

Pewniaki – pytania potwierdzone na wejściówkach w 2023:

Proszę wymienić nazwę jednej klasy z pakietu `java.util.concurrent` wykorzystującą tablicę kopiowaną przy zapisie (proszę wpisać nazwę klasy bez nazwy pakietu):

Poprawne odpowiedzi (zapisujemy jedną z nich, bez spacji, wielkość liter prawdopodobnie MA znaczenie):

- `CopyOnWriteArrayList`
- `CopyOnWriteArraySet`

Dla której z podanych poniżej kolekcji nie można wywoływać metody `remove()`?

Wybierz jedną odpowiedź:

- a. `ConcurrentMap`
- b. `unmodifiableCollection`
- c. `synchronizedCollection`
- d. `Collection`

Pojęcie „widoczność” w odniesieniu do PW oznacza:

Wybierz jedną odpowiedź:

- a. rodzaj klasy nadrzędnej
- b. typ danych zwracanych przez metody danego obiektu
- c. `problematykę odczytu i zapisu obiektów współdzielonych`
- d. rodzaj użytego modyfikatora: `private`, `protected` lub `public`

Instrukcja atomowa to taka która:

Wybierz wszystkie poprawne:

- a. `Jest niepodzielna`
- b. `wykonuje się w jednym cyklu procesora`
- c. `Jest rozszczepialna jak atom`
- d. `Wymaga synchronizowania za pomocą semaforów`

Która z poniższych metod powoduje zablokowanie procesu do czasu skutecznego pobrania danych z kolejki `BlockingQueue`?

Wybierz jedną odpowiedź:

- a. `remove()`
- b. `poll(time, unit)`
- c. `take()`
- d. `poll()`

Podaj nazwę metody zwracającej pustą kolekcję:

`emptyList`

Napisz klasę niezmienną:

W skrócie: robimy klasę w której wszystkie pola są final private, nie robimy żadnych setterów, żadna metoda nie może zostać przesłonięta. Jeśli chcemy zrobić coś w rodzaju settera to robimy metodę która zwróci nowy obiekt z ustawioną daną wartością.

Przykład klasy niezmiennej:

```

public class Prostokat {
    private final int posX;
    private final int posY;

    public Prostokat(int posX, int posY) {
        this.posX = posX;
        this.posY = posY;
    }

    public int getPosX() {
        return posX;
    }

    public Prostokat setPosX(int posX) {
        return new Prostokat(posX, this.posY);
    }

    public int getPosY() {
        return posY;
    }

    public Prostokat setPosY(int posY) {
        return new Prostokat(this.posX, posY);
    }
}

```

Napisz klasę z metodą synchronizowaną:

W skrócie: robimy dowolną klasę, z metodą która współbieżnie mogłaby być niebezpieczna (np. inkrementacja), przez modyfikatorem dostępu metody dopisujemy „synchronized”. Tyle.

Przykład klasy z metodą synchronizowaną:

```

public class KlasaABC {
    private int i = 0;

    synchronized public int incrementI() {
        return ++i;
    }
}

```

Napisz co to jest klasa niezmienna:

Klasa niezmienna to klasa, której obiekty nie mogą być modyfikowane. Wszystkie dane zawarte w każdym obiekcie tej klasy są podawane w czasie tworzenia obiektu i pozostają niezmiennie przez cały czas istnienia tego obiektu.

Istnieje 5 reguł koniecznych do utworzenia klasy nieziennej:

1. Nie twórz żadnych metod modyfikujących obiekt.
2. Upewnij się, że żadna metoda nie może zostać przesłonięta.
3. Zadeklaruj wszystkie pola jako final.
4. Zadeklaruj wszystkie pola jako private.
5. Zapewnij wyłączny dostęp do dowolnych modyfikowanych komponentów.

Opisz co to jest final i po co się go stosuje:

Final to słowo kluczowe, które zależnie od użytego kontekstu oznacza niezmiennosc deklarowanego elementu. Finalne mogą być zarówno zmienne, metody, jak i klasy:

- Zmiennej finalnej można tylko raz przypisać wartość, co z reguły ma miejsce podczas jej deklaracji.
- Metody finalnej nie można zaimplementować w klasie pochodnej – można to zastosować, aby zapobiec niespodziewanemu zachowaniu się metody w podklasach.
- Klasa finalna nie może być dziedziczona przez inne klasy, co może być podyktowane względami bezpieczeństwa lub efektywności. Wszystkie metody w klasie finalnej są automatycznie finalne.

Napisz metodę z blokiem synchronizowanym:

```
public class TestSynchronizedBlock extends Thread {

    static MyObject value = new MyObject();

    @Override
    public void run() {
        while (true) {
            synchronized (value) {
                value++;
            }
        }
    }

    public static void main(String[] args) {
        TestSynchronizedBlock[] t = new TestSynchronizedBlock[2];

        t[0] = new TestSynchronizedBlock();
        t[1] = new TestSynchronizedBlock();

        t[0].start();
        t[1].start();
    }
}
```

Podaj nazwę klasy zwracającą słabo spójne iteratory:

WeakHashMap

Pozostałe pytania – mogą się trafić:

Interfejs Callable zwraca obiekt:

Wybierz jedną odpowiedź:

- klasy String
- interfejsu Runnable
- dowolnej, ale zdefiniowanej klasy lub interfejsu
- klasy V

Metodą oczekiwania na zakończenie procesu klasy Process jest:

Wybierz jedną odpowiedź:

- a. proces.close()
- b. proces.waitForClose()
- c. **proces.waitFor()**
- d. proces.waitForEnd()

Czy każda klasa niezmienna jest statyczna?

Prawda

Fałsz

Wyjaśnienie: Klasa niezmienna może być statyczna lub nie-statyczna.

Obiekty, które w sposób formalny nie są niezmiennie, ale których stanu nie można zmienić po opublikowaniu:

Wybierz jedną odpowiedź:

- a. nie mogą być używane razem z dodatkową synchronizacją
- b. mogą być bezpiecznie używane przez dowolny wątek pod warunkiem dodatkowej synchronizacji
- c. **mogą być bezpiecznie używane przez dowolny wątek bez dodatkowej synchronizacji**
- d. nie mogą być bezpiecznie używane przez dowolny wątek

Czy klasę *ThreadMXBean*, można wykorzystać do pobrania informacji o aktualnym procesie?

Prawda

Fałsz

Czy możliwa jest zmiana wartości pól obiektu, do którego referencja znajduje się w polu finalnym?

Prawda

Fałsz

Czy obiekty, do których referencje znajdują się w polach finalnych mogą być później modyfikowane?

Prawda

Fałsz

Czy klasa *HashMap* jest bezpieczna wielowątkowo?

Prawda

Fałsz

Jakie warunki muszą być spełnione w celu bezpiecznej publikacji obiektów?

Wybierz jedną odpowiedź:

- a. Żaden - wszystkim zajmuje się maszyna Javy
- b. Obiekt musi być widoczny dla innych wątków
- c. **Zarówno obiekt i referencja do niego muszą być widoczne dla innych wątków dokładnie w tym samym momencie**
- d. Referencja do obiektu musi być widoczna dla innych wątków

Czy tworzenie nowych zadań wykonywanych współbieżnie możliwe jest jedynie poprzez implementację interfejsu Runnable?

Prawda

Fałsz

Wyjaśnienie: Tworzenie nowych zadań wykonywanych współbieżnie jest możliwe przez interfejs Runnable oraz poprzez dziedziczenie po klasie Thread.

Czy blok synchronized działa jak semafor binarny (muteks)?

Prawda

Fałsz

Do uzyskania obiektu klasy ThreadMXBean służy:

Wybierz jedną odpowiedź:

- a. new ThreadMXBean()
- b. ThreadMXBean.newInstance()
- c. ManagementFactory.getThreadMXBean()
- d. ThreadMXBean.getInstance()

Czy wartości pól z modyfikatorem volatile mogą być modyfikowane jedynie w konstruktorze?

Prawda

Fałsz

Wyjaśnienie: Pola z modyfikatorem volatile mogą być modyfikowane w dowolnym momencie.

Czy wartości pól z modyfikatorem final mogą być modyfikowane jedynie w konstruktorze?

Prawda

Fałsz

Czy metodą klasy Runtime do uruchamiania procesów jest exec()?

Prawda

Fałsz

Executor jest interfejsem, a nie klasą:

Prawda

Fałsz

Czy synchronizacja danych musi być w Javie stosowana w każdym (dosłownie) przypadku?

Prawda

Fałsz

Czy wznowienie pracy procesu, który wywołał Thread.sleep(), jest możliwe dopiero po zadeklarowanym czasie?

Prawda

Fałsz

Wartością zwracaną przez Thread.setUncaughtExceptionHandler() jest:

Wybierz jedną odpowiedź:

- a. RuntimeException
- b. Thread
- c. void
- d. Throwable

Czy obiekt klasy CountdownLatch może być używany wielokrotnie?

Prawda

Fałsz

Czy kolekcja HashSet podczas obsługi współbieżnej musi być synchronizowana w celu zapewnienia pełnej spójności danych?

Prawda

Fałsz

Co się stanie, jeżeli podczas iteracji instancji klasy Vector, element kolekcji zostanie współbieżnie usunięty?

- a. Zgłoszony zostanie wyjątek ConcurrentModificationException
- b. Zgłoszony zostanie wyjątek ArrayOutOfBoundsException
- c. Usuwany element zostanie jedynie pominięty podczas iteracji
- d. Wynik operacji będzie nieświeży.

CopyOnWriteArraySet nie wymaga synchronizacji, ponieważ:

Wybierz jedną odpowiedź:

- a. Podczas iteracji kolekcji nie możliwe jest jej modyfikowanie
- b. Podczas odczytu tworzony jest obraz kolekcji, aby dany proces posiadał własną kopie
- c. Posiada wbudowany mechanizm synchronizacji
- d. Podczas zapisu tworzona jest nowa kopia kolekcji

Ile procesów na raz może wymieniać się buforami przy pomocy obiektu klasy Exchanger?

Wybierz jedną odpowiedź:

- a. 3
- b. 4
- c. dowolna, parzysta ilość
- d. 2
- e. 1
- f. tyle, ile było zarejestrowanych

Czy dostęp współbieżny do zapisu kolekcji ConcurrentHashMap jest ograniczony dla pewnej puli wątków?

- a. Prawda
- b. Fałsz