03.12.2023 r.

Michał Iwicki, Michał Kukla, Michał Zajączkowski

Konspekt projektu „Kontrola budżetu”

Chcemy stworzyć aplikację, która będzie pomagać w zarządzaniu budżetem domowym. Ułatwi ona śledzenie wydatków i będzie podsumowywać, ile zostało pieniędzy w gospodarstwie domowym pod koniec każdego miesiąca.

Podstawową klasą aplikacji będzie klasa Wydatek z atrybutami:

* kwota – ile zapłaciliśmy,
* data – kiedy zapłaciliśmy,
* użytkownik – kto zapłacił,
* opis – za co zapłaciliśmy.

Klasy dziedziczące po klasie Wydatek:

* WydatekStały – powtarzające się wydatki ze stałą kwotą (np. czynsz),
* WydatekZmienny – powtarzające się wydatki ze zmienną kwotą (np. prąd, gaz),
* WydatekNieprzewidziany (jak lekarz, hydraulik),
* WydatekCodzienny (np. jedzenie, picie),
* WydatekNiekonieczny (jak np. wyjście do kina, na basen, zakup książki, itd.),

- WydatekDoZwrotu (np. wydatek jednego użytkownika na drugiego, który musi być zwrócony) – dziedziczy po WydatekNiekonieczny.

* WydatekInwestycyjny (np. komputer, meble, ubrania).

Kolejną główną klasą jest Wpływy z atrybutami:

* kwota,
* data,
* użytkownik,
* opis.

Klasy dziedziczące po Wpływy:

* WpływRegularny (tyle, ile zarabiamy co miesiąc – inwestycje + praca),
* WpływNieregularny (premia w pracy),
* WpływZPoprzedniegoMiesiąca (jaki bilans przechodzi z poprzedniego miesiąca na nowy).

Klasa BudżetMiesięczny z atrybutami:

* bilans - ile zostało nam pieniędzy pod koniec miesiąca,
* listaWydatków,
* listaWpływów,
* sumaWydatków,
* sumaWpływów.

Metody klasy BudżetMiesięczny:

* skumulujWydatkiZDanejKategorii,
* podsumujBudżet,
* skumulujWydatkiUżytkownika.

Klasa Użytkownik zawiera atrybuty:

* nazwa,
* oszczędności.

Metody klasy Użytkownik:

* dodajWydatek,
* usuńSwójWydatek.
* modyfikujWydatek.
* getWydatki – wypisuje listę wydatków.
* podsumujWydatkiZaMiesiąc.

Klasy dziedziczące po Użytkownik to UżytkownikZarabiający z dodatkowymi metodami:

* zróbOpłatę,
* zasilBudżet.

Mamy jedną instancję klasy BudżetMiesięczny i wiele instancji klasy Użytkownik, które go modyfikują. Do aplikacji będzie dodane GUI. Z upływem czasu projekt będzie dalej rozwijany.

I:

* backend
* użytkownicy i wydatki

K:

* GUI – logowanie użytkownika, wyświetlanie, obsługa danych

Z:

* wizualizacja

Wstępna dokumentacja do backendu:

Wszystkie wpływy i wydatki dziedziczą po głównej klasie Transakcja.

Można w niej modyfikować kwote date opis i uwzględnianie w budzecie.

Po nich dziedziczą Wpływy i Wydatki. Wpływy dzielą się na Cykliczne i Nieregularne.

Cykliczne Wpływy i Wydatki implementują interface Cykliczny zawierający metody realizuj , przypomnij i setZrealizowany.

Transakcje cykliczne znajdują się na oddzielnej liście w budżecie i dopiero po zrealizowaniu są przenoszone do listy z wszystkimi transakcjami, robi to metoda realizuj.

Przypomnij będzie przy wejściu w aplikacje przypominało o nadchodzących opłatach i wpływach. setZrealizowany obsługuje zmienną zrealizowany.

WydatekDoZwrotu to rozszezony WydatekNieregularny który ma dodatkową zmienną dłużny która przechowuje który użytkownik ma oddac pieniądze temu który wykonał transakcje. Można potem powiadomienia wysyłać(trzeba coś w tym wymyślić). Oraz metodą zapisującą date oddania pieniędzy przez dłużnego. Po oddaniu pieniędzy transakcja przechodzi na dłużnego i jest przypisana do jego konta.

Wydatek nieregularny ma na razie 4 typy jak z wcześniejszego punktu dokumentacji.

Zmienna uwzglednicWBudzecie rozdziela transakcje na te prywatne dla danego użytkownika i te które opisują wydatki w budżecie całej rodziny(głównie cykliczne wydatki oraz te niezbędne).

Transakcje dodaje się do budżetu za pomocą użytkownika.

UzytkownikZarzadzajacy może dodatkowo zarządzać cyklicznymi transakcjami, otworzyć nowy miesiąc archiwizując stary, i generować analizę całego budżetu.

Budzet ma 2 listy zwykłych transakcji i tych cyklicznych reszta

Przykładowy wygląd:

