

Programowanie 2

Zadanie 5

Piotr Błaszyński

27 marca 2019

Proszę przygotować prostą klasę do przechowywania opisu pomieszczeń mających po 4 możliwe wyjścia. Klasa powinna zawierać następujące pola: opis, współrzędne x i y , wskaźniki na pomieszczenia znajdujące się po wszystkich stronach świata (n-północ, e-wschód, s-południe, w-zachód), przedmiot (dla przedmiotu zdefiniować dodatkową klasę z dwoma polami: nazwa i wartość). Metody dostępne w klasie opisującej pomieszczenie: pobranie opisu, pobranie wskaźnika do pomieszczenia sąsiedniego (4 metody), sprawdzenie czy w pomieszczeniu jest przedmiot, zwrócenie przedmiotu. Przykładowy kod demonstracyjny:

- utworzyć 18 pomieszczeń (18 obiektów) i ustawić im wskaźniki na sąsiadów wg takiego wzoru:

```
##X#X###  
X#####X#  
##X#X###
```

- przechować je w tablicy 3x6 (tablica zawiera **wskaźniki** do pomieszczeń, tworzymy obiekty przy pomocy `new`),
- wyświetlić pomieszczenia w następujący sposób:
 - iterujemy po tablicy:
 - jeżeli wskaźnik na sąsiednie pomieszczenie:
 - * jest ustawiony - nie wyświetlamy w tym miejscu nic,
 - * nie jest ustawiony - wyświetlamy znak 'X' - czyli ścianę
 - same pomieszczenia wyświetlamy jako znak '#'
- utworzyć obiekty dla trzech przedmiotów,

- wylosować położenie tych trzech przedmiotów, zapisać to do odpowiedniego obiektu pomieszczenia, (jeżeli ktoś ma problem z kodem do losowania, może te przedmioty utworzyć i dopisać do wybranych pomieszczeń),
- przejść przez pomieszczenia, zaczynając od lewego górnego rogu (nw) i idąc wg schematu: e, s, s, w, e, n, e, e, s, n, n, s, e, e, n, e, e, w, w, s, s, e, e
 - zliczać kroki,
 - sprawdzać w każdym pomieszczeniu, czy jest przedmiot
 - w momencie napotkania przedmiotu wyświetlić numer kroku i nazwę przedmiotu,
- wyświetlić pomieszczenia raz jeszcze, podobnie jak za pierwszym razem, z pokazaniem, gdzie są przedmioty (zamiast '#' wyświetlić pierwszą literę nazwy przedmiotu).

Korzystać z trybu tekstowego, przed wyświetleniem można wypełnić sobie tablicę znaków o rozmiarach 3x8 i w niej przechowywać poszczególne wartości do wyświetlania pomieszczeń.

Przypominam o stosowaniu angielskich nazw klas, pól i metod (szczególnie powtórki typu `print_zawodnik` będę oceniał negatywnie).

Dla ambitnych i szybkich,

- napisać kod znajdujący najkrótszą możliwą drogę pozwalającą na znalezienie wszystkich przedmiotów lub przedmiotów dających najwyższą wartość w stosunku do liczby wykonanych kroków,
- napisać kod generujący takie pomieszczenia o dowolnej długości (powtarza się środkowy fragment),
- do przechowywania wskaźników użyć sprytnych wskaźników (ang. smart pointers).

Przykładowe użycie new:

```
#include <iostream>
using namespace std;
class Simple
{
public:
    Simple() {cout << "ctor_called" << endl;}
    void doSomething() {cout << "doSth_called" << endl;}
};
int main(void)
{
    Simple *simplePointer = new Simple;
    simplePointer->doSomething();
    Simple *pointerToTab = new Simple[3];
    Simple *pointersTab[3];
    pointersTab[0] = new Simple;
    pointersTab[1] = new Simple;
    pointersTab[2] = new Simple;
    delete simplePointer;
    delete pointersTab[0];
    delete pointersTab[1];
    delete pointersTab[2];
    delete [] pointerToTab;
    return 0;
}
```

Przykładowy kod pokazujący użycie konstruktora z listą inicjalizacyjną:

```
#include <iostream>
using namespace std;
class Simple
{
public:
    Simple() {cout << "ctor_called" << endl;}
    Simple(int field, float value):field(field), value(value)
    {cout << "ctor_called:" << field << endl;}
    void doSomething() {cout << "doSth_called" << endl;}
private:
    int field, value;
};
int main(void)
{
    Simple *simplePointer = new Simple(42, 3.14);
    simplePointer->doSomething();
    delete simplePointer;
    return 0;
}
```