PROGRAMOWANIE ZORIENTOWANE OBIEKTOWO

ĆWICZENIA DODATKOWE

MODUŁ 5

Altkom Akademia S.A., materiały własne

1 KLASY I OBIEKTY

ĆWICZENIE 1.1:

Definiowanie klas i obiektów

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętność:
  - definiowania klas i ich atrybutów
  - tworzenia obiektów na podstawie klas
  - dostępu do atrybutów klas

CELE I ZADANIA:

- Utwórz dwie współpracujące ze sobą klasy
- Zdefiniuj sposób tworzenia obiektów i zachowanie klas

ALGORYTM WYKONANIA:

- Utwórz klasę o nazwie *Point*
  - każdy punkt jest opisywany przez dwie współrzędne: x oraz y
  - jeżeli podczas tworzenia obiektu zostaną one pominięte, to powinien powstać punkt o współrzędnych (0, 0)
  - zdefiniuj metodę zwracającą reprezentację tekstową punktu o postaci: *P(współrzędna_x, współrzędna_y)*
- Utwórz klasę o nazwie *Circle*
  - koło jest opisywane poprzez dwa atrybuty: *center* oraz *radius*
  - środek jest punktem centralnym koła
  - standardowo tworzone jest zawsze koło o środku w punkcie o współrzędnych (0, 0) oraz o promieniu równym 1
  - zmieniając wartości atrybutów można przesunąć koło i zmienić jego rozmiar
  - zdefiniuj metodę zwracającą reprezentację tekstową koła o postaci: *Circle[P(współrzędna_x, współrzędna_y), r = promień]*
  - zdefiniuj w klasie metody obliczające obwód i pole koła
  - utwórz metodę wypisującą pełne informacje o kole (środek, promień, obwód i pole)
- Przetestuj działanie klas – utwórz koło i wypisz informacje o nim
- Zmień parametry koła (współrzędne środka/promień) i ponownie wypisz informacje
- Czy taka konstrukcja klas chroni nas przez wprowadzeniem bezsensownych wartości (np. ujemnego promienia)?

2 HERMETYZACJA

ĆWICZENIE 2.1:

Hermetyzacja

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - hermetyzowania klas

CELE I ZADANIA:

- Dokonaj hermetyzacji klas z poprzedniego ćwiczenia dodatkowego

ALGORYTM WYKONANIA:

- Zmodyfikuj klasy z poprzedniego ćwiczenia dodatkowego: (*Point* oraz *Circle*)
- Dokonaj ich hermetyzacji
- W tym celu atrybuty zadeklaruj jako prywatne
- Modyfikacja wartości atrybutów będzie możliwa za pomocą dodatkowych metod dostępowych
- Podczas zmiany wielkości promienia, zagwarantuj, że przyjmowane będą tylko wartości dodatnie (bez zera)
- Jak zmieni się kod testujący?

ĆWICZENIE 2.2:

Wykorzystanie właściwości

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - tworzenia i wykorzystania właściwości

CELE I ZADANIA:

- Korzystając z rozwiązania poprzedniego ćwiczenia dodatkowego, przebuduj obie klasy, tak aby ich atrybuty były właściwościami

ALGORYTM WYKONANIA:

- Zmodyfikuj klasy z poprzedniego ćwiczenia dodatkowego: (*Point* oraz *Circle*)
- Atrybuty zadeklaruj jako właściwości – to spowoduje, że dostęp do nich, choć wygląda jak bezpośredni, faktycznie będzie się odbywał poprzez metody dostępowe
- Przetestuj działanie programu
- Upewnij się, że inaczej niż to miało miejsce w pierwszym ćwiczeniu, teraz można kontrolować wartości promienia (walidować wartości atrybutów)

ĆWICZENIE 2.3:

Model obiektowy stacji pomiarowych*

Rozwiązanie zadania może się przydać do projektu zaliczeniowego :-)

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - modelowania danych za pomocą klas
  - tworzenia i wykorzystania właściwości

CELE I ZADANIA:

- Zrealizuj te same zadania, co w ćwiczeniu dodatkowym 2.1 z modułu *PYTH_DANE* wykorzystując model obiektowy
- Atrybuty klasy zdefiniuj jako właściwości

ALGORYTM WYKONANIA:

- Zaprojektuj zestaw współpracujących klas koniecznych do opisania stacji pomiarowych
  - klasę *Commune* reprezentującą gminę
  - klasę *City* reprezentującą miejscowość
  - klasę *Station* reprezentującą stację pomiarową
- W klasach atrybuty zdefiniuj jako właściwości (*properties*)
- Dane docelowo będą pobierane z usługi, więc nie będą wymagać walidacji, ani późniejszej modyfikacji
- Zastanów się w jaki sposób zaprojektować klasy niemutowalne
- Klasy będą ze sobą powiązane relacją asocjacji (stacja pomiarowa znajduje się w jakiejś miejscowości, a miejscowość w jakiejś gminie)
- Zastanów się, jak przenieść dane stacji znajdujące się w tej chwili w słownikach do instancji klas
- Zadbaj o to, aby nie tworzyć wielu instancji o identycznym stanie, przykładowo:
  - wiele miejscowości znajduje się w tej samej gminie – wtedy powinna być tylko jedna instancja reprezentująca daną gminę
  - podobnie wiele stacji pomiarowych może znajdować się w tej samej miejscowości – tu też należy użyć pojedynczej instancji miejscowości
- Takie podejście zaoszczędzi pamięć, a ponadto ułatwi dalszą rozbudowę aplikacji, gdy zechcemy zebrane dane przechować w relacyjnej bazie danych
- Pamiętaj także o zdefiniowaniu metody *__str__*, aby w czytelny sposób przedstawić dane o stacjach i ich lokalizacjach
- Przetestuj działanie modelu obiektowego
- Zrealizuj analogiczne zapytania, jak w ćwiczeniu dodatkowym 2.1 z modułu *PYTH_DANE* wykorzystując teraz model obiektowy

ĆWICZENIE 2.4:

Model obiektowy stanowisk pomiarowych*

Rozwiązanie zadania może się przydać do projektu zaliczeniowego :-)

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - modelowania danych za pomocą klas
  - tworzenia i wykorzystania właściwości

CELE I ZADANIA:

- Zrealizuj te same zadania, co w ćwiczeniu dodatkowym 2.2 z modułu *PYTH_DANE* wykorzystując model obiektowy
- Atrybuty klasy zdefiniuj jako właściwości

ALGORYTM WYKONANIA:

- Zaprojektuj zestaw współpracujących klas koniecznych do opisania stanowisk pomiarowych
  - klasę *Param* reprezentującą parametr pomiarowy
  - klasę *Sensor* reprezentującą stanowisko pomiarowe (czujnik)
- W klasach atrybuty zdefiniuj jako właściwości (*properties*)
- Podobnie jak w poprzednim ćwiczeniu instancje, po ich utworzeniu, powinny być niemodyfikowalne
- Utwórz funkcję, która przeniesie dane stanowisk pomiarowych znajdujące się w tej chwili w słownikach do instancji klas
- Zadbaj o to, aby nie tworzyć wielu instancji o identycznym stanie
- Pamiętaj także o zdefiniowaniu metody *__str__*, aby w czytelny sposób przedstawić dane o stanowiskach pomiarowych
- Przetestuj działanie modelu obiektowego
- Zrealizuj analogiczne zapytania, jak w ćwiczeniu dodatkowym 2.2 z modułu *PYTH_DANE* wykorzystując teraz model obiektowy

ĆWICZENIE 2.5:

Model obiektowy danych pomiarowych*

Rozwiązanie zadania może się przydać do projektu zaliczeniowego :-)

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - modelowania danych za pomocą klas
  - tworzenia i wykorzystania właściwości

CELE I ZADANIA:

- Zrealizuj te same zadania, co w ćwiczeniu dodatkowym 2.3 z modułu *PYTH_DANE* wykorzystując model obiektowy
- Atrybuty klasy zdefiniuj jako właściwości

ALGORYTM WYKONANIA:

- Zaprojektuj zestaw współpracujących klas koniecznych do opisania danych pomiarowych
  - klasę *Value* reprezentującą pojedynczy pomiar
  - klasę *Data* reprezentującą zestaw danych pomiarowych
- W klasach atrybuty zdefiniuj jako właściwości (*properties*)
- Dokonaj odpowiedniej konwersji typów danych, np.
  - wartości pomiarowe powinny być liczbami
  - rozważ także, czy nie zdefiniować osobnych właściwości dla daty (możesz użyć typu *date*) i czasu (typ *time*)
- Podobnie, jak w poprzednim ćwiczeniu, instancje, po ich utworzeniu, powinny być niemodyfikowalne
- Utwórz funkcję, która przeniesie dane pomiarowe znajdujące się w tej chwili w słownikach do instancji klas
- Pamiętaj także o zdefiniowaniu metody *__str__*, aby w czytelny sposób przedstawić dane pomiarowe
- Przetestuj działanie modelu obiektowego
- Zrealizuj analogiczne zapytania, jak w ćwiczeniu dodatkowym 2.3 z modułu *PYTH_DANE* wykorzystując teraz model obiektowy

3 DZIEDZICZENIE

ĆWICZENIE 3.1:

MRO

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętność:
  - określania kolejności wyszukiwania atrybutów klas będących w relacji dziedziczenia

CELE I ZADANIA:

- Ustal dla podanej hierarchii klas porządek MRO

ALGORYTM WYKONANIA:

- Przeanalizuj poniższą hierarchię klas:

```
O = object
class F(O): pass
class E(O): pass
class D(O): pass
class C(D, F): pass
class B(E, D): pass
class A(B, C): pass
```

- Określ porządek przeszukiwania (MRO) dla klasy A
- Zweryfikuj swój wynik