# KodeMorph: Automatická refaktorizace kódu na základě datových transformací

GitHub repozitář

Autor: **Le Duc Hung** | Rok obhajoby: 2026

## PROBLÉM A CÍL

Změna datového schématu (přejmenování či přesun polí) nevyžaduje jen migraci dat, ale i úpravu veškerého kódu, který s daty pracuje. Ruční oprava všech referencí je zdlouhavá a náchylná k chybám.

**Cíl:** z jediné definice transformace automaticky a konzistentně upravit data *i* kód — tzv. *sdružená evoluce* — a tím eliminovat lidskou chybovost a zrychlit celý proces.

## OČEKÁVANÁ APLIKACE

Nástroj cílí na jednoduché *headless* aplikace:

- **data:** MongoDB kolekce nebo lokální JSON
- **frontend:** Svelte (zobrazovací vrstva)
- **backend** předává *raw* data beze změny

Žádné ORM mapování ani mezivrstva transformující data — proto stačí upravit pouze databázi a Svelte šablony.

## PODPOROVANÉ TRANSFORMACE

Operation-based sada s jasným sémantickým záměrem — aplikuje se na data *i* kód.

| Transformace | Příkaz | Před → Po |
|---|---|---|
| **wrap field** | `wrap /a as b` | `{"a":10}` → `{"b":{"a":10}}` |
| **wrap as list** | `wrap /a as list` | `{"a":10}` → `{"a":[10]}` |
| **rename** | `rename /a to b` | `{"a":10}` → `{"b":10}` |
| **move** | `move /a to /x/y` | `{"a":1}` → `{"x":{"y":{"a":1}}}` |
| **map list values** | `map /tags[] to { "label": value }` | `["foo"]` → `[{"label":"foo"}]` |

### Wildcard operace: `*` a `[]`

`*` zastupuje klíče v objektu, `[]` prvky v seznamu; výsledek navíc závisí na pozici znaku v cestě. Podporují je `wrap` a `move`:

| | |
|---|---|
| `wrap /*/id as identifier` | klíč `id` kdekoliv pod kořenem |
| `wrap /users[]/id as identifier` | `id` uvnitř každého prvku seznamu |
| `move /users/* to /archive` | hromadný přesun všech klíčů kontejneru |

`rename` a `map` je **záměrně nepodporují** — vznikla by nejednoznačnost (více klíčů přejmenovaných na jeden cíl, skládání šablon přes neznámý počet prvků).

## PŘÍKLAD: REFAKTORIZACE STRUKTURY

Plochá struktura uživatele se přeskládá do vnořené — KodeMorph na základě níže uvedené posloupnosti příkazů transformuje data *a* zároveň aktualizuje všechny odpovídající reference v Svelte šabloně.

### Posloupnost příkazů

```
1. wrap / as user
2. rename /user/user_id to id
3. rename /user/username to name
4. wrap /user/email as contact
5. wrap /user/role as access
6. wrap /user/tags as access
7. map /user/access/tags[] to { "label": value }
```

### JSON data

*Před*

```
{
  "user_id": 123,
  "username": "john",
  "email": "john@example.com",
  "role": "admin",
  "tags": ["active", "premium"]
}
```

*Po*

```
{
  "user": {
    "id": 123,
    "name": "john",
    "contact": { "email": "john@example.com" },
    "access": {
      "role": "admin",
      "tags": [{ "label": "active" },
               { "label": "premium" }]
    }
  }
}
```

### Svelte šablona

*Před*

```
<p>{userData.user_id}</p>
<p>{userData.username}</p>
<p>{userData.email}</p>
<p>{userData.role}</p>
{#each userData.tags as tag}
  <li>{tag}</li>
{/each}
```

*Po*

```
<p>{userData.user.id}</p>
<p>{userData.user.name}</p>
<p>{userData.user.contact.email}</p>
<p>{userData.user.access.role}</p>
{#each userData.user.access.tags as tag}
  <li>{tag.label}</li>
{/each}
```

## SROVNÁNÍ PŘÍSTUPŮ

KodeMorph = kompromis mezi kontrolou manuálního přístupu a rychlostí AI: rychlý, deterministický a bezpečný.

| Kritérium | Manuální | KodeMorph | AI asistent |
|---|---|---|---|
| Rychlost | nízká | **vysoká** | vysoká |
| Riziko chyb | vysoké | **žádné** | střední |
| Migrační skript | ručně | **automaticky** | generuje AI |
| Konzistence kódu | dle vývojáře | **garantovaná** | dle promptu |
| Determinismus | – | **plný** | ne |
| Kontrola / bezpečnost | plná, pracná | **vysoká** | riziko delegace |

## Výsledek a závěr

Výsledkem práce je nástroj KodeMorph, který na základě podporované sady datových transformací automaticky refaktorizuje kód, aby odpovídal změněnému datovému schématu. KodeMorph nabízí jednoduché CLI rozhraní, které umožňuje vývojářům snadno integrovat tento nástroj do svého vývojového workflow, a díky modulární architektuře je otevřen pro budoucí rozšíření o další transformace.

Vedoucí práce:
**doc. Mgr. Tomáš Petříček, Ph.D.**