**Projekt: Symulator Automatu Do Napojów**

**1. Założenia:**Program ma pozwalać na korzystanie z wirtualnego automatu do napojów, symulować różne stany – m.in. awarii, braku produktu czy pustej kasetki z pieniędzmi.  
Automat można serwisować, uzupełniać, opróżniać czy psuć. Jest również tryb serwisanta, w którym użytkownik po podaniu odpowiedniego PIN-u ma bezpośredni dostęp do wielu czynności serwisowych, aczkolwiek jego rola ograniczona jest do prostego utrzymania maszyny. W celu uzyskania dostępu do trybu serwisowego należy znać PIN, który w tym przypadku wynosi „5425”.  
Dane o automacie są wczytywane z pliku stan.txt. Jeśli takowy nie istnieje bądź wystąpił problem z jego odczytem, ładowane są dane domyślne. Stan automatu jest zapisywany zawsze przed zamknięciem programu w pliku stan.txt, którego ręczna modyfikacja umożliwia na bardziej zaawansowaną konfigurację dostępnych produktów.   
Można kupić dwa rodzaje produktów: zwykły napój (zimny) i napój ciepły. Domyślnie w napoju ciepłym znajduje się skonfigurowana wcześniej ilość cukru. Użytkownik przy zakupie ma dodatkowo możliwość wybrania ilości cukru – bez, ze standardową ilością, z dodatkowym cukrem. Dodatkowy cukier wiąże się z dodatkową opłatą 20 groszy.  
Użytkownik o wydarzeniach informowany jest głównie przez kolejne okna dialogowe, a sam automat wyposażony jest w prosty wyświetlacz informujący chociażby o zaistnieniu usterki czy zwracanej reszcie.

Symulowane są m.in.: ilości dostępnych napojów, kubków i cukru, maksymalne pojemności magazynków na napoje, kubków i cukru, ograniczone zasoby automatu do wydawania reszty (aczkolwiek przepełnienie kasetki nie występuje), losowo występujące usterki automatu (uniemożliwiające dalszą pracę aż do jej usunięcia), możliwość kopnięcia w automat ze skutkiem uniemożliwiającym dalszą pracę automatu (aż do usunięcia usterki).

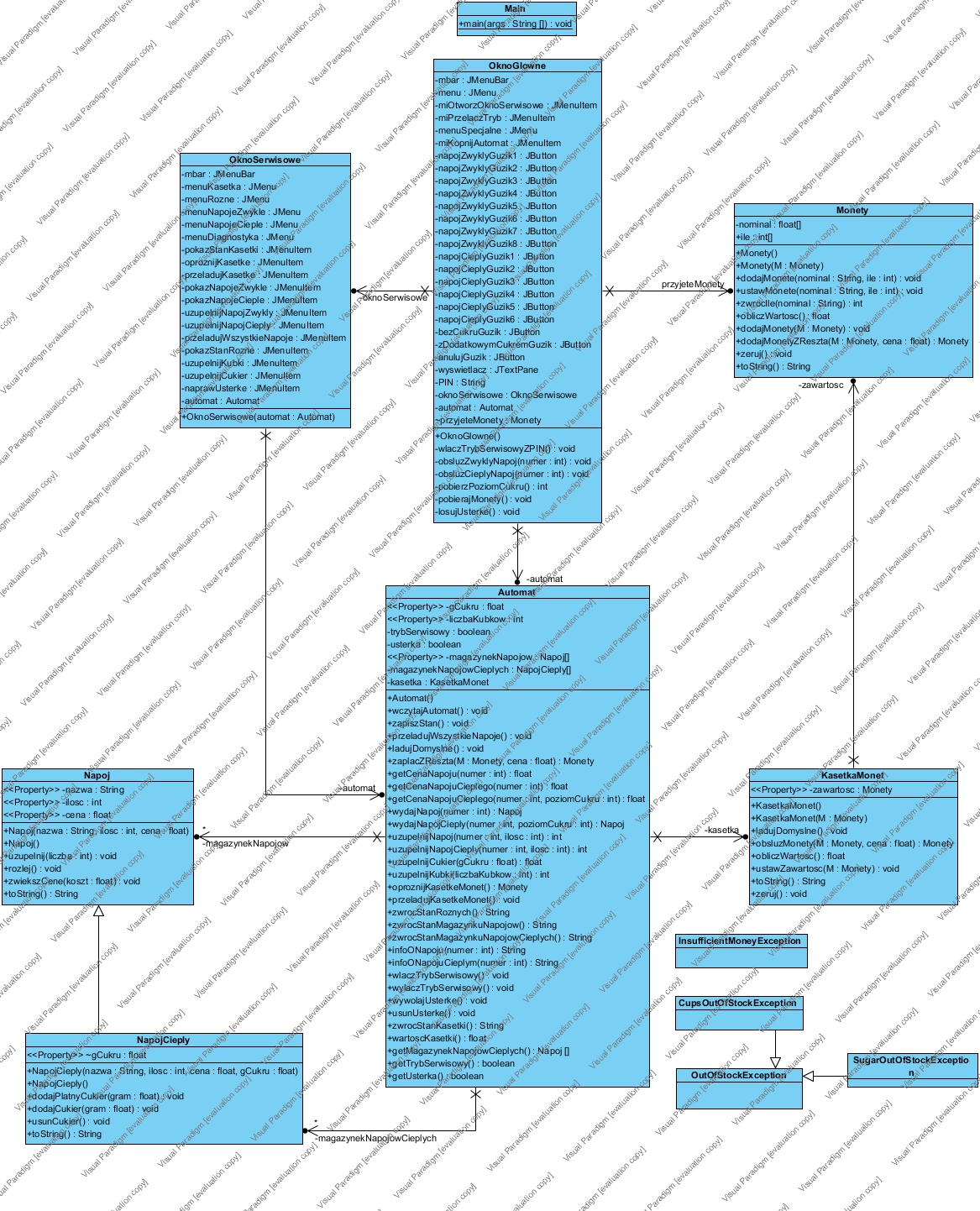
Automat jest ograniczony do posiadania w magazynku maksymalnie 8 napojów zwykłych, 6 napojów ciepłych, 5kg cukru, 150 kubełków

Wszystko oparte jest na prostym interfejsie graficznym z walidacją danych wejściowych.

**2. Wewnątrz programu:***Program został napisany w języku Java z JDK 1.8 i skompilowany w systemie Windows 7 z użyciem IntelliJ IDEA 15.0.4.*

W projekcie jest łącznie zaimplementowanych 12 klas, włącznie z klasą testową i czterema własnymi wyjątkami (alfabetycznie):

* *Automat* – model automatu ze wszystkimi jego stanami – magazynki/zasobniki, tryb serwisanta, stan usterki; metody do zmiany/ładowania/zapisywania stanów.
* *CupsOutOfStockException* – dziedziczone po *OutOfStockException*
* *InsufficientMoneyException* – dziedziczone po IllegalArgumentException
* *KasetkaMonet* – zbiór monet dostępnych w automacie
* *Main –* zawiera pojedyncze wywołanie konstruktora *OknoGlowne*
* *Monety* – prywatna tablica float **nominal[9]** z nominałami i publiczna tablica int **ile[9]** z ilością monet dla poszczególnych nominałów
* *Napoj* – private: String **nazwa**, int **ilość**, float **cena**
* *NapojCieply* – dziedziczone po *Napoju* +private float**gCukru**
* OknoGlowne – główne okienko, wirtualny frontowy panel automatu
* *OknoSerwisowe* – okienko poboczne dla serwisanta; umożliwia serwisowanie automatu przez użytkownika
* *OutOfStockException* – dziedziczone po IllegalStateException
* *SugarOutOfStockException* – dziedziczone po *OutOfStockException*



W celu obliczenia reszty zastosowano prosty algorytm zachłanny, który po kolei od największego nominału sprawdza, ile maksymalnie razy można danym nominałem wypłacić, po czym odkłada wyliczone monety na zwrotną kupkę monet.  
Same podstawowe operacje na monetach są zaimplementowane w specjalnej klasie Monety, w której są metody odpowiedzialne za obliczanie reszty czy całkowitej wartości zgromadzonych monet.

Wydawanie napojów i reszty kontrolowane jest w Oknie Głównym. Tam są pobierane wszelkie dane od użytkownika i wyliczana ostateczna cena, natomiast sytuacje braku koniecznych produktów są załatwiane za pomocą obsługi wyjątków. Automatyczne zapisywanie stanu przy wyjściu jest zrealizowane za pomocą WindowListenera – na starcie zdefiniowane jest DO\_NOTHING\_ON\_CLOSE, ale nasłuchiwacz próbę zamknięcia przechwytuje, zapisuje stan i przywraca DefaultCloseOperation jako JFrame.EXIT\_ON\_CLOSE, zamykając ostatecznie aplikację.  
  
**Struktura stan.txt:**Program zapisuje i odczytuje do/z pliku stan.txt o ściśle określonej strukturze (przykład poniżej):

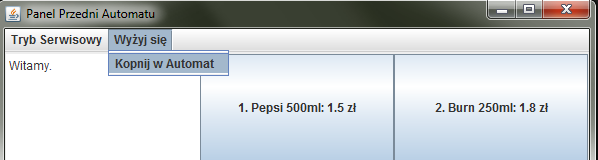
2425.0 110 true true  
50 30 27 18 11 14 14 11 5   
Pepsi 500ml  
10 1.5   
Burn 250ml  
10 1.8   
Coca-Cola 500ml  
10 1.65   
Fanta 500ml  
10 1.55   
Mirinda 500ml  
10 1.55   
7-Up 500ml  
10 1.55   
Tiger 250ml  
10 1.75   
Nestea 500ml  
10 1.5   
Gorąca czekolada  
20 1.5 15.0   
Cappuccino  
15 2.0 10.0   
Espresso  
15 1.9 10.0   
Kawa czarna  
20 1.75 8.0   
Kawa z mlekiem  
20 1.75 15.0   
Herbata cytrynowa  
20 1.2 15.0

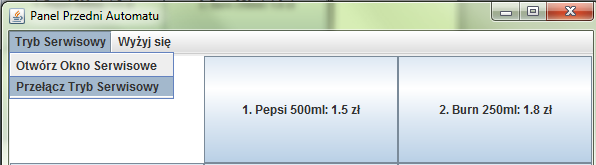
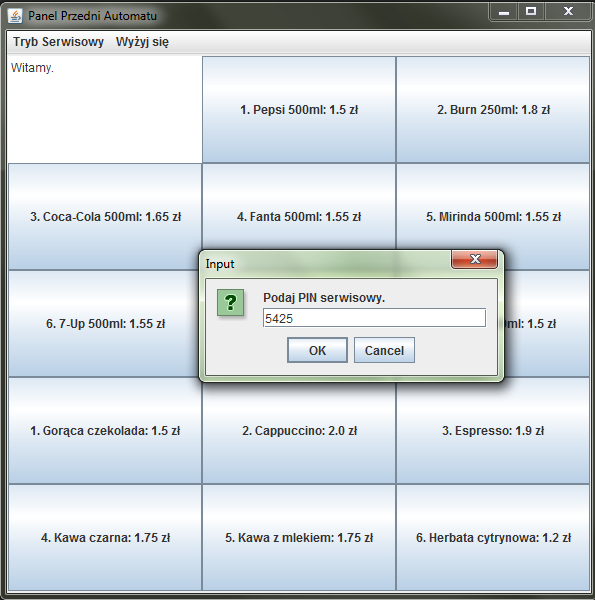
W pierwszej linijce: <float>**gCukru** <int>**liczbaKubkow** <true/false>**trybSerwisowy** <true/false>**usterka**  
W drugiej linijce są po kolei zapisane liczby monet w kasetce, licząc od najniższego nominału:  
1gr, 2gr, 5gr, 10gr, 20gr, 50gr, 1zł, 2zł, 5zł.  
W kolejnych linijkach znajdują się nazwy napojów i ich atrybuty (<int> **ilość** i <float> **cena** w przypadku napojów zwykłych oraz ilość, cena i wymagane do sporządzenia <float> **gCukru**  w przypadku napojów ciepłych). Po kolei wypisywane dane odnośnie ośmiu napojów zwykłych i sześciu napojów ciepłych.

**Swing:**Główne okno składa się prawie wyłącznie z przycisków rozmieszczonych za pomocą GridLayout – 3 kolumny, dowolne (0) wiersze. Jako wyświetlacz służy JTextPane z wyłączoną możliwością edycji. Akcje niezwiązane bezpośrednio z symulowanym przednim panelem automatu wyprowadzono na pasek narzędziowy/menu.  
Okno serwisowe w całości składa się z elementów menu. Jest automatycznie ukrywane poza trybem serwisowym.

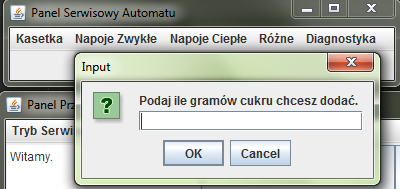
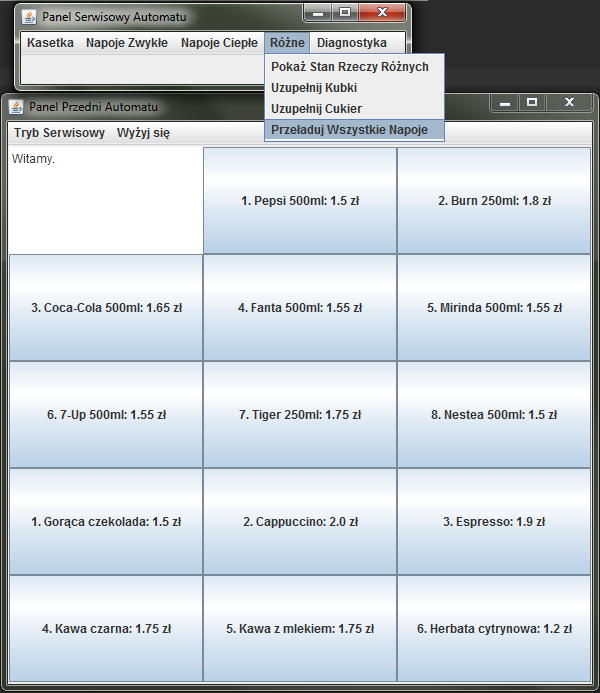
**3. Interfejs:**

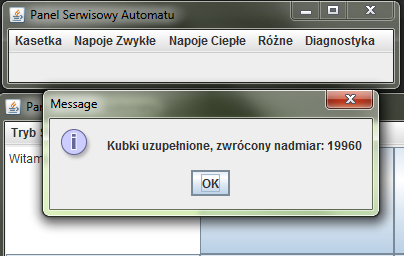
Panel frontowy

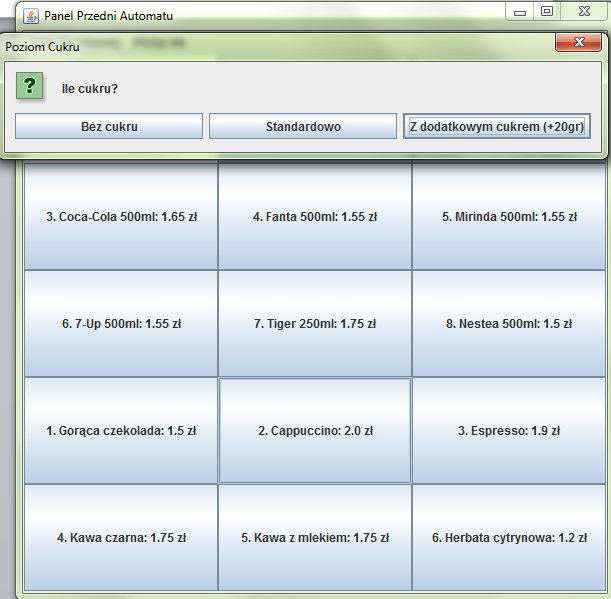
Poniżej uwidoczniona możliwość kopnięcia w automat w celu wywołania usterki:  


Tutaj dostęp do trybu serwisowego:  
  
Jeśli tryb serwisowy nie był włączony, użytkownik zostanie poproszony o PIN: 

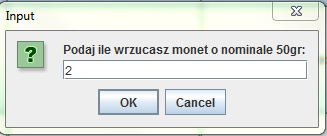
Tryb serwisowy pozwala na bezpośredni odczyt i edycję stanów automatu. Po kliknięciu na dowolną opcję, program będzie prowadzić użytkownika ‘za rączkę’:



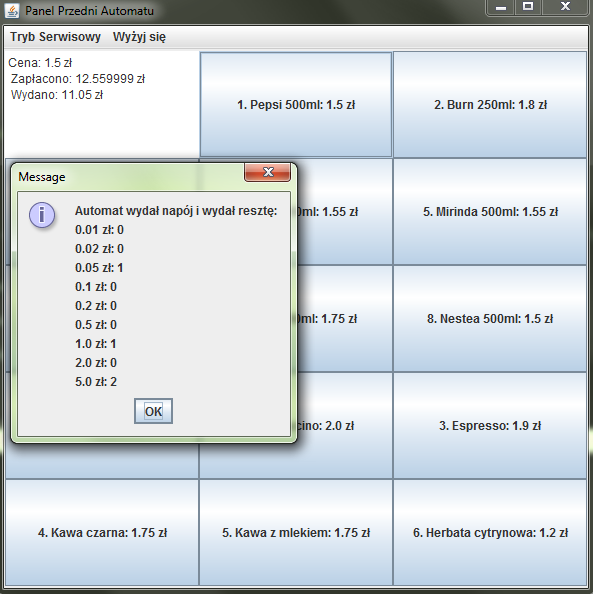
Interfejs graficzny zwraca również informację o nadmiarze uzupełnianego produktu.

Po kliknięciu na jeden z sześciu napojów ciepłych automat pyta się o poziom cukru:

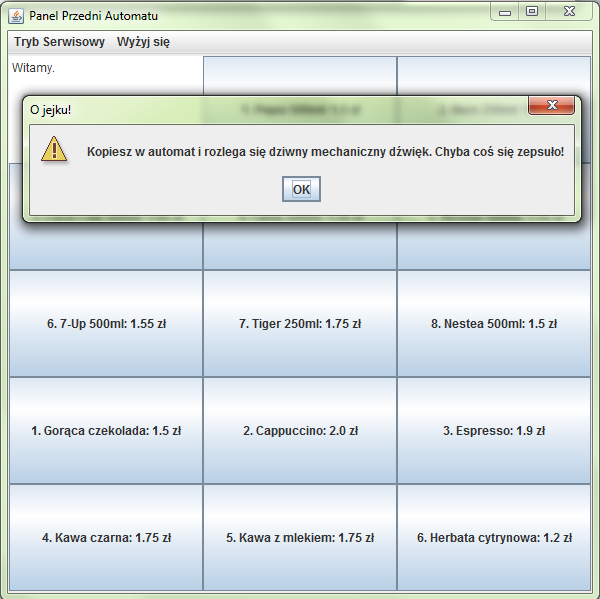
A następnie wymaga wrzucenia monet:

\

Po udanej transakcji urządzenie zwraca przydatne informacje na wyświetlaczu, a okno dialogowe opisuje świat zewnętrzny.

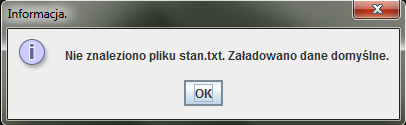


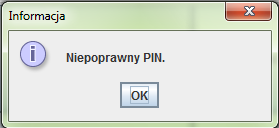
**4. Testy:**Część dokumentacji odnośnie interfejsu sama w sobie stanowiła pewnego rodzaju testy.  
Poniżej zawarte zostały pozostałe przykłady działania aplikacji.

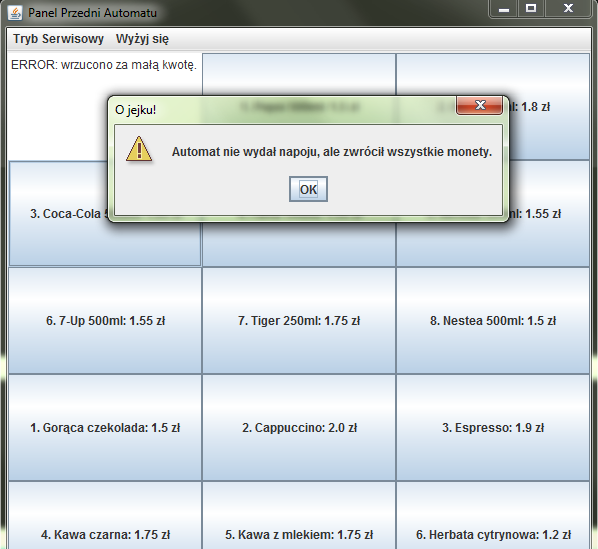
Po kilku kopnięciach doszło do awarii:

Po próbie zamówienia jakiegokolwiek napoju, automat odmawia posłuszeństwa:

Program ładuje dane domyślne, jeśli nie znajdzie pliku stan.txt:

  
Jeśli wpisze się niepoprawny PIN:



Jeśli wrzuci się zbyt małą kwotę:  
Nie da się również ‘uzupełnić’ automatu o negatywną liczbę towaru: