

## Zadanie 3 - Gra Krwiożercza Biedronka

(praca z tablicami, używanie C# do programowania obiektowego i stosowanie zasad OOP)

### Opis

Stwórz grę 2D w formie aplikacji konsolowej w języku C#. Gra opowiada o biedronce jedzącej mszyce. Gracz przesuwając po ekranie biedronkę w kierunku, w którym pojawiają się mszyce. Z każdym ruchem biedronki, ta traci punkty energii (robi się zmęczona). Kiedy gracz znajdzie biedronkę na polu z mszycą, mszyca zostaje zjedzona, a biedronka zyskuje więcej energii. Gra kończy się, gdy wszystkie mszyce zostaną zjedzone lub gdy biedronka nie ma już energii do poruszania się.

### Tablica

Aby zagrać w grę, należy wyświetlić planszę. Jest to dwuwymiarowa macierz wypełniona znakami reprezentującymi pole w grze planszowej. Pozwól graczowi określić rozmiar matrycy. Przykład: jeśli gracz wstawi cyfrę 5, a twoim znakiem pola jest tylda, powinna wyświetlić się następująca plansza:

```
~ ~ ~ ~ ~  
~ ~ ~ ~ ~  
~ ~ ~ ~ ~  
~ ~ ~ ~ ~  
~ ~ ~ ~ ~
```

### Gracz - biedronka

Gracz może chodzić biedronką po planszy. Biedronka jest reprezentowana przez specjalny znak na ekranie. Biedronka na starcie pojawia się losowo na planszy. Oznacza to, że z każdym startem gry od nowa, biedronka startuje z innego pola.

Jeśli biedronka jest reprezentowana przez hash, tablica będzie wyglądać następująco:

```

~ ~ ~ ~ ~
~ # ~ ~ ~
~ ~ ~ ~ ~
~ ~ ~ ~ ~
~ ~ ~ ~ ~

```

### Ruchy gracza

Gracz może przesuwać biedronkę po planszy. Biedronka może poruszać się tylko w 4 kierunkach: w górę, w dół, w lewo lub w prawo. Gracz porusza biedronką, naciskając określone klawisze (np. W w górę, A w lewo, S w dół i D w prawo). Z każdym ruchem gracz traci energię. Licznik energii powinien być wyświetlany i odświeżany przy każdym ruchu.

Jeśli gracz naciśnie klawisz, aby przesunąć się w prawo, przejście biedronki zostanie wyświetlone w następujący sposób:

```

~ ~ ~ ~ ~      ~ ~ ~ ~ ~
~ # ~ ~ ~      ~ ~ # ~ ~
~ ~ ~ ~ ~      -> ~ ~ ~ ~ ~
~ ~ ~ ~ ~      ~ ~ ~ ~ ~
~ ~ ~ ~ ~      ~ ~ ~ ~ ~
Energia: 5      Energia: 4

```

### Mszyce, wróg

Po rozpoczęciu gry mszyce powinny być wyświetlane na planszy w losowych miejscach. Mszyca jest reprezentowana przez określony znak (inny niż biedronka). Z każdym ruchem gracza mszyce przemieszczają się chaotycznie na najbliższe pole. Mszyca może poruszać się w 8 kierunkach: w górę, na lewe-górne pole, w lewo, na lewe-dolne pole, w dół, na prawe-dolne pole, w prawo, na prawe-górne pole. Możesz zdecydować, ile mszyc ma zostać wygenerowanych na początku gry.

Jeśli zdecydujesz się rozpocząć z 2 mszycami, a znakiem reprezentującym mszycę jest @, plansza może wyglądać następująco:

```

~ ~ ~ ~ @
~ # ~ ~ ~
~ ~ ~ ~ ~
~ ~ @ ~ ~
~ ~ ~ ~ ~

```

Kiedy gracz porusza się z biedronką, mszyce również losowo przemieszczają się na jedno z najbliższych pól. Powiedzmy, że po pierwszym ruchu gracza (przemieszczenia biedronki) pierwsza mszyca przesunie się w lewy-dolny róg, a druga w prawo. Wcześniejsza tablica będzie wyglądać następująco:

```

~ ~ ~ ~ ~
~ # ~ @ ~
~ ~ ~ ~ ~
~ ~ ~ @ ~
~ ~ ~ ~ ~

```

## Rozgrywka

Kiedy biedronka „wchodzi” na mszycę, mszyca zostaje zjedzona. Oznacza to, że mszyca znika z ekranu, a biedronka zyskuje punkty energii.

~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ # @ ~ ~	~ ~ # ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
<i>Energia: 5</i>	<i>Energia: 6</i>

Jeśli biedronka zje wszystkie mszyce, gra się kończy i gracz widzi podsumowanie gry (ilość ruchów, punkty energii itp.). Jeśli biedronce skończą się punkty energii, gra się kończy, a gracz przegrywa.

**Wymagania techniczne:**

1. Utwórz klasę Game z właściwościami (properties) do przechowywania stanu gry (takimi jak zmienne przechowujące rozmiar planszy) i metodami specyficznymi dla gry (na przykład osobna metoda, która rysuje planszę na ekranie jako macierz znaków wypełnioną tyldami).
2. Utwórz klasę Insect jako klasę bazową dla klasy Ladybird (biedronka) i klasy Aphid (mszyca).
3. Aby wygenerować losowy ruch mszyc, użyj klasy Random w .NET
4. Pamiętaj o zaimplementowaniu metod w klasach Game i Insect oraz wywołaniu tych metod w Main(). Innymi słowy: brak logiki i algorytmów w Main()!
5. Ty decydujesz, ile mszyc ma zostać wygenerowanych na początku gry, ile punktów energii ma gracz, ile punktów energii gracz traci przy każdym ruchu, jakimi klawiszami należy naciskać, aby się poruszać biedronka itp.

**Do wykonania:**

1. Wystać na platformę uczelnianą w zasobie Zadanie 3 kompilujący się i działający kod gry.
2. Prześlij raport na platformę uczelnianą w zasobie Zadanie 3 zawierający:
  - A. Zrzuty ekranu z gry (menu, kilka pierwszych ruchów, podsumowanie na końcu) z opisem przebiegu gry
  - B. Wstaw przykłady ze swojego kodu pokazujące, że stworzyłeś grę zgodnie z zasadami programowania obiektowego:
    - I. Podzieliłeś swój program na klasy
    - II. Użyłeś dziedziczenia
    - iii. Użyłeś polimorfizmu