УДК 004.1

Оврамець І. В., *здобувач 4 курсу спеціальності 122 Комп'ютерні науки,*
Антонов Ю. С., *канд. фіз.-мат. наук, доцент, доцент кафедри інформаційних технологій*

АРХІТЕКТУРНІ ОСОБЛИВОСТІ МЕСЕНДЖЕРА З БАГАТОРІВНЕВИМ ШИФРУВАННЯМ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному світі питання шифрування, захисту інформації та протидії кіберзагрозам є доволі актуальними [1–2], оскільки кожна людина постійно комунікує зі своїми колегами, друзями або близькими, використовуючи для цього месенджери або інші засоби зв'язку.

Під час розробки різноманітних додатків важливим складником є архітектура програмного забезпечення [3–4], особливо коли йдеться про великі проєкти зі значною кількістю кодів. Необхідно зробити функціонал достатньо простим для розширення та виправлення помилок. Можливість повторного використання коду в інших проєктах можна реалізувати за допомогою багатошарової архітектури.

Цю роботу присвячено створенню кросплатформного месенджера з багаторівневим шифруванням та особливостям його архітектурної реалізації.

Для коректної роботи месенджера виникає потреба уніфікувати методи шифрування, тобто зробити так, щоб за потреби додати новий тип шифрування в месенджер було достатньо реалізувати відповідний інтерфейс і додати певні налаштування у файл конфігурації. Вирішення цього завдання можна продемонструвати у вигляді схеми взаємодії класів між собою.

На рис. 1 наведено схему діаграми класів для реалізації криптографічних алгоритмів [5] у месенджері з багаторівневим шифруванням. Вона складається з декількох класів та інтерфейсів:

ICrypto – основний інтерфейс, який містить методи, що дають змогу здійснювати шифрування (Crypt) та дешифрування (Decrypt) повідомлень.

SymmetricCryptoAlg та AsymmetricCryptoAlg – абстрактні класи, що забезпечують базове функціонування для симетричних та асиметричних криптографічних алгоритмів відповідно.

PermutationCryptoAlg – конкретна реалізація алгоритму перестановки для реалізації шифрування повідомлень на основі класу SymmetricCryptoAlg.

ShanonFanoAlg – реалізація симетричного алгоритму шифрування, яка використовує реалізовані класи та методи, що безпосередньо не знаходяться в цьому класі, проте викликаються, обгортка реалізується на основі класу SymmetricCryptoAlg.

RSACryptoAlg – реалізація асиметричного алгоритму шифрування на основі абстрактного класу AsymmetricCryptoAlg.

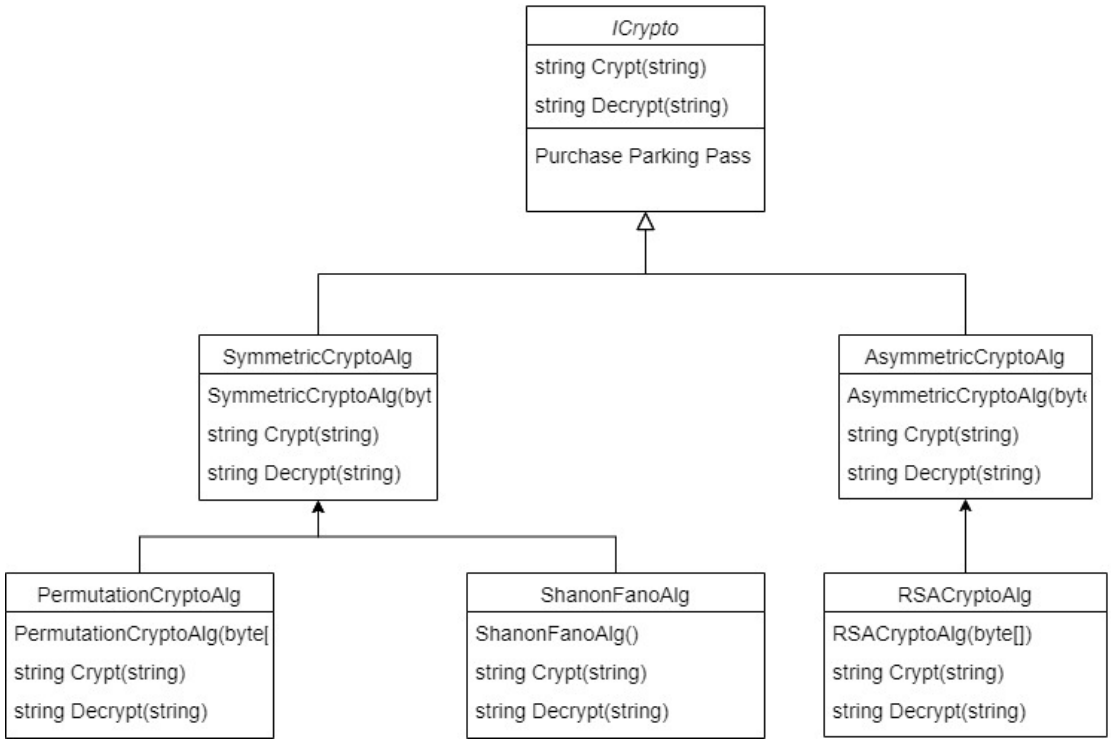

Рис. 1. Діаграма класів криптографічної компоненти

На рис. 2 наведено діаграму фабричних методів, що складається з таких класів:

ICryptoFactory – визначає інтерфейс для створення криптографічних об’єктів через два методи:

– ICrypto Create() – створює об’єкт без додаткових параметрів;

– ICrypto Create(string[] param) – створює об’єкт, приймаючи масив параметрів.

CryptoPermutationFactory – створює об’єкти, що реалізують криптографію з використанням перестановок.

ShanonFanoCryptoFactory – створює об’єкти, що реалізують криптографію з використанням методу Шеннона-Фано.

RSACryptoFactory – створює об’єкти, що реалізують RSA-криптографію.

Отже, ця діаграма демонструє застосування шаблону Factory Method для створення різних видів криптографічних об’єктів через абстрактну фабрику та її конкретні реалізації.

Ця архітектура забезпечує кілька важливих переваг:

Розширюваність – завдяки використанню інтерфейсу ICrypto й абстрактних класів SymmetricCryptoAlg та AsymmetricCryptoAlg додавання нового алгоритму шифрування стає простим завданням. Потрібно лише створити новий клас, який реалізує відповідний інтерфейс, і програма автоматично зможе використовувати цей новий алгоритм.

Гнучкість конфігурації – параметри алгоритмів та послідовність їх застосування може бути індівідуальними для кожного абонента.

Надійність шифрування – для шифрування повідомлення, може використовуватись одночасно декілька різних алгоритмів, що буде підвищувати захищеність повідомлення загалом.

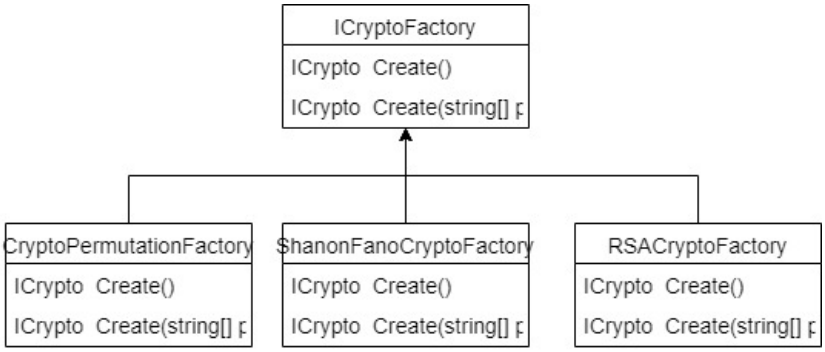

Рис. 2. Діаграма фабричних класів

Повторне використання коду – оскільки алгоритми шифрування реалізовані як окремі класи, їх можна легко повторно використовувати в інших проєктах.

Завдяки цій архітектурі розробники можуть легко додавати, видаляти або змінювати алгоритми шифрування без необхідності внесення змін у весь код програми. Це робить систему гнучкою, масштабованою та простою у підтримці, особливо для великих проєктів з багатьма залежностями.

Список використаних джерел

1. Демашкевич А. В., Антонов Ю. С. Розробка програмного забезпечення для приховування повідомлення в цифрових зображеннях за допомогою методу LSB. Матеріали IV Всеукраїнської науково-практичної конференції *«Комп'ютерні технології обробки даних»*. Вінниця, ДонНУ імені Василя Стуса. 2023. С. 242–245.

2. Антонов Ю. С., Римар П. В., Антонова О. Г. Проблема DoS/DDoS атак навчальних ресурсів здобувачами. *Сучасний захист інформації*. 2019. № 4(40). С. 52–62.

3. Tune N., Perrin J. Architecture Modernization. 2024. 488 с.

4. Design Patterns. URL: https://refactoring.guru/design-patterns

5. Wong D. Real-World Cryptography. 2021. 400 с.

УДК 004.1

Явгусішин Б. А., *здобувач 4 курсу спеціальності 122 Комп'ютерні науки,*
Антонов Ю. С., *канд. фіз.-мат. наук, доцент, доцент кафедри інформаційних технологій*

РЕАЛІЗАЦІЯ КАЗУАЛЬНОЇ ГРИ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ З ВИКОРИСТАННЯМ ПЛАТФОРМИ .NET

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасна ігрова індустрія залучає мільйони гравців з усього світу та отримує великі прибутки. Попри глобальну рецесію та зменшення ігрового ринку через введення обмежень у Китаї, ринок був у передбаченому падінні після двох років зростання через пандемію. Наразі ринок стабілізується, що є гарною можливістю для просунення власних ігрових додатків [1–2]. Враховуючи те, що на світовому ринку ігрової індустрії найуспішнішим та найприбутковішим сектором є ігри для мобільних пристроїв, тема роботи є актуальною.