розрахунку прогнозу на наступний крок.

Крім того, у статті детально проаналізовано використання принципу Dependency inversion при реалізації подібних сервіс-орієнтованих архітектурних рішень. В кінці приведені результати порівняння моделей для оцінки та прогнозу ринкових ризиків. Вибрані кращі моделі.

Ключові слова: IoC; DI; DIP; SOLID; ESB; SOA; VaR.

This paper considers the use of technology of loosely coupled software modules of an automated information system based on injection of dependencies to assess and forecast market risks. Based on the implemented automated system for basic parametric VaR methods, calculations were performed and modeling was performed, in particular for the exponentially smoothed model, variation-covariance (multicomponent portfolio method), volatility model and heuristic model. According to the results of modeling among the classical parametric VaR methods, the GARCH model (8, 8) with a horizon of 60 working days showed the best forecast. This is due to the fact that in periods of increased market turbulence, the use of constant volatility values leads to an underestimation of VaR, and in periods of stable market, the use of high volatility leads to an overestimation of VaR in the calculations.

© А. Ю. Зінченко, 2021

However, the disadvantages of classical VaR methods are that the random heteroskedastic process is first approximated by a normal law, and then the approximation determines the confidence interval and the quantile, respectively. In this approach, "heavy tails" that cause extreme losses of capital are not taken into account outside the approximation. Therefore, based on the theory of interval estimation, it is proposed to use a heuristic model, the idea of which is that the coefficients of the forecast model are also predicted. So, the model does not work on coefficients adapted to previous statistics, but it works on coefficients that are projected to calculate the forecast for the next step. According to estimates of the optimality criterion recommended by the Basel Committee, the heuristic model based on the proposed approach proved to be the best and showed fewer breakdowns.

In addition, the article analyzes in detail the use of various injections of dependencies (constructor injection, parameter injection, setter injection, interface injection) in the implementation of weakly related software components of the information system. This ensures the implementation of the basic principle of SOLID – The Dependency Inversion Principle through IoC control inversion.

Ключові слова: *IoC; DI; DIP; SOLID; ESB; SOA; VaR.*

Постановка проблеми. Останні три роки для світу в цілому та для України зокрема ознаменувалися глобальними подіями (пандемія коронавірусу, війна в Україні), які пришвидшили процес інформатизації в разі, зокрема, у фінансово-економічному секторі, наданні державних послуг, у галузі освіти тощо. Тому логічно, що спостерігається тенденція переходу в розробці програмного забезпечення (ПЗ) від клієнт-серверних до сервіс-орієнтованих архітектур (service-oriented architecture – SOA) з використанням сервісів, мікросервісів чи гібридних хмарних технологій надання послуг. Дані сервіси будуються з використанням протоколів SOAP, XML-RPC, JSON-RPC або ж архітектурного стилю REST. Як приклад, можна навести “Єдиний веб-портал державних послуг Дія” – розподілену масштабовану SOA. Але як тоді бути з уже впровадженими CRM-системами (системами управління відносинами з клієнтами) чи ERP-системами (системами планування ресурсів підприємства), в які на початку десятих років чи раніше було вкладено десятки мільйонів грошей? Як правило, такі автоматизовані системи управління (АСУ) проектували у двохланковій (інколи, навіть у трьохланковій з сервером застосувань) клієнт-серверній архітектурі. Вихід єдиний – використання SOA з розподіленими компонентами. Сучасні перспективні SOA повинні підтримувати та супроводжувати користувача на кожному етапі процесу workflow (системи управління процесами, які повторюються, і завданнями, які виконуються в певному порядку) й мати комплексний набір хмарних рішень, із підтримкою штучного інтелекту, який може

оперувати онлайн, офлайн та сторонніми джерелами даних для повного та динамічного представлення їх клієнту. Серед останніх найпоширенішою моделлю хмарних рішень є SaaS (software as a service), доступ до програмного забезпечення якого надається віддалено мережевими каналами через веб-інтерфейс чи термінал.

Оскільки механізм слабкої зв'язаності є основним принципом об'єктно-орієнтованого програмування (ООП) SOLID – основою принципу інверсії DIP (The Dependency Inversion Principle), а також одним із дев'яти шаблонів об'єктно-орієнтованого конструювання (ООК) Крейга Лармана GRASP для визначення відповідальності між класами та їх об'єктами, то побудова слабо-зв'язаних системних модулів під час проектування SOA, зокрема, у фінансовому секторі економіки для оцінки і прогнозу ринкових ризиків на основі параметричних методів VaR (Value at Risk або вартісна міра банківського ризику) є темою актуальною.

Аналіз останніх досліджень і публікацій. Основні принципи слабкої зв'язаності та пов'язаності ПЗ були закладені ще Ларрі Константином наприкінці 1960-х років як частина структурованого дизайну для зменшення витрат на обслуговування та модифікацію [1, 2]. Основний принцип об'єктно-орієнтованого конструювання DIP – детально проаналізований у [3], а також в класичній книжці “Банди 4-ох” “Design Patterns: Elements of Reusable Object-Oriented Software”.

Основні принципи управління ризиками детально подані в матеріалах Базельського Комітету, зокрема в Базель 1 та Базель 2. Теорія інтервального оцінювання проаналізована в праці [4] та [5]. Крім того, класичні параметричні методи VaR для оцінки і прогнозу ринкових ризиків добре описані в [5] і [6].

Мета статті: розглянути реалізацію технології слабозв'язаних компонентів ПЗ на основі інверсії управління, тобто реалізацію механізму Dependency Injection (DI), з метою побудови інформаційної технології (IT) для створення адекватних моделей оцінювання ринкових ризиків і прогнозу за її допомогою.

Виклад основного матеріалу. Зв'язаність програмного забезпечення є метрикою програмного забезпечення та одним із основних принципів, який закладений в основі ООП. Даний принцип описує взаємозалежність між програмними компонентами. Слабка зв'язаність ПЗ є ознакою добре структурованої і добре спроектованої АСУ. Тут слід відрізнити низьку пов'язаність між програмними модулями, яка є іншим поняттям та описує наскільки пов'язані функції в межах одного модуля. Жорстка пов'язаність є серйозним недоліком програмного забезпечення, оскільки утруднює розуміння логіки модулів, їх модифікацію, автономне тестування, і навіть повторне використання. Як зазначалося вище, слабка зв'язаність ПЗ є основ-

ним принципом OOK SOLID, тобто, по суті, основою DIP, а також одним із дев'яти шаблонів OOK GRASP.

Зменшення зв'язаності програмних модулів під час проектування сучасних ІТ базується на шаблонах проектування багатопланової архітектури (наприклад, Model-View-Controller, Model-View-Presenter, Model-View-ViewModel та ін.). Воно досягається шляхом функціонального дизайну – підходу, основою якого є обмеження обов'язків модулів із функціональністю. Ключовим методом такої технології слабкої зв'язаності компонентів ПЗ є використання інверсії управління (IoC) через реалізацію механізму DI. При цьому основним механізмом безпеки даних під час взаємодії рівня даних (Resource Layer) з рівнем бізнес-логіки (Business Layer) та інтегрованими корпоративними сервісами в SOA є використання сервісної шини підприємства (ESB), яка слугує фреймворком для обміну повідомленнями та реалізує технологію слабозв'язаних компонентів в ІТ. ESB забезпечує обмін даними між різними сервісами SOA. Найпоширенішими її варіантами використання є Oracle Enterprise Service Bus та Microsoft Azure Service Bus.

Сучасне проектування архітектури ІТ в розподілених АСУ потребує глибокого розуміння не тільки ООП та 23 основних принципів OOK, а й розуміння різниці між IoC, DI та DIP, між паттернами, фреймворками та бібліотеками, а також різниці проектуванням взаємозв'язків між класами. Не будемо зупинятися на основних принципах OOK, деталях проектування класів у контексті реалізації, наслідування чи асоціації (агрегації і композиції відповідно). Перейдемо до опису проектування слабозв'язаності програмних модулів.

IoC – це досить загальне поняття, яке відрізняє бібліотеку динамічної компановки від фреймворка. На відміну від традиційного потоку управління IoC інвертує управління потоком. У класичній моделі архітектури ПЗ код, який викликається, сам контролює зовнішнє оточення, а також час і порядок викликів бібліотечних методів. Проте у випадку з фреймворком обов'язки міняються місцями: фреймворк надає деякі точки розширення, через які він викликає певні методи користувацького коду.

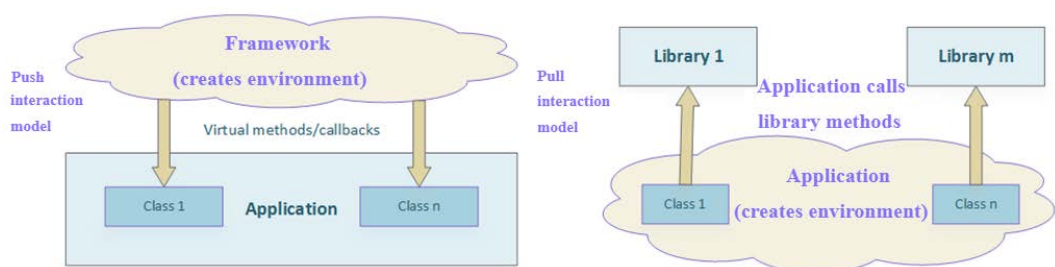


Рис. 1. Інвертація потоку управління інверсією управління

IoC використовується для підвищення модульності програми і робить її розподіленою і розширюваною. Вперше термін був використаний Майклом Меттсоном у праці [7]. Прикладами шаблонів проектування, які дотримуються принципу інверсії керування, є зворотні виклики, планувальники, цикли подій, ін'єкція залежностей, шаблонний метод і паттерн Наглядач.

Основною технологією реалізації IoC є використання механізму ін'єкції залежностей через конструктор (Constructor injection), параметр (Parameter injection), інтерфейс (Interface injection) чи сеттер (Setter injection). Існують також інші варіанти – використання паттерну Локатора служби, контекстного пошуку (contextualized lookup), паттерну Стратегія та інших. Найефективнішим є DI.

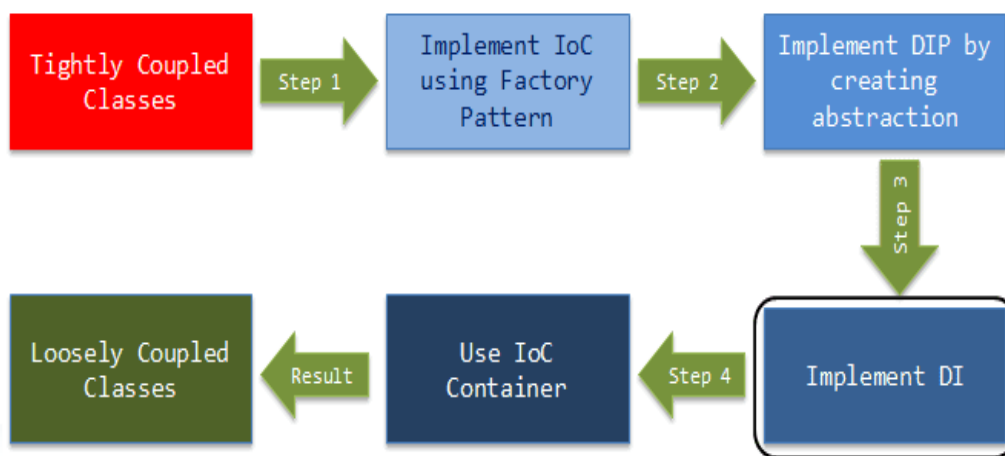


Рис. 2. Приклад реалізації IoC, з використанням шаблонного методу

Основним принципом SOLID є DIP, який дає рекомендації, якими повинні бути залежності між високорівневими та низькорівневими абстракціями:

- компоненти верхніх рівнів не повинні залежати від компонентів нижніх рівнів. Обидва повинні залежати від абстракцій;
- абстракції не повинні залежати від деталей. Проте самі деталі повинні залежати від абстракцій;
- компоненти вищого рівня реалізують бізнес-правила або логіку в системі. Низькорівневі компоненти займаються більш детальними операціями.

В принципі DIP залежності між класами повинні розташовуватись на поточному або вищому рівні абстракції. Тобто будь-який клас, який потребує наявності інтерфейсу в конструкторі, не обов'язково слідуватиме принципу інверсії залежностей.

Основний механізм реалізації IoC – DI – полягає у передачі класу його залежностей. Він слугує певним фреймворком. Як зазначалося вище, існує декілька конкретних способів або паттернів застосування DI. Наведемо їх нижче на прикладі логування для реалізації параметричних методів VaR:

```
class ReportProcessor
{
    private readonly IReportSender _reportSender;
    // Ін'єкція конструктора: передача обов'язкової залежності
    public ReportProcessor(IReportSender reportSender)
    {
        _reportSender = reportSender;
        Logger = LogManager.DefaultLogger;
    }
    // Ін'єкція методу: передача обов'язкових залежностей методу
    public void SendReport(Report report, IReportFormatter formatter)
    {
        Logger.Info("Sending report...");
        var formattedReport = formatter.Format(report);
        _reportSender.SendReport(formattedReport);
        Logger.Info("Report has been sent");
    }
    // Ін'єкція властивостей: встановлення необов'язкових інфраструктурних
    залежностей
    public ILogger Logger { get; set; }
}
// Ін'єкція на основі сеттеру: Іос-контейнер викликає методи сеттера класа
Provider
// після виклику конструктора для створення екземпляра bean
public Provider provider ()
{
    Provider provider = new Provider ();
    provider.setItem(item1());
    return provider;
}
<bean id = "provider" class= "org.baeldung.provider.Provider" >
    <property name = "item" ref= "item1" />
</bean>
```

Розглянемо тепер застосування розробленої ІТ, а також АСУ на її основі, до задачі оцінки та прогнозу ринкових ризиків з використанням параметричних методів VaR. Зауважимо, що в реалізації технології слабозв'язаних програмних

модулів з використанням ін'єкцій залежностей був використаний контейнер Unity – контейнер IoC з відкритим програмним кодом для платформи NET.

Фінансові ринки, зокрема в Україні в нинішнього часу, – це досить складна, нестабільна, високотехнологічна сфера, яка безпосередньо пов'язана з найрізноманітнішими фінансовими ризиками. Ризик є мірою ймовірності настання якої-небудь несприятливої події або її наслідків, що призводить до прямих втрат або непрямого збитку. Найбільш значущі види ризику (кредитний, інвестиційний, валютний) можуть привести не тільки до серйозного погіршення фінансового стану кредитної організації, але в найгіршому випадку – до втрати капіталу і банкрутства. Тому головна задача ризик-менеджменту полягає у виявленні і запобіганні можливих несприятливих подій, знаходженні шляхів мінімізації їх наслідків, створенні математичних методологій оцінок та прогнозування ризиків. Крім того, недооцінка ризиків призводить до втрат капіталу за рахунок їх реалізації, переоцінка – до надмірного резервування, внаслідок чого надмірна частина капіталу виводиться з обороту, що також призводить до втрат доходу.

Міра ризику VaR – це величина збитків, яка з імовірністю, що дорівнює рівню довіри для заданого довірчого інтервалу (наприклад, 99 %), не буде перевищена. Обчислення величини VaR проводиться з метою твердження про те, що ми впевнені на $X\%$ (з імовірністю $X/100$), що наші втрати не перевищать Y умовних одиниць протягом наступних N днів. У даному твердженні VaR – це невідома величина Y , яку потрібно обчислити. За рекомендацією Базельського комітету, міра ризику $VaR = K\sigma V$, де K – квантіль стандартного нормального розподілу при ймовірності 0,95 або 0,99; σ – середньоквадратичне відхилення, яке вираховується на вибірці зміни курсів акцій чи валют; V – розмір відкритої валютної позиції. Якщо одноденні витрати капіталу перевищували прогнозоване значення VaR за останні 250 торгових днів то для $K \leq 4$ – визначається “зелена зона”; $5 \leq K \leq 9$ – визначається “жовта зона”; $K \geq 10$ – “червона зона”, але в більшості випадків такі моделі відкидаються. При цьому зміни курсів логарифмуються, тобто

$X_t = \log\left(\frac{R_t}{R_{t-1}}\right)$, де R_t, R_{t-1} – офіційні курси акцій чи валют за певну послідовність днів відповідно.

Якщо величина VaR (а саме: другий член $k\sigma$ моделі VaR) може бути обчислена через параметри розподілу, то такі методи обчислення називаються параметричними. Дана методика визначає два основні типи моделей – класичну (варіаційно-коваріаційну) і експоненціальну (експоненційну і експоненційно згладжену). Крім того, виділяють модель на основі прогнозування волатильності за допомогою ARCH/GARCH. Класична VAR спира-

ється на припущення нормального закону розподілу флуктуацій випадкового процесу, середньоквадратичне відхилення якого визначається як

$$\sigma_t = \sqrt{\frac{1}{t-1} \sum_{i=1}^t (x_i - m)^2},$$

де квантіль k для імовірності 0,95 дорівнює 1,95.

Тобто стверджується, що з імовірністю 95 % процес не вийде з діапазону $k\sigma$. З іншого боку, нерівність Б'єнеме–Чебешева стверджує, що

$$P(|x - m| \geq k\sigma) \leq \frac{1}{k^2},$$

де x – випадкова змінна; m – математичне сподівання; σ – середньоквадратичне відхилення. Тоді при $k=1,95$ імовірність

$P(|x - m| \geq 1.95\sigma) \leq 0.263$ з точністю до четвертого знаку, тобто для реального процесу імовірність виходу з діапазону $k\sigma$ досить велика.

Адаптивна модель багатокomпонентного портфеля класичного VAR для визначення середньоквадратичного відхилення σ використовує формулу, наведену вище, та загальну процедуру адаптації і прогнозу коефіцієнта α граничного інтервалу. Тобто $VaR_i = \alpha * \tilde{\sigma}_i V_i, i = 1, 2, \dots, n$, де α – значення квантиля довірчого інтервалу; $\tilde{\sigma}_i$ – прогнозоване значення відхилення.

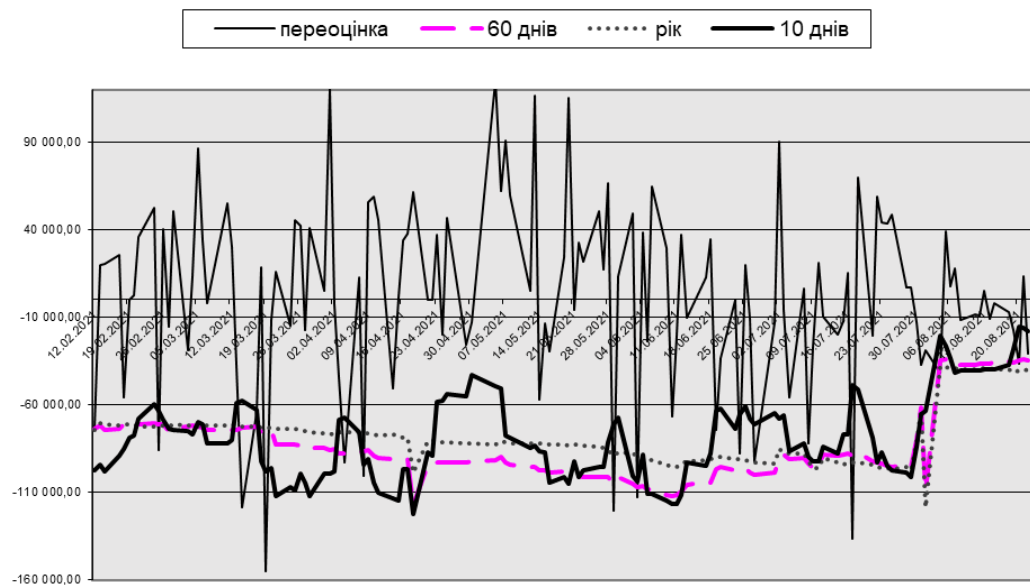


Рис. 3. Бек-тестінг варіаційно-коваріаційної моделі для різних горизонтів

Для проведеного дослідження ціни акцій на основі моделі багатокомпонентного портфеля з горизонтами в 1 рік (250 робочих днів), 60 і 10 робочих днів відповідно кількість пробітів становила 9, 8 та 14 відповідно. Тому згідно з Базельським комітетом модель з горизонтом у 60 днів є більш адекватною. При цьому з графіка видно, що графік з горизонтом у 10 днів непогано враховує тренд.

Ефективнішою оцінкою є експоненційно зважене середньоквадратичне відхилення, яке визначається як $\sigma_t = \sqrt{(1-\lambda) \sum_{i=1}^t \lambda^{i-1} (x_i - m)^2}$, де λ – коефіцієнт експоненційного згладжування. Дана модель в рекурсивній формі записується як $\hat{\sigma}_{t+1} = \sqrt{\lambda \sigma_t^2 + (1-\lambda) \sigma_t^2}$.

Недоліками класичної та експоненціальної моделей є те, що випадковий гетероскедастичний процес спочатку апроксимується нормальним законом, а потім за апроксимацією визначається довірчий інтервал і відповідно квантиль. За такого підходу за межами апроксимації залишаються неврахованими “важкі хвости”, які викликають екстремальні втрати капіталу. Тому було прийнято рішення розробити в архітектурі ІТ методику, яка спирається на оцінки та прогноз параметрів випадкового процесу безпосередньо в динаміці змін курсів.

Оцінки та прогнози міри ризику зводяться до математичної задачі інтервального оцінювання флуктуацій випадкового процесу. В [5] показано, що ця задача зводиться до задачі визначення толерантних границь випадкового процесу у формі: $U_1 = \hat{x} - k\sigma$ та $U_2 = \hat{x} + k\sigma$, де \hat{x} – середнє значення параметра за вибіркою; σ – середньоквадратичне відхилення; коефіцієнт k – визначається як $k = k_\infty \left(1 + \frac{x_\gamma^2}{\sqrt{2n}} + \frac{5x_\gamma^2 + 10}{12n}\right)$. У свою чергу, k_∞ визначається

зі співвідношення $\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-\frac{t^2}{2}} dt = 2\Phi_0(k_\infty) = P$, де P – частка вибірки, що повинна попасти в інтервал, заданий толерантними границями, і визначається з умови конкретної задачі. Оскільки $2\Phi_0(k_\infty)$ визначається з табличних даних, то, визначивши P , автоматично визначається і коефіцієнт k_∞ .

Подібним чином знаходиться коефіцієнт x_γ зі співвідношення $\frac{1}{\sqrt{2\pi}} \int_{x_\gamma}^{+\infty} e^{-\frac{t^2}{2}} dt = 0.5 - \Phi_0(x_\gamma) = 1 - \gamma$, де нормована функція Лапласа визнача-

ється з табличних значень. При заданій величині параметра γ для умов конкретної задачі обчислюється x_γ . При цьому γ визначає задану імовірність того, що доля P буде в межах толерантного інтервалу. Виходячи з економічних припущень щодо γ і P , побудований таким способом інтервал в межах $(u_1; u_2)$ визначає міру флуктуацій випадкового процесу. Тобто γ і P визначають випадкові границі $(u_1; u_2)$. У свою чергу, для зворотної задачі – змінюючи крок.

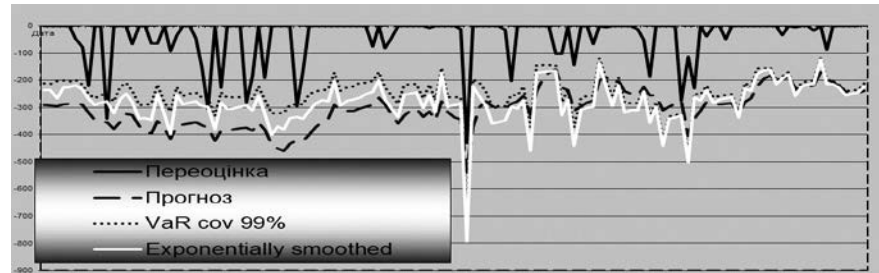


Рис. 4. Порівняння різних методів оцінки VaR

Таблиця 1

Результати розрахунку критеріїв порівняння

Моделі VaR	Кількість пробоїв	Недорезерв, грн	Перерезерв, грн	Оцінка критерію
Варіаційно-коваріаційна	1	27,66	34712,23	1,1328
Експоненціально згладжена	2	72,79	31470,48	1,0659
Евристична узагальнена	1	0,08022	33989,91	1,0497

З отриманих результатів можна зробити висновок, що варіаційно-коваріаційна модель та евристична дали однакову кількість пробоїв, а експоненціальна модель дала 2 пробої, але ця кількість вкладається у вимоги Базельського комітету: 4 пробої за 250 днів (брався проміжок в 130 днів). Найменший недорезерв показала евристична модель, а найбільший – експоненціальна модель. А з перерезервом навпаки, у експоненціальної моделі він найменший, а ось у моделі варіаційно-коваріаційного VaR – найбільший.

Згідно з оцінкою критеріїв, евристична модель – найкраща, бо її коефіцієнт дорівнював 1,0497 і є найменшим.

На останок побудуємо модель VaR з урахуванням волатильності на основі класичної моделі GARCH(p,q) та порівняємо її з моделлю варіаційно-коваріаційного VaR. Адже у періоди підвищеної ринкової турбулентності використання сталих значень волатильності призводить до недооцінки VaR. І навпаки, у періоди стабільного ринку використання завищеної волатильності у розрахунках призводить до переоцінки VaR. Будемо коригувати волатильність, використовуючи ковзну модель GARCH(8,8) з горизонтом 60 робочих днів. Для перевірки наявності гетероскедастичності у виборці даних використовувалася Q-статистика Льюнга-Бокса, яка обчислюється за формулою

$$Q = N(N+2) \sum_{i=1}^n \rho(i) / (N-i), \text{ де } n = N/4; \quad \rho(i) = \frac{\sum_{k=i+1}^N [\hat{\varepsilon}^2(k) - \hat{\sigma}_\varepsilon^2][\hat{\varepsilon}^2(k-i) - \hat{\sigma}_\varepsilon^2]}{\sum_{k=1}^N [\hat{\varepsilon}^2(k) - \hat{\sigma}_\varepsilon^2]}$$

рати залишків, у яких $\hat{\sigma}_\varepsilon^2 = \frac{1}{N-1} \sum_{k=1}^N \hat{\varepsilon}^2(k)$; збурення $\hat{\varepsilon}(k) = \varepsilon(k)$ з моделі ARCH, N – число залишків після побудови моделі AR(p). Якщо значення $\hat{\varepsilon}^2(k)$ не корелюються між собою, то Q-статистика має розподіл χ^2 з n ступенями свободи. При цьому розрахунок волатильності відбувається за такою рекурентною формулою:

$$\sigma^2(k) = a_0 + \sum_{i=1}^q \alpha_i \varepsilon^2(k-i) + \sum_{i=1}^p \beta_i \sigma^2(k-i),$$

де константа a_0 – базова волатильність; ε – попередні зміни цін; q – порядок членів r^2 моделі GARCH, тобто зміни курсу, які впливають на волатильність; p – порядок членів σ^2 моделі ARCH; α_i – вагові коефіцієнти для визначення ступеня впливу змін на поточне значення волатильності. Тоді модель прогнозу на наступний крок матиме такий вигляд:

$$h(k) = a_0 + \sum_{i=1}^q \alpha_i \varepsilon^2(k-i) + \sum_{i=1}^p \beta_i h(k-i).$$

Хоча кількість пробів для даної моделі становить 6, на відміну від варіаційно-коваріаційної моделі VaR із горизонтом у 60 днів та 8 пробоями, проте точність прогнозу в моделі GARCH (8, 8) була не набагато більшою. Згідно з оцінкою критерію оптимальності модель GARCH (8, 8) є кращою. Проте порівнюючи дані моделі з евристичною моделлю, яка побудована на запропонованому підході, що коефіцієнти прогнозної моделі адаптуються

до попередніх вибірок і використовуються в моделі для прогнозу на наступний крок, евристична модель дала меншу кількість пробоїв та виявилася найкращою за оцінкою критерію оптимальності, який рекомендований Базельським комітетом.

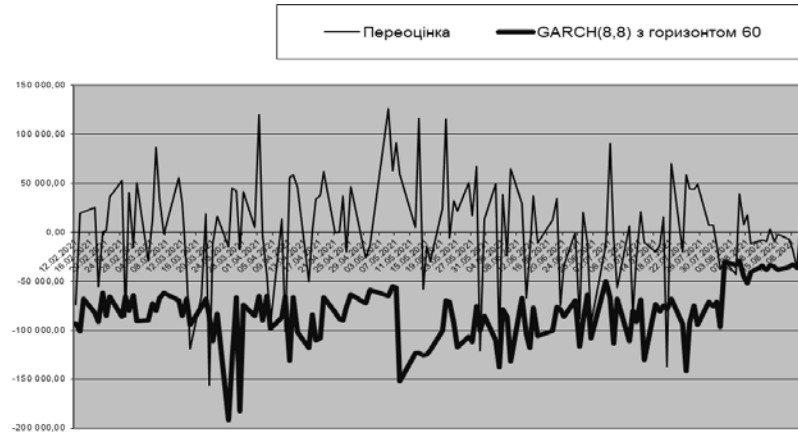


Рис. 5. Модель GARCH(8, 8) з горизонтом 60 робочих днів

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі. Використання ін'єкції залежностей у реалізації технології слабозв'язаних програмних компонентів інформаційної системи забезпечує імплементацію основного принципу SOLID – DIP через інверсію управління IoC. У статті детально розглянуто реалізацію DI через реалізацію IoC в проектуванні сучасних IT. На основі реалізованої IT з інкапсульованим механізмом DI для забезпечення слабозв'язаності між окремими компонентами SOA було виконано порівняння деяких параметричних методів VaR для оцінки і прогнозу ринкових ризиків. При цьому для побудови евристичної моделі було запропоновано власну методику оцінювання і прогнозу, яка спирається на оцінки та прогноз параметрів випадкового процесу безпосередньо в динаміці змін курсів. За результатами моделювання згідно з критерієм оптимальності, рекомендованим Базельським комітетом, евристична модель виявилася кращою. Поміж класичних моделей кращою була модель VaR з урахуванням волатильності.

У перспективі планується розглянути сучасні механізми безпеки в SOA, зокрема застосування блокчейн у фінансових операціях та його інтеграцію з корпоративною шиною Oracle, яка реалізує технологію слабозв'язаності на рівні даних SOA. Крім того, планується реалізувати запропоновану методику оцінки ринкових ризиків для інших параметричних методів VaR.

Список використаних джерел:

1. *Stevens Wayne P., Myers Glenford J., Constantine Larry LeRoy* (June 1974). Structured design. *IBM Systems Journal*. 13 (2). P. 115–139. doi:10.1147/sj.132.0115.
2. *Yourdon Edward, Constantine Larry LeRoy* (1979). *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*. Yourdon Press. Bibcode: 1979sdfd. ISBN 978-0-13-854471-3.
3. *Philip A. Laplante*, What Every Engineer Should Know about Software Engineering. CRC Press, 2007. P. 105–106. ISBN 978-1-4200-0674-2.
4. *Лукашин Ю. П.* Прогнозирование временных рядов с помощью моделей авторегрессии-скользящего среднего первого и второго порядка. – Москва: ИМЭМО, 1983. С. 107.
5. *Браунли К. А.* Статистическая теория и методология в науке и технике. Москва: Наука, 1977. С. 407.
6. *Ульянова М.* Управление рыночным риском. Молодой ученый. 2014. № 1.2. С. 99–102.
7. *Mattsson Michael* (February 1996). Object-Oriented Frameworks, A survey of methodological issues. Department of Computer Science, Lund University. URL : <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.1127>

References:

1. *Stevens Wayne P., Myers Glenford J., Constantine Larry LeRoy* (June 1974). Structured design. *IBM Systems Journal*. 13 (2). С 115–139. doi:10.1147/sj.132.0115.
2. *Yourdon Edward, Constantine Larry LeRoy* (1979). *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*. Yourdon Press. Bibcode: 1979sdfd. ISBN 978-0-13-854471-3.
3. *Philip A. Laplante*. What Every Engineer Should Know about Software Engineering. CRC Press, 2007. P. 105–106. ISBN 978-1-4200-0674-2.
4. *Lukashin Yu. P.* Prognozirovanie vremennykh ryadov s pomoshch'yu modo-ley avtoregressii-skol'zyashchego srednego pervogo i vtorogo poryadka. М.: ИМЭМО, 1983. С. 107.
5. *Braunli K. A.* Statisticheskaya teoriya i metodologiya v nauke i tekhnike. М.: Nauka, 1977. С. 407.
6. *Ul'yanova M.* Upravlenie rynochnym riskom. 2014. № 21.2. С. 99–102.
7. *Mattsson Michael* (February 1996). Object-Oriented Frameworks, A survey of methodological issues. Department of Computer Science, Lund University. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.1127>