PoolTool

PROJEKT REALIZOWANY W RAMACH PIPR

OPIS PROJEKTU I JEGO CEL

PoolTool jest programem działającym w terminalu, który realizuje projekt mający na celu stworzenie systemu obsługi pływalni. Program przechowuje informacje o nazwie danej pływalni, liczbie jej torów, a także o godzinach jej pracy i cenniku. Umożliwia on dodawanie dwóch rodzajów nowych rezerwacji – dla klientów indywidualnych, a także dla szkółek pływackich. Oprócz tego generuje podsumowanie przychodów pływalni w danym dniu.

Program umożliwia podgląd dostępnej liczby biletów oraz dostępnych torów dla danych przedziałów czasowych. Uwzględnia on ich dostępność podczas dodawania nowych rezerwacji, a w przypadku jej braku, automatycznie proponuje najbliższy możliwy termin rezerwacji na tych samych warunkach (tj. na identyczną ilość czasu oraz, w przypadku szkółek pływackich, na taki sam tor).

Dodatkowo PoolTool posiada specjalny panel administracyjny, który pozwala na ustawienie obecnego dnia. Informacja ta jest wykorzystywana dla wszystkich pływalni. Program jest odporny na wszelkie nieprawidłowości przy wprowadzaniu danych – w przypadku ich wystąpienia, wyświetla stosowne komunikaty i umożliwia dalszą pracę.

ARCHITEKTURA PROGRAMISTYCZNA

Realizacja projektu PoolTool opiera się o architekturę **model-widok**, która umożliwia separację warstwy logicznej aplikacji od warstwy obsługującej interakcje z użytkownikiem. Dodatkowo program posiada niezależny moduł *io_manager*, który wspomaga obsługę plików.

Kod źródłowy programu składa się z następujących klas i modułów:

- ***Model:***
  - Moduł *value_types*:
    - **Price** – reprezentuje cenę, osobno przechowuje informacje o części złotych i groszy.
    - **HoursRange** – reprezentuje zakres godzin, przechowuje informacje o czasie początkowym i końcowym.
    - **Services** – typ wyliczeniowy posiadający literały reprezentujące rodzaj usługi (klient indywidualny lub szkółka pływacka).
    - **WeekDay** – typ wyliczeniowy posiadający literały reprezentujące kolejne dni tygodnia.
  - Moduł *price_list_model*:

        - **PriceListPosition** – reprezentuje pojedynczą pozycję na cenniku.
        - **PriceListModel** – reprezentuje cały cennik pływalni.
    - Moduł *reservations_model*:
        - **Reservation** – przechowuje informacje o pojedynczej rezerwacji dla klienta indywidualnego.
        - **SchoolReservation** – klasa dziedzicząca po *Reservation*, dodatkowo przechowuje informacje o przypisanym numerze toru, zatem przechowuje informacje o rezerwacji dla szkółki pływackiej.
        - **ReservationSystemModel** – reprezentuje cały system rezerwacji – przechowuje je, umożliwia ich dodawanie, a także zlicza całkowity przychód.
    - Moduł *pool_model*:
        - **PoolModel** – klasa łącząca całą warstwę logiczną aplikacji w całość. Przechowuje informacje o pływalni, a także posiada referencje do systemu rezerwacji i cennika.
- ***Widok:***
    - Moduł ***admin_view*** – zawiera funkcje obsługujące interakcje z użytkownikiem dla panelu administracyjnego
    - Moduł ***pool_view*** – zawiera funkcje obsługujące interakcje z użytkownikiem dla obsługi pływalni.
- ***Moduł io_manager*** – zawiera funkcje zapisujące i odczytujące z plików
- ***Moduł admin*** – zawiera klasę **Admin**, przechowuje ona informacje o konfiguracji programu i umożliwia zarządzanie nią.
- ***Plik pooltool.py*** – stanowi punkt wejścia dla całego programu. Obsługuje argumenty uruchomienia programu i wywołuje funkcje startowe widoku.

OBSŁUGA PROGRAMU

URUCHAMIANIE

PoolTool może pracować w dwóch trybach – trybie administracyjnym oraz trybie obsługi pływalni. Będąc w głównym katalogu projektu (***pooltool***) możemy wywołać w terminalu następujące komendy:

- **`python ./pooltool.py -a`** – uruchamia tryb administracyjny. Tryb ten pozwala na zmianę obecnego dnia na pływalniach, można go ustawić na następny lub poprzedni.
- **`python ./pooltool.py -p [ścieżka do pliku]`** – uruchamia tryb obsługi pływalni, która jest wczytywana z podanego pliku.

ODCZYT I ZAPIS PLIKÓW

PoolTool pracuje z plikami w formacie JSON. Do poprawnego działania programu wymagany jest plik ***config.json*** znajdujący się w katalogu głównym projektu. Przechowuje on informację o obecnym dniu na pływalniach. Nie musi być on jednak tworzony ręcznie – gdy program wykryje, że taki plik nie istnieje, pozwoli na jego szybkie utworzenie po podaniu startowego numeru dnia, miesiąca i roku po uruchomieniu trybu administracyjnego. Przykładowy plik *config.json* znajduje się w katalogu ***example_files***.

Plik, który podawany jest dla programu w trybie obsługi pływalni, musi zawierać początkowe informacje o basenie – jego nazwę, liczbę torów, godziny pracy oraz cennik. Przykładowy plik wejściowy basenu (w katalogu przykładowych plików) ma nazwę ***valid_pool.json***. Dodatkowo umieszczona jest także skrócona wersja przykładowego pliku wejściowego (***valid_pool_template.json***), która pozwala na szybkie stworzenie własnej konfiguracji basenu, dzięki kopiowaniu pozycji cennika i godzin pracy i wpisywaniu tam własnych wartości.

Kilka informacji dot. plików:

- Wpisywane godziny (zarówno w cenniku jak i w godzinach pracy) muszą mieć wartość minut równą **0 lub 30**. Ograniczenie to dotyczy całego programu – oznacza to, że pracuje on **w półgodzinnych odstępach czasu**. Basen musi być otwarty przez minimum 1 godzinę w dniu.
- Klucze w słowniku *working_hours* są cyframi oznaczającymi numer dnia tygodnia, gdzie **0** – poniedziałek, a np. **6** – niedziela.
- Liczba torów (*lanes_amount*) musi wynosić **co najmniej 3**, aby można było w ogóle dokonać rezerwacji dla szkółek pływackich.
- Klucz *service* dla pojedynczej pozycji w cenniku (*price_list*) oznacza rodzaj klienta, którego ta pozycja dotyczy. Wartość **0** oznacza klienta indywidualnego, a **1** – szkółkę pływacką. Klucz *day* przyjmuje takie wartości dnia tygodnia, jak opisano wyżej.
- Cennik musi całkowicie pokrywać godziny pracy basenu, a także musi być określony dla każdej usługi. Oznacza to, że może być on zróżnicowany nawet względem godzin (można to określić odpowiednio ustalając wartości dla klucza *hours_range*), ale nie może być niezgodny z godzinami pracy basenu, godziny określone w cenniku

dla danej usługi nie mogą się nakładać ani być rozdzielne, a także wszystkie pozycje cennika razem muszą odpowiednio pokrywać godziny pracy basenu.

- Pliki basenów możemy wczytywać z dowolnej lokalizacji. Po wczytaniu pliku i dodaniu nowych rezerwacji dla basenu, program zapisze plik, ale **w katalogu głównym projektu**, pod tą samą nazwą jak wczytany plik. Dzięki temu program nie będzie nadpisywał utworzonych startowych plików wejściowych, a także wszystkie pliki basenów zawierające rezerwacje znajdują się w jednym miejscu.

W katalogu z przykładowymi plikami znajdują się również przykładowe niepoprawne pliki startowe basenów (***invalid_pool_X.json***). Plik nr 1 ma niepoprawne i brakujące klucze, nr 2 zawiera cennik niepokrywający godzin pracy i nieokreślony dla wszystkich usług, nr 3 ma zbyt małą liczbę torów, a basen w pliku nr 4 ma zbyt krótkie godziny otwarcia. Próba wczytania tego typu plików spowoduje wyświetlenie odpowiedniego komunikatu o błędzie.

OBSŁUGA BASENU

Po poprawnym wczytaniu basenu program nas przywita, wyświetli informację o nazwie basenu i obecnym dniu, a także wyświetli dostępne akcje. Akcję wybieramy przez wprowadzenie odpowiadającej jej liczby. W zależności od wybrania odpowiednich opcji, program będzie pytał o dane do wprowadzenia (np. numer dnia, miesiąca lub roku dla nowej rezerwacji) lub wyświetli odpowiednie dane. Istnieje możliwość anulowania dodawania nowej rezerwacji (ze względu na złożoność tej akcji), a także istnieje akcja wyjścia z programu.

W przypadku chęci przejścia do innego dnia, należy uruchomić program w trybie administracyjnym, zmienić dzień, a następnie uruchomić program w trybie obsługi basenu – można wówczas kontynuować pracę.

PODSUMOWANIE PRACY NAD PROJEKTEM

Całość pracy nad projektem rozpoczęła się od dokładnego rozplanowania działania programu, a także utworzenia pierwszego planu architektury programistycznej aplikacji. Praca nad kodem źródłowym projektu przebiegała z użyciem systemu Git, a sam projekt już od początkowego etapu był zarządzany na platformie GitLab. Pozwoliło to na dobrą organizację zadań i ich rozplanowanie w czasie, dzięki wykorzystaniu funkcji dostępnych na tej platformie.

Pierwszy plan projektu zakładał wykorzystanie architektury MVVM, jednak w nieco późniejszym etapie przekształciła się ona w architekturę model-widok, z pominięciem warstwy modelu widoku. Zmiana ta była nieoczekiwana, jednak okazała się bardzo dobra w kontekście sprawności realizacji całego projektu. Początkowy pomysł nie zakładał również

istnienia panelu administracyjnego – zmiana ta również przyniosła korzyści, w tym przypadku związane z ujednoliceniem czasu na wszystkich pływalniach.

PoolTool nie jest pozbawiony wad – w moim odczuciu największym problemem związanym z programem jest konieczność utworzenia dość skomplikowanego pliku z informacjami o basenie, a w przypadku gdy zechcemy utworzyć cennik zróżnicowany godzinowo, plik wejściowy staje się naprawdę złożony. Dodatkowo program niszczy wygodne wizualnie formatowanie pliku JSON po zapisie rezerwacji – w takim stanie możliwa jest np. zmiana cennika basenu, ale staje się ona wtedy niewygodna. Problem ten nie został rozwiązany ze względu na konieczność utworzenia dość skomplikowanego systemu inicjalizacji basenu i edycji informacji przede wszystkim o cenniku, który musiałby weryfikować na bieżąco pokrycie jego godzin z godzinami pracy. Próba zaimplementowania takiego systemu mocno wydłużyłaby czas pracy nad projektem.

Pomimo pewnych niedoskonałości, PoolTool bardzo dobrze radzi sobie z powierzanymi mu zadaniami i spełnia wszystkie założenia projektowe. Obsługa pływalni jest szybka i wygodna, a sam program jest odporny na błędy, wobec czego trudno o nieoczekiwane przerwanie jego pracy. Kod źródłowy został napisany zgodnie z dobrymi praktykami oraz spełnia zasady rozgraniczenia logiki aplikacji od jej interfejsu. Dodatkowo program posiada dużo szczegółowych testów, dzięki czemu jego działanie jest dobrze sprawdzone i gwarantuje niezawodność.

BARTOSZ KISŁY, NR INDEKSU 318670