

Zaznacz prawdziwe zdania dotyczące dziedziczenia w języku java.

clear

Klasa potomna ma zawsze dostęp do wszystkich składowych klasy rodzica.

clear

Można dziedziczyć po dowolnej liczbie klas.

done

Klasa potomna nie musi mieć dostępu do wszystkich składowych obiektu rodzica.

done

Można dziedziczyć tylko po jednej klasie.

clear

Klasa potomna dziedziczy wszystkie konstruktory po klasie rodzica.

Zaznacz prawdziwe zdania dotyczące zasady otwarte-zamknięte (Open-Close Principle).

DO7I2NPPM849LE96VEOB1B51GYPW9G6VVUYDPRQX2G9D6WXM

clear

Zasada mówi, że powinno się korzystać wyłącznie z oprogramowania dostarczonego na otwartej licencji.

clear

Zasada mówi, że wszystkie pola klasy muszą być prywatne, a wszystkie metody publiczne.

done

Zasada mówi, że elementy systemu powinny być zamknięte na modyfikacje , a otwarte na rozszerzanie.

done

Można zmieniać zachowanie obiektu bez zmieniania jego kodu.

done

Zasadę można zastosować w praktyce wykorzystując dziedziczenie.

done

Zasadę można zastosować w praktyce wykorzystując kompozycję.

clear

Zasada mówi, że elementy systemu powinny być zamknięte na rozszerzanie , a otwarte na modyfikacje.

Zaznacz prawdziwe zdania dotyczące słowa kluczowego this w języku java.

0EUS4XB78AQE87BKJVIYF1I1TJMTT5QRIPMQW7BH65D6H4SJ

done

Możemy za jego pomocą wywołać konstruktor danej klasy.

clear

Możemy za jego pomocą wywołać konstruktor z klasy rodzica.

clear

Oznacza wskaźnik do obiektu, na rzecz którego wykonujemy metodę.

done

Oznacza referencję do obiektu, na rzecz którego wykonujemy metodę.

Zaznacz prawdziwe zdania dotyczące klas abstrakcyjnych w języku java.

5KDO03XXV14G8JFFZMFMT2UMGJWVB0Y5Q7DTE532U2T65GHH

clear

Nie mogą dziedziczyć po innych klasach.

clear

Nie można po nich dziedziczyć.

done

Nie można utworzyć obiektów tych klas.

clear

Muszą zawierać conajmniej jedną metodę abstrakcyjną.

done

Jeśli klasa zawiera conajmniej jedną metodę abstrakcyjną, to musi być klasą abstrakcyjną.

Zaznacz wszystkie prawdziwe zdania dotyczące hermetyzacji (enkapsulacji).

88KDNAQJBZZWFTJUWR66WXKU44KBV24YGRYYRHBQZ3HQ8LOH

done

Zasadę hermetyzacji narusza uczynienie wszystkich pól klasy polami publicznymi.

done

Pozwala na ukrycie przed użytkownikiem szczegółów implementacji.

done

Jej celem jest oddzielenie implementacji od interfejsu.

done

Jednym z przykładów jej zastosowania jest ograniczenie dostępu do pewnych pól/metod.

clear

Hermetyzację możemy osiągnąć wtedy i tylko wtedy gdy wszystkie pola i metody uczynimy prywatnymi.

clear

Każdy obiektowo zorientowany język programowania musi wspierać hermetyzację.

Zaznacz nieprawdziwe zdania dotyczące zmiennych statycznych w Java.

NU1GHJNRSWWIPBWCNLG0JCUF4XVVF397HGH1LOF6ABS7LNT0

done

Są inicjalizowane w momencie tworzenia nowego obiektu.

done

Nie można zmienić ich wartości.

clear

są inicjalizowane w momencie ładowania klasy.

clear

Są wspólne dla wszystkich obiektów danej klasy.

Zaznacz prawdziwe zdania dotyczące dziedziczenia.

4RY9F4NN3ILUFVGI5LOQ32WXLJN2JIVCWD34X6DZSW0VCKM3

done

Jest to jeden ze sposobów powtórzenia raz napisanego kodu.

done

Obiekt klasy potomnej zawiera w sobie podobieństwo klasy rodzica.

clear

Klasa rodzica jest szczególnym przypadkiem klasy potomka.

done

Zgodnie z przyjętą konwencją, klasę rodzica umieszczamy na diagramach UML ponad klasami potomnymi.

done

Na diagramach UML oznaczamy je za pomocą strzałki z trójkątnym grotem, który nie jest wypełniony żadnym kolorem.

done

Jest to jeden ze sposobów realizacji reguły "Don't Repeat Yourself"

W wersjonowaniu semantycznym każdej z liczb odpowiada odpowiednia interpretacja. Dopasuj liczbę do jej interpretacji. (//W teście jest wybier, tu zaznacz prawdziwe//komentarz autora testownika).

A82TN0LOTJK0DANJWS9JFPA619XGJKNPHJEAU3A2LOLE8PG7

done

Patch - drobna poprawka, nie zmienia API.

done

Minor Version - interfejs zostaje rozszerzony, ale pozostaje kompatybilny wstecznie.

done

Major Version - dodana została nowa funkcjonalność, interfejs traci kompatybilność wsteczną

Zaznacz prawdziwe zdania dotyczące słowa kluczowego super w języku java.

UIFVDWFPRJPRBK8QCTY1SJGJY5K2TUCGVFE30TSJNM8GF3XS

clear

Możemy za jego pomocą wywołać konstruktor danej klasy.

clear

Oznacza referencję do obiektu, na rzecz którego wykonujemy metodę.

done

Oznacza referencję do podobiektu klasy rodzica.

done

Możemy za jego pomocą wywołać konstruktor pochodzący z klasy rodzica.

clear

Oznacza wskaźnik do podobiektu klasy rodzica.

Zaznacz prawdziwe zdania dotyczące prywatnych pól klasy.

WYNFMO3FCFC68D551CIDLEE23VLK5PACJTHIVP9MHTIIIOVIQ

clear

Widoczne są w klasach dziedziczących po klasie macierzystej.

clear

Dowolny obiekt może mieć do nich dostęp.

clear

Widoczne są dla innych klas znajdujących się w tym samym pakiecie.

done

Można mieć do nich dostęp tylko w metodach należących do tej klasy.

Zaznacz prawdziwe zdania dotyczące zasady odwracania zależności (Dependency Inversion Principle).

K5GD4AFQQH5UY6XG1QC11ZB3ANNYJ60C9WC26YZZ58AWQJKU

clear

Zależności pomiędzy klasami powinno się definiować na poziomie konkretnych klas (Concrete Classes).

done

Zależności pomiędzy klasami powinno się definiować na jak najwyższym poziomie hierarchii dziedziczenia.

clear

Powinno się unikać stosowania interfejsów i klas abstrakcyjnych.

done

Zależności pomiędzy klasami powinny wynikać z abstrakcji

Zaznacz prawdziwe zdania dotyczące konstruktorów w języku java.

0IQP3UW81J6EMK2G3AKT517DYGUDK5ADFFD5COFZK94REQIJ

done

Wywołany jest w momencie użycia operatora new.

clear

Inicjuje tylko statyczne zmienne klasy.

clear

Ładuje wskazaną klasę do pamięci.

done

Tworzy nowy obiekt danej klasy.

Zaznacz wszystkie prawdziwe zdania dotyczące kompozycji.

PQ1KWYACPNRJVDM34N7ORW7YR5GCZU7CYJMKDZE0T1V4J8G

done

Obiekty zawarte w kompozycji są niszczone razem z kompozytem.

clear

Obiekty zawarte w kompozycji nie są niszczone razem z kompozytem.

clear

Jest to szczególny przypadek dziedziczenia.

done

Obiekt należący do kompozytu zawiera w sobie obiekty pochodzące z innych klas.

done

Jest to szczególny przypadek agregacji.

Zaznacz prawdziwe zdania dotyczące pól klasy o dostępie pakietowym.

clear

Dowolny obiekt może mieć do nich dostęp.

clear

Widoczne są w w klasach dziedziczących po ich klasie macierzystej.

clear

Można mieć do nich dostęp tylko w metodach należących do danej klasy.

done

Widoczne są dla innych klas znajdujących się w tym samym pakiecie.

Zaznacz wszystkie prawdziwe pytania dotyczące zasady pojedynczej odpowiedzialności (Single Responsibility Principle).

Y2MDP99JIUD2H8XXNKY7RUDDOQD1SMDYX4YLG39BKNHUHAWH

done

Pozwala na zachowanie spójności klasy.

clear

Zasada zabrania by klasa implementowała więcej niż jeden interfejs.

done

Zasada mówi że "Powinien być tylko jeden powód do modyfikacji klasy".

clear

Zasada zabrania by klasa miała więcej niż jedną publiczną metodę.

clear

Zasada mówi że klasa powinna posiadać tylko prywatne metody.

Zaznacz prawdziwe zdania dotyczące odśmieczacza (garbage collector) w języku Java.

1PSETSWFROIY0HXMILPJDSJB6DP5MLGXTZEEBUE615G4FMB

done

Zwalnia zasoby pamięciowe wewnątrz maszyny wirtualnej java.

clear

Programista ma pełną kontrolę nad jej działaniem.

clear

Zwalnia zasoby znajdujące się poza maszyną wirtualną java.

clear

Usuwa obiekt z pamięci natychmiast po wyzerowaniu licznika referencji wskazujących na obiekt.

Zmienna double[] x; reprezentuje typ: (java).

3X5IP1N7OIQY0YO8CRK4GXFBP8X8CF2YTBAD2TPNOW10WHZ8

clear

Prymitywny.

done

Obiektowy

Zaznacz prawdziwe zdania dotyczące metod statycznych w języku java.

clear

Mogą być wywołane dla obiektów statycznych.

done

Są wspólne dla wszystkich obiektów danej klasy.

clear

Ich argumentami mogą być tylko zmienne statyczne.

done

Mogą być wywołane nawet gdy nie istnieje żaden obiekt klasy do której należą.

Zaznacz wszystkie prawdziwe zdania dotyczące zasady segregacji interfejsów (Interface Segregation Principle).

Q5NVMBWIOCA9TRV1270MSI6TP7EUHEPP6JK4FUF70Y5QC3Y7

clear

Nazwy metod zadeklarowanych w interfejsie powinny być posortowane w kolejności alfabetycznej.

done

Preferowane są drobnoziarniste interfejsy.

done

Klasy nie powinny być zmuszane do implementowania metod, których nie używają.

clear

Klasa może implementować tylko jeden interfejs.

Zaznacz prawdziwe zdania dotyczące klas anonimowych w języku Java.

HNTTCCCVEBKDFHHRPC2F0C0EQZ0N62V034EBERTLJ1ACHS6N

clear

Mogą posiadać nieprzeciążone metody abstrakcyjne.

done

Nie posiadają własnej nazwy.

done

Nie można w nich zadeklarować konstruktora.

clear

W ogóle nie posiadają konstruktora.

Zaznacz wszystkie prawdziwe zdania dotyczące interfejsów w języku Java (java 8 i nowsze).

CY7GY38BTOY25RF2F4ND6LUPX6N4T3XXAUQB5Z5JW222KHTX

clear

Klasa może implementować tylko jeden interfejs.

done

Może zawierać metody statyczne.

done

Interfejs może zawierać domyślne implementacje metod.

clear

Interfejs może zawierać pola.

Zaznacz prawdziwe zdania dotyczące chronionych (protected) pól klasy.

done

Widoczne są w klasach dziedziczących po ich klasie macierzystej.

clear

Dowolny obiekt może mieć do nich dostęp.

done

Można mieć do nich dostęp tylko w metodach należących do danej klasy.

done

Widoczne są dla innych klas znajdujących się w tym samym pakiecie.

Zaznacz prawdziwe zdania dotyczące agregacji.

ANUZPPYE12KGBCO65OFUE59CVSBNQVG2C3PG2MNTKYLD0YM4

done

Obiekty zawarte w agregacie nie są niszczone razem z agregatem.

done

Obiekt należący do klasy agregatu zawiera w sobie obiekty pochodzące z innych klas.

clear

Jest to szczególny przypadek dziedziczenia.

clear

Obiekty zawarte w agregacie są niszczone razem z agregatem.

Zaznacz prawdziwe zdania dotyczące finalnych argumentów metod (java).

9QUF5VT70JQYGF5S9CP10WK6XXHHSHTK3EYUPUOAT4G6L29VE

done

Zapobiegają zmianie wartości argumentu.

done

Mogą być użyte w dowolnych metodach.

Zaznacz prawdziwe zdania dotyczące wzorca projektowego zwanego delegacją.

1H99UO7KFZ8MEXUUJXS1OIZH400O9FKP3IODHKDRBBO033XQ

done

Wykorzystuje kompozycję lub agregację.

done

Pozwala na zmianę zachowania się obiektu bez tworzenia nowej klasy potomnej.

clear

Wykorzystuje wyłącznie dziedziczenie.

done

Pozwala na przekazanie części odpowiedzialności klasy na obiekty innych klas.

Zmienna double x; reprezentuje typ (Java)

KHGR0HD0YL5SX4YNH8FMWHTDUVLF9KGFJ83KOYAC9FECK01Q

done

Prymitywny

clear

Obiektowy

Zaznacz prawdziwe zdania dotyczące metody clone z interfejsu Cloneable.

8VWP1IF07EJ6GO4P9UJOVSGD9J6NQ1ILJS5FY6TI4XV6H5DJ

done

Domyślna implementacja metody dostarczana jest przez klasę object.

clear

Domyślna implementacja dostarczana jest przez sam interfejs Cloneable.

done

Domyślna implementacja wykonuje płytką (shallow) kopię obiektu.

clear

Domyślna implementacja wykonuje głęboką (deep) kopię obiektu.

Zaznacz prawdziwe zdania dotyczące publicznych pól klasy.

C3M1EXQO7QOZBL2DB89JHSI5DQJDQYWX1C52YPN5D7UT0J7G

done

Wszystkie są widoczne w klasach dziedziczących po ich klasie macierzystej.

done

Dowolny obiekt może mieć do nich dostęp.

clear

Można mieć do nich dostęp tylko w metodach należących do danej klasy.

done

Widoczne są dla innych klas znajdujących się w tym samym pakiecie.