

Automat do napojów

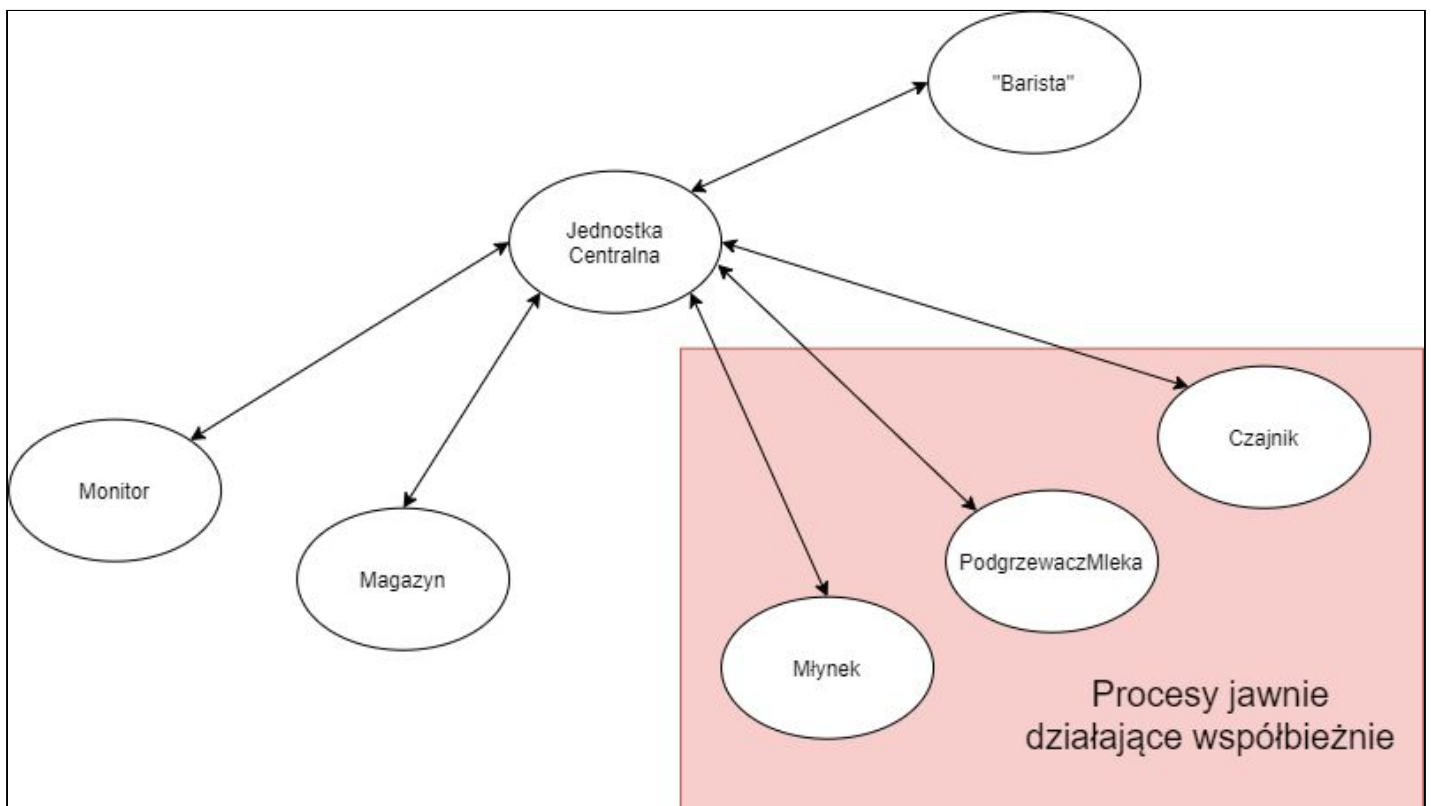
Szymon Borowy, Marcin Kozak
WEAiIB, Informatyka, rok 3. grupa 4a

Data oddania projektu: 16.01.2020 r

Cel programu, opis

Stworzony przez nas program ma w uproszczeniu symulować logikę automatu do napojów. Automat wyświetla użytkownikowi menu z listą możliwych produktów do wyboru i czeka na wybór użytkownika. Po wczytaniu wejścia program je przetwarza i albo zwraca błąd albo normalnie kontynuuje działanie. Przy normalnym funkcjonowaniu wyświetla użytkownikowi postęp poszczególnych etapów tworzenia produktu (np: mielenie kawy)

Opis i schemat struktury zadaniowej programu



Przykładowe działanie programu:

Zainicjowanie wszystkich procesów w funkcji start/0 i podanie wszystkich procesów do jednostki sterującej

Legenda:

JC → Jednostka Centralna

1. JC inicjalizuje Monitor
 - 1.1. Monitor czyści cały ekran i melduje JC wykonanie "zadania"
2. JC odbiera meldunek od Monitora i inicjalizuje Magazyn
 - 2.1. Inicjalizacja Magazynu i meldunek do JC
3. JC każde Monitorowi wyświetlić menu
 - 3.1. Monitor wyświetla menu i czeka na **INPUT** od użytkownika
 - 3.2. Po pewnym czasie użytkownik wprowadza Numer chcianego napoju.
 - 3.3. Monitor zwraca Numer do JC
 - 3.4. Monitor wywołuje "się rekurencyjnie, żeby być w stanie przyjąć kolejny komunikat"
4. JC każde Monitorowi wyświetlić w 3 różnych wierszach miejsce na paski postępu przygotowania poszczególnych części Napoju.
5. JC konwertuje INPUT do liczby naturalne
6. Jeżeli użytkownik podał niewłaściwy znak (np: literę "b") to JC wysła do Monitora rozkaz o wyświetlenie komunikatu o błędzie. Zakładamy że taka sytuacja w tym przypadku nie ma miejsca.
7. JC zapytuje Magazyn czy jest odpowiednia liczba dostępnych surowców
 - 7.1. Magazyn odejmuje od aktualnych zapasów ilości potrzebne do wyprodukowania Napoju i jeżeli któraś z wartości jest mniejsza od 0 to zwraca błąd o braku danego składnika
 - 7.2. Magazyn wysła do JC wiadomość o wyświetleniu aktualnego stanu Magazynu. Jest to funkcjonalność potrzebna jedynie w celach prezentacji projektu. W finalnej wersji nie ma potrzeby żeby użytkownik wiedział jaki jest stan zapasów w magazynie
 - 7.3. Magazyn wysła do JC wiadomość że jest w stanie obsłużyć dane zamówienie- mamy wystarczającą liczbę surowców.
8. JC wywołuje 3 współbieżne procesy- Czajnik, Młynek i Podgrzewacz Mleka i podaje im potrzebne ilości surowców.
 - 8.1. Każdy z tych 3 procesów wysła co jakiś czas (obliczany na podstawie potrzebnych przetwarzanych przez niego surowców) informacje o wykonaniu części zadania do JC, która przesyła ją do Monitora, który wyświetla paski postępu.
 - 8.2. UWAGA: Zaimplementowany sposób ma na celu tylko ogólne pokazanie jak taki proces mógłby przebiegać. W bardziej realnym rozwiązaniu każdy z tych 3 procesów prawdopodobnie nie raportowałby do JC o wykonaniu części zadania, aby w przypadku większej liczby procesów nie zablokować JC. Każdy z tych 3 procesów obsługiwałby elementy mechaniczne urządzenia.
9. JC do odebraniu jakiegoś sygnału o zakończeniu całości jednego z 3 procesorów przetwarzających surowce (np: grzanie wody) przekazuje ją do Baristy.
 - 9.1. Zostało to zaimplementowane w taki sposób żeby Barista był procesem który niejako zbiera informację o ukończeniu wszystkich 3 etapów i jak wszystkie 3 zostaną ukończone daje znać JC

- 9.2. Taka implementacja miała na celu uniknięcie blokowania JC co zdarzyłoby się gdyby miała ona czekać np na ukończenie 2. z 3 etapów
10. Kiedy wszystkie 3 procesy zakończyły swoje zadanie i Barista się o tym dowiedział, przekazuje JC informację że proces został zakończony.
11. JC po wyświetleniu komunikatu o końcu procesu powraca do stanu czekania na INPUT użytkownika

Instrukcja obsługi

Po kompilacji program uruchamiamy za pomocą polecenia `main:start()`. Następnie możemy wybierać Napój za pomocą klawiszy **1-9**. Jeżeli zakończyliśmy interakcję z programem należy zamiast liczby podać symbol **k**. Klawisz **s** służy do pokazania stanu magazynu, a klawisz **r** do resetu magazynu.

Możliwe rozszerzenia programu

1. Automat można przerobić z biurowego automatu do kawy na bardziej komercyjny-sprzedający produkty. Wystarczy dodać funkcjonalności odpowiedzialne za płatności kartą/gotówką i wydawanie reszty.
2. Dodanie bardziej szczegółowej informacji o błędach → Jeżeli nie ma zarówno wody jak i kawy to wyświetli się tylko błąd o braku wody (case za to odpowiedzialny jest sprawdzany wcześniej i wcześniej raportuje swój błąd)