

Programowanie Współbieżne i Rozproszone

Piotr Paruch, gr. 3a

Symulacja tablicy Kanban

Aplikacja składa się z kilku modułów:

- KanbanBoard – odpowiada za stworzenie tablicy i wywoływanie zadań wykonujących funkcje operujące elementami tablicy
- KanbanCard - moduł "karteczki", jej budowa i funkcje operujące na niej
- Worker - moduł pracownika, jego budowa i funkcje operujące pracownikami
- Supervisor – nadzorca uruchamiający procesy jako jego "dzieci"
- CLI - mały interfejs wiersza poleceń

Możliwe stany karteczek to: "Backlog", "To Do", "In Progress", "Testing" oraz "Done".

Tworzenie karteczki polega na:

- a) Losowym wybraniu stanu początkowego ("Backlog" lub "To Do")
- b) Wybraniu zadania z listy zadań
- c) Wylosowaniu stopnia trudności zadania (wartości koniecznej do osiągnięcia w celu przekazania zadania do kolejnego etapu)
- d) Dwa ostatnie elementy listy to stan początkowy zadania - 0 oraz wartość false - świadcząca o braku przyporządkowanego pracownika

Tworzenie pracownika polega na:

- a) Wygenerowaniu losowego imienia i nazwiska za pomocą modułu Faker.Person
- b) Wylosowaniu "umiejętności" pracownika, tj. przedziału, z którego później losujemy efektywność w danym kroku
- c) Dodania wartości *nil* – brak przypisanej karteczki

Po uruchomieniu aplikacji i wpisaniu komendy "start" pojawia się nam wygenerowana tablica (jej stan początkowy) i losowo zostają przypisane zadania do pracowników.

```
> +-----+-----+-----+-----+-----+
| Backlog | To Do | In Progress | Testing | Done |
+-----+-----+-----+-----+-----+
| Scale bananas | Enter cheat codes | | | |
| Free Hong Kong | Load simulation | | | |
| | Generate buttons procedurally | | | |
| Finish the sente | Decrypt Engrams | | | |
| Initiate launch sequence | Extract more minerals | | | |
| | Recombobulate discombobulators | | | |
+-----+-----+-----+-----+-----+

> Task "Load simulation" is assigned to Soledad Abbott
> Task "Recombobulate discombobulators" is assigned to Davon Brakus
> Task "Decrypt Engrams" is assigned to Mrs. Audra Stiedemann
> Task "Scale bananas" is assigned to Lizeth Runolfsdottir IV
> Task "Finish the sente" is assigned to Mckenzie Kreiger
```

W każdym kroku zostają przypisane zadania (jeśli są dostępni pracownicy), a następnie pracownicy wykonują akcję, której wynikiem może być np. częściowa realizacja danego zadania bądź ukończenie go, co skutkuje przeniesieniem karteczki do następnego etapu.

W etapach Backlog oraz Testing akcja polega na losowym "przepuszczeniu" zadania do następnego etapu z pewnym prawdopodobieństwem. W przypadku braku powodzenia akcji Testing zadanie wraca do etapu In Progress.

Używając komendy "next" przechodzimy do kolejnych kroków:

```
> Gwen McLaughlin Jr. googled the task "Generate buttons procedurally" and decided to pretend it doesn't exist.
> Ms. Rebeka Gutmann worked on task "Free Hong Kong" and progressed it 6 pts.
> Dr. Alexandria Casper worked on task "Finish the sente" and progressed it 3 pts.
> Mr. Berniece Mante PhD worked on task "Extract more minerals" and progressed it 3 pts.
> Mustafa Kris worked on task "Scale bananas" and progressed it 4 pts.
>
```

Backlog	To Do	In Progress	Testing	Done
Enter cheat codes	Scale bananas Free Hong Kong			
Load simulation Generate buttons procedurally	Finish the sente			
Initiate launch sequence Decrypt Engrams	Extract more minerals Recombobulate discombobulators			

W każdym kroku wypisane są wykonane czynności, tzn. przypisanie karteczek do pracownika, ukończone zadania czy postępy w ich wykonaniu/testach.

```
> Task "Extract more minerals" is assigned to Steve Morissette DVM
> Task "Free Hong Kong" is assigned to Calista Kirlin
> No available tasks - Worker stays idle
> No available tasks - Worker stays idle
> Adella Wisoky worked on task "Scale bananas" and progressed it 4 pts.
> Adella Wisoky finished working on task "Scale bananas"
> Task "Extract more minerals" didn't pass the tests and must go back to development phase again.
> Calista Kirlin worked on task "Free Hong Kong" and progressed it 6 pts.
> Calista Kirlin finished working on task "Free Hong Kong"
> Demetrius Maggio had no task assigned thus didn't work
> Stephon Davis had no task assigned thus didn't work
>
```

Backlog	To Do	In Progress	Testing	Done
			Scale bananas Free Hong Kong	Enter cheat codes
				Load simulation Generate buttons procedurally Finish the sente Initiate launch sequence Decrypt Engrams
		Extract more minerals		Recombobulate discombobulators

Po wykonaniu ostatniego kroku widzimy ostateczny stan tablicy:

```
> Task "Scale bananas" is assigned to Adella Wisoky
> No available tasks - Worker stays idle
> No available tasks - Worker stays idle
> No available tasks - Worker stays idle
> No available tasks - Worker stays idle
> Task "Scale bananas" passed tests successfully!
> Steve Morissette DVM had no task assigned thus didn't work
> Calista Kirlin had no task assigned thus didn't work
> Demetrius Maggio had no task assigned thus didn't work
> Stephon Davis had no task assigned thus didn't work
>

> +-----+-----+-----+-----+-----+
| Backlog | To Do | In Progress | Testing | Done |
+-----+-----+-----+-----+-----+
|         |       |             |         | Enter cheat codes |
|         |       |             |         | Scale bananas     |
|         |       |             |         | Free Hong Kong    |
|         |       |             |         | Load simulation   |
|         |       |             |         | Generate buttons procedurally |
|         |       |             |         | Finish the sente   |
|         |       |             |         | Initiate launch sequence |
|         |       |             |         | Decrypt Engrams    |
|         |       |             |         | Extract more minerals |
|         |       |             |         | Recombobulate discombobulators |
+-----+-----+-----+-----+-----+

> All tasks are done!
```

Tworzenie tablicy i wykonywanie operacji tablicowych

Tablica tworzona jest przez nadzorcę jako proces potomny:

```
@spec start_link([{:name, atom} | {:global, any} | {:via, atom, any}]) :: :ignore | {:error, any}
def start_link(opts) do
  Supervisor.start_link(__MODULE__, :ok, opts)
end

@impl true
@spec init(:ok) :: {:ok, %{intensity: any, period: any, strategy: any}, [any]}
def init(:ok) do
  children = [
    KanbanSim.KanbanBoard
  ]
  Supervisor.init(children, strategy: :one_for_one)
end
```

W wyniku tej operacji wszystkie kolejne operacje wywołujące nowe procesy "podpinają" je pod drzewo nadzorcze.

Większość operacji tablicowych GenServer (tablica) oddelegowuje nowo stworzonym procesom jako zadanie(Task) wywołujące funkcję z modułów Worker czy KanbanCard. Taski tworzone są tylko i wyłącznie w celu wykonania danej funkcji. W przypadku wywołania funkcji bezpośrednio w GenServerze (tablicy) aplikacja działałaby synchronicznie.

Przykład wywołania zadania:

```
def handle_cast(:set_rows, kboard) do
  {cards, workers, _rows} = kboard

  rows =
    cards
    |> Task.async_stream(&KanbanSim.KanbanCard.map_row/1)
    |> Enum.map(fn {:ok, result} -> result end)

  kboard = {cards, workers, rows}
  {:noreply, kboard}
end
```