Java Programming 1 Project Description – Gloire Mpoyi

Project Title: Creative Habit Tracker App

Abstract

This app is for someone who wants to commit to hobbies or a craft whilst they continue to be accountable and organized to complete their tasks on time. The application is meant to help with organization and consistency by introducing a gamified approach to maintain engagement using a reward system and customizable avatars.

Object-oriented design

The main classes expected for my implementation - Task, User, Avatar, and Reward class.

Reward class – defines common properties and methods for rewards that can be granted to users. This class is abstract because the specific reward could vary for each task or avatar, but the concept remains the same.

Task class - Represents a task that the user needs to complete. This class extends the base class reward, so it is represented as a concrete class because specific instances of tasks are required.

User class - Represents the user of the program who has tasks to complete and an avatar to update based on rewards. This concrete class manages and avatar states

Avatar class - Represents the user's avatar that changes based on completed tasks. This concrete class modifies avatar properties as tasks are completed.

Class Relations

User → Task - Aggregation — The User maintains a list of Task objects that can exist independently.
User → Reward - Aggregation — The User keeps track of Reward objects that are unlocked.
Task → Reward - Composition — Certain tasks have embedded reward logic, meaning Reward objects are tied to specific tasks.

Avatar → User - Composition — The Avatar is a part of the User and is dependent on it.
Reward → Avatar - Association — Rewards interact with the Avatar to update properties.

Inheritance and polymorphism applications

Reward Class - Extended by BasicReward, MilestoneReward, and CustomItemReward, each with custom behaviors (e.g., applyReward()).

Polymorphism - The User class can handle lists of tasks and rewards polymorphically, allowing seamless iteration and method calls such as task.completeTask() or reward.applyReward().