Projekt Świat v2

Wyobraźmy sobie płaski dwuwymiarowy świat o określonych rozmiarach, który składa się z organizmów podzielonych na różne rodzaje, m.in. rośliny (plants) oraz zwierzęta (animals).

W katalogu LivingWorld:

https://github.com/ZeissIQSPL/CppAtAcademia/tree/main/CPP_01_OOP/LivingWorld

znajduje się szkic takiego projektu. Poniższe zadania dotyczą głównie rozbudowy oraz refaktoringu zaprezentowanego kodu.

Zadania

1. **[5 punktów]** Każdy Organizm powinien posiadać pole przechowujące historię jego przodków. Historia jest zaimplementowana jako lista numerów tur narodzin i śmierci przodków. Wyposaż Organizmy w odpowiednie konstruktory (kopiujące, przenoszące), oraz destruktory. Obiekty powinny posiadać również odpowiednie mechanizmy aktualizacji nowych danych. Jak powinny się zachowywać operatory przypisania?
2. **[5 punktów]** Do świata dodaj nowe organizmy zgodnie z opisem świata z projektu PythonWorld https://github.com/tborzyszkowski/PythonWorld/tree/master/Laboratorium

   (katalogi od `World01` do `World04`). Wszystkie klasy (także te istniejące) wyposaż w odpowiedni model dziedziczenia oraz odpowiednie metody polimorficzne.
3. **[5 punktów]** Po realizacji zadania z poprzedniego punktu, spraw aby klasy Organism, Animal i Plant były klasami abstrakcyjnymi.
4. **[5 punktów]** W klasie World mamy naiwny mechanizm serializacji świata. Nie jest on zgodny z SOLID. Przebuduj ten mechanizm tak, aby każdy obiekt sam zajmował się swoją serializacją i deserializacją. Przetestuj nowy mechanizm na wybranych przykładach.
5. **[5 punktów]** Domyślnie program będzie wyświetlał kolejne epoki w konsoli. Zaprogramuj okienkową wersję programu w dowolnej technologii (SDL, Qt, OpenGL, DirectX lub inne).

Termin ostateczny: 12 czerwca 2025 r.

Kod w szablonie projektu nie zawsze napisany jest zgodnie z zaleceniami nowoczesnego programowania w C++. W ramach refaktoringu kodu popraw i oznacz te fragmenty, które mogłyby być napisane bardziej nowocześnie. Opowiesz o nich prowadzącemu podczas obrony. Dodaj własne autorskie komentarze, które pomogą w zrozumieniu kodu.

Podczas obrony projektu należy znać podstawowe zasady programowania obiektowego oraz mechanizmy języka C++ omawiane na wykładach. Prowadzący może poprosić o małe zmiany w kodzie mające na celu sprawdzenie samodzielności oraz zrozumieniu projektu. Ważne aby orientować się w kodzie całego projektu.

Uwaga! Niniejsza specyfikacja może ulec nieznacznym zmianom, aby dokładniej przedstawić wymagania.

Katalog World01 (Opis gatunków)

https://github.com/tborzyszkowski/PythonWorld/tree/master/Laboratorium/World01

Opis świata

W katalogu `begin` znajduje się definicja świata, w którym rządzą następujące zasady:

- świat jest płaski i posiada wysokość i szerokość
- każdy organizm na świecie posiada:
    - power: zwiększa się co jedną turę o 1; decyduje o sile organizmu
    - initiative: priorytet decyduje o kolejności wykonania ruchu w ramach jednej tury
    - position: położenie w świecie
    - liveLength: liczba tur do końca życia
    - powerToReproduce: granica dolna siły, powyżej której może się rozmnażać; po rozmnożeniu traci połowę siły
    - sign: znak reprezentujący organizm w świecie
    - world: świat, w którym żyje organizm

Zwierze

- bazując na definicji organizmu napisać definicję zwierzęcia, które prócz własności organizmu potrafi się przemieszczać i pamięta swoją ostatnią pozycję
- w wyniku przemieszczania organizm może spotkać inny organizm; w wyniku takiego spotkania zachodzą konsekwencje opisane w organizmie (sprawdź szczegóły w pliku źródłowym)

Owca

- bazując na definicji zwierzęcia zdefiniować owcę
- owca przemieszcza się i je trawę
- owca posiada następujące atrubuty:
    - power = 3
    - initiative = 3
    - liveLength = 10
    - powerToReproduce = 6
    - sign = 'S'

Zadanie

- do świata dodać owcę i obserwować jak się rozwija świat w poszczególnych turach.

Katalog World02 (Opis gatunków)

https://github.com/tborzyszkowski/PythonWorld/tree/master/Laboratorium/World02

Mlecz

- bazując na definicji rośliny dodać miniszka lekarskiego
- miniszek lekarski posiada następujące atrubuty:
    - power = 0
    - initiative = 0
    - liveLength = 6
    - powerToReproduce = 2
    - sign = 'D'

Zadanie

- do świata dodać miniszka lekarskiego i obserwować jak się rozwija świat w poszczególnych turach
- czy świat jest dobrze zbalansowany?
- co dzieje się na świecie, na którym jest tylko trawa i miniszek lekarski?

Katalog World03 (Opis gatunków)

https://github.com/tborzyszkowski/PythonWorld/tree/master/Laboratorium/World03

Wilk

- bazując na definicji zwierzęcia dodać wilka
- wilk posiada następujące atrubuty:
    - power = 8
    - initiative = 5
    - liveLength = 20
    - powerToReproduce = 16
    - sign = 'W'

Zadanie

- do świata dodać wilka i obserwować jak się rozwija świat w poszczególnych turach
- czy świat jest dobrze zbalansowany?
- co dzieje się na świecie, na którym jest tylko trawa, miniszek lekarski oraz wilk?

Katalog World04 (Opis gatunków)

https://github.com/tborzyszkowski/PythonWorld/tree/master/Laboratorium/World04

Muchomor

- bazując na definicji rośliny dodać muchomor
- muchomr posiada następujące atrubuty:
    - power = 0
    - initiative = 0
    - liveLength = 12
    - powerToReproduce = 4
    - sign = 'T'
- organizm, który zje muchomora ginie niezależnie od siły

Zadanie

- do świata dodać muchomor i obserwować jak się rozwija świat w poszczególnych turach
- czy świat jest dobrze zbalansowany?
- co dzieje się na świecie, na którym jest tylko muchomor, owca i wilk?