

Programowanie obiektowe – zadanie oceniane (1/4)

09-04-2018

Należy wysłać rozwiązanie w 90 minut po rozpoczęciu.
Liczba punktów do zdobycia: 25.

1. Wysyłanie rozwiązania

Należy spakować projekt w archiwum zip i nadać mu nazwę: login.zip

Należy wysłać email:

To: bednarzm@student.mini.pw.edu.pl

Subject: [PO] Zadanie1/4 2018

Załączniki: login.zip

2. Opis ogólny

Należy stworzyć aplikację symulującą znaną grę w statki. Istnieje kilka rodzajów okrętów mających pewne własności. Strzelaniem do nich zajmować się będzie Działo. W każdej turze oddanych zostanie pewna liczba strzałów. Okręty mają swoją wytrzymałość, więc niektóre toną dopiero po przyjęciu 5-ciu strzałów, a inne już po jednym. Będzie to gra turowa.



3. Opis szczegółowy

Część A (18 pkt).

Celem tego etapu jest stworzenie aplikacji posiadającej hierarchię potrzebnych klas, zdolnej do utworzenia planszy do gry i prawidłowego wypełnienia jej okrętami.

Pole na którym będzie rozgrywana batalia będzie miało 100 x 100 komórek, z których każda będzie referencją, która może odnosić się do każdego z trzech wymienionych poniżej rodzajów okrętów:

➤ Niszczyciel

Zawiera informacje takie jak:

- ➔ Nazwa
- ➔ Unikalny numer okrętu (dla wszystkich okrętów)
- ➔ Unikalny numer niszczyciela (dla wszystkich niszczycieli)
- ➔ Pewna liczba rakiet (1-20)
- ➔ Wytrzymałość (5)
- ➔ Liczba trafień (0)
- ➔ Początkowa pozycja na planszy (dwa indeksy opisujące położenie)

➤ Fregata

Zawiera informacje takie jak:

- ➔ Nazwa
- ➔ Unikalny numer okrętu (dla wszystkich okrętów)
- ➔ Unikalny numer fregaty (dla wszystkich fregat)
- ➔ Pewna liczba dział (1-6)
- ➔ Wytrzymałość (3)
- ➔ Liczba trafień (0)
- ➔ Początkowa pozycja na planszy (dwa indeksy opisujące położenie)

➤ Korweta

Zawiera informacje takie jak:

- ➔ Nazwa
- ➔ Unikalny numer okrętu (dla wszystkich okrętów)
- ➔ Unikalny numer korwety (dla wszystkich korwet)
- ➔ Pewna liczba bomb głębinowych (1-4)
- ➔ Wytrzymałość (1)
- ➔ Liczba trafień (0)
- ➔ Początkowa pozycja na planszy (dwa indeksy opisujące położenie)

Nazwa jest konkatencją nazwy klasy okrętów z numerem okrętu konkretnej klasy.

Pozostałe pola z wyjątkiem pozycji na planszy, są tworzone przez obiekt w momencie jego tworzenia. Części wspólne powinny być umieszczone w nadklasach i tam inicjalizowane.

Powinien zostać stworzony generator do okrętów

Pozycja na planszy (wiersz, kolumna) jest losowana i dostarczana do obiektu w momencie jego tworzenia przez generator okrętów. Obiekt jest wstawiany do planszy na wylosowanej pozycji o ile dane miejsce nie jest zajęte. Jeżeli jest, pozycja losowana jest na nowo. Do skutku.

Każdy okręt powinien mieć dostęp do planszy.

Całą grą powinien zarządzać obiekt "Gra", który:

- Posiada metodę inicjalizującą (int liczbaNiszczycieli, int liczbaFregat, int liczbaKorwet), niewidoczną z zewnątrz, która tworzy obiekty okrętów, rozmieszcza je na planszy i jest niedostępna z zewnątrz.
- Posiada konstruktor pobierający te same parametry, co metoda inicjalizująca i wywołujący ją.
- Posiada konstruktor bezparametrowy, który wywołuje metodę inicjalizującą z następującymi parametrami ustawionymi "na sztywno": 100 (niszczycieli), 500 (fregat), 1000 (korwet).
- Ma w sobie planszę z okrętami i inne potrzebne obiekty.

Część B (4 pkt).

Celem niniejszego fragmentu jest dodanie obiektu Działo do obiektu Gra, które będzie miało możliwość oddawania strzałów i zatapiania trafionych okrętów.

Obiekt "Działo" powinien zostać dodany do obiektu "Gra". W każdej turze działo losuje pozycję na planszy, którą chce ostrzelać. Jeżeli jest pusta (null), nie oddaje strzału. W przeciwnym przypadku (jeżeli jest tam referencja na któryś z okrętów), wywołuje na nim metodę: void ostrzal(). Metodą tą muszą dziedziczyć wszystkie okręty. Okręt odnotowuje w polu "liczbaTrafien" ile razy został trafiony i za każdym razem wypisuje na konsolę: "Okręt Niszczyciel/Fregata/Korweta nr [numer okrętu] został trafiony". W przypadku, gdy liczba trafień jest równa wartości "wytrzymałosc", dany okręt:

- ➔ wypisuje na konsolę: "Okręt Niszczyciel/Fregata/Korweta nr [numer okrętu] został trafiony i ZATOPIONY"
- ➔ Usuwa się z planszy (w miejsce w którym był wstawia wartość null)

Zmiany w obiekcie "Gra".

Obiekt ten powinien implementować interfejs z metodą public abstract void graj(int liczbaTur).

Tyle razy ile wynosi parametr "liczbaTur" powinna być powtarzana następująca sekwencja:

1. Na konsoli powinna być wyświetlona informacja: "Tura nr 1/2/3....."
2. Działo strzela 10 razy na turę.

Część C (3 pkt).

Celem tego etapu jest dodanie możliwości ruchu okrętów na planszy oraz jej prostej wizualizacji.

Zmiany w obiekcie "Gra".

Obiekt ten powinien implementować interfejs (może być ten sam, co w poprzednim etapie tylko z dodatkową metodą), która wypisuje na konsoli całą planszę. W efekcie powinien być widoczny prostokątny obszar. Tam, gdzie nic nie ma, pojawia się znak: "~". Tam, gdzie jest jakiś okręt, na referencji, która może wskazywać na dowolny z tych obiektów należy wywołać metodą implementowaną przez każdy z nich: String getSymbol(). Niszczyciel ma symbol "N", fregata "F", a korweta "K". Taki symbol należy wypisać na konsolę.

Zmiany w obiekcie reprezentującym okręt.

Okręt po trafieniu przemieszcza się w losowym kierunku: góra lub dół oraz lewa lub prawa. Jeżeli jest na brzegu planszy (np. o indeksie 0), to naturalnym jest, że nie może popłynąć w lewo, znalazłby się poza polem gry. Wtedy przesuwa się tylko w drugim kierunku. Za przemieszczenie odpowiada funkcja void move(), która jest wywoływana z referencji na wszystkie obiekty. Losuje ona w którym kierunku okręt ma się przemieścić po trafieniu. Jeżeli nie ma takiej możliwości (jak

już zostało wspomniane powyżej), np. okręt jest na lewej granicy planszy, a z prawej strony jest inny okręt, to nie rusza się wcale.

4. Uwagi

Proszę niepotrzebnie nie eksponować pól ani metod, które nie powinny być widoczne bardziej niż muszą. Należy też stosować dziedziczenie tam, gdzie jest ono potrzebne. Dobrze jest rozlokować klasy po pakietach w sensowny sposób i pamiętać o tym, żeby używać typów o najmniejszej możliwej wielkości. Odstępstwa mogą skutkować ew. obniżeniem punktacji.

W kwestiach nie regulowanych niniejszymi specyfikacjami obowiązuje dowolność.