Autor: Piotr Sawicki

Projekt: Symulacja oddziału banku

Założenia:

1. W banku zarejestrowanych jest n klientów, w oddziale znajduje się m kas, symulacja trwa p minut – parametry n, m, p mogę być podane jako argumenty uruchomienia programu lub mogą być wczytane z pliku, którego nazwa podana jest jako pierwszy argument. Nazwa nie może zaczynać się od cyfry.
2. Klienci banku dzielą się na indywidualnych oraz biznesowych. Mogą oni wykonywać te same operacje. Obsługa klientów biznesowych trwa 5 minut dłużej. Jeżeli w banku jest więcej niż 1 kasa to 10% kas jest dostępna tylko dla klientów biznesowych (przy czym zawsze jest 1 taka kasa).
3. W każdym banku znajduje się dokładnie jeden wpłatomat oraz jeden bankomat.
4. Dostępne operacje dla klientów to: sprawdzenie informacji dot. konta, zmiana PIN, wpłata gotówki (tylko wpłatomaty i kasy), wypłata gotówki (tylko bankomaty i kasy) oraz wzięcie pożyczki (tylko kasy). Czas obsługi jest różny w zależności od operacji.

Klasy:

1. Account - reprezentuje klienta banku. Zawiera ID klienta, stan konta, typ(indywidualny lub biznesowy) oraz stan(wolny, zajęty, czekający na rozpatrzenie prośby o pożyczkę).
2. BankElement (abstrakcyjna) – interfejs dla elementów obsługi klienta. Zawiera ID elementu oraz kolejkę (std::queue) klientów.
3. ATM – rozszerza BankElement, reprezentuje maszynę do obsługi klienta.
4. InputTM – rozszerza ATM, reprezentuje wpłatomat.
5. OutputTM – rozszerza ATM, reprezentuje bankomat.
6. Teller – rozszerza BankElement, reprezentuje kasę.
7. BankBranch – reprezentuje oddział banku, zawiera wektor klientów, wektor kas, instancję wpłatomatu i instancję bankomatu. Zawiera informacje o ilości pieniędzy, jaką dysponuje oddział.
8. Logger<> - klasa pomocnicza do zapisu przebiegu informacji do strumieni. Zrealizowana jako szablon, parametrem którego jest typ strumienia.

Przebieg symulacji:

Symulacja uruchamiania jest za pomocą funkcji symulate() klasy BankBranch. Przed rozpoczęciem symulacji konstruktor klasy inicjalizuje wektory klientów i kas oraz instancje wpłatomatu i bankomatu. Oddział banku ma na początku $10.000.000. Podczas inicjalizacji klientów losowany jest ich typ (10% szansy na zostanie biznesowym) oraz stan konta (między $0 a $10.000, x1000 jeżeli biznesowy). Suma stanów konta klientów jest dołączana do pieniędzy banku. Podczas symulacji wykonywana jest pętla, która trwa 1 sekundę i reprezentuje 1 minutę działania banku. W każdym przebiegu pętli losowany jest numer klienta, który przychodzi do oddziału, przy czym jeżeli wylosowany zostanie klient, który już znajduje się w banku, uznawane jest, że nikt nowy nie przychodzi. Następnie losowana jest operacja (20% szansy na każdą). Klient wybiera między bankomatem, wpłatomatem a kasami porównując długości kolejek do każdego miejsca, w którym może wykonać wybraną operację. Idzie zawsze do kolejki najkrótszej wg ilości osób już w niej stojących, przy czym w pierwszej kolejności wybierany jest wpłatomat lub bankomat.

Elementy STL:

1. W klasach dziedziczących po BankElement znajduje się kolejka reprezentująca czekających do elementu klientów. Elementem kolejki jest std::tuple, które łączy klienta z długością operacji, którą wykonuje lub chce wykonać.
2. Klasa BankBranch używa std::vector do przechowywania informacji o klientach oraz kasach.

Sytuacje wyjątkowe:

1. Jeżeli dojdzie do próby wypłacenie pieniędzy z wpłatomatu lub wpłacenia do bankomatu rzucany jest wyjątek std::logic_error. Symulacja jest kontynuowana z pominięciem klienta, który próbował tego dokonać.
2. Jeżeli zapis do pliku log.txt nie powiedzie się rzucany jest wyjątek std::runtime_error. Użytkownik jest informowany o błędzie i program się zamyka.
3. Jeżeli stan konta oddziału spadnie poniżej $0 rzucany jest wyjątek std::runtime_error. Dochodzi wtedy do udzielenia pomocy przez rząd, tak aby bank miał do dyspozycji znów $10.000.000. Symulacja jest wtedy kontynuowana ze zresetowanym licznikiem czasu.