PRACA Z PLIKAMI, OBSŁUGA BŁĘDÓW, DOSTĘP DO SYSTEMU PLIKÓW

ĆWICZENIA DODATKOWE

MODUŁ 6

Altkom Akademia S.A., materiały własne

1 MODUŁY I PAKIETY

ĆWICZENIE 1.1:

Model obiektowy stacji pomiarowych – modularyzacja*

Rozwiązanie zadania może się przydać do projektu zaliczeniowego :-)

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętność:
  - podziału aplikacji na warstwy
  - użycia instrukcji importu

CELE I ZADANIA:

- Wykorzystaj rozwiązanie ćwiczenia dodatkowego 2.3 z projektu *PYTH_UML_PROG_OO*
- Dokonaj podziału kodu pomiędzy kilka modułów
- Użyj instrukcji importu, aby uzyskać dostęp do kodu z innych modułów

ALGORYTM WYKONANIA:

- Wykorzystaj rozwiązanie ćwiczenia dodatkowego 2.3 z projektu *PYTH_UML_PROG_OO*
- Umieść definicje klas (stacji, gminy, miejscowości) w osobnych modułach
- Utworzone moduły zgrupuj w pakiecie *model*
- Utwórz kolejny pakiet o nazwie *convert*
- Umieść w nim kod umożliwiający konwersję danych do modelu obiektowego
- Na koniec w pakiecie głównym utwórz moduł zawierający kod testujący poprawność działania utworzonych klas i funkcji

ĆWICZENIE 1.2:

Model obiektowy stanowisk pomiarowych – modularyzacja*

Rozwiązanie zadania może się przydać do projektu zaliczeniowego :-)

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętność:
  - podziału aplikacji na warstwy
  - użycia instrukcji importu

CELE I ZADANIA:

- Wykorzystaj rozwiązanie ćwiczenia dodatkowego 2.4 z projektu *PYTH_UML_PROG_OO*
- Dokonaj podziału kodu pomiędzy kilka modułów
- Użyj instrukcji importu, aby uzyskać dostęp do kodu z innych modułów

ALGORYTM WYKONANIA:

- Wykorzystaj rozwiązanie ćwiczenia dodatkowego 2.4 z projektu *PYTH_UML_PROG_OO*
- Umieść definicje klas (parametrów pomiarowych i stanowisk pomiarowych) w osobnych modułach
- Utworzone moduły zgrupuj w pakiecie *model*
- Utwórz kolejny pakiet o nazwie *convert*
- Umieść w nim kod umożliwiający konwersję danych do modelu obiektowego
- Na koniec w pakiecie głównym utwórz moduł zawierający kod testujący poprawność działania utworzonych klas i funkcji

ĆWICZENIE 1.3:

Model obiektowy danych pomiarowych – modularyzacja*

Rozwiązanie zadania może się przydać do projektu zaliczeniowego :-)

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętność:
  - podziału aplikacji na warstwy
  - użycia instrukcji importu

CELE I ZADANIA:

- Wykorzystaj rozwiązanie ćwiczenia dodatkowego 2.5 z projektu *PYTH_UML_PROG_OO*
- Dokonaj podziału kodu pomiędzy kilka modułów
- Użyj instrukcji importu, aby uzyskać dostęp do kodu z innych modułów

ALGORYTM WYKONANIA:

- Wykorzystaj rozwiązanie ćwiczenia dodatkowego 2.5 z projektu *PYTH_UML_PROG_OO*
- Umieść definicje klas (zestaw danych i pojedynczy pomiar) w osobnych modułach
- Utworzone moduły zgrupuj w pakiecie *model*
- Utwórz kolejny pakiet o nazwie *convert*
- Umieść w nim kod umożliwiający konwersję danych do modelu obiektowego
- Na koniec w pakiecie głównym utwórz moduł zawierający kod testujący poprawność działania utworzonych klas i funkcji

2 WYJĄTKI

ĆWICZENIE 2.1:

Figury – użycie wyjątków standardowych

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętność:
  - użycia wyjątków standardowych
  - obsługi wyjątków

CELE I ZADANIA:

- Wykorzystaj rozwiązanie dodatkowego ćwiczenia 2.2 z projektu *PYTH_UML_PROG_OO*
- Użyj wyjątków standardowych do sygnalizacji sytuacji wyjątkowych
- Dodaj obsługę wyjątków

ALGORYTM WYKONANIA:

- Wykorzystaj rozwiązanie dodatkowego ćwiczenia 2.2 z projektu *PYTH_UML_PROG_OO*
- Użyj mechanizmu wyjątków do zasygnalizowania wystąpienia sytuacji wyjątkowej
- W tym przypadku sytuacją wyjątkową będzie próba ustawienia niedozwolonej wartości atrybutu
- Zastanów się, która klasa wyjątku może być użyta do wskazania, że wartość promienia okręgu nie może być ujemna, ani zerowa
- Wykryj taką sytuację i wyrzuć obiekt wyjątku
- Uruchom program i zobacz, co się stanie, gdy spróbujesz podać niewłaściwą wartość promienia
- Dodaj obsługę wyjątku i nie dopuść do zmiany wartości promienia
- Wypisz na ekranie komunikat wyjaśniający, dlaczego walidacja danych zezwoliła na przyjęcie nowych danych
- W jaki sposób można związać obiekt wyjątku z komunikatem?
- Jak można dotrzeć do tego komunikatu?
- Uruchom ponownie program i sprawdź jego działanie
- Co się stanie, jeśli jako wartość promienia zostanie podana wartość nieliczbowa?
- Dodaj obsługę nowego wyjątku i uruchom ponownie program

ĆWICZENIE 2.2:

Figury – własne klasy wyjątków

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - definiowania własnych klas wyjątków
  - wyrzucania wyjątków i ich obsługi

CELE I ZADANIA:

- Wykorzystaj rozwiązanie poprzedniego ćwiczenia
- Zastąp standardową klasę sygnalizującą błędną wartość promienia – własną klasą

ALGORYTM WYKONANIA:

- Zmodyfikuj rozwiązanie poprzedniego ćwiczenia
- Utwórz dedykowaną klasę wyjątku o nazie *RadiusError*
- Klasa powinna mieć wbudowany komunkat błędu, informujący o niewłaściwej wartości promienia
- Wartość promienia należy przekazać do instancji tworzonego wyjątku
- Zastąp standardową klasę wyjątku własną i sprawdź działanie programu