

Języki programowania, środa 20.01.2021, Kolokwium

Program wysyłamy na MS Teams najpóźniej o 11:45

Tworzymy dwie klasy: abstrakcyjną Bitwa oraz dziedziczącą z niej klasę Czolgi. Mapa bitwy jest reprezentowana przez dwuwymiarową tablicę przechowującą liczby całkowite `int`. Tablicę charakteryzują liczby `kol` i `wier` (liczba kolumn i wierszy) oraz nazwa.

1. Zaimplementować klasę abstrakcyjną Bitwa

przechowującą pola (dostępne tylko dla klas pochodnych):

- pole `nazwa` – typ `string` przechowujące nazwę mapy
- liczby całkowite `kol` i `wier` – rozmiar mapy (liczba kolumn i wierszy)
- mapa – tablica dwuwymiarowa typu `int` o wymiarach `kol` x `wier` (alokowana dynamicznie podczas tworzenia obiektu klasy) typu `int` przechowująca położenie czołgów; konstruktor domyślnie ustawia wszystkie pola tablicy na 0

konstruktory:

- bezparametrowy (ustawia pola `kol=15`, `wier=10` i `nazwa="RowninaWroclawska"`)
- z parametrami
- kopiujący

destruktor (jeśli konieczny) oraz metody:

- `GetNazwa()` - zwraca pole `nazwa`
- `Wypisz()` - metoda czysto wirtualna

1. Zaimplementować klasę Czolgi, dziedziczącą z klasy abstrakcyjnej

przechowującą pole prywatne:

- pole `przeciwnik1` – typ `string` przechowujący nazwę pierwszego przeciwnika
- pole `przeciwnik2` – typ `string` przechowujący nazwę drugiego przeciwnika

Powinna zawierać konstruktor bezparametrowy (tworzy mapę 20x25 oraz ustawia nazwę "RowninaWroclawska") oraz z parametrami a także destruktor, jeśli potrzebny. W obu konstruktorach klasy podrzędnej używamy odpowiedniego konstruktora klasy nadrzędnej w liście inicjalizacyjnej. Klasa zawiera także deklarację metod:

- `Wypisz()` wypisuje nazwę, liczby `n` i `m` oraz planszę (w formacie `kol` x `wier`), np.:

```
Nazwa: RowninaWroclawska
Przeciwnik 1: Rosja
Przeciwnik 2: Polska
0 0 2 0 0 0 0 0 0 1 1 0
0 0 1 1 1 1 1 1 0 0 0 0
0 0 0 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 0
0 0 1 1 1 1 1 2 0 0 2 0
0 0 1 1 1 1 1 1 1 0 0 0
```

Przeciwnik 1 (Rosja) wygrywa bitwę czołgów!

Bitwę wygrywa ten przeciwnik, który ma więcej czołgów na mapie!

- `Zapisz(char* nazwapliku)` – zapisuje do pliku (nazwa pliku podana jako argument) planszę w postaci takiej samej jak wypisanie jej na ekran metodą `Wypisz`
- `Losuj(int liczbaCzolgow1, int liczbaCzolgow2)` – losuje `liczbaCzolgow1` oraz `liczbaCzolgow2` razy współrzędne (x,y) położenia czołgów obu państw i ustawia je na „1” lub „2”. **Uwaga!** Jeżeli wylosowane zostanie miejsce już zajęte, losowanie należy powtórzyć; oczywiście nie możemy przekroczyć rozmiaru tablicy
- `Rozszerz(int col, int row, int val)` – metoda przyjmuje dwie liczby typu `int` i rozszerza tablicę o podaną liczbę odpowiednio kolumn i wierszy. Nowe elementy powinny mieć ustawione pola na wartość podaną jako trzeci argument metody. **Wskazówka:** w tym celu należy stworzyć tymczasowy obiekt (przez wskaźnik) - tablicę o poszerzonym rozmiarze, przepisać ze starej tablicy istniejące elementy do tych samych pól w nowym obiekcie, wypełnić dodatkowe pola ustawiając je na `val` i usunąć stary obiekt. Na końcu należy wskaźnikowi (czyli polu klasy) przypisać nowo utworzoną tablicę.

3. Napisać program, testujący działanie utworzonych klas

Program powinien wykonywać kolejne instrukcje w zadanej kolejności: tworzy obiekty typu `Czolgi`: `c1`, `c2`, odpowiednio przy użyciu konstruktora domyślnego i z trzema parametrami (`n=10`, `m=15`, `nazwa: "NizinaMazowiecka"`). Następnie w tablicy obu obiektów ustawiamy losowo położenia czołgów. Wypisujemy oba utworzone obiekty na ekran przy użyciu metody `Wypisz`. Następnie rozszerzamy tablicę `c2` o dodatkowe pola – 2 kolumny i 3 wiersze i ponownie wypisujemy ją na ekran oraz zapisujemy oba obiekty do odpowiednich plików (podawanych jako parametry 1 i 2 uruchomienia programu).

Uwaga!

1. Każda klasa, powinna posiadać plik `.cpp` oraz plik `.h`. Deklaracja klasy musi być obłożona dyrektywami `ifndef`.
3. Program, który się nie kompiluje będzie oceniony **maksymalnie na 40% maksymalnej liczby punktów**.