

Imię i nazwisko .....

### Pytanie 1/64

**przedmiot: Projektowanie aplikacji internetowych**

Co to jest timeout (dot. aplikacji webowych)?

- A. jest to maksymalny czas bez aktywności użytkownika, po którym wygasa sesja.
- B. jest to maksymalny czas życia serwletu.
- C. jest to maksymalny czas na załadowanie aplikacji przez kontener.
- D. jest to czas przeznaczony na poprawne zakończenie się czasochłonnych metod.

### Pytanie 2/64

**przedmiot: Projektowanie aplikacji internetowych**

HttpServlet (dot. technologii Java) to:

- A. klasa, po której dziedziczy serwlet.
- B. Interfejs implementowany przez serwlet.
- C. metoda generująca odpowiedź HTTP.
- D. obiekt żądania, który otrzymuje serwlet.

### Pytanie 3/64

**przedmiot: Projektowanie aplikacji internetowych**

Cykl życia serwletu (dot. technologii Java) jest kontrolowany przez:

- A. kontener internetowy.
- B. kontener biznesowy.
- C. deskryptor wdrożenia.
- D. programistę w kodzie aplikacji webowej.

### Pytanie 4/64

**przedmiot: Projektowanie aplikacji internetowych**

Jakie oprogramowanie jest niezbędne (dot. minimalnych wymagań), aby tworzyć i uruchamiać aplikacje internetowe w technologii Java?

- A. Java EE Software Development Kit oraz Java Platform Standard Edition
- B. Java EE Software Development Kit
- C. Java Development Kit oraz zintegrowane środowisko programistyczne (IDE)
- D. Java EE Software Development Kit oraz zintegrowane środowisko (IDE)

**Pytanie 5/64****przedmiot: Projektowanie aplikacji internetowych**

Które środowisko programowania aplikacji internetowych nie jest językiem skryptowym?

- A. Java
- B. JavaScript
- C. Python
- D. PHP

**Pytanie 6/64****przedmiot: Projektowanie aplikacji internetowych**

Która z dostępnych technologii Java dostarcza komponenty interfejsu użytkownika działające w pełni po stronie serwera aplikacji?

- A. JavaServer Faces
- B. Enterprise JavaBeans
- C. JavaFX
- D. Swing

**Pytanie 7/64****przedmiot: Projektowanie aplikacji internetowych**

Sposób odwzorowania obiektowej architektury (Object-Relational Mapping) aplikacji internetowej na bazę danych o relacyjnym charakterze jest możliwy za pomocą następującego mechanizmu dostępnego w języku Java:

- A. Java Persistence API
- B. Java DataBase Connectivity
- C. Dependency Injection
- D. Java Transaction API

**Pytanie 8/64****przedmiot: Projektowanie aplikacji internetowych**

Komponent Javy działający w modelu żądanie-odpowiedź i rozszerzający możliwości serwera aplikacji to:

- A. servlet
- B. applet
- C. skryptlet
- D. JavaBean

**Pytanie 9/64****przedmiot: Projektowanie aplikacji internetowych**

Proszę uzupełnić następujące zdanie.

Mechanizm ... używany jest głównie do utrzymywania sesji, np. poprzez wygenerowanie i odesłanie tymczasowego identyfikatora po zalogowaniu użytkownika do aplikacji internetowej.

- A. Cookie
- B. HTTPS
- C. DNS
- D. URI

**Pytanie 10/64****przedmiot: Projektowanie aplikacji internetowych**

W celu bezpiecznego przesłania danych z formularza (np. loginu oraz hasła) za pośrednictwem przeglądarki internetowej do serwera aplikacji webowej należy użyć następującej metody protokołu HTTP:

- A. POST
- B. GET
- C. PUT
- D. TRACE

**Pytanie 11/64****przedmiot: Projektowanie aplikacji internetowych**

Które stwierdzenie dotyczące filtru (dot. technologii Java EE) nie jest prawdą?

- A. filtr można przypisać tylko do jednego zasobu.
- B. filtr można łączyć z dowolnym typem zasobu.
- C. filtr nie tworzy odpowiedzi dla klienta.
- D. filtr można zastosować do zapisu informacji w logach.

**Pytanie 12/64****przedmiot: Projektowanie aplikacji internetowych**

Obiektu JavaBean można używać określając jeden z 4 różnych zasięgów. Który z nich jest najmniejszy?

- A. strona
- B. żądanie
- C. sesja
- D. aplikacja

**Pytanie 13/64****przedmiot: Programowanie obiektowe 2**Co robi metoda `__init__()` w Pythonie?

- A. Ta metoda jest automatycznie wywoływana, gdy tworzony jest nowy obiekt i inicjalizuje go.
- B. Inicjalizuje klasę do użycia.
- C. Inicjalizuje moduł do użycia.
- D. Inicjalizuje wszystkie atrybuty danych zerem po wywołaniu.

**Pytanie 14/64****przedmiot: Programowanie obiektowe 2**Niech **Person** będzie pewną klasą w języku Python. Rozważając deklarację `p = Person()`, które z następujących stwierdzeń jest najbardziej dokładne?

- A. `p` jest odniesieniem do instancji klasy **Person**
- B. `p` jest obiektem typu **Person**
- C. `p` jest instancją typu **Person**
- D. `p` jest podklasą klasy **Person**

**Pytanie 15/64****przedmiot: Programowanie obiektowe 2**Protokół *menadżera kontekstu* w języku Python składa się z metod:

- A. `__enter__()` i `__exit__()`
- B. `__enter__()` i `__with__()`
- C. `__with__()` i `__exit__()`
- D. `__enter__()` i `__call__()`

**Pytanie 16/64****przedmiot: Programowanie obiektowe 2**

Wskaż zdanie prawdziwe dotyczące klas w języku Python.

- A. Python zezwala na wielokrotne dziedziczenie.
- B. W Pythonie nie można tworzyć własnych klas na podstawie typów takich jak **UserList**, **UserString** lub **UserDict**, które są przeznaczone do bezpośredniego użycia.
- C. W Pythonie nie można tworzyć własnych klas na podstawie typów wbudowanych takich jak **list**, **str** lub **dict**.
- D. W Pythonie nie można tworzyć klas, których instancjami są inne klasy.

**Pytanie 17/64****przedmiot: Programowanie obiektowe 2**

Metoda specjalna `__repr__()` w języku Python musi:

- A. zwracać ciąg Unicode (**str**)
- B. zwracać ciąg Unicode (**str**) lub sekwencję bajtów (**bytes**)
- C. wyświetlać ciąg Unicode (**str**) lub sekwencję bajtów (**bytes**)
- D. zwracać sekwencję bajtów (**bytes**)

**Pytanie 18/64****przedmiot: Programowanie obiektowe 2**

Wywołanie klasy w Pythonie przypomina wywołanie funkcji i zwykle tworzy nową instancję tej klasy. Takie zachowanie jest skutkiem tego, że:

- A. sama klasa jest instancją pewnej metaklasy, która definiuje metodę `__call__()`
- B. sama klasa jest podklasą klasy **object**, która definiuje metodę `__call__()`
- C. sama klasa jest instancją pewnej metaklasy, która definiuje metodę `__get__()`
- D. sama klasa jest podklasą klasy **object**, która definiuje metodę `__get__()`

**Pytanie 19/64****przedmiot: Programowanie obiektowe 2**

Które z poniższych wyrażeń można wykorzystać do sprawdzenia, czy obiekt **obj** jest instancją klasy **A**, czy nie w języku Python?

- A. `isinstance(obj, A)`
- B. `isinstance(A, obj)`
- C. `obj.isinstance(A)`
- D. `A.isinstance(obj)`

**Pytanie 20/64****przedmiot: Programowanie obiektowe 2**

Wskaż zdanie fałszywe dotyczące języka Python:

- A. **object** jest podklasą **type**
- B. **type** jest swoją własną instancją
- C. **object** jest instancją **type**
- D. **type** jest podklasą **object**

**Pytanie 21/64****przedmiot: Programowanie obiektowe 2**

Terminem *kacze typowanie* w języku Python określamy:

- A. Formę polimorfizmu, gdzie funkcje operują na dowolnych obiektach implementujących odpowiednie metody, bez względu na ich klasy lub jawne deklaracje interfejsu.
- B. Praktykę tworzenia kodu, w którym sugerujemy, że może być coś złego w projekcie.
- C. Praktykę pisania programów, które używają informacji czasu wykonywania o sobie, aby zmienić swoje zachowanie.
- D. Budowę bardziej złożonych obiektów z prostszych.

**Pytanie 22/64****przedmiot: Programowanie obiektowe 2**

Co pojawi się na ekranie po wykonaniu poniższego programu:

```
class Point:
    def __init__(self, x = 0, y = 0):
        self.x = x
        self.y = y

    def __sub__(self, other):
        return Point(self.x + other.x, self.y + other.y)

p1 = Point(3, 4)
p2 = Point(1, 2)
result = p1 - p2
print(result.x, result.y)
```

- A. 4 6
- B. 2 2
- C. 0 0
- D. 1 1

## Pytanie 23/64

### przedmiot: Programowanie obiektowe 2

Przeanalizuj poniższy kod programu i zaznacz prawidłową odpowiedź (gdy program zostanie uruchomiony).

```
class Vector:
    def __init__(self, x = 0, y = 0):
        self.x = x
        self.y = y
    def __eq__(self, other):
        return self.x == other.y and self.y == other.x

w = Vector(4,3)
v = Vector(3,4)
print(v == w, v != w)
```

- A. Na ekranie wyświetli się: **True False**
- B. Na ekranie wyświetli się: **False True**
- C. Na ekranie wyświetli się: **True None** ponieważ brakuje zdefiniowanej metody specjalnej `__ne__()` wywoływanej podczas obsługi operatora nierówności (`!=`).
- D. Zostanie zgłoszony wyjątek **TypeError** ponieważ brakuje zdefiniowanej metody specjalnej `__ne__()` wywoływanej podczas obsługi operatora nierówności (`!=`).

## Pytanie 24/64

### przedmiot: Programowanie obiektowe 2

Przeanalizuj poniższy kod programu i zaznacz prawidłową odpowiedź (gdy program zostanie uruchomiony).

```
class A:
    def __init__(self, i = 0):
        self.i = i

class B(A):
    def __init__(self, j = 0):
        self.j = j

b = B()
print(b.i)
print(b.j)
```

- A. Program nie będzie działał, ponieważ klasa **B** dziedziczy po klasie **A**, ale atrybut **i** nie jest dziedziczony.
- B. Program nie będzie działał, ponieważ atrybut **j** nie jest dostępny dla obiektu **b**.
- C. Program będzie działał poprawnie, a klasa **B** dziedziczy po klasie **A** wraz ze wszystkimi atrybutami.
- D. Program nie będzie działał poprawnie, gdyż podczas tworzenia obiektu **b** nie dostarczono argumentu dla konstruktora (na przykład do **b** należy przypisać **B(5)**).

## Pytanie 25/64

### przedmiot: Programowanie obiektowe 2

Przeanalizuj poniższy kod programu i zaznacz prawidłową odpowiedź (gdy program zostanie uruchomiony).

```
class A:
    def __iter__(self):
        i = 10
        while i > 4:
            yield i
            i -= 2
```

```
g = A()
print(next(g))
print(next(g))
print(next(g))
```

- A. Zostanie zgłoszony wyjątek **TypeError**, ponieważ **g** nie jest iteratorem.
- B. Na ekranie wyświetli się:

```
9
7
5
```

- C. Na ekranie wyświetli się:

```
10
8
6
```

- D. W programie jest błąd, ponieważ brakuje instrukcji **return** w metodzie **\_\_iter\_\_()** co spowoduje nieskończoną iterację.

## Pytanie 26/64

### przedmiot: Programowanie obiektowe 2

Załóżmy, że klasa **A** implementuje m.in. metody specjalne **\_\_sub\_\_()** i **\_\_rsub\_\_()** oraz **obj** jest instancją klasy **A**, a **x** instancją jakiejś innej klasy. Kiedy Python wywoła metodę specjalną **\_\_rsub\_\_()** dla obiektu **obj**?

- A. gdy wykonamy działanie:

```
x - obj
```

- B. gdy wykonamy działanie:

```
obj - obj
```

- C. gdy wykonamy działanie:

```
obj - x
```

- D. gdy wykonamy działanie:

```
obj + -obj
```



## Pytanie 27/64

### przedmiot: Programowanie obiektowe 2

Spójrz na poniższy kod w języku Python i wskaż prawidłową odpowiedź.

```
import collections

Card = collections.namedtuple('Card', ['rank', 'suit'])

class Deck:
    RANKS = [str(n) for n in range(2,11)] + list('JQKA')
    SUITS = 'kier karo trefl pik'.split()
    def __init__(self):
        self._cards = [Card(suit,rank) for suit in
                        Deck.RANKS for rank in Deck.SUITS]
    def __len__(self):
        return len(self._cards)

    def __getitem__(self, index):
        return self._cards[index]

def set_card(deck,index,card):
    deck._cards[index] = card

Deck.__setitem__ = set_card
```

- A. Powyższy fragment kodu jest klasycznym przykładem *małego łatania*.
- B. Powyższy fragment kodu jest klasycznym przykładem *kaczego typowania*.
- C. Powyższy fragment kodu jest klasycznym przykładem *gęsiego typowania*.
- D. Powyższy fragment kodu jest klasycznym przykładem użycia *metaprogramowania*.

## Pytanie 28/64

### przedmiot: Programowanie obiektowe 2

Przeanalizuj poniższy kod programu i zaznacz prawidłową odpowiedź (gdy program zostanie uruchomiony).

```
class A:
    counter = 0
    def __init__(self, i = 1):
        self.i = i
        A.count(self.i)

    @classmethod
    def count(cls, j):
        cls.counter += j

a1 = A(4)
a2 = A()
a3 = A(6)
print(a1.counter, a3.counter)
```

A. Na ekranie wyświetli się:

11 11

B. Na ekranie wyświetli się:

4 11

C. Na ekranie wyświetli się:

0 6

D. Na ekranie wyświetli się:

4 6

## Pytanie 29/64

### przedmiot: Java - programowanie sieciowe

Poprawna kolejność wywołani konstruktorów w celu zapisania skompresowanych obiektów do pliku w języku Java wygląda następująco:

- A. new ObjectOutputStream( new ZipOutputStream( new FileOutputStream("nazwa\_pliku")));
- B. new ZipOutputStream(new ObjectOutputStream(new FileOutputStream("nazwa\_pliku")));
- C. new FileOutputStream(new ZipOutputStream(new ObjectOutputStream("nazwa\_pliku")));
- D. new ObjectOutputStream(new ZipOutputStream("nazwa\_pliku"));

**Pytanie 30/64****przedmiot: Java - programowanie sieciowe**

Aby w języku Java obiekty danej klasy były serializowalne klasa ta musi spełniać następujący warunek :

- A. musi implementować interfejs **Serializable**
- B. wszystkie jej pola muszą być typów niezmiennych
- C. musi implementować interfejs **Closable**
- D. wszystkie jej pola muszą być publiczne

**Pytanie 31/64****przedmiot: Java - programowanie sieciowe**

W Javie dostępne są dwa rodzaje wątków, proszę wskazać które to są

- A. Callable, Runnable
- B. Threading, Processable
- C. Threading, HttpCilent
- D. Threading, Runnable

**Pytanie 32/64****przedmiot: Java - programowanie sieciowe**

Tworząc aplikację serwerową w języku Java, oczekujemy na nowe połączenia. Wskaż za pomocą jakiej metody przyjmowane są nowe połączenia

- A. `serverSocket.accept()`
- B. `socket.accept()`
- C. `serverSocket.connect()`
- D. `socket.connect()`

**Pytanie 33/64****przedmiot: Java - programowanie sieciowe**

Za pomocą, której metody w języku Java można utworzyć nowy obiekt **InetAddress** do lokalizacji *example.com*?

- A. `InetAddress ip = InetAddress.getByName("www.example.com");`
- B. `InetAddress ip = new InetAddress.getByName("www.example.com");`
- C. `InetAddress ip = InetAddress.getHost("www.example.com");`
- D. `InetAddress ip = new Socket();`

**Pytanie 34/64****przedmiot: Java - programowanie sieciowe**

Wskaż za pomocą jakiej klasy w języku Java można stworzyć gniazdo do nawiązania połączenia klienta - serwer?

- A. `Socket()`
- B. `SocketServerFactory()`
- C. `socketServer()`
- D. `OutputStream()`

**Pytanie 35/64****przedmiot: Java - programowanie sieciowe**

Czy w Javie można uruchomić wątek **demon**?

- A. Tak, można uruchomić wątek **demon**.
- B. Tak, ale tylko w wersji 8 Javy.
- C. Zależy od platformy na której uruchomiona jest Java.
- D. Nie, Java nie daje takiej możliwości.

**Pytanie 36/64****przedmiot: Java - programowanie sieciowe**

Czy wątki uruchomione w Javie mogą zwracać wartości do wątku głównego?

- A. Tak, wątki typu **Callable** oraz w wyjątkowych sytuacjach wątki typu **Runnable**.
- B. W Javie wątki nie mogą zwracać wartości do wątku głównego.
- C. Tylko wątki typu **Callable** i tylko wartość numeryczne.
- D. Tylko wątki typu **Runnable**.

**Pytanie 37/64****przedmiot: Java - programowanie sieciowe**

Za pomocą której klasy można stworzyć plik archiwum w języku Java ?

- A. `ZipOutputStream()`
- B. `FileOutputStream()`
- C. `ZipInputStream()`
- D. `InputStream()`

**Pytanie 38/64****przedmiot: Java - programowanie sieciowe**

Wskaż, które z metod wykorzystasz do wstrzymania i odblokowania wątku w Javie:

- A. `wait()`, `notify()`
- B. `blok()`, `unblok()`
- C. `stop()`, `start()`
- D. `pause()`, `unlock()`

**Pytanie 39/64****przedmiot: Java - programowanie sieciowe**

W języku Java do pracy z bazą danych wymagane jest:

- A. Sterownik **JDBC/ODBC** do wybranej bazy danych.
- B. Implementacja kodu Java w miejscu, gdzie znajduje się baza danych.
- C. Nic nie jest wymagane, środowisko Java posiada wszystkie wymagane komponenty.
- D. Biblioteki **Database**, która zawiera wszystkie wymagane informacje o bazach danych.

**Pytanie 40/64****przedmiot: Java - programowanie sieciowe**

Czy wykorzystując środowisko Java można odczytać zawartość strony internetowej?

- A. Tak, wykorzystując odpowiednie metody można odczytać zawartość strony internetowej.
- B. Tylko wybrane strony, które wykorzystują framework oparty o Javę.
- C. Tylko wybrane strony, które wykorzystują framework oparty o Javę.
- D. Tak, jeżeli serwer odpowiada na porcie 8081.

**Pytanie 41/64****przedmiot: Podstawy programowania współbieżnego**

Które z podanych określeń definiują pojęcie program współbieżny?

- A. Program złożony ze skończonej liczby programów sekwencyjnych.
- B. Program wykonywany na maszynach wieloprocessorowych.
- C. Program wykonywany przez jeden proces.
- D. Program, w którym kolejność wykonywanych instrukcji jest zdefiniowana.

**Pytanie 42/64****przedmiot: Podstawy programowania współbieżnego**

Co oznacza pojęcie „proces” w abstrakcji współbieżności ?

- A. Program sekwencyjny załadowany do pamięci komputera.
- B. Fragment programu wykonywany współbieżnie
- C. Metoda synchronizująca program współbieżny
- D. Wykonywanie się programu współbieżnego.

**Pytanie 43/64****przedmiot: Podstawy programowania współbieżnego**

Co oznacza określenie „instrukcja atomowa”?

- A. Instrukcja, której wykonanie nie może zostać przerwane.
- B. Instrukcja, która musi być wykonana przez wszystkie procesy wykonujące program.
- C. Instrukcja, której wykonanie nie powinno zostać przerwane, ponieważ spowodowałoby to błąd programu.
- D. Instrukcja zapisu obiektu do pamięci.

**Pytanie 44/64****przedmiot: Podstawy programowania współbieżnego**

Kiedy program współbieżny wykonuje się równolegle?

- A. Gdy liczba procesów wykonujących program nie jest większa od liczby fizycznych procesorów w systemie komputerowym.
- B. Gdy jest uruchomiony na komputerze wieloprocessorowym.
- C. Gdy w trakcie wykonywania programu współbieżnego dochodzi do przełączania kontekstu.
- D. Gdy kilka procesów wykonuje w tym samym czasie daną instrukcję.

**Pytanie 45/64****przedmiot: Podstawy programowania współbieżnego**

Co nazywamy sekcją krytyczną w programie współbieżnym?

- A. Fragment kodu programu współbieżnego, który nie może być wykonywany w tym samym czasie przez więcej niż jeden proces.
- B. Miejsce w programie współbieżnym, w którym dochodzi do przepłotu.
- C. Fragment kodu programu współbieżnego zawierający instrukcje, które nie wykonują się atomowo.
- D. Miejscem w programie współbieżnym, w którym dochodzi do komunikacji między procesami.

**Pytanie 46/64****przedmiot: Podstawy programowania współbieżnego**

W jaki sposób weryfikuje się poprawność programu współbieżnego?

- A. Sprawdza się, czy własność bezpieczeństwa i własność żywotności są spełnione.
- B. Uruchamia się program i analizuje się wynik działania programu.
- C. Sprawdza się, czy nie wystąpi stan zakleszczenia procesów.
- D. Sprawdza się, czy każdy proces wykona swoje zadanie.

**Pytanie 47/64****przedmiot: Podstawy programowania współbieżnego**

Które z podanych scenariuszy wykonania programu współbieżnego zapisanego w pseudokodzie są prawdziwe?

Program współbieżny:

```
Semafor s 1  
  
p1: wait (s)    q1: wait(s)  
p2: signal(s)   q2: exit  
p3: exit
```

- A. Program się nie zakończy, a wartość semafora s będzie równa 0.
- B. Program się zakończy z wartością semafora s równą 1.
- C. Program się nie zakończy, a wartość semafora s będzie równa 1.
- D. Program się zakończy z wartością semafora s równą -1.

**Pytanie 48/64****przedmiot: Podstawy programowania współbieżnego**

Bariera to narzędzie synchronizacji procesów używane w sytuacji wymagającej, aby

- A. zadania zaczęły się wykonywać równocześnie
- B. zadania zakończyły się w tym samym momencie
- C. zadania były szeregowane według ustalonej kolejności
- D. zadania mogły wykonywać się równolegle

**Pytanie 49/64****przedmiot: Podstawy programowania współbieżnego**

Program współbieżny jest wykonywany przez cztery procesy, z których każdy wykonuje instrukcje p1... p9 zapisane w pseudokodzie:

Program współbieżny:

```
semafor s1
semafor b0
integer m ?

p1: wait(s)
p2: m m-1
p3: if (m=0)
p4:     signal(b)
p5: else
p6:     signal(s)
p7: wait(b)
p8: signal(b)
p9: exit
```

Dla jakiej wartości zmiennej m program się zakończy

- A.  $m = 4$
- B.  $m < 4$
- C.  $m > 4$
- D. Program zakończy się dla każdej wartości  $m > 0$

**Pytanie 50/64****przedmiot: Podstawy programowania współbieżnego**

Jaką rolę pełnią zmienne warunkowe w monitorach klasycznych?

- A. Służą do warunkowego zatrzymania procesu w trakcie wykonywania operacji na monitorze.
- B. Służą do rozstrzygania, który proces ma wykonać operację na monitorze w sytuacji, gdy kilka procesów równocześnie próbuje ją wykonać.
- C. Definiują warunki, przy których proces powinien być zatrzymany.
- D. Gwarantują wzajemne wykluczanie podczas wykonywania operacji na monitorze.

**Pytanie 51/64****przedmiot: Podstawy programowania współbieżnego**

W problemie synchronizacyjnym znanym jako „czytelnicy i pisarze” własność bezpieczeństwa jest naruszona w przypadku, gdy

- A. w czytelni jest kilka procesów, a jeden z nich zapisuje dane
- B. procesy odczytują te same dane w czytelni
- C. proces wykonuje odczyt danych przed procesem zapisującym dane
- D. procesy współbieżnie odczytują dane w czytelni



**Pytanie 52/64****przedmiot: Podstawy programowania współbieżnego**

Do czego służy algorytm Dekkera?

- A. Do zapewnienia wzajemnego wykluczania w dostępie do sekcji krytycznej, w przypadku rywalizacji dwóch procesów.
- B. Do zapewnienia wzajemnego wykluczania w dostępie do sekcji krytycznej, w przypadku rywalizacji wielu procesów.
- C. Służy do szeregowania procesów w przypadku, gdy procesy oczekują na wolny procesor.
- D. Służy do wyboru procesu zarządcy w programie współbieżnym.

**Pytanie 53/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Podstawowe przekształcenia geometryczne obrazu to

- A. Przesunięcia, obroty, odbicia obrazu oraz transformacje jego geometrii.
- B. Przesunięcia, przewroty, odbicia obrazu oraz transformacje jego geometrii.
- C. Przesunięcia, obroty, podbicia obrazu oraz transformacje jego geometrii.
- D. żadna z pozostałych odpowiedzi nie jest poprawna

**Pytanie 54/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Do punktowych przekształceń obrazu zalicza się

- A. Przekształcenia logiczne, arytmetyczne, look-up table, wyrównywanie histogramu i binaryzacja.
- B. Przekształcenia logiczne, asymetryczne, look-up table, wyrównywanie histogramu i binaryzacja.
- C. Przekształcenia logistyczne, arytmetyczne, look-up table, wyrównywanie histogramu i binaryzacja.
- D. żadna z pozostałych odpowiedzi nie jest poprawna

**Pytanie 55/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Do operacji logicznych zaliczane są

- A. Negacja obrazu, suma logiczna obrazów, równoważność logiczna.
- B. Negacja obrazu, suma logiczna obrazów, wyrównanie histogramu.
- C. Negacja obrazu, wyrównanie histogramu, równoważność logiczna.
- D. żadna z pozostałych odpowiedzi nie jest poprawna

**Pytanie 56/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Do podstawowych operacji arytmetycznych realizowanych na obrazach należą.

- A. Kombinacja liniowa dwóch obrazów, przemnożenie obrazu przez stałą.
- B. Binarystacja, przemnożenie obrazu przez stałą.
- C. Kombinacja liniowa obrazów, binarystacja.
- D. żadna z pozostałych odpowiedzi nie jest poprawna

**Pytanie 57/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Różniczkowanie funkcji dwuwymiarowej w przypadku gradientu Robertsa odbywa się wzdłuż kierunku pod kątem

- A. 45 stopni
- B. 30 stopni
- C. 60 stopni
- D. 120 stopni

**Pytanie 58/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Dwa podstawowe sposoby rozmieszczania cyfrowych elementów obrazu to

- A. Siatka heksagonalna i siatka kwadratowa.
- B. Siatka heksagonalna i macierz konwolucji.
- C. Macierz konwolucji i siatka kwadratowa.
- D. Macierz heksagonalna i macierz konwolucji.

**Pytanie 59/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Przekształcenie obrazu RGB na obraz czarno-biały nazywamy

- A. binarystacją, progowaniem
- B. filtracją, manipulacją
- C. skalowaniem
- D. histogramem

**Pytanie 60/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Celem filtrów górnoprzepustowe jest

- A. Zwiększenie szczegółów w obrazie w tym ewentualnych szumów.
- B. Zmianę kolorów obrazu.
- C. Zmniejszenie szczegółów w obrazie.
- D. Rozmycie niektórych fragmentów obrazu i jednocześnie wyostrenie pozostałych.

**Pytanie 61/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Stosowanie filtrów dolnoprzepustowych na dowolnych obrazach powoduje

- A. Usunięcie lub zminimalizowanie szumów oraz rozmycie obrazu.
- B. Zwiększenie szczegółów w obrazie w tym nasilenie szumu w obrazie.
- C. Wyostrenie obrazu i rozmycie kolorów.
- D. Zwiększenie kontrastu i zmianę kolorów.

**Pytanie 62/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Do podstawowych operacji kontekstowych zaliczane są

- A. filtracja liniowa i nieliniowa, wyszukiwanie narożników / krawędzi
- B. filtracja linowa, binaryzacja obrazu i operacje asymetryczne
- C. zmianę wartości piksel, wyrównanie histogramu
- D. operacje arytmetyczne i logiczne, przekształcenia geometryczne

**Pytanie 63/64****przedmiot: Przetwarzanie obrazów cyfrowych**

Na czym polega binaryzacja?

- A. Redukcji ilości informacji zawartej w obrazie.
- B. Zamianie dowolnych kolorów na inne kolory.
- C. Wyodrębnieniu niewidocznych szczegółów z obrazów kolorowych.
- D. Zachowaniu ilości informacji zawartej w obrazie.

**Pytanie 64/64****przedmiot: Przetwarzanie obrazów cyfrowych**

W jaki sposób wykrywane są krawędzie z obrazu

- A. Na podstawie macierzy obliczającej różnice wartości pikseli w skali szarości.
- B. Na podstawie wzorców za pomocą których dopasowywane są krawędzie.
- C. Na podstawie podobieństwa kolorów.
- D. Na podstawie histogramu.