Nastupujte, prosím!

Každý, kto cestuje lietadlom zažije pocit pri odletovej bráne, že rad na lietadlo sa vôbec nepohne a určite musí existovať rýchlejší spôsob nastupovania cestujúcich. V tomto zadaní sa spustíte do hľadania takého spôsobu a porovnáte zaužívané spôsoby. Inšpiráciou pre zadanie je video „The Better Boarding Method Airlines Won't Use" od CGP Grey, ktoré je dostupné na: https://www.youtube.com/watch?v=oAHbLRjF0vo.

Vo vašom riešení vytvoríte simuláciu procesu nastupovania do lietadla. V každej simulácii vytvoríte model lietadla, ktorý postupne naplníte cestujúcimi. Rôzne spôsoby nastupovania budú určené poradím pridávania cestujúcich na základe ich miesta. Ako aj v skutočnosti, každé sedadlo bude reprezentované dvojicou hodnôt: rad (napr. 12) a miesto (napr. C). Pre jednoduchosť riešenia budeme predpokladať, že každé lietadlo má miesta od A po F (v každom rade sú umiestnené dve skupiny sedadiel po tri). Takisto predpokladáme, že všetky sedadlá majú byť obsadené cestujúcimi.

Simulácie a modely vytvoríte podľa paradigmy objektovo orientovaného programovania, pričom váš projekt musí obsahovať aspoň tri triedy:

- `Passenger` (v súbore `passenger.py`) – trieda reprezentuje jedného cestujúceho s pridelením miestom a batožinou.
- `Plane` (v súbore `plane.py`) – trieda reprezentuje lietadlo s určenou dĺžkou. Z implementačného pohľadu bude lietadlo reprezentované ako zoznam zoznamov, napr. lietadlo s 8 radmi bude reprezentované nasledovne

X	1F	2F	3F	4F	5F	6F	7F	8F	X
X	1E	2E	3E	4E	5E	6E	7E	8E	X
X	1D	2D	3D	4D	5D	6D	7D	8D	X
X	1C	2C	3C	4C	5C	6C	7C	8C	X
X	1B	2B	3B	4B	5B	6B	7B	8B	X
X	1A	2A	3A	4A	5A	6A	7A	8A	X

  kde stredný prázdny riadok je chodba a dva krajné stĺpce (označené X) sú iba pomocné stĺpce a zjednodušujú nám simuláciu. Na každej pozícii v tomto poli bude zoznam cestujúcich, ktorí sa nachádzajú na danej pozícii. **Pozor:** pri reprezentácii lietadla pomocou dvojrozmerného poľa sú v riadkoch sedadlá s rovnakým písmenom, rady sedadiel sú vo stĺpcoch!
- `Boarding` a podtriedy (všetky v súbore `boarding.py`) – trieda reprezentuje spôsob nastupovania do lietadla. Trieda `Boarding` definuje všeobecnú funkcionalitu, podtriedy špecifikujú už konkrétny spôsob nastupovania, resp. pridávania cestujúcich do lietadla.

V rámci zadania nasimulujete šesť spôsobov nastupovania, pričom pre každý spôsob zadefinujete novú podtriedu `Boarding`:

1. back-to-front – cestujúcich rozdelíte na štyri skupiny na základe toho, v ktorom rade sedia. Na lietadlo najprv spustíte cestujúcich z najzadnejšej skupiny (v náhodnom poradí), potom cestujúcich z druhej najzadnejšej skupiny, atď. Ako poslední nastupujú cestujúci, ktorí sedia vpredu. Môžete predpokladať, že dĺžka lietadla, t.j. počet radov, bude deliteľná 4.
2. front-to-back – cestujúcich rozdelíte na štyri skupiny na základe toho, v ktorom rade sedia. Na lietadlo najprv spustíte cestujúcich z najprednejšej skupiny (v náhodnom poradí), potom cestujúcich z druhej najprednejšej skupiny, atď. Ako poslední nastupujú cestujúci, ktorí sedia na konci lietadla. Môžete predpokladať, že dĺžka lietadla, t.j. počet radov, bude deliteľná 4.
3. window-to-aisle – cestujúcich rozdelíte na tri skupiny na základe toho, kde sedia v rámci troch sedadiel (pri okne, v strede, alebo pri chodbe). Na lietadlo najprv spustíte cestujúcich sediacich pri okne, teda sedadlá A a F (v náhodnom poradí), potom cestujúcich sediacich v strede, teda sedadlá B a E, a na záver cestujúcich sediacich pri chodbe, teda sedadlá C a D.
4. aisle-to-window – cestujúcich rozdelíte na tri skupiny na základe toho, kde sedia v rámci troch sedadiel (pri okne, v strede, alebo pri chodbe). Na lietadlo najprv spustíte cestujúcich sediacich pri chodbe, teda sedadlá C a D (v náhodnom poradí), potom cestujúcich sediacich v strede, teda sedadlá B a E, a na záver cestujúcich sediacich pri okne, teda sedadlá A a F.
5. random – cestujúcich nerozdelíte do žiadnych skupín, spustíte ich na lietadlo v náhodnom poradí.
6. Steffen's perfect – cestujúcich spustíte na lietadlo podľa postupu definovaného Jasonom Steffenom: od okna po chodbu, zozadu každý druhý rad, striedavé strany. To znamená, že na lietadlo najprv spustíte cestujúcich na sedadlách v párnom rade a na sedadle A. Potom prídu cestujúci v párnom rade na sedadle F. V ďalšom kroku nastupujú cestujúci v nepárnych radoch na sedadle A. Nastupovanie cestujúcich sediacich pri okne ukončíte skupinou v nepárnych radoch na sedadle F. Nastupovanie pokračuje podľa rovnakého pravidla pre sedadlá B a E, resp. C a D:

X	16	8	15	7	14	6	13	5	X
X	32	24	31	23	30	22	29	21	X
X	48	40	47	39	46	38	45	37	X
X	44	36	43	35	42	34	41	33	X
X	28	20	27	19	26	18	25	17	X
X	12	4	11	3	10	2	9	1	X

Trieda `Passenger` (3 body)

V súbore `passenger.py` nájdete kostru kódu triedy `Passenger`.
Konštruktor triedy má tri parametre: `row` (číslo radu), `seat` (písmeno sedadla cestujúceho) a `no_of_bags` (počet batožín). Trieda má nasledovné vnútorné premenné:

- `self.row` – číslo radu, kde cestujúci má miesto (parameter `row`)
- `self.seat` – písmeno sedadla (parameter `seat`)
- `self.bags` – vyjadruje počet krokov, ktorý cestujúci potrebuje na to, aby svoju batožinu dal do odkladacieho priestoru nad hlavou (parameter `no_of_bags * 4`)
- `self.plane` – smerník na lietadlo, v ktorom sa nachádza cestujúci; v konštruktore nastavený na `None`
- `self.current_position` – vyjadruje aktuálnu polohu cestujúceho v lietadle, v konštruktore nastavená na `[None, None]`

Vašou úlohou je doplniť funkcionalitu nasledovným spôsobom:

`get_position(self)` – metóda vracia pozíciu cestujúceho v poli (riadok a stĺpec v zozname zoznamov), ktoré reprezentuje lietadlo v ktorom sa cestujúci nachádza. Ak cestujúci ešte nebol pridaný do lietadla, metóda vracia `None`.
Pozor: číslo radu miesta cestujúceho nezodpovedá číslu radu v poli, ktoré reprezentuje lietadlo!

`get_seat(self)` – metóda vracia pozíciu sedadla cestujúceho v lietadle (riadok a stĺpec v zozname zoznamov).
Pozor: číslo radu miesta cestujúceho nezodpovedá číslu radu v poli, ktoré reprezentuje lietadlo!

`add_to_plane(self, plane)` – metóda pridá cestujúceho do lietadla. Metóda má jeden dodatočný parameter: smerník na lietadlo, do ktorého chceme pridať cestujúceho. V metóde aktualizujte hodnotu vnútornej premennej `plane` a dvojicu čísel `current_position` na `[0, 3]` (0 reprezentuje nultý rad sedadiel, 3 je chodba – index stĺpca a riadku v dvojrozmernom poli reprezentujúcom lietadlo).

`can_sit(self)` – metóda určí, či cestujúci má voľnú cestu ku svojmu miestu ak už stojí v danom rade. Ak cestujúci má miesto pri chodbe, metóda vracia vždy `True`, v opačnom prípade vráti `True` ak sedadlá medzi sedadlom cestujúceho a chodbou nie sú obsadené (napr.: ak cestujúci si chce sadnúť na 4E, ale niekto sedí na 4D, metóda vracia `False`). Funkcia vracia `False` ak cestujúci ešte nebol pridaný do lietadla, alebo nestojí na chodbe.

`forced_to_move(self, x, y)` – metóda reprezentuje vynútený pohyb cestujúceho (samotný cestujúci sa nechce pohnúť, ale je to nevyhnutné pretože prekáža niekomu inému). Metóda má dva dodatočné parametre – `x` a `y` –, ktoré reprezentujú novú polohu cestujúceho v lietadle. V metóde máte aktualizovať hodnotu dvojice `current_position`.
Dávajte si pozor na správnu reprezentáciu aktuálnej pozície!

`move(self)` – metóda reprezentuje pohyb cestujúceho v lietadle. Ak cestujúci ešte nebol pridaný do lietadla, funkcia vyhodí `TypeError`. Ak cestujúci sa nachádza v lietadle, metóda vráti vždy dve hodnoty – novú pozíciu cestujúceho (rad sedadiel a číslovanie sedadiel resp. chodby) Pohyb cestujúceho vieme popísať s niekoľkými pravidlami:

1. kým cestujúci nie je v rade svojho miesta, ostane na chodbe a vždy urobí jeden krok dopredu ak nasledujúca pozícia je voľná
2. ak cestujúci už stojí v rade svojho miesta, najprv uloží svoju batožinu (znížte hodnotu `self.bags` o 1, až kým nedosiahne 0, cestujúci ostane na svojej pozícii)
3. ak cestujúci už stojí v rade svojho miesta a nemá batožinu, pozrie sa, či má voľnú cestu k svojmu miestu. Ak áno, tak si sadne, ak nie, poprosí ďalších cestujúcich, aby mu uvoľnili cestu (viď `move_row` a `return_row` v triede `Plane`).

Trieda `Plane` (3 body)

V súbore `plane.py` nájdete kostru kódu triedy `Plane`.
Konštruktor triedy má jeden parameter: `length` (počet radov v lietadle). Trieda má nasledovné vnútorné premenné:

- `self.length` – počet radov v lietadle (parameter `length`)
- `self.seats` – dvojrozmerný zoznam zoznamov, kde každý riadok reprezentuje sedadlá s rovnakým písmenom a každý stĺpec reprezentuje jeden rad sedadiel. Na každej pozícii tohto poľa je zoznam cestujúcich, ktorí sa nachádzajú na danej pozícii.

Vašou úlohou je doplniť funkcionalitu nasledovným spôsobom:

`print_plane(self)` – metóda vykreslí na obrazovku jednoduchú reprezentáciu lietadla. (Metóda slúži ako pomôcka pre vás, nebude hodnotená).

`add_passengers(self, psg_list)` – metóda pridá cestujúcich zo zoznamu do lietadla. Má jeden parameter, `psg_list`, ktorý je zoznam s cestujúcimi. Pre každého cestujúceho zavolajte metódu z triedy `Passenger`, ktorá ho pridá do lietadla a následne pridajte cestujúceho aj do zoznamu na pozícii [3, 0] (opačné poradie ako reprezentácia pozície v triede `Passenger`).

`isEmpty(self, row, seat)` – metóda vracia `True` ak zoznam na pozícii `[seat, row]` je prázdna, `False` v opačnom prípade. Ak daná pozícia neexistuje, vráti `True`.

`move_row(self, row, seat_letter)` – metóda slúži na posunutie cestujúcich zo sedadiel s účelom uvoľnenia cesty pre ďalšieho cestujúceho k svojmu miestu. Metóda má dva parametre – rad a písmeno sedadla, do ktorého si chce cestujúci sadnúť. Metóda najprv zistí, či pozícia [seat, row + 1] je voľná, a ak áno, posunie každého sediaceho cestujúceho na túto pozíciu (pridá ich do príslušného zoznamu). Ak pozícia nie je voľná, metóda nič nerobí, a nový cestujúci musí ďalej čakať.

`return_row(self, row)` – metóda je opakom metódy `move_row`, t. j. vráti všetkých cestujúcich z pozície [3, row + 1] na svoje miesta v jednom kroku, ak majú sedieť v danom rade.

`move_passengers(self)` – metóda posunie všetkých cestujúcich na chodbe a aktualizuje ich pozíciu v dvojrozmernom poli. Pre zistenie novej pozície cestujúceho použite metódu `move` z triedy `Passenger`. Pre predídeniu deadlockov začnite s aktualizáciou pozície cestujúcich na konci lietadla (posledné miesto na chodbe).

`boarding_finished(self)` – metóda zistí, či je nastupovanie dokončené. Vráti `True`, ak na chodbe už nie sú cestujúci a všetky sedadlá sú obsadené. V opačnom prípade vráti `False`.

Trieda `Boarding` (1 bod + 0.5 bodov za každú podtriedu)

V súbore `boarding.py` nájdete kostru kódu triedy `Boarding`.
Konštruktor triedy nemá žiaden parameter, trieda má jednu vnútornú premennú:

- `self.plane` – smerník na lietadlo, v konštruktore nastavený na None

Vašou úlohou je doplniť funkcionalitu nasledovným spôsobom:

`generate_boarding(self, plane_length)` – metóda, ktorá vygeneruje prípad nastupovania daným spôsobom. V hlavnej triede `Boarding` je to prázdna metóda, v podtriedach má vygenerovať cestujúcich podľa pravidiel spôsobu nastupovania (viď vyššie) a pridať ich do lietadla (zavolajte metódu `add_passengers` z `Plane`). Pri generovaní cestujúcich najprv vždy vytvorte jednotlivé skupiny cestujúcich podľa ich miesta, následne náhodne premiešajte skupinu cestujúcich a až potom ich pridajte do lietadla. Metóda má jeden parameter, a to dĺžku lietadla, pre ktoré chcete vytvoriť simuláciu.

`run_simulation(self, plane_length)` – metóda, ktorá spustí simuláciu nastupovania cestujúcich do lietadla. Metóda má jeden parameter, a to dĺžku lietadla (počet radov), pre ktoré chcete vytvoriť simuláciu. V metóde máte vygenerovať poradie nastupovania cestujúcich (pomocou `generate_boarding`). Nastupovanie prebieha opätovným posúvaním cestujúcich až kým nastupovanie nie je dokončené – všetky miesta sú obsadené. Metóda vracia jednu hodnotu, počet potrebných krokov pre ukončenie nástupu. Metódu implementujte iba v triede `Boarding`.

`test_boarding_method(self, plane_length, no_simulation)` – metóda spustí niekoľko simulácií nastupovania danou metódou. Metóda má dva parametre: `plane_length` (počet radov v lietadle) a `no_simulation` (počet simulácií). Metóda vracia dve hodnoty: priemerný počet krokov potrebných na ukončenie nástupu, a zoznam výsledkov jednotlivých simulácií. Metódu implementuje priamo v triede `Boarding`.

Následne implementuje metódy `generate_boarding` v podtriedach `BoardingFTB` (front to back), `BoardingBTF` (back to front), `BoardingWTA` (window to aisle), `BoardingATW` (aisle to window), `BoardingRandom` (úplne náhodné poradie), `BoardingSteffen` (Steffen's perfect).