

Wstęp do programowania

Pracownia 13

Uwaga: Wszystkie zadania z tej listy będzie można oddawać do końca semestru. Na liście nie ma premii, żadne z zadań nie wlicza się do maksimum.

Zadanie 1.(1pkt) Popraw klasę `LinkedList` z wykładu 13, by umożliwiała dodawanie list (zarówno `+` jak i `+=`, oraz mnożenie przez liczbę. Postaraj się, żeby dodawać do listy (`LinkedList`) można było również inne iterowalne rzeczy, na przykład zwykłe listy. Musisz samodzielnie znaleźć odpowiednie `__nazwy__` metod.

Zadanie 2.(1pkt) Popraw metodę `__delitem__` klasy `LinkedList` z wykładu, by umożliwiała poprawne usunięcie dowolnego elementu (nie tylko pierwszego). Dodaj również obsługę ujemnych indeksów w metodzie `__getitem__`

Zadanie 3.(1pkt) Wykorzystaj funkcję `reachable` z wykładu 12 do tworzenia map na siatce kwadratowej, przypominających rysunek umieszczony na Skosie. Mapy powinny mieć następujące właściwości:

- Powinny być narysowane na kwadracie, czarne pola to ściany, pozostałe pola – podłoga
- Podłoga w jednej „lokacji” powinna mieć ten sam kolor (lokacją nazwiemy maksymalny zbiór pól, w którym z każdego pola można dojść do każdego innego wykonując ruchy o jedno pole w lewo, prawo, górę lub dół ¹)
- Różne lokacje mają różne kolory

(zwróć uwagę na to, że mapa ze skosu zawiera „brzeg”, tzn. zewnętrzne kwadraciki są czarne. Użycie takich „wartowników” upraszcza kod, a wygląda w miarę estetycznie).

Zadanie 4.(0.5-1*pkt) Zmodyfikuj program z poprzedniego zadania w ten sposób, by rysował bardziej atrakcyjne wizualnie mapy, zawierające mniej regularne komnaty, jakieś korytarze, etc. Ocena zależy od efektu końcowego (0.5 oznacza: ok, rzeczywiście jest mniej regularnie, 1.0 – no faktycznie, ładne).

Zadania z kolokwium

Dodatkowo do listy będą dołączone (wybrane) zadania z kolokwium. Dokładna specyfikacja pojawi się po kolokwium.

¹ Czyli lokacja to spójna składowa grafu 4-sąsiedztwa pól