

# Hore, dole...

24. 5. 2024

Pri návrhu business centier je dôležitá otázka počet a kapacita výťahov. Vašou dnešnou úlohou je implementovať simuláciu, ktorá čiastočne odpovie na túto otázku; nasimulujete pohyb niekoľkých výťahov a vykreslite, ako sa mení počet ľudí čakajúcich na výťah pri príchode do práce. V skripte pre implementáciu simulácie (0524-1.py) nájdete tri triedy:

## Worker

Definuje pracovníka v business centre, pričom pre každého pracovníka sú určené dve hodnoty:

- `start_floor` (int) – na ktorom podlaží sa pracovník nachádza (v simulácii vždy 0)
- `goal_floor` (int) – na ktoré podlažie sa chce pracovník dostať

Do triedy nemusíte doplniť žiadne metódy (ak to vy nepotrebuje).

## Elevator

Definuje výťah, pre ktorý sú určené dva parametre a dve ďalšie členské premenné:

- `capacity` (int) – kapacita výťahu, t. j. koľko ľudí sa do výťahu zmestí
- `current_floor` (int) – na ktorom podlaží sa výťah práve nachádza; defaultne je 0
- `moving_up` (bool) – vyjadruje, či sa výťah pohybuje hore (True) alebo dole (False)
- `passengers` (list) – zoznam objektov typu `Worker`, ktorí sú vo výťahu

Doplňte metódy do triedy podľa nasledovných špecifikácií:

- **`move()`** – metóda reprezentuje posunutie výťahu o jedno podlažie; inkrementujte, resp. dekrementujte hodnotu aktuálneho podlažia (`current_floor`) o 1 v závislosti od smeru výťahu (`moving_up`)
- **`get_passenger_count()`** – metóda vráti počet cestujúcich vo výťahu
- **`change_direction()`** – metóda zmení smer výťahu (zneguje hodnotu `moving_up`)
- **`add_passenger(passenger)`** – metóda pridá do výťahu pracovníka, ak sa tam zmestí a vráti hodnotu True; ak sa pracovník do výťahu nezmestí, metóda vráti False; metóda má jeden parameter: objekt typu `Worker`, ktorý chce vstúpiť do výťahu
- **`remove_passenger(passenger)`** – metóda odstráni z výťahu pracovníka; následne zmeníte smer výťahu, ak výťah sa pohybuje smerom hore a už sa v ňom nenachádzajú žiadni pracovníci (výťah sa vráti na prízemie); metóda má jeden parameter: objekt typu `Worker`, ktorý chceme odstrániť z výťahu. V metóde ošetríte prípad, keď dostaneme ako parameter objekt `Worker`, ktorý sa v zozname nenachádza. V tomto prípade metóda nič neurobí.

## Building

Trieda reprezentuje budovu, ktorá je definovaná nasledujúcimi premennými:

- `floor_count` (int) – počet podlaží v budove
- `elevators` (list) – zoznam objektov typu `Elevator`, ktorý sa vygeneruje na základe parametrov `elevator_count` (počet výťahov) a `elevator_capacity` (kapacita výťahov)
- `floors` (list) – zoznam zoznamov, ktorý reprezentuje zoznam čakajúcich pracovníkov na jednotlivých podlažiach; pri vytvorení objektu je to zoznam `floor_count` prázdnych zoznamov

Do triedy doplňte nasledujúce metódy:

- **add\_worker\_to\_floor(worker, floor)** – metóda pridá objekt typu `Worker` (daný parametrom `worker`) do zoznamu čakajúcich na danom podlaží (parameter `floor`)
- **is\_waiting()** – metóda slúži ako ukončovacia podmienka simulácie; vráti počet ľudí, ktorí čakajú na výťah na prízemí (nulté podlažie)
- **time\_step()** – metóda reprezentuje hlavnú metódu simulácie, urobí jeden časový krok, a posunie každý výťah nasledovne:
  1. získa číslo podlažia kde sa výťah práve nachádza
  2. pre každého pasažiera výťahu zistí, či aktuálne podlažie je jeho cieľové podlažie; ak áno, odstráni pracovníka z výťahu
  3. pre každého pracovníka, ktorý čaká na danom podlaží, ak výťah ide smerom hore, skúste pridať pracovníka do výťahu; ak sa to podarí, vymažte pracovníka zo zoznamu čakajúcich ľudí na podlaží
  4. posuňte výťah
  5. ak po posune je výťah na nultom alebo najvyššom podlaží, otočte ho

Skript obsahuje metódu `simulate_workday()`, ktorá nasimuluje príchod pracovníkov ráno do budovy. Metóda má nasledovné parametre:

- `number_of_workers (int)` – celkový počet pracovníkov v budove
- `floor_count (int)` – počet podlaží (prízemie + všetky poschodia)
- `number_of_elevators (int)` – počet výťahov v budove
- `elevator_capacity (int)` – kapacita výťahov, t. j. koľko ľudí sa zmestí do jedného výťahu

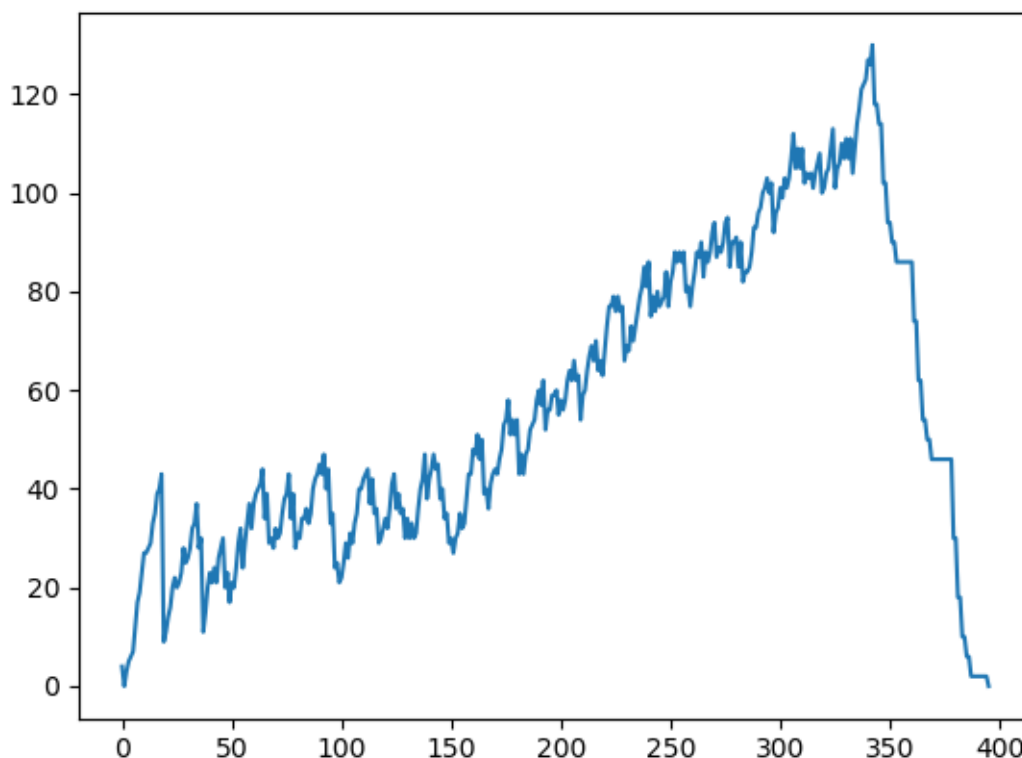
Implementujte metódu **`simulate_workday()`** podľa nasledovnej špecifikácie:

- inicializujte budovu na základe zadaných parametrov
- na začiatku vygenerujte 1 až 5 pracovníkov, pre ktorých náhodne vygenerujte cieľové podlažie (1 až výška budovy - 1); následne pridajte vygenerovaného pracovníka do zoznamu čakajúcich na nultom podlaží
- kým ste nevygenerovali `number_of_workers` pracovníkov, alebo stále niekto čaká na nultom podlaží, v cykle urobte nasledovné kroky:
  1. zavolajte metódu pre posun výťahov
  2. zaznamenajte počet čakajúcich ľudí po posune (do zoznamu `waiting_counts`)
  3. podobne ako na začiatku simulácie, ak ste doteraz nevygenerovali každého pracovníka, vygenerujte 1 až 5 pracovníkov, pre ktorých vygenerujte cieľové podlažie; budete potrebovať pomocnú premennú na zaznamenanie počtu pracovníkov, ktorých ste už vygenerovali
- po ukončení simulácie funkcia vráti počet čakajúcich ľudí po jednotlivých krokoch

**Vo funkcii main() odpovedzte na otázku:**

**Ako sa mení počet čakajúcich ľudí počas toho, ako ľudia prichádzajú do práce?**

Pre získanie odpovede vykonajte jednu simuláciu. Výsledky znázorníte pomocou grafu, ktorý musí mať názov, a pomenované osi. Samotný graf môže vyzeráť nasledovne:



**Pri riešení dodržujte nasledujúce zásady:**

- riešenie môžete rozšíriť o ľubovoľné metódy
- pri riešení nemusíte postupovať presne podľa návodu, mali by ste ale ponechať funkcionality
- predpripravený skript môžete ľubovoľne opravovať, nemali by ste ale meniť funkcionality
- nepristupujte priamo k členským premenným objektov a tried
- hlavná funkcia môže obsahovať iba volanie funkcie `main()`

*Dĺžka kódu (bez komentárov): cca. 120 riadkov.*