**JAVA**

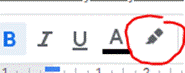
**II KOLOKWIUM  
   
 jak śpią programiści?**

**Na JAVIE**

**XD**

**Dobry programista wiesza się razem z programem**

**Wklejamy tu treść pytania, jeśli pytanie się powtarza usuwamy drugie takie same.**

****

**Kolorem Zielonym zaznaczamy tą odpowiedź której jesteśmy PEWNI**

**Kolorem Żółtym zaznaczamy odpowiedź która może być prawidłowa ale nie jesteśmy pewni**

**Kolorem czerwonym zaznaczamy odpowiedź która NA PEWNO jest nieprawidłowa.**

**Na ten plik wchodzimy za pomocą trybu incognito wtedy jesteśmy anonimowo podpisani**

**1. Wykonanie kodu:**

**StringBuilder sB = new StringBuilder();**

*for(int i=0;i<100;i++)  
 sB.append("Ala ma kota ("+(i+1)+").\n");  
try(BufferedWriter bw = new BufferedWriter(new FileWriter(filePath))) {  
 bw.write(sB.toString());  
}catch(IOException e) {  
 log.error(e,e);  
}*

**Spowoduje:  
Wybierz jedną lub więcej odpowiedzi**

a) wypisanie na konsoli napisu "Ala ma kota 100"

b) wypisanie na konsoli napisów od "Ala ma kota 1" do "Ala ma kota 100"

c) utworzenie pliku w ścieżce wskazującej przez zmienną filePath jeśli nie zostanie wygenerowany wyjątek

d) wypisanie na konsoli zawartości obiektu sB

**2. Dla obiektu klasy Thread lub pochodnej:   
Wybierz jedną lub więcej odpowiedzi**

a) wywołanie metody start powoduje wykonanie w wątku głównym aplikacji

b) wywołanie metody start powoduje wykonanie w oddzielnym wątku

c) wywołanie metody run powoduje wykonanie w oddzielnym wątku

d) wywołanie metody run powoduje wykonanie w wątku głównym aplikacji

**3. Dla klas:**

*public class Aaaa {  
 int x;  
 public Aaaa(int x) {this.x=x;}*

*public void show() {  
 System.out.println("SHOW A");  
 }  
 }*

**oraz**

*public class Bbbb extends Aaaa {  
 int y;*

*public Bbbb(int x, int y) {  
 super(x);  
 this.y=y;  
 }*

*public void show() {  
 System.out.println("SHOW B");  
 }  
 }*

**wykonanie kodu:**Aaaa a = new Aaaa(9);  
Bbbb b = new Bbbb(3,4);  
  
a=b;  
a.show();

**Wybierz jedną lub więcej odpowiedzi**

a) umożliwi dostęp do obiektu klasy "Bbbb" w zakresie widocznym przez klasę "Aaaa"

b) spowoduje wypisanie na konsolę "SHOW B"

c) spowoduje wygenerowanie wyjątku klasy Exception

d) spowoduje wypisanie na konsolę "SHOW A"

**4. Typ opakowujący to:  
Wybierz jedną lub więcej odpowiedzi**

a) float

b) String

c) Double

d) Boolean

**5. Wykonanie kodu:** -- powtórka 1 **StringBuilder sB = new StringBuilder();**

*for(int i=0;i<100;i++)  
 sB.append("Ala ma kota ("+(i+1)+").\n");  
 try(BufferedWriter bw = new BufferedWriter(new FileWriter(filePath))) {  
 bw.write(sB.toString());  
 }catch(IOException e) {  
 log.error(e,e);  
 }*

**spowoduje:  
Wybierz jedną lub więcej odpowiedzi**

a) wypisanie na konsoli napisów od "Ala ma kota 1" do "Ala ma kota 100"

b) wypisanie na konsoli zawartości obiektu sB

c) wypisanie na konsoli napisu "Ala ma kota 100"

d) utworzenie pliku w ścieżce wskazującej przez zmienną filePath jeśli nie zostanie wygenerowany wyjątek

**6. Iterowanie zbioru o deklaracji Set<String> set = new HashSet<String>();  
Wybierz jedną lub więcej odpowiedzi**

a) można wykonać poprzez for(Integer element:set){}

b) można wykonać z użyciem iteratora

c) zapewnia kolejność elementów zgodnie z ich wstawianiem

d) można wykonać poprzez Iterator it = set.iterator(); i pętle while

**57. Deklaracja klasy Parameter w postaci:**

*public final class Parameter{  
 [puste jest intended]  
}*

**Wybierz jedną lub więcej odpowiedzi**

a) zakazuje zmian wartości wszystkich zmiennych tej klasy

b) zakazuje definiowania konstruktora publicznego

c) zabrania dziedziczenia z tej klasy

d) nie umożliwia dokonania przesłonięcia dowolnej z metod tej klasy

**58. W klasie java.util.Collections  
Wybierz jedną lub więcej odpowiedzi**

a) metoda sort sortuje zgodnie z porządkiem naturalnym elementy listy

b) metoda disjoint zwraca true jeśli 2 kolekcje nie mają wspólnych elementów

c) metoda disjoint zwraca true jeśli 2 kolekcje mają wspólne elementy

d) metoda shuffle losowo układa elementy listy

**59. Instrukcja Integer iObject =5; iObject++;   
Wybierz jedną lub więcej odpowiedzi**

1. **int -> Integer**
2. **Integer -> int**
3. **int -> Integer**

a) wykona tylko 1 operację automatycznego wypakowania

b) wykona 2 operacje automatycznego opakowania

c) wykona 2 operacje automatyczngo wypakowania

d) wykona tylko 1 operację automatycznego opakowania

**60. (API strumieni) Operacja redukcji w strumieniach   
Wybierz jedną lub więcej odpowiedzi**

a) to operacja kończąca

b) umożliwia przeszukiwanie strumienia

c) to operacja zwracająca inny strumień

d) to zwrócenie wartości typu innego niż strumień

**61.Strumień FileInputStream:  
Wybierz jedną lub więcej odpowiedzi**

a) czyta pojedynczy bajt danych za pomocą metody read()

b) czyta bajty danych za pomocą metody readLine()

c) kończy czytanie poprzez zwrócenie null

d) kończy czytanie poprzez zwrócenie wartości -1

**63. Interfejs BaseStream w API strumieni  
Wybierz jedną lub więcej odpowiedzi**

a) nie rozszerza interfejsu AutoCloseable

b) definiuje podstawowe metody dla wszystkich strumieni

c) znajduje się w pakiecie java.util.stream

d) jego klasy implementujące mogą być zarządzane przez instrukcje try-catch dla zasobów

**64. Predicate<T>  
Wybierz jedną lub więcej odpowiedzi**

a) metoda interfejsu zwraca typ T

b) umożliwia sprawdzanie warunków logicznych z użyciem wyrażenia lambda

c) udostępnia metodę verify()

d) metoda interfejsu zwraca typ boolean

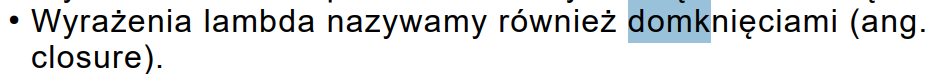
**65. Wyrażenie lambda:  
Wybierz jedną lub więcej odpowiedzi**

a) dzielone jest przez operator lambda (->) na 2 części

b) nie może przyjmować parametrów

c) to inaczej domknięcie

d) jest metodą bez nazwy



**66. Wykonanie kodu:**

*StringBuilder sB = new StringBuilder();  
String line;  
try (BufferedReader br= new BufferedReader(new FileReader(filePath))){  
 while((line=br.readLine())!=null) {  
 sB.append(line);  
 }  
 } catch(IOException e) {  
 log.error(e,e);  
 }*

**przy braku wygenerowania wyjątków:  
Wybierz jedną lub więcej odpowiedzi**

a) odczyt linia po linii zawartości pliku

b) spowoduje zwolnienie dostepu do pliku "filePath" po wyjściu z bloku try

c) spowoduje wczytanie zawartości pliku do obiektu klasy StringBuilder

d) zasilenie zmiennej "line" ostatnią linią zawartości pliku

**68. Metoda isAlive() w klasie wątku  
Wybierz jedną lub więcej odpowiedzi**

a) może być wykorzystywana do ustalenia konieczności oczekiwania przez wątek główny na   
 zakończenie działania wątków potomnych

b) sprawdzenia czy wątek nadal działa

c) pozwala na określenie maksymalnego czasu oczekiwania na wątek

d) powoduje włączenie oczekiwania na wątek

**69. Tworzenie własnej klasy wyjątków:  
Wybierz jedną lub więcej odpowiedzi**

a) umożliwia przesłanianie metod

b) wymaga rozszerzenia klasy Exception

c) zabrania wywoływania konstruktora klasy bazowej

d) umożliwia tworzenie własnych metod w klasie wyjątku

**70. Throwable to:  
Wybierz jedną lub więcej odpowiedzi**

a) klasa pochodna klasy Error

b) klasa bazowa dla klas wyjątków

c) klasa na szczycie hierarchii dziedziczenia wszystkich klas

d) klasa bazowa dla klasy Exception

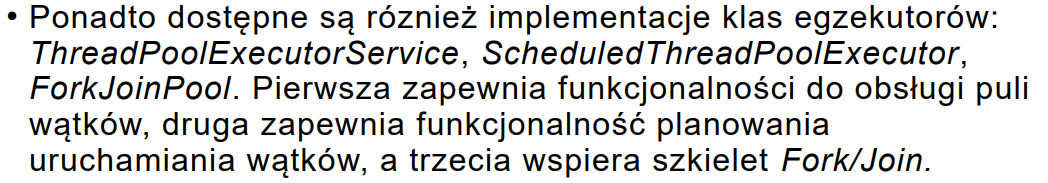
**71. Egzekutory //todo  
Wybierz jedną lub więcej odpowiedzi**

a) wymagają zamknięcia po użyciu za pomocą metody close()

b) w celu uruchomienia dają możliwość przekazania tylko obiektu klasy Thread lub pochodnej

c) dają możliwość pracy z ustaloną pulą wątków

d) zapewniają obsługę puli wątków



**72. Interfejs Runnable  
Wybierz jedną lub więcej odpowiedzi**

a) poprzez metodę run definiuje główne działanie wątku

b) definiuje metodę start() rozpoczynającą działanie wątku

c) definiuje wiele metod

d) wymusza implementację metody run()

**73. (API strumieni) Filtrowanie w strumieniach  
Wybierz jedną lub więcej odpowiedzi**

a) realizowane jest poprzez przekazanie wyrażenia lambda z 1 parametrem

b) zwraca obiekt strumienia ograniczonego poprzez predykat

c) przyjmuje zmienną interfejsu funcyjnego BinaryOperator

d) nie może być wykonywane wielokrotnie (na wyniku poprzedniego filtrowania)

**74. Klasa FileOutputStream  
Wybierz jedną lub więcej odpowiedzi**

a) udostępnia metodę write z 1 argumentem do zapisu danych

b) umożliwia zapis danych do pliku

c) nie generuje żadnych wyjątków

d) umożliwia bezpośredni zapis dowolnego typu danych

**75. Klasa ResourceBundle: //todo  
Wybierz jedną lub więcej odpowiedzi**

a) umożliwia pobieranie różnych wersji językowych komunikatów dla użytkownika

b) umożliwia pobieranie różnych wersji językowych etykiet interfejsu

c) nie umożliwia dostosowania komunikatów w zależności od systemu operacyjnego

d) umożliwia zapisywanie przez program komunikatów wyjątków w pliku properties

**76. Synchronizacja instrukcji:  
Wybierz jedną lub więcej odpowiedzi**

a) możliwa jest przy użyciu synchronizowanego bloku instrukcji

b) nie jest możliwa w JAVA

c) wymusza zmianę deklaracji metody na synchronizowaną

d) możliwa jest jedynie dla wybranych instrukcji

**77. W klasie stałych wyliczeniowych (enum):  
Wybierz jedną lub więcej odpowiedzi**

a) można dziedziczyć po dowolnej innej klasie

b) nie można definiować dodatkowych atrybutów powiązanych ze stałymi wyliczeniowymi

c) można definiować konstruktory prywatne

d) można definiować prywatne zmienne składowe klasy

**78. Klasa java.util.Formatter //todo  
Wybierz jedną lub więcej odpowiedzi**

a) zawsze zwraca ten sam wynik formatowania niezależnie od przekazanych ustawień   
 lokalizacyjnych (Locale)

b) obsługuje formatowanie dla typów danych: String, Date oraz typów numerycznych

c) zawsze zwraca ten sam wynik formatowania niezależnie od ustawień lokalizacyjnych (Locale)   
 systemu operacyjnego

d) pozwala na indeksowanie przekazywanych argumentów

**79. Interfejsy Callable i Future:  
Wybierz jedną lub więcej odpowiedzi**

a) są sparametryzowane typem danych, który będzie zwrócony

b) umożliwiają działanie na wartościach obliczanych przez wątek w przyszłości

c) umożliwiają podzielenie obliczeń na wiele wątków

d) używają egzekutorów to załadowania zadania

**80. (API strumieni) Odwzorowywanie w strumieniu  
Wybierz jedną lub więcej odpowiedzi**

a) nie może być wykonywane dla typów prostych

b) to zmiana typu elementu strumienia

c) wykorzystuje wyrażenie lambda przekazywane jako argument metody map

d) używa metody map() dla typów obiektowych

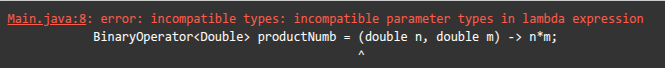
**81. Instrukcja: BinaryOperator<Double> productNumb = (n,m) -> n\*m;  
Wybierz jedną lub więcej odpowiedzi**

a) powoduje wykonanie metody o ciele: n\*m

b) definiuje parametry m i n jako double -- a nie Double?

c) productNumb.apply(3d, 4d); spowoduje obliczenie iloczynu 3\*4 -- 3d\*4d

d) wykorzystuje interfejs funkcyjny użytkownika



**82. Instrukcja try-catch dla zasobów:  
Wybierz jedną lub więcej odpowiedzi**

a) zwalnia zadeklarowany zasób po wyjściu z bloku try

b) umożliwia automatyczne zamykanie dowolnych zasobów

c) umożliwia automatyczne zamykanie obiektów klas implementujący AutoCloseable

d) umożliwia zdefiniowanie nazwanego bloku resources

**83. Strumień FileInputStream  
Wybierz jedną lub więcej odpowiedzi**

a) wymusza przeczytanie pliku od początku aż do żądanego bajtu

b) daje swobodny dostęp do pliku

c) nie generuje żadnych wyjątków

d) daje możliwość zadeklarowania obiektu w try-catch dla zasobów

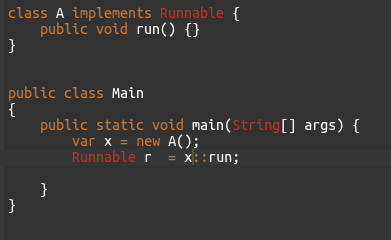
**84. Interfejs Runnable  
Wybierz jedną lub więcej odpowiedzi**

a) umożliwia uruchomienie wątku poprzez powiązanie obiektu klasy implementującej ten interfejs z   
 obiektem klasy Thread

b) daje możliwość powołania zmiennej referencyjnej -- Runnable::run ?

c) daje możliwość przesłaniania metod klasy Thread

d) to jedno z możliwych rozwiązań do zastosowania w programach wielowątkowych do obsługi   
 działania wątku



**85. Metoda join() w klasie wątku:  
Wybierz jedną lub więcej odpowiedzi**

a) z parametrem typu String włacza oczekiwanie na wątek o podanej nazwie

b) pozwala na określenie maksymalnego czasu oczekiwania na wątek

c) powoduje włączenie oczekiwania na wątek

d) umożliwia ustalenie stanu wątku

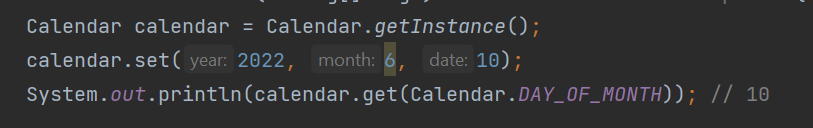
**86. W klasie java.util.Calendar:  
Wybierz jedną lub więcej odpowiedzi**

metoda set pozwala ustawić datę z podaniem wartości dnia, miesiąca i roku

dla metody set wartość miesiąca podajemy w zakresie 0-11

istnieje konstruktor publiczny udostępniony do tworzenia obiektu

obiekt kalendarza tworzymy z użyciem getInstance()

Dlaczego nienawidzę javy:  


**87. Wyjątek w ciele wyrażenia lambda:   
Wybierz jedną lub więcej odpowiedzi**

a) musi być obsłużony instrukcją try-catch

b) nie jest możliwy do wygenerowania

c) możliwy jest do wygenerowania ale wymaga stosownej deklaracji w interfejsie funkcyjnym

d) który jest generowany jest zawsze klasą pochodną klasy bazowej Throwable

D - zależy od humoru prowadzącego w trakcie egzaminu

**88. Function<T,R>  
Wybierz jedną lub więcej odpowiedzi**

a) to jeden z predefiniowanych interfejsów funkcyjnych wbudowany w JAVA

b) to klasa bazowa wszystkich funkcji w języku JAVA

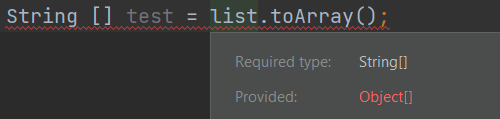
c) udostępnia metodę apply

d) to interfejs funkcyjny który przyjmuje 1 parametr typu R i zwraca wartość typu T

**89. Dla listy o deklaracji List<String> list = new ArrayList<String>();  
Wybierz jedną lub więcej odpowiedzi**

a) nie można dodawać elementów

b) możliwe jest iterowanie przy użyciu strumienia

c) nie można skonwertować elementów do tablicy typu String[]  chop ma raka

d) można wykonać iterowanie z użyciem pętli for(String item: list){}



**90. Klasa Semaphore w JAVA:  
Wybierz jedną lub więcej odpowiedzi**

a) pozwala na kontrolowanie dostępu do zasobu za pomocą licznika.

b) w celu wydania pozwolenia udostępnia metodę get()

c) w celu zwolnienia pozwolenia udostępnia metodę put()

d) wydając zezwolenie zmniejsza swój licznik semafora

**91. Strumień   
Wybierz jedną lub więcej odpowiedzi**

a) umożliwia zapis danych tekstowych w formacie UNICODE

b) umożliwia odczyt danych z pliku tekstowego

c) może być łączony z obiektem klasy BufferedReader

d) nie może być deklarowany w sekcji zasobów w instrukcji try-catch

**92. (API strumieni) Strumień można utworzyć  
Wybierz jedną lub więcej odpowiedzi**

a) jako pusty z użyciem metody Stream.empty()

b) poprzez metodę stream() bezpośrednio na obiekcie każdej tablicy

c) dla typów prostych

d) z obiektu dowolnej kolekcji

**93. Priorytet wątku:  
Wybierz jedną lub więcej odpowiedzi**

a) przyjmuje wartości od 1 do 10

a) przyjmuje wartości z zakresu [MIN\_PRIORITY,MAX\_PRIORITY]

a) całkowicie determinuje kolejność wykonywania wątków

a) może być ustawiony przy użyciu metody setPriority

**94. CountDownLatch  
Wybierz jedną lub więcej odpowiedzi**

a) posiada przeciążoną wersję metody await()

b) nie posiada konstruktora

c) to klasa zatrzasku zapewniająca aby wątek poczekał na zajście pewnego zdarzenia

d) bazuje jedynie na limicie i zliczaniu zajścia zdarzeń

**95. Instrukcja Integer iObject = Integer.valueOf(5);  
Wybierz jedną lub więcej odpowiedzi**

a) przedstawia operację manualnego wypakowania

b) przedstawia operację manualnego opakowania

c) porównuje zmienną Integer z wartością 5

d) przypisuje dla obiektu klasy Integer wartość 5

**96. Synchronizacja:  
Wybierz jedną lub więcej odpowiedzi**

a) może być zastosowana na poziomie definicji klasy

b) może być zastosowana na poziomie metody

c) to mechanizm zapewniający dostęp do zasobu w trybie izolowanym dla wątku

d) to zastosowanie koncepcji mechanizmu monitora i blokad

**97. W klasie Parameter2 zdefiniowanej jako:***Notka: kod niżej był nie pełny brać z ziarnem soli*

*public class Parameter2{  
 private String name = null;  
 public Parameter2(String name){  
 this.name = name;  
 }  
 public final String getName() {  
 return name;  
 }  
}*

**Wybierz jedną lub więcej odpowiedzi**

a) można dodać konstruktor prywatny

b) metoda "getName()" musi zostać przeciążona

c) wartość zmiennej "name" nie może zostać zmieniona

d) metoda "getName()" nie może zostać przesłonięta

**98. Dla deklaracji zmiennej klasowej zdefiniowanej jako**

*private Map<String, Integer> mapCarPower = null;*

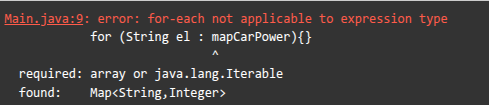
**Wybierz jedną lub więcej odpowiedzi**

a) można dokonać iteracji jako: "for (String el : mapCarPower){//instrukcje}"

b) można w konstruktorze utworzyć obiekt poprzez: "new Map<>();"

c) można utworzyć obiekt poprzez: "new HashMap();"

d) można utworzyć obiekt poprzez: "new LinkedHashMap<>();"



**99. Testy jednostkowe z użyciem pakietu jUnit 4.0  
Wybierz jedną lub więcej odpowiedzi**

a) zaleca się pisać z uwzględnieniem wzorca Arrange Act Assert

b) wymuszają użycie metody z dekoratorem @Before

c) wymagają dekoratora @Test przy definiowaniu metody testowej

d) zaleca się pisać z uwzględnieniem wzorca AAA

**100. Dla następującego kodu:**

*try {  
 jakasMetoda();  
 } catch (MyException e) {  
 //obługa  
 }*

**Wybierz jedną lub więcej odpowiedzi**

a) jeśli jakasMetoda() wygeneruje wyjątek MyException to zostanie on obsłużony w bloku catch

b) jeśli jakasMetoda() wygeneruje wyjątek klasy pochodnej MyException to zostanie on obsłużony   
 w bloku catch

c) jeśli jakasMetoda() nie wygeneruje wyjątku to kod z bloku catch nie zostanie wykonany

d) jeśli jakasMetoda() wygeneruje wyjątek Exception to zostanie on obsłużony w bloku catch

**101. Zjawisko przechwytywania zmiennych dla wyrażeń lambda:  
Wybierz jedną lub więcej odpowiedzi**

a) dotyczy zmiennych składowych klasy

b) powoduje konieczność traktowania zmiennej lokalnej jako praktycznie sfinalizowanej

c) uniemożliwia czytania zawartości zmiennych zdefiniowanych poza ciałem wyrażenia lambda

d) nie ma wpływu na zmienne składowe klasy

**102. BinaryOperator<T>  
Wybierz jedną lub więcej odpowiedzi**

a) Udostępnia metodę T apply(T a, T b);

b) Może być poprawnie użyty dla typu Boolean

c) Umożliwia przypisanie wyrażenia lambda bezparametrowego

d) Może być zdefiniowany dla typu double

**103. Referencja do metody  
Wybierz jedną lub więcej odpowiedzi**

a) Może być pobrana ale tylko do metody statycznej

b) Może być pobrana ale tylko do metody instancyjnej

c) Umożliwia wywoływanie metody z użyciem zmiennej interfejsu funkcyjnego

d) Instancyjnej nie może być pobrana jeśli nie jest zdefiniowany objekt

**104. Dla listy o deklaracji List list = new ArrayList()   
 oraz zbioru Set set = new HashSet()**

**Wybierz jedną lub więcej odpowiedzi**

a) Poprawna jest instrukcja: list = new HashSet(set);

b) Poprawna jest instrukcja: set = new ArrayList(set);

c) Poprawna jest instrukcja: set = new HashSet(list);

d) Poprawna jest instrukcja: list = new ArrayList(set);

**105. Klasa wyliczeniowa może:  
Wybierz jedną lub więcej odpowiedzi**

a) Dziedziczyć z innej klasy

b) Być klasą bazową dla innej klasy pochodnej

c) Definiować własne metody

d) Definiować własne konstruktory publiczne

**106. Zmienna referencyjna interfejsu  
Wybierz jedną lub więcej odpowiedzi:**

**customMagicInterface num = randomClass::randomFun**

a) Umożliwia wywołanie implementacji metody zadeklarowanej w interfejsie

b) Nie może być zdefiniowana

c) Umożliwia zapisanie zmiennej referencyjnej klasy implementującej dany interfejs

d) Umożliwia zapisanie dowolnej innej zmiennej referencyjnej

**107. Interfejs w języku JAVA  
Wybierz jedną lub więcej odpowiedzi:**

a) umożliwia definiowanie implementacji domyślnych metod publicznych

b) umożliwia deklarowanie stałych

c) umożliwia implementowanie metod prywatnych

d) umożliwia implementowanie metod statycznych

**108. Komponenty Swing są klasami pochodnymi klasy JComponent:   
Wybierz jedną lub więcej odpowiedzi**

a) tylko JFrame

b) wszystkie

c) tylko kontenery szczytowe

d) wszystkie poza kontenerami szczytowymi

**109. Referencja do konstruktora**

**Wybierz jedną lub więcej odpowiedzi**

a) Odwołuje się do konstruktora zgodnego z sygnaturą metody abstrakcyjnej interfejsu funkcyjnego

b) Pobierana jest poprzez wyrażenie postaci "NazwaKlasy::new"

c) Nie jest możliwa do pobrania w JAVA 11

d) Może być przypisana do dowolnej zmiennej referencyjnej typu interfejsu funkcyjnego

**111. Interfejs w JAVA 11**

**Wybierz jedną lub więcej odpowiedzi**

a) Może zawierać tylko jedną metodę abstrakcyjną

b) Może używać słowa kluczowego "default" przy deklaracji metody, ale pod warunkiem  
 zdefiniowania jej ciała

c) Nie może definiować metod "protected"

d) Obsługuje wielokrotne dziedziczenie z innych interfejsów

**112. W klasie StringUtils:**

**Wybierz jedną lub więcej odpowiedzi**

a) metoda difference nie uwzględnia wielkości znaków

b) metoda isBlank jest odporna na null podany w parametrze

c) metoda join pozwala na złączenie listy elementów typu String w jeden łańcuch znaków

d) metoda isNumeric zwróci true jeśli przekazany String jest możliwy do zmiany na typ liczbowy//-1

**113. Interfejs SortedMap**

**Wybierz jedną lub więcej odpowiedzi**

a) definiuje metodę tailMap(start) zwracającą posortowaną podmapę

b) definiuje metodę sort() sortującą klucze mapy

c) definiuje metodę subMap(start,end) zwracającą posortowaną podmapę

d) rozszerza interfejs Map

**114. Klasa TreeMap**

**Wybierz jedną lub więcej odpowiedzi**

a) jest klasa pochodna AbstractMap

b) dostarcza konstruktor przyjmujący kolekcję

c) używa struktury drzewiastej do przechowywania wpisów

d) umożliwia sterowanie porządkiem poprzez przyjmowanie komparatora w konstruktorze

**115.Klasa LinkedHashMap**

**Wybierz jedną lub więcej odpowiedzi**

a) udostępnia konstruktor przyjmujący komparator

b) daje możliwość uzyskania dostępu do wpisów zgodnie z kolejnością ostatniego dostępu do nich

c) daje możliwość iterowania po parach zgodnie z kolejnością ich wstawiania // iterowanie po Mapie? || ale ma metodę “foreach((key,value)->{})”

d) udostępnia metodę getAllLinks zwracającą listę pozycji kluczy

**116. Klasa RandomAccessFile**

**Wybierz jedną lub więcej odpowiedzi**

a) Jest klasą pochodną OutputStream

b) Umożliwia dostęp do danych plikowych tylko w dostępie sekwencyjnym

c) Daje możliwość dostępu do pliku do zapisu i odczytu

d) Jest klasą pochodną klasy InputStream

**117.Biblioteka commons email**

**Wybierz jedną lub więcej odpowiedzi**

a) pozwala na wysyłanie tego samego e-maila do więcej niż jednego adresata z użyciem klasy   
 SimpleEmail

b) pozwala wysyłać e-maila z osadzonymi obrazkami w wiadomości

c) pozwala na wysyłanie maili z obrazkami z użyciem klasy SimpleEmail

d) pozwala na wysyłanie maili z dowolnymi załącznikami z użyciem klasy SimpleEmail

**118. Interfejs NavigableMap**

**Wybierz jedną lub więcej odpowiedzi**

a) definiuje metodę pobierającą ostatnią parę mapy

b) rozszerza interfejs SortedMap

c) definiuje zachowanie wyszukiwania najbliższego dopasowania do danego klucza

d) definiuje metodę pobierającą pierwszą parę mapy

przez interfejs sortedMap, musi być posortowana, a to oznacza, że najmniejszy klucz jest pierwszy, a największy ostatni

**119. Klasa abstrakcyjna:**

**Wybierz jedną lub więcej odpowiedzi**

a) nie może posiadać metod prywatnych

b) tworzona jest poprzez dziedziczenie z klasy Abstract

c) może dziedziczyć z innej klasy

d) jest klasą pochodną klasy Object

**120. Klasa java.time.LocalDate**

**Wybierz jedną lub więcej odpowiedzi**

a) udostępnia metodę getMonthValue() zwracającą 11 jako wartość grudnia

b) udostępnia metodę getMonthValue() zwracającą 12 jako wartość grudnia

c) udostępnia metodę getMonthValue() zwracającą 0 jako wartość stycznia

d) udostępnia metodę getMonthValue() zwracającą 1 jako wartość stycznia