Nastupujte, prosím!

Každý, kto cestuje lietadlom zažije pocit pri odletovej bráne, že sa rad na lietadlo 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 nasledovné koncepty:

- `Position` – trieda reprezentuje pozíciu cestujúceho v lietadle. Budeme uvažovať o dvoch pohľadoch: z pohľadu pasažiera je x-ová súradnica rad, v ktorom práve stojí a y-ová súradnica vyjadruje sedadlo (písmeno, resp. príslušný index). Z pohľad lietadla bude prvá súradnica číslo riadku, teda sedadiel s rovnakým písmenom, a druhá súradnica bude vyjadrovať rad sedadiel (ako na obrázku nižšie). Trieda je plne implementovaná, nemusíte ju upravovať.
- `Passenger`– trieda reprezentuje jedného cestujúceho s pridelením miestom a batožinou.
- `Plane` – trieda reprezentuje lietadlo s určenou dĺžkou. Z implementačného pohľadu bude lietadlo reprezentované ako dvojrozmerné pole 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!

- `AbstractBoarding` a podtriedy – trieda reprezentuje spôsob nastupovania do lietadla. Trieda `AbstractBoarding` 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 `AbstractBoarding`:

1. back-to-front – cestujúcich rozdelíte do štyroch skupín na základe toho, v ktorom rade sedia. Na lietadlo najprv pustíte 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 do štyroch skupín na základe toho, v ktorom rade sedia. Na lietadlo najprv pustíte 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 do troch skupín na základe toho, kde sedia v rámci troch sedadiel (pri okne, v strede, alebo pri chodbe). Na lietadlo najprv pustíte 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 do troch skupín na základe toho, kde sedia v rámci troch sedadiel (pri okne, v strede, alebo pri chodbe). Na lietadlo najprv pustíte 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, pustíte ich na lietadlo v náhodnom poradí.
6. Steffen's perfect – cestujúcich pustíte 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)

Cestujúci je definovaný nasledovnými hodnotami:
- `row` (`int`) – číslo radu, kde cestujúci má miesto;
- `seat` (`char`) – písmeno sedadla;
- `bags` (`int`) – vyjadruje počet krokov, ktorý cestujúci potrebuje na to, aby svoju batožinu dal do odkladacieho priestoru nad hlavou;
- `plane` (`Plane`) – smerník na lietadlo, do ktorého chce cestujúci nastúpiť;
- `currentPosition` (`Position`) – aktuálna pozícia cestujúceho z jeho pohľadu.

Vašou úlohou je doplniť funkcionalitu nasledovným spôsobom:

Implementujte konštruktor triedy, ktorý nastaví hodnoty členských premenných `row`, `seat`, a `bags` podľa parametrov (na odkladanie každého kusu batožiny potrebuje cestujúci 4 kroky). Zvyšné členské premenné inicializujte na nulu. Ak písmeno sedadla nie je platné, nech sa vygeneruje výnimka typu `ArgumentException`.

`Position? GetPosition()` – 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 `null`.
Pozor: číslo radu miesta cestujúceho nezodpovedá číslu radu v poli, ktoré reprezentuje lietadlo! Nezabudnite vymeniť poradie súradníc v návratovej hodnote.

`Position GetSeatPosition()` – metóda vracia pozíciu sedadla cestujúceho v lietadle (riadok a stĺpec v dvojrozmernom poli).
Pozor: číslo radu miesta cestujúceho nezodpovedá číslu radu v poli, ktoré reprezentuje lietadlo! Nezabudnite vymeniť poradie súradníc v návratovej hodnote.

`void AddToPlane(Plane 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 pozíciu `currentPosition` na `[0, 3]` (0 reprezentuje nultý rad sedadiel, 3 je chodba – index stĺpca a riadku v dvojrozmernom poli reprezentujúcom lietadlo).

`void ForcedToMove(int x, int 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 `currentPosition`.
Dávajte si pozor na správnu reprezentáciu aktuálnej pozície a poradie súradníc!

`bool CanSit()` – metóda určí, či cestujúci má voľnú cestu k 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.

`Position Move()` – metóda reprezentuje pohyb cestujúceho v lietadle. Ak cestujúci ešte nebol pridaný do lietadla, funkcia vyhodí `Exception`. Ak sa cestujúci nachádza v lietadle, metóda vráti vždy novú pozíciu cestujúceho po presune (reprezentácia z pohľadu cestujúceho). Pohyb cestujúceho vieme popísať 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 si ukladá svoju batožinu (znížte hodnotu `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 `MoveRow` a `ReturnRow` v triede `Plane`).

Trieda `Plane` (3 body)

Lietadlo je definované dvomi hodnotami:

- `length` (`int`) – dĺžka lietadla;
- `seats` (`List<Passenger>[,]`) – dvojrozmerné pole, kde na každej pozícii sa nachádza zoznam pasažierov, ktorí stoja na danom mieste.

Vašou úlohou je doplniť funkcionalitu nasledovným spôsobom:

Konštruktor triedy má jeden parameter: `length` (počet radov v lietadle). Na základe tejto hodnoty inicializujte členské premenné, dvojrozmerné pole nech má rozmery 7x(dĺžka lietadla + 2), a nech na každej pozícii sa nachádza prázdny zoznam pasažierov.

`void PrintPlane()/PrintSeats()/PrintCorridor()` – metódy vykresľujú na obrazovku jednoduchú reprezentáciu lietadla. (Slúžia ako pomôcka pre vás, nebudú hodnotené a nemusíte ich upravovať).

`void AddPassengers(List<Passenger> passengerList)` – metóda pridá cestujúcich zo zoznamu do lietadla. Má jeden parameter, `passengerList`, 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`).

`bool IsEmpty(int row, int col)` – metóda vracia `true` ak zoznam na pozícii `[col, row]` je prázdna, `false` v opačnom prípade. Ak daná pozícia neexistuje, vráti `true`. Poradie je pomenené kvôli rôznej reprezentácie pozície pasažiera.

`void MoveRow(int row, char seat)` – 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.

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

`void MovePassengers()` – 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).

`bool BoardingFinished()` – 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 `AbstractBoarding` (1 bod + 0.5 bodov za každú podtriedu)

Konštruktor triedy je defaultný, trieda má jednu vnútornú premennú:

- `plane` (`Plane`) – smerník na lietadlo, pri inicializácii ponechaný ako `null`.

Vašou úlohou je doplniť funkcionalitu nasledovným spôsobom:

`GenerateBoarding(int planeLength)` – metóda, ktorá vygeneruje prípad nastupovania daným spôsobom. V hlavnej triede `AbstractBoarding` je to abstraktná metóda, v podtriedach má vygenerovať cestujúcich podľa pravidiel spôsobu nastupovania (viď vyššie) a pridať ich do lietadla (zavolajte metódu `AddPassengers` 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.

`int RunSimulation(int planeLength)` – 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 `GenerateBoarding`). 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 `AbstractBoarding`.

`List<int> TestBoardingMethod(int planeLength, int simulationNo)` – metóda spustí niekoľko simulácií nastupovania danou metódou. Metóda má dva parametre: `planeLength` (počet radov v lietadle) a `simulationNo` (počet simulácií). Metóda vracia a zoznam výsledkov jednotlivých simulácií (teda počet krokov potrebných na ukončenie nastupovania). Metódu implementuje priamo v triede `AbstractBoarding`.

Následne implementuje metódy `GenerateBoarding` 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).