|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Информационная безопасность

**ОСНОВЫ ООП В ИБ**

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ НА ТЕМУ Паттерны «шаблонный метод» и «стратегия»

Студент:

А.П. Васютина, группа ИУ8-114

Преподаватель:

А.Ю. Быков

Москва, 2022 г.

Цель работы:освоить паттерн программирования с шаблонами на практике.

Постановка задачи:

Реализовать с помощью двух паттернов шаблонный метод и стратегия пример получения хешей некоторых данных с использование не менее двух разных хеш-функций

# Выполнение работы

1. Объявление родительской сущности и реализация шаблонных методов

Листинг 1 – Реализация

**type** HashInterface **interface** {  
 GenerateHash(string) string  
}  
  
**type** SHA256Hash **struct**{}  
  
**func** (sha \*SHA256Hash) GenerateHash(text string) string {  
 byteHash := sha256.Sum256([]byte(text))  
 **return** hex.EncodeToString(byteHash[:])  
}  
  
**type** SHA512Hash **struct**{}  
  
**func** (sha \*SHA512Hash) GenerateHash(text string) string {  
 byteHash := sha512.Sum512([]byte(text))  
 **return** hex.EncodeToString(byteHash[:])  
}

1. Использование родительской сущности для универсализации генерации хэша:

Листинг 2 – Использование родительской сущности

**type** Service **struct** {  
 Storage storage.StorageInterface  
 HashService crypto.HashInterface  
}  
  
**func** NewService(userStorage storage.StorageInterface, hashInterface crypto.HashInterface) \*Service {  
 **return** &Service{  
 Storage: userStorage,  
 HashService: hashInterface,  
 }  
}

1. Вызов генерации хэша родительской сущности:

Листинг 3 – Генерация

**func** (as \*Service) AddUser(user \*entities.User) error {  
 user.Password = as.HashService.GenerateHash(user.Password)  
 **return** as.Storage.AddUser(user)  
}  
  
**func** (as \*Service) ChangePassword(user \*entities.UserPasswordChange) error {  
 user.Password = as.HashService.GenerateHash(user.Password)  
 user.NewPassword = as.HashService.GenerateHash(user.NewPassword)  
 **return** as.Storage.ChangePassword(user)  
}  
  
**func** (as \*Service) Authorize(user \*entities.User) (\*entities.AccessToken, error) {  
 user.Password = as.HashService.GenerateHash(user.Password)  
 **return** as.Storage.Authorize(user)  
}

1. Пример генерации хэша SHA-512 при добавлении пользователя с паролем ‘**123**’:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – SHA-256

1. Пример генерации хэша SHA-256 при добавлении пользователя с паролем ‘**123**’:

Изображение выглядит как текст

Автоматически созданное описание