Lab Report

Project Title: CityCompass – Explore The World
Course Title: Object-Oriented Programming Language Lab
Course Code: CSE0613122
Submitted To:

Mr. Al-Imtiaz
Associate Professor & Head, Dept. Of CSE

University of Information Technology & Sciences (UITS)

Propa Punam

Lecturer, Dept. Of CSE

University of Information Technology & Sciences (UITS)

Submitted By:

Meherab Hasan Sabit – 0432410005101008
Jawad-Ul-Karim – 0432410005101029
Arafat Rahman – 0432410005101030
Atika Akter Mitu – 0432410005101050

Date: 15-12-2024

Abstract

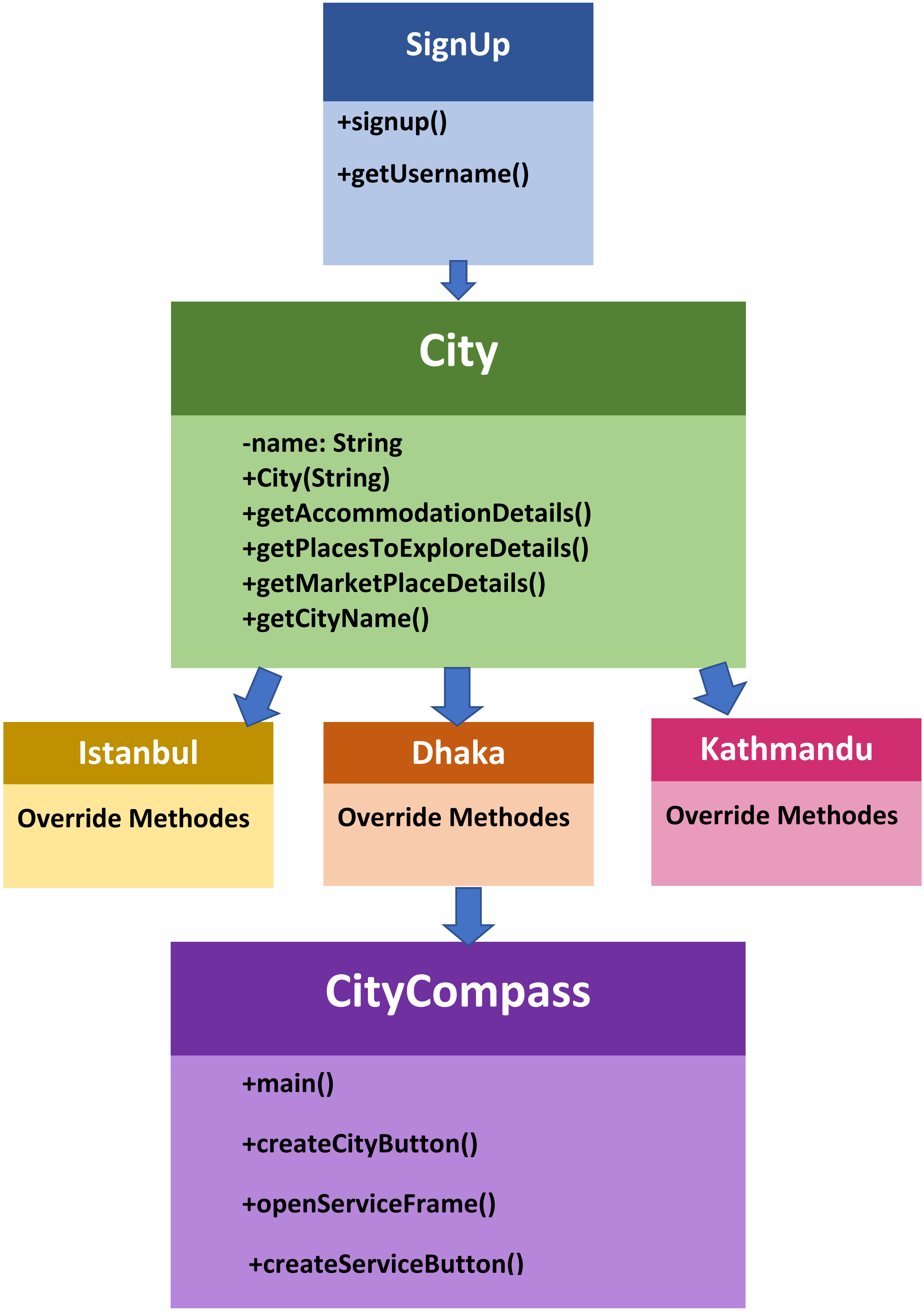
CityCompass is a Java-based interactive console application designed to help users explore information about various cities. It enables users to sign up, choose a city, and explore details about accommodations, tourist attractions, and marketplaces in three cities: Istanbul, Dhaka, and Kathmandu. The project demonstrates the use of object-oriented programming principles, including inheritance, abstraction, encapsulation, and polymorphism.

Objectives

1. To implement object-oriented programming concepts in Java.
2. To design and develop an interactive console-based application.
3. To explore the practical application of abstract classes and inheritance.
4. To create a modular, maintainable, and user-friendly Java program.

Equipment and Components

- **Hardware**:
  - Computer with a Java Development Kit (JDK) installed.
- **Software**:
  - Integrated Development Environment (IDE) – NetBeans & VsCode
  - Java Runtime Environment (JRE).
- **Programming Language**: Java



SignUp

+signup()

+getUsername()

City

-name: String
+City(String)
+getAccommodationDetails()
+getPlacesToExploreDetails()
+getMarketPlaceDetails()
+getCityName()

Istanbul

Override Methodes

Dhaka

Override Methodes

Kathmandu

Override Methodes

CityCompass

+main()

+createCityButton()

+openServiceFrame()

+createServiceButton()

Theory

The project implements key object-oriented programming (OOP) principles:

1. **Encapsulation**:
   - The `SignUp` class uses private attributes for user information, ensuring data integrity.
2. **Inheritance**:
   - The `City` abstract class serves as the parent class, and subclasses (`Istanbul`, `Dhaka`, `Kathmandu`) inherit and implement its methods.
3. **Abstraction**:
   - The `City` class defines abstract methods for accommodation, places to explore, and marketplaces, requiring subclasses to provide specific implementations.
4. **Polymorphism**:
   - The application dynamically determines the behavior of methods based on the city object selected by the user.

Methodology

1. **User Registration**:
   - The `SignUp` class collects and stores user details.
2. **City Selection**:
   - The `createCityButton()` method allows users to select one of the predefined cities.
3. **Module Selection**:
   - The `createServiceButton()` method enables users to explore accommodations, tourist spots, or marketplaces.
4. **Dynamic Method Calls**:
   - Based on the selected city, relevant details are displayed using overridden methods in subclasses.

Code:

```
package CityCompassProject;
import javax.swing.*;
import java.awt.*;
class SignUp {
    private static String userName;
    private static String userEmail;
    public static void signUp() {
        while (true) {
            try {
                userName = JOptionPane.showInputDialog("Enter your name:");
                if (userName == null || userName.isEmpty()) throw new Exception("Name cannot be empty!");
                userEmail = JOptionPane.showInputDialog("Enter your email:");
                if (!userEmail.contains("@")) throw new Exception("Email must contain '@'!");
                JOptionPane.showMessageDialog(null, "Welcome to CityCompass, " + userName + "!");
                break;
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
            } } }
    public static String getUserName() {
        return userName;
    }
}
abstract class City {
    protected String name;
    public City(String name) {
        this.name = name;   }
    public abstract String getAccommodationDetails();
    public abstract String getPlacesToExploreDetails();
    public abstract String getMarketPlaceDetails();
    public String getCityName() {
        return name;    }}
class Istanbul extends City {
    public Istanbul() {
        super("Istanbul");    }
    @Override
    public String getAccommodationDetails() {
        return """
            1. Ramada Plaza - A luxurious 5-star hotel located in the Sisli district, offering fine dining and a rooftop pool.
                2. Grand Faith Hotel - A cozy 3-star hotel in Refik Caddesi, ideal for budget travelers.
 3. The Marmara Taksim - A premium hotel at the heart of Istanbul, offering stunning city views and modern
amenities.      """;
   }
   @Override
    public String getPlacesToExploreDetails() {
        return """
                1. Galata Tower - A medieval stone tower with panoramic views of the Bosphorus.
            2. The Blue Mosque - An architectural marvel known for its blue tiles and domes.
            3. Topkapi Palace - A historical palace showcasing Ottoman artifacts and grandeur.
            """;   }
   @Override
   public String getMarketPlaceDetails() {
      return """
            1. Istinye Park Mall - A modern mall with luxury brands and entertainment.
            2. The Grand Bazaar - One of the world's oldest covered markets with thousands of shops.
            3. Spice Bazaar - Famous for exotic spices, Turkish delights, and dried fruits           """;
   }}
class Dhaka extends City {
   public Dhaka() {
      super("Dhaka");   }
  @Override
   public String getAccommodationDetails() {
      return """
            1. Pan Pacific Sonargaon Hotel - A luxurious 5-star hotel with traditional hospitality.
            2. Hotel 71 - A budget-friendly option in the heart of Dhaka.
            3. Radisson Blu Water Garden - A premium hotel with serene gardens and modern facilities.
            """; }
   @Override
   public String getPlacesToExploreDetails() {
      return """
            1. Lalbagh Fort - A Mughal fort offering a glimpse into Dhaka's rich history.
            2. Ahsan Manzil - The Pink Palace, a historic residence of Dhaka's Nawabs.
            3. Hatirjheel - A beautiful lake area, perfect for evening strolls and boating.
            """; }
   @Override
   public String getMarketPlaceDetails() {
      return """
            1. Bashundhara City - One of South Asia's largest malls with shops and eateries.
            2. New Market - A bustling hub for bargain shopping.
            3. Jamuna Future Park - A modern shopping mall with entertainment and dining options.
            """   }}
class Kathmandu extends City {
   public Kathmandu() {
      super("Kathmandu");  }
   @Override
   public String getAccommodationDetails() {
      return """
            1. Hyatt Regency - A luxurious 5-star resort offering Himalayan views.
            2. Hotel Yak & Yeti - A heritage hotel blending modern amenities with Nepalese charm.
            3. Fairfield by Marriott - A modern hotel offering comfort and convenience.
            """; }
  @Override
   public String getPlacesToExploreDetails() { return """
            1. Swayambhunath Stupa - The Monkey Temple with panoramic valley views.
            2. Pashupatinath Temple - A UNESCO World Heritage Site and a sacred Hindu temple.
            3. Durbar Square - A historical complex of palaces, temples, and courtyards.  """;    }
```

```
] @Override
    public String getMarketPlaceDetails() {
        return """
                1. Thamel Market - A tourist area with shops selling handicrafts and trekking gear.
                2. Durbar Square Market - Surrounded by ancient temples and palaces.
                3. Asan Bazaar - A vibrant local market for spices, textiles, and fresh produce.
                """;    }}
public class CityCompass {
    public static void main(String[] args) {
        SignUp.signUp();
        JFrame mainFrame = new JFrame("CityCompass - Explore the World");
        mainFrame.setSize(700, 500);
        mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mainFrame.setLayout(new GridLayout(2, 2, 10, 10));
        JButton istanbulButton = createCityButton("Explore Istanbul", new Istanbul(), new Color(173, 216, 230));
        JButton dhakaButton = createCityButton("Explore Dhaka", new Dhaka(), new Color(221, 160, 221));
        JButton kathmanduButton = createCityButton("Explore Kathmandu", new Kathmandu(), new Color(240, 128,
128));
        mainFrame.add(istanbulButton);
        mainFrame.add(dhakaButton);
        mainFrame.add(kathmanduButton);
        mainFrame.setVisible(true); }
    private static JButton createCityButton(String label, City city, Color color) {
        JButton button = new JButton(label);
        button.setBackground(color);
        button.addActionListener(e -> openServiceFrame(city));
        return button;  }
    private static void openServiceFrame(City city) {
        JFrame serviceFrame = new JFrame("CityCompass - " + city.getCityName());
        serviceFrame.setSize(700, 500);
        serviceFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        serviceFrame.setLayout(new GridLayout(1, 3, 10, 10));
        JButton accommodationButton = createServiceButton("Accommodation", city.getAccommodationDetails(),
new Color(135, 206, 235));
        JButton placesButton = createServiceButton("Places to Explore", city.getPlacesToExploreDetails(), new
Color(216, 191, 216));
        JButton marketButton = createServiceButton("Marketplace", city.getMarketPlaceDetails(), new Color(255,
182, 193));
        serviceFrame.add(accommodationButton);
        serviceFrame.add(placesButton);
        serviceFrame.add(marketButton);
        serviceFrame.setVisible(true);    }
    private static JButton createServiceButton(String label, String details, Color color) {
        JButton button = new JButton(label);
```

○

Observations

1. The program successfully prompts the user for their name and email during signup.
2. The application provides a graphical user interface (GUI) for city selection and exploration using JFrame and JButton.
3. Clicking on city-specific buttons (e.g., "Explore Istanbul") opens new windows with options for exploring accommodations, tourist spots, and marketplaces.
4. Dynamic data is displayed using the overridden methods in the respective city classes (Istanbul, Dhaka, Kathmandu).
5. Exception handling ensures the program prompts the user correctly when invalid input is entered during signup.

Results

1. The application efficiently collects and validates user details during the signup phase.
2. The GUI is interactive and user-friendly, providing clear navigation for exploring city features.
3. The abstract class City and its subclasses (Istanbul, Dhaka, Kathmandu) work cohesively to deliver accurate and city-specific information.
4. Each button click dynamically loads appropriate content using method overrides.

Discussion and Analysis

1. **Strengths**: The program demonstrates strong adherence to object-oriented principles like abstraction, inheritance, and polymorphism.

   - GUI components such as JFrame and JButton enhance user interactivity and engagement.
   - The modular structure of the code ensures scalability. Additional cities can be easily added by creating new subclasses of the City abstract class.
2. **Areas for Improvement**:
   - The design could benefit from more advanced error handling during city exploration to address unforeseen issues (e.g., UI freezing or missing data).
   - User preferences or login sessions could be saved for repeated usage using file handling or a database.
   - More cities or additional details (e.g., cuisine, transport options) could enrich the user experience further.
3. **Performance**:
   - The GUI operations are lightweight and responsive for the given functionalities.
   - The program structure is efficient and avoids redundancy due to the use of abstract classes and method overriding.

Conclusion

The CityCompass project is a robust demonstration of object-oriented programming principles combined with basic GUI development in Java. It effectively engages users in exploring cities with dynamic and user-friendly features. The modular code design ensures easy maintenance and future scalability. By incorporating additional features like persistent data storage or more cities, the application could be further enhanced to deliver an even more engaging user experience.

References

Mr. Al-Imtiaz

Associate Professor & Head, Dept. Of CSE University of Information Technology & Sciences (UITS)

Propa Punam

Lecturer, Dept. Of CSE

University of Information Technology & Sciences (UITS)