

Centrum obsługi kart płatniczych obsługuje płatności kartami kredytowymi, debetowymi i bankomatowymi. Z żądaniem autoryzacji każdej płatności mogą zwracać się sklepy, zakłady usługowe i firmy transportowe (to są klienci centrum). Centrum autoryzuje płatność po wysłaniu zapytania do banku który wydał daną kartę płatniczą a informację o tym archiwizuje w postaci wpisu w swoim systemie bazodanowym, które można przeszukiwać.

Uwaga: klienci firm są też klientami banku i tam mają konta - nie muszą być rejestrowani w firmach, płatność odbywa się (jak w rzeczywistości) podając dane karty i kwotę. Są 2 relacje klient: firma jest klientem centrum, osoba jest klientem firmy i zarazem klientem banku, w którym ma konto.

Minimalny zakres funkcjonalności:

- a. zarządzanie firmami korzystającymi z centrum centrum i bankami (dodawanie, usuwanie, przegląd)
- b. zarządzanie kartami (przypisana do klienta banku, wydana przez bank)
- c. obsługa żądań autoryzacji (kwota, data, dane karty) - decyzja zależna od rodzaju karty
- d. zapis i odczyt stanu systemu na dysk (realizacja archiwum operacji w postaci odrębnego pliku)
- e. przeszukiwanie archiwum za pomocą złożonych warunków (firma, bank, numer karty, właściciel, kwota, warunki OR i AND)

Kardynalne wskazówki dotyczące kodu:

1. Stosuj znaczące nazwy
2. Oddzielaj interfejs od logiki ! :
 - o używaj parametrów i wyników zamiast czytać z klawiatury/pisać na ekran)
 - o klasy logiki niezależne od klas interfejsowych, klasy interfejsowe zależne od klas logiki
3. Długie metody powinny być podzielone (zasada 1 ekranu)
4. Ograniczaj używanie konstrukcji 'if' - maks. 3 poziomy zagłębienia
5. Komentarze - kod samo-dokumentujący

Wskazówki dotyczące prezentacji gotowych programów

1. Programy prezentowane są publicznie z użyciem projektora na komputerach pracowni lub studentów.
2. Prezentowany program musi być zgodny z kodem załadowanym do Moodle.
3. Prezentowany program jest obsługiwany przez prowadzącego - ale może on poprosić o to jego autorów.
4. Wskazane jest dostarczenie przykładowego, zapisanego stanu programu, aby można było go wczytać i uniknąć żmudnego wprowadzania danych początkowych.