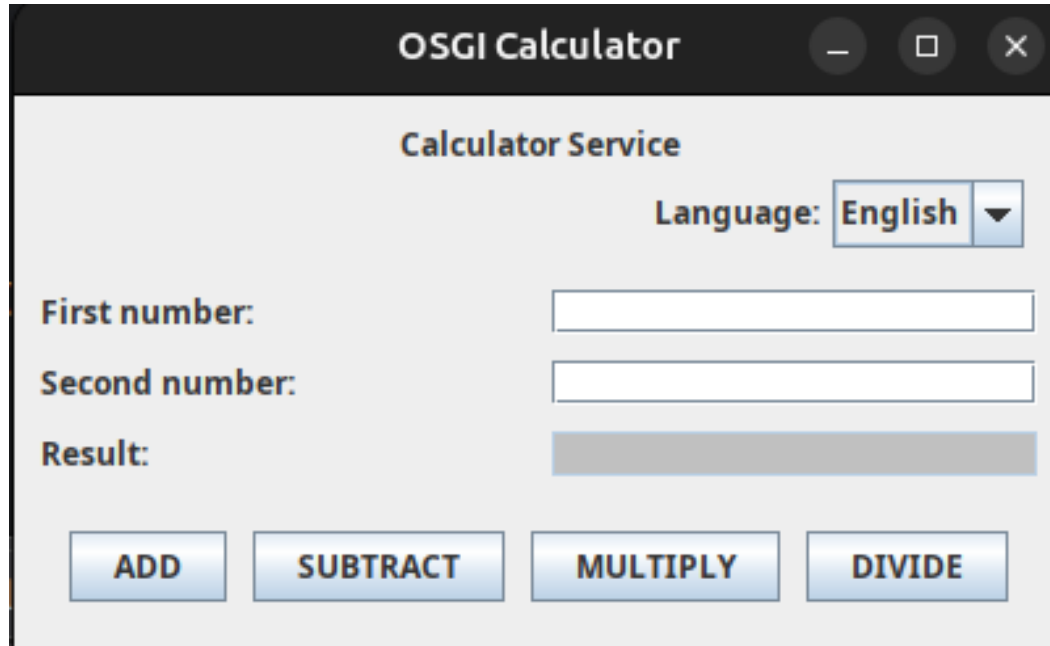


OSGI Calculator



The image shows a screenshot of a Java Swing window titled "OSGI Calculator". The window has a standard macOS-style title bar with minimize, maximize, and close buttons. Inside the window, the text "Calculator Service" is centered at the top. Below this, there is a "Language:" label followed by a dropdown menu currently set to "English". Further down, there are three labels: "First number:", "Second number:", and "Result:". Each label is followed by a text input field. The "Result:" field is currently empty and has a light gray background. At the bottom of the window, there are four buttons labeled "ADD", "SUBTRACT", "MULTIPLY", and "DIVIDE" arranged horizontally.

OSGI Calculator

Calculator Service

Language: English ▼

First number:

Second number:

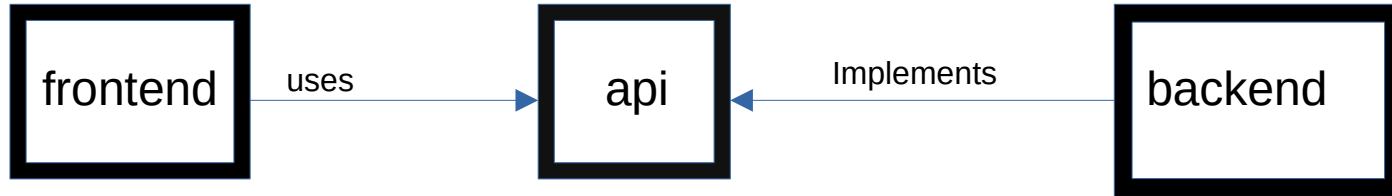
Result:

ADD SUBTRACT MULTIPLY DIVIDE

Can Beydoğan

High-Level Architecture

- Modules:
 - **api** → interfaces
 - **backend** → language logic & services
 - **frontend** → UI & user interaction



API Module (api)

- **No implementations**
- Only interfaces
 - DataService
 - NumberLanguage
- Shared **dependency** for both backend and frontend

Backend Module (backend)

- An OSGi bundle that **implements** the interfaces from api
 - DataServiceImpl
- **Language-specific** implementations
 - TurkishNumbers
 - EnglishNumbers
- **Map-driven** number logic
- Language rules **isolated** per class
 - **Easy to add new** languages

Multilingual Number Conversion

- Converting numbers between:
 - Text \rightarrow Integer
 - Integer \rightarrow Text

Text → Integer

- **Goal** : Convert written numbers into integers
 - Normalize input (lowercase, trim)
 - Split sentence into words
 - Map each word to numeric value
 - Build number using scale logic (hundred, thousand, million)
 - Uses two accumulators: **current**, **result**
- **Example:**

"two thousand five hundred six" → 2506

Integer → Text

- **Goal:** Convert integer into words
 - Handle special cases (0, negative)
 - Process large scales first (million → thousand → hundred)
 - Language-specific rules (Turkish):
 - No “bir yüz”, no “bir bin”
 - But “bir milyon” is allowed
 - Use recursion for each part
 - Lookup words for small numbers
- **Example:**

2345 → two thousand three hundred forty five

Frontend Module (frontend)

- **Purpose** : User interface & localization
- **Contents:**
 - MainWindow.java
 - UI.java
 - properties
- **Responsibilities:**
 - UI rendering
 - Locale detection
- **Localization:**
 - messages_en.properties
 - messages_tr.properties
 - UI text separated from logic

Examples

OSGI Hesap Makinesi

Calculator Service

Dil: Türkçe ▼

Birinci sayı: iki yüz bir

İkinci sayı: on bir

Sonuç: iki bin iki yüz on bir

TOPLA ÇIKAR ÇARP BÖL

OSGI Calculator

Calculator Service

Language: English ▼

First number: two hundred one

Second number: eleven

Result: two thousand two hundred eleven

ADD SUBTRACT MULTIPLY DIVIDE

Default Language: "tr"
Locale: "en"

Any Question

- GitHub repo:

<https://github.com/Cann23/OSGI-Calculator>