

Marcin Jarzębski

Opis rozwiązania zadania rekrutacyjnego dla Junior Scala Developera

Realizacja funkcjonalności zawarta jest w module obsługującym zadania (MenedzerZadan), oraz klasach reprezentujące zadania.

Moduł „DostepDoBazy” pełni funkcję modułu który pozwala głównemu programowi na wczytywanie oraz zapisywanie danych do ewentualnej bazy danych. W rozwiązaniu rolę bazy danych pełni plik tekstowy, z przykładowymi danymi, aby łatwiej było testować aplikację.

Program zawarty w main.scala umożliwia prostą komunikację z aplikacją tak, aby można było przetestować działanie kodu.

Na uwagę zasługuje metoda „zwrocListeZadan” obiektu MenedzerZadan. Realizuje ona większość wymagań co do odfiltrowania zadań z systemu. Pobiera ona w parametrze funkcję, która ma za zadanie ocenić czy dane zadanie powinno znaleźć się w aktualnie generowanej liście wynikowej.

Wymagania co do możliwych operacji na zadaniach są zrealizowane w formie odpowiednio nazwanych metod.

ID nie jest na stałe przypisane do zadania. Jest ono unikalne w obrębie jednej sesji programu, i służy tylko łatwemu wskazywaniu zadań do modyfikacji.

W rozwiązaniu jest wiele niedociągnięć, których nie byłem w stanie poprawić ze względu na brak czasu, lub nieznajomość określonych mechanizmów języka Scala. Niedociągnięcia z których zdaję sobie sprawę:

- Pola zadań powinny być prywatne, z odpowiednio utworzonymi metodami Get i Set.
- Zadanie cykliczne nie ma reprezentacji cykliczności w logice programu (jest ono reprezentowane poprzez oddzielny typ zadania, i opisową definicję cykliczności).
- Status zadania powinien być typu wyliczeniowego, teraz reprezentowany jest on napisem.
- Dane wczytywane z bazy nie są sprawdzane (założyłem, że moduł zawsze dostarczy poprawną listę zadań).
- Listy tworzone są najprawdopodobniej w niezbyt optymalny sposób. Dałoby się to poprawić używając list które mogą zmieniać swój stan, jednak nie zdążyłem na tyle je opanować by móc ich użyć w programie.