Даалгавар 1: Type system

Энэ даалгавар нь Python дээр **төрлийн тэмдэглэгээ бичиж (writing type annotations)** сурах зорилготой. Python-ийн сүүлийн үеийн хувилбарууд нь функцэд (Python 3.5-аас хойш) болон локал хувьсагчид (Python 3.6-аас хойш) төрлийн тэмдэглэгээ бичихийг дэмждэг. Тухайн программыг ажиллуулахын өмнө third-party программаар (жишээ нь, mypy) шалгасан байна. Та тэмдэглэгээний формат болон боломжит төрлүүдийн талаар товч мэдээллийг эндээс авах боломжтой.

Таны даалгавар бол Python 3 программыг төрлийн тэмдэглэгээ нэмэн өргөтгөнө. Программыг well-formed байлгахын тулд хамгийн тодорхой төрлүүдийг нэмэх хэрэгтэй. Жишээ нь, some_variable = 3 утга олголтыг Any-ээр биш харин int төрлөөр тэмдэглэсэн байх ёстой (Any нь ерөнхийдөө зөв боловч тийм ч ашигтай биш). Өөрөөр хэлбэл, some_variable-ын зөв шийдэл: int = 3. task1.py файлын методууд (аргумент болон буцаах төрлүүд) дээр төрөл тэмдэглэнэ. Локал хувьсагчдад тайлбар хийх шаардлагагүй. Төрөл нь нийтлэг байж болно, жишээ нь, List[str] нь хүчинтэй гэж үзнэ. Ийм нийтлэг төрлүүдийн хувьд зөвхөн Dict биш Dict[int, int] гэх мэт төрлийн бүх аргументуудыг зааж өгнө.

Санамж: Та кодоо default тохиргоотой mypy 1.3.0 -оор зөв бичигдсэн эсэхийг нь шалгасан байна. Нэмж хэлэхэд, өгөгдсөн төрлийн тэмдэглэгээ бүрийг тухайн аргумент/буцах утгын зөв, хамгийн тохирохтой нь харьцуулсан байна. (Хэрэв олон төрөл эквивалент байвал тэдгээрийг бас зөвшөөрнө.) Үүсгэсэн программ тань Python 3.9-ын программ байх ёстой бөгөөд ямар ч зарлалтыг устгах эсвэл ямар нэгэн үйлдлийг өөрчлөх ёсгүй.

Даалгавар 2: Functional programming

Хөгжүүлэлтийн орчныг бэлтгэх

Өөрийн компьютер дээрээ Haskell суулгасан эсэхийг шалгаарай. Visual Studio Code дээр Haskell программ хөгжүүлэлтийн орчныг бэлтгэх хэрэгтэй. Жишээ нь: vscode in win, vscode on mac

Факториал тодорхойлолт

Факториал нь сөрөг бус бүхэл n-ийн n-ээс бага буюу тэнцүү байх бүх эерэг тооны үржвэр бөгөөд $n!$ гэж тэмдэглэдэг. Жишээлбэл, $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$.

Haskell функцийн кодчилол

Өгөгдсөн сөрөг бус бүхэл тооны факториалыг тооцоолох Haskell функц бичнэ.

1. Факториал функцийг тодорхойлохдоо рекурс ашиглана.
2. Паттерн тааруулах (Pattern matching) ашиглан факториал функцийг хэрэгжилтийг сайжруулна.

Жишээ код:

```haskell
    -- Define factorial function using recursion
    factorial :: Integer -> Integer
    factorial 0 = 1
    factorial n = n * factorial (n - 1)
```

Санамж: Факториал функцийнхээ үр ашгийг дээшлүүлэх аргууд судалж үзээрэй (боломжтой бол).

Тестчилэл

Факториал функцийн зөв эсэхийг шалгахын тулд тестийн тохиолдлыг бич. $0, 1, 5, 10$ гэх мэт янз бүрийн оролтыг турших. Жишээ тохиолдлууд:

```haskell
-- Test cases
main :: IO ()
main = do
    putStrLn "Factorial of 0 is: " ++ show (factorial 0)
    putStrLn "Factorial of 5 is: " ++ show (factorial 5)
    putStrLn "Factorial of 10 is: " ++ show (factorial 10)
```

ТАНД АМЖИЛТ ХҮСЬЕ!