|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | |  | |
| Test końcowy | | | *PS-06-01-Z1/03*  Ważne od: 2021.07.02 | |
| ***Programowanie Python na poziomie średnio zaawansowanym. Akademia IT***  ***SK-PRO-ORANGE-PYSZ***  ***Data: 20-23.11.2023***  ***Wykładowca: Radosław Janiak***  ***TES\_231120\_2551500PYT\_140408AK\_*** | | |  | |
| ***Dane uczestnika testu*** | ***Wyniki – punktacja*** | |
| ***Imię: proszę uzupełnić*** | ***Maksymalna:*** *….* ***pkt.*** | |
| ***Nazwisko: proszę uzupełnić*** | ***Uzyskana:*** *….* | |
| ***Data wypełnienia testu: proszę wybrać datę*** |  | |
| ***Miejscowość: proszę uzupełnić*** |  | |

**1) Jak nazywamy koncept pozwalający na ukrywanie szczegółów implementacji i wystawianie tylko potrzebnych funkcjonalności w Pythonie?**

Enkapsulacja

Dziedziczenie

Polimorfizm

Abstrakcja

Liczba punktów: ***\_\_\_\_\_***

**2) W jakim celu używa się metody** **\_\_missing\_\_()** **w Pythonie?**

Aby zdefiniować zachowanie słownika, gdy wyszukiwany klucz nie istnieje

Aby zdefiniować zachowanie klasy, gdy brakuje jednej z jej metod

Aby zdefiniować, co się dzieje, gdy brakuje wartości w liście

Aby zdefiniować, co się dzieje, gdy brakuje modułu w importowanych bibliotekach

Liczba punktów: ***\_\_\_\_\_***

**3) Jak nazywamy funkcję, która jest definiowana wewnątrz innej funkcji?**

Funkcja zagnieżdżona

Funkcja anonimowa

Funkcja wyższego rzędu

Funkcja rekurencyjna

Liczba punktów: ***\_\_\_\_\_***

**4) Co oznacza, gdy mówimy, że Python ma "silne typowanie"?**

Zmienne są automatycznie typowane na podstawie wartości, które otrzymują

Typy zmiennych można zmieniać dynamicznie

Wartości zmiennych nie są automatycznie konwertowane na inne typy podczas operacji

Typy zmiennych są definiowane przed użyciem

Liczba punktów: ***\_\_\_\_\_***

**5) Co to jest dekorator w Pythonie?**

Wzorzec projektowy

Funkcja, która przyjmuje inną funkcję jako argument i zwraca nową funkcję

Sposób dodawania metadanych do funkcji

Funkcja, która wykonuje wiele ról

Liczba punktów: ***\_\_\_\_\_***

**6) Który z poniższych nie jest wbudowanym typem danych w Pythonie?**

List

Tuple

Set

Array

Liczba punktów: ***\_\_\_\_\_***

**7) Jaka jest różnica między \_\_str\_\_ a \_\_repr\_\_?**

\_\_str\_\_ jest używany do reprezentacji łańcuchowej, \_\_repr\_\_ do reprezentacji słownika

\_\_str\_\_ jest używany do reprezentacji łańcuchowej, \_\_repr\_\_ do reprezentacji obiektu w sposób czytelny dla dewelopera

\_\_str\_\_ jest używane dla wszystkich obiektów, \_\_repr\_\_ tylko dla obiektów klasy

Brak różnic

Liczba punktów: ***\_\_\_\_\_***

**8) Co to jest "list comprehension"?**

Wzorzec projektowy dla list

Sposób na szybsze przetwarzanie list

Skrócona składnia do tworzenia nowych list na podstawie istniejących kolekcji

Nazwa metody, która wykonuje operacje na listach

Liczba punktów: ***\_\_\_\_\_***

**9) Co to jest "context manager" w Pythonie?**

Specjalna klasa służąca do zarządzania zasobami, takimi jak pliki, połączenia sieciowe czy blokady

Mechanizm zarządzania pamięcią używany przez interpreter Pythona

Menadżer modułów używany do importowania bibliotek

Nazwa metody, która wykonuje operacje na listach

Liczba punktów: ***\_\_\_\_\_***

**10) Co oznacza słowo kluczowe yield w Pythonie?**

Zwraca wartość z funkcji i kończy jej wykonywanie

Wstrzymuje wykonywanie funkcji, zwracając wartość, a następnie wznawia działanie funkcji skąd się zatrzymała

Wstrzymuje wykonywanie bieżącej iteracji pętli

Wskazuje, że funkcja jest dekoratorem

Liczba punktów: ***\_\_\_\_\_***

**11) Jakiego słowa kluczowego używamy do tworzenia własnych wyjątków?**

raise

assert

except

class

Liczba punktów: ***\_\_\_\_\_***

**12) Jak wywołać konstruktor klasy nadrzędnej w Pythonie 3?**

super().\_\_init\_\_()

super(self).init\_\_()

super(self, Class).init()

super(Class, self).init()

Liczba punktów: ***\_\_\_\_\_***