SDU UNIVERSITY 1996

FINAL PROJECT

BY MANSUR ZHUMAZHAN, KAISAR ORAKOV, MEIRZHAN ANARBEKULY

CONTENT

OUR TEAM

Kaisar Orakov

Project Designer

Mansur Zhumazhan

Project Manager

Meirzhan Anarbekuly

Developer

Introduction

The EngBro project is designed to help people learn languages in a non-standard way. Our team strives to make the language learning process not only effective, but also exciting. The main goal is to demonstrate the effective use of the six design patterns that we reviewed during the course "Design Patterns", emphasizing their importance for improving code maintainability, flexibility and scalability.

SDU University

ENGBRO
LANGUASE LEARNING

THE PURPOSE

The purpose of the EngBro project is to offer an innovative and engaging approach to language learning, leveraging the principles of software design patterns. This project aims to demonstrate the practical application and benefits of six specific design patterns—Singleton, Strategy, Decorator, Adapter, Observer, and Factory Method in the console version and Singleton, Command, Builder patterns—in the GUI version using Spring Boot framework. By integrating these design patterns, the project seeks to showcase how they can enhance code maintainability, flexibility, and scalability, thereby improving the overall effectiveness and reliability of software development in the context of language learning tools

Problem Statement

Our university only teaches in English and it seems that our students should have at least a B1 level, but we feel that many students still have problems with the English language. We may notice that it is difficult for them to study, for example, discrete mathematics from an English book

Solution

ENGBRO

LANGUASE LEARNING

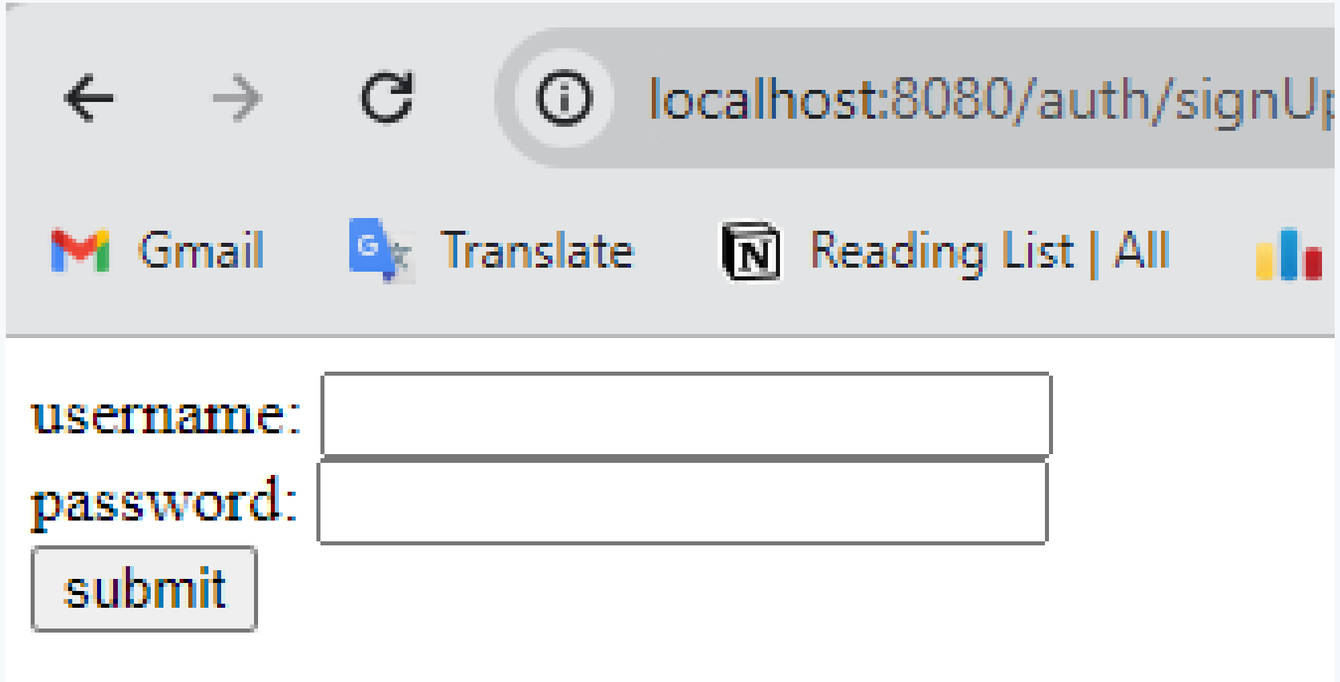
Our project offers a comprehensive solution combining language assessment, personalized training programs and practical exercises to develop skills. We will develop an algorithm that analyzes the user's language abilities and suggests a personalized set of exercises aimed at improving weaknesses in language skills. In addition, we will provide an extensive list of words and phrases important for everyday communication and professional activities that will help learners expand their vocabulary. This approach will allow students to learn English more effectively and purposefully, tailoring the learning process to their individual needs and goals.

final result

- Language Learning Through Interactive Exercises: The project focuses on English language learning through two engaging exercises, "Guess the Word" and "Match Translations," providing users with an interactive and practical way to enhance their language skills.
- Knowledge Assessment Feature: The inclusion of a knowledge assessment feature allows users to evaluate their English proficiency by taking a test. The result provides users with a level indication (A1, A2, B1, etc.), offering personalized feedback on their language proficiency.
- Personalized Dictionary Management: Users can add words to a personal dictionary and easily search for translations between English and Russian. This feature enhances the learning experience by allowing users to build a customized vocabulary.
- Profile Page: All user information, including exercise progress, test results, and personalized dictionary entries, is stored on a user's profile page, providing a centralized hub for tracking and managing language learning activities.

Singleton Pattern

- The purpose of the `UserManagerSingleton` class is to provide a global access point to user management in the application using a single instance of the `UserManager` class. This ensures uniformity in user management, reduces the risk of conflicts and increases resource efficiency.
- The static `getInstance` method provides a global access point to a single instance of the `UserManagerSingleton' class. If an instance has not been created yet, the method creates it.



Observer and Strategy

ExerciseObserver Interface:

• Represents an observer that can be notified about the user's exercise performance.

• Contains the update method, which takes the user's score and the total number of words in the exercise.

Game to guess the word in English. User can choose the difficulty, which represented by Strategy pattern. By the end of the game user gets the number of correct answers by notification, represented by Observer pattern.

```java
public interface ExerciseObserver {
    1 usage   1 implementation
    void update(int score, int totalWords);
}
```

```java
private void playGuessWordGame(Scanner scanner) {
    int numCorrect = 0;

    System.out.println("Выберите сложность:");
    System.out.println("1) Стандартная");
    System.out.println("2) Невозможная");

    int strategyChoice = scanner.nextInt();

    if (strategyChoice == 1) {
        wordMaskingContext.setStrategy(new DefaultWordMaskingStrategy());
    } else if (strategyChoice == 2) {
        wordMaskingContext.setStrategy(new AllLettersMaskingStrategy());
    } else {
        System.out.println("Неверный выбор сложности");
        return;
    }

    System.out.println("Угадайте слово:");

    for (WordToGuess word : wordsToGuess) {
        String maskedWord = wordMaskingContext.maskWord(word.getEnglishWord());
        System.out.print(word.getRussianWord() + ": " + maskedWord + " ");
        String userGuess = scanner.next();

        if (userGuess.equalsIgnoreCase(word.getEnglishWord())) {
            System.out.println("Правильно!");
            numCorrect++;
        } else {
            System.out.println("Неверно. Правильный ответ: " + word.getEnglishWord());
        }
    }

    notifyObservers(numCorrect, wordsToGuess.size());
}
```

```java
private void matchTranslations(ArrayList<WordPair> pairsToMatch) {
    Scanner scanner = new Scanner(System.in);
    int numCorrect = 0;

    System.out.println("Найдите соответствие для каждого русского слова:");

    for (WordPair pair : pairsToMatch) {
        System.out.print(pair.getRussianWord() + ": ");
        String userGuess = scanner.next();

        if (userGuess.equalsIgnoreCase(pair.getEnglishWord())) {
            System.out.println("Правильно!");
            numCorrect++;
        } else {
            System.out.println("Неверно. Правильный ответ: " + pair.getEnglishWord());
        }
    }

    notifyObservers(numCorrect, pairsToMatch.size());
}
```

Adapter

The `DictionaryAdapter` is an adapter for the `Translator` interface, providing an adaptation to the use of the `Dictionary` class as a translation source. This adapter provides a link between the interface and the `Dictionary` functionality, allowing the latter to be used in a context focused on the expected `Translator` interface. It tracks the time spent on each translation, providing additional functionality for performance monitoring.

```java
public class DictionaryAdapter implements Translator {
    2 usages
    private Dictionary dictionary;
    2 usages
    private long lastTranslationTime;

    no usages
    public DictionaryAdapter(Dictionary dictionary) { this.dictionary = dictionary; }

    2 usages
    @Override
    public String translateToRussian(String englishWord) {
        long startTime = System.currentTimeMillis();
        String translation = dictionary.translate(englishWord);
        long endTime = System.currentTimeMillis();
        lastTranslationTime = endTime - startTime;
        return translation;
    }

    2 usages
    @Override
    public long getLastTranslationTime() { return lastTranslationTime; }
}
```

THANK'S FOR ATTENTION

We really tried. Hope for a good grade<3