

Sprawozdanie AiSD
lab4_22.10.2024
Andrzej Janaszek gr 2.
27.10.2024

Pliki:

- zadanie{i}.cpp
- priority_queue.h, queue.h, deque.h - pliki z kolejkami

Implementacje kolejek na szablonach (templates) zostały napisane w 3 osobnych plikach nagłówkowych.

Zadania 1-4:

względnie proste w porównaniu do 5 więc brak komentarza.

Zadanie 5:

opis

LiftManager przyjmuje requesty czyli naciśnięcia przycisków przywołania windy na danym piętrze. Program jest uproszczony - zakłada że od razu można zakolejkować również destynacje tego requesta (osoba naciska na N piętrze przycisk i chce dostać się na M piętro, więc od razu możemy dodać N i M).

Na jednym piętrze może wejść do windy więcej niż jedna osoba - kilka zgłoszeń. Program usuwa z kolejki wpisy dla powtarzających się pięter *from*. Można Było również zastosować kolejkę bez duplikatów.

Kolejkowanie zgłoszeń:

Bezczynna winda zaczyna jechać - trywialny

Winda która jedzie sprawdza czy zgłoszenie jest *osiągalne* i w dobrą stronę, jeżeli tak zostaje dodane do kolejki, jeżeli jest w stronę przeciwną to ląduje na kolejce przeciwnego kierunku, a jeżeli jest nieosiągalne ale w tym samym kierunku to ląduje w kolejce oczekującej (pending) , która jest rozpatrywana przy zmianie kierunku windy.

uwagi:

- metoda request(int, int, Direction) niepotrzebnie przyjmuje wartość destination, ponieważ można ją wyliczyć z from i destination. Jest to zaimplementowane w taki sposób, ponieważ w prawdziwym świecie dostalibyśmy jedynie informacje o piętrze z którego zostało zgłoszone żądanie i kierunek w którym osoba chce się udać (góra, dół)

Wywołanie

Program wypisuje przy każdym ruchu windy (wywołaniu move()) jej stan - każde piętro i pozycje widny (X - jedzie, S - zatrzymana/przystanek)

Dobrze przegląda się wywołanie/wywnik programu debugując go (breakpoint na printState() i puszczając program od wywołania do wywołania metody)