

TEMAT: Pakiety. Wyjątki. Asercje.

Zadanie 1.

Na podstawie fragmentu kodu:

```
public class Wyjatek2 {  
    public static void main( String... args ) {  
        int d, a;  
        System.out.println( "Start" );  
        try {  
            d = 0;  
            a = 42 / d;  
            System.out.println( "Nieosiągalne" );  
        }  
        catch (ArithmeticException e) {  
            System.out.println( "Wyjatek: " + e );  
        }  
    }  
}
```

wskaż, w której klasie przesłaniana jest metoda `toString()` wyświetlająca opis przechwyconego wyjątku `e`.

- a) `ArithmeticException`
- b) `Exception`
- c) `Throwable`

Zadanie 2.

Blok `finally` jest wykonywany bez względu na to, czy wewnątrz bloku kodu instrukcji `try` zostanie wyrzucony wyjątek.

- a) Prawda
- b) Fałsz

Zadanie 3.

Wskaż, które z poniższych stwierdzeń jest/są poprawne.

- a) Po bloku `try` powinien zawsze wystąpić blok `catch`
- b) Po bloku `try` powinny zawsze wystąpić co najmniej dwa bloki `catch`
- c) Po bloku `try` powinien zawsze wystąpić blok `finally`
- d) Po bloku `try` powinien zawsze wystąpić blok `catch` lub blok `finally`
- e) Po bloku `try` powinien zawsze wystąpić blok `catch` i blok `finally`

Zadanie 4.

Wskaż, który/które z poniższych wyjątków jest/są typu RuntimeException.

- a) ClassCastException
- b) FileNotFoundException
- c) NullPointerException
- d) SecurityException
- e) Wszystkie powyższe

Zadanie 5.

Na podstawie fragmentu kodu:

```
class Main {
    public static void main(String args[]) {
        try {
            throw 10;
        }
        catch(int e) {
            System.out.println("Got the Exception " + e);
        }
    }
}
```

wskaż, jaki będzie wynik wykonania.

- a) Zostanie wyświetlone Got the Exception 10
- b) Zostanie wyświetlone Got the Exception 0
- c) Wystąpi błąd w trakcie kompilacji. Wyjaśnij przyczynę!

Zadanie 6.

Na podstawie fragmentu kodu:

```
class Test extends Exception {}

public class Main {
    public static void main(String args[]) {
        try {
            throw new Test();
        }
        catch(Test t) {
            System.out.println("Got the Test Exception");
        }
        finally {
            System.out.println("Inside finally block");
        }
    }
}
```

wskaż, jaki będzie całościowy wynik wykonania.

- a) Zostanie wyświetlone Got the Test Exception
 Inside finally block
- b) Zostanie wyświetlone wyłącznie Got the Test Exception
- c) Zostanie wyświetlone wyłącznie Inside finally block
- d) Wystąpi błąd w trakcie kompilacji

Zadanie 7.

Na podstawie fragmentu kodu:

```
public class Test {  
    public static void main(String[] args) {  
        try {  
            int a[] = {1, 2, 3, 4};  
            for (int i = 1; i <= 4; i++)  
            {  
                System.out.println("a[" + i + "]=" + a[i] + "\n");  
            }  
        }  
        catch (Exception e) {  
            System.out.println ("error = " + e);  
        }  
        catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println ("ArrayIndexOutOfBoundsException");  
        }  
    }  
}
```

wskaż, jaki będzie wynik wykonania. W przypadku wskazania wystąpienia błędu lub wyrzucenia wyjątku należy zamieścić poprawiony kod (całościowo).

- a) Wystąpi błąd w trakcie kompilacji. Wyjaśnij przyczynę!
- b) Wystąpi błąd w trakcie uruchomienia. Wyjaśnij przyczynę!
- c) Zostanie wyrzucony wyjątek `ArrayIndexOutOfBoundsException`. Wyjaśnij przyczynę.
- d) Zostanie wyświetlony kod błędu
- e) Zostaną wyświetlone elementy tablicy

Zadanie 8.

Na podstawie fragmentu kodu:

```
public class Test {  
    String str = "a";  
  
    void A() {  
        try {  
            str += "b";  
            B();  
        }  
        catch (Exception e) {  
            str += "c";  
        }  
    }  
}
```

```
    }  
}  
  
void B() throws Exception {  
    try {  
        str += "d";  
        C();  
    }  
    catch(Exception e) {  
        throw new Exception();  
    }  
    finally {  
        str += "e";  
    }  
  
    str += "f";  
}  
  
void C() throws Exception {  
    throw new Exception();  
}  
  
void display() {  
    System.out.println(str);  
}  
  
public static void main(String[] args) {  
    Test object = new Test();  
    object.A();  
    object.display();  
}  
}
```

wskaż, jaki będzie wynik wykonania.

- a) abdef
- b) abdec
- c) abdefc

Zadanie 9.

Wskaż, które z poniższych zdań są prawdziwe.

- 1) Każda klasa należy do jakiegoś pakietu
- 2) Wszystkie klasy z tego samego pliku źródłowego należą do jednego pakietu
- 3) Jeśli dla klasy nie został wyspecyfikowany żaden pakiet wówczas klasa zostanie przydzielona do specjalnego pakietu nieposiadającego nazwy
- 4) Jeśli dla klasy nie został wyspecyfikowany żaden pakiet wówczas tworzony jest nowy pakiet o nazwie identycznej z nazwą klasy i klasa jest w nim umieszczana

- a) Tylko 1, 2 i 3
- b) Tylko 1, 2 i 4
- c) Tylko 4
- d) Tylko 1 i 3

Zadanie 10.

Na podstawie fragmentu kodu:

```
public class Main {  
    public static void main(String[] args)  
    {  
        int x = 0;  
        assert (x > 0) ? "assertion failed" : "assertion passed" ;  
        System.out.println("finished");  
    }  
}
```

wskaż, jaki będzie wynik wykonania (uruchomienie JVM z przełącznikiem -ea).

- a) Zostanie wyświetlone `finished`
- b) Wystąpi błąd w trakcie kompilacji. Wyjaśnij przyczynę!
- c) Wystąpi `AssertionError` a następnie zostanie wyświetlone `finished`
- d) Wystąpi `AssertionError` z komunikatem `assertion failed`

Zadanie 11.

Na podstawie fragmentu kodu:

```
public class Test {  
    public static int y;  
    public static void foo(int x) {  
        System.out.print("foo ");  
        y = x;  
    }  
    public static int bar(int z) {  
        System.out.print("bar ");  
        return y = z;  
    }  
    public static void main(String [] args ) {  
        int t = 0;  
        assert t > 0 : bar(7);  
        assert t > 1 : foo(8); /* Line 18 */  
        System.out.println("done ");  
    }  
}
```

wskaż, jaki będzie wynik wykonania (uruchomienie JVM z przełącznikiem -ea).

- a) Zostanie wyświetlone `bar`
- b) Zostanie wyświetlone `bar done`
- c) Zostanie wyświetlone `foo done`
- d) Wystąpi błąd w trakcie kompilacji. Wyjaśnij przyczynę!

Zadanie 12.

Którego/których z poniższych modyfikatorów nie należy stosować dla elementów klasy, aby zapewnić dostęp do nich innym klasom z tego samego pakietu.

- a) public
- b) protected
- c) Brak modyfikatora dostępu
- d) private

Zadanie 13.

Pakiet `java.lang` jest niejawnie (domyślnie) importowany w każdym pliku źródłowym.

- a) Prawda
- b) Fałsz

Zadanie 14.

Archiwa JAR wykorzystują kompresję przy użyciu formatu:

- a) TAR
- b) ZIP
- c) ARJ
- d) CAB

Zadanie 15.

Wyjątki niekontrolowane (unchecked) są:

- a) Sprawdzane przez kompilator
- b) Sprawdzane przez JVM
- c) Obie powyższe odpowiedzi
- d) Żadna z powyższych odpowiedzi

Uwaga: Każdy punkt proszę traktować rozłącznie, jako wyczerpującą odpowiedź

Zadanie 16.

Konstrukcja `try-with-resources` została wprowadzona w:

- a) Java2 SE 1.2
- b) Java SE 5
- c) Java SE 7
- d) Java SE 8
- e) Java SE 9

Zadanie 17.

Na podstawie fragmentu kodu:

```
try { int x = Integer.parseInt("two"); }
```

wskaż wyjątek, który mógłby zostać użyty w sekcji catch (zaznacz wszystkie możliwe odpowiedzi):

- a) ClassCastException
- b) IllegalStateException
- c) NumberFormatException
- d) IllegalArgumentException
- e) ExceptionInInitializerError
- f) ArrayIndexOutOfBoundsException

Zadanie 18.

Na podstawie fragmentu kodu:

```
1. // insert code here
2. class StatTest {
3.     public static void main(String[] args) {
4.         System.out.println(MAX_VALUE);
5.     }
6. }
```

wskaż instrukcję, która wstawiona w linii 1 wciąż umożliwi poprawną kompilację kodu (zaznacz wszystkie możliwe odpowiedzi):

- a) import static java.lang;
- b) import static java.lang.Integer;
- c) import static java.lang.Integer.*;
- d) import static java.lang.Integer.*_VALUE;
- e) import static java.lang.Integer.MAX_VALUE;
- f) Żadna z powyższych instrukcji nie posiada poprawnej składni.

Zadanie 19.

W poniższym szkielecie kodu:

```
public class Test {
    public static void main(String[] args)
    {
        try {
            Scanner scn = new Scanner(System.in);
            ...
        }
        //TODO: fill missing statements
    }
}
```

proszę uzupełnić:


- 1) włączyć (import) do kodu statyczne pole System.out;
- 2) uzupełnić instrukcję try o jednakową procedurę obsługi:

- `out.println(ex.getMessage());`

dla wyjątków `NoSuchElementException` oraz `IllegalStateException` (proszę zastosować `multi-catch`) oraz:

- `System.exit(0);`

dla pozostałych niepoprawnych sytuacji (w tym wyjątków `Exception` i błędów `Error`) – proszę zastosować w tym przypadku jeden typ.



<i>Poprawnych odpowiedzi</i>	<i>Ocena</i>
<0>	n/k
<1, 10>	2.0
<11, 12>	3.0
<13, 14>	3.5
<15, 16>	4.0
<17, 18>	4.5
<19>	5.0