

Oto zapis do pliku (dokumentu), zawierający wszystkie pytania z kolokwium Java wraz z poprawnymi odpowiedziami:

1. „Dana jest następująca klasa (obrazek). Którą wersję należy wstawić, aby program wypisał „Hello”?”

```
public class Hello {  
    {  
        // WSTAWIĆ TUTAJ  
        System.out.println("Hello");  
    }  
}
```

Możliwe odpowiedzi:

1. `public void main (String[] args)`
2. `static public void main (String[] array)`
3. `static public main (String args[])`
4. `public static void main (String args)`
5. `public void main(String args[])`

Poprawna odpowiedź: **static public void main (String[] array)** (czyli odpowiedź nr 2).

2. „Dana jest klasa przedstawiona na obrazku. Program uruchomiono w następujący sposób:

java Program jeden dwa trzy cztery piec szesc

Co zostanie wypisane na ekranie?"

### Przykładowy kod fragmentu main:

```
class Program {  
  
    public static void main(String[] args) {  
        System.out.println(args[1] + "/" + args[2] + "/" + args[3]);  
    }  
}
```

Możliwe odpowiedzi:

1. java/jeden/dwa
2. jeden/dwa/trzy
3. dwa/trzy/cztery

4. java/Program/jeden

Poprawna odpowiedź: **dwa/trzy/cztery** (odpowiedź nr 3).

---

3. „Dany jest program przedstawiony na obrazku. Jaki będzie wynik działania powyższego programu?”

```
public class Operatory {  
    public static void main(String[] argumenty) {  
        int x = 10;  
        long y = 20;  
        short z = 30;  
        System.out.println(++z + y++ * z);  
    }  
}
```

Możliwe odpowiedzi: 611, 651, 930, 960

**Obliczenie:**

- ++z ustawia z=31, w wyrażeniu to 31
- y++ używa 20, potem y=21
- nowa wartość z to 31

Więc mamy  $31 + 20 * 31 = 31 + 620 = 651$ .

Poprawna odpowiedź: **651**.

---

4. „Dany jest kod źródłowy (obrazek). Co się stanie, jeśli spróbujemy skompilować i uruchomić ten program?”

```
class Test {  
    static int x;  
    int k;  
  
    public Test(int n, int m) {  
        x = n;  
        k = m;  
    }  
}
```

```

public static void main(String[] args) {
    Test t1 = new Test(10, 20);
    Test t2 = new Test(30, 40);

    System.out.print(t1.x + " "); // tu x jest 30, bo x jest static
    System.out.print(t1.k + " "); // k=20 dla t1
    System.out.print(t2.x + " "); // x=30
    System.out.println(t2.k);    // k=40 dla t2
}
}

```

Możliwe odpowiedzi:

- Nie będzie błędów, program wypisze 30 20 30 40
- Nie będzie błędów, program wypisze 30 40 30 40
- Wystąpi błąd kompilacji bo nie ma zdefiniowanego konstruktora bezparametrowego
- Wystąpi błąd podczas uruchomienia bo nie ma zdefiniowanego konstruktora bezparametrowego
- Nie będzie błędów, program wypisze 10 20 30 40

Poprawna odpowiedź: **Nie będzie błędów, program wypisze 30 20 30 40.**

---

5. „W jakich nawiasach podajemy typ podczas definiowania klasy uogólnionej (generycznej)?”

Możliwe odpowiedzi: ( ), [ ], { }, < >

Poprawna odpowiedź: < >

---

6. „Czy klasa może mieć więcej niż jeden konstruktor bezparametrowy?”

Możliwe odpowiedzi: Tak, Nie wiem, Nie

Poprawna odpowiedź: **Nie**

---

7. „Jak nazywa się własność każdej tablicy języka Java, która umożliwia odczytanie jej aktualnego rozmiaru?”

Możliwe odpowiedzi: capacity, dimension, length, size

Poprawna odpowiedź: **length**

---

8. „Dany jest kod źródłowy (obrazek). Fragment ten jest przykładem:”

```
class A {  
    int n;  
  
    public A(int n) { this.n = n; }  
  
    public int suma(int n1, int n2) {  
        //...  
        return 0;  
    }  
}
```

```
class B extends A {  
    public B(int n) { super(n); }  
  
    public int suma(int k1, int k2) {  
        //...  
        return 0;  
    }  
}
```

Możliwe odpowiedzi: przeciążania metod (overloading), przestaniania metod (overriding).

Ponieważ sygnatury metoda `suma(int,int)` są takie same w klasie bazowej i pochodnej, jest to **nadpisywanie metod** (overriding).

Poprawna odpowiedź: **przestaniania metod (overriding)**.

---

9. „Dany jest kod źródłowy (obrazek). Jaki jest rezultat uruchomienia tego programu?”

```
class A {  
    public A(String s) { System.out.print("1"); }  
}
```

```
public class B extends A {  
    public B(String s) {  
        super(s);  
    }  
}
```

```
        System.out.print("3");
    }
    public static void main(String[] args) {
        new B("2");
        System.out.println(" ");
    }
}
```

Możliwe odpowiedzi:

- Program wypisze 13
- Program wypisze 312
- Program wypisze 31
- Program wypisze 132
- Kod się nie skompiluje

Tworząc obiekt `new B("2")`: najpierw wywołanie `super("2")` wypisze "1", potem konstruktor B wypisze "3". Finalny ciąg to "13" i potem spacja i nowy wiersz.

Poprawna odpowiedź: **Program wypisze 13.**

---

10. „Dane są klasy (obrazek). Co się stanie po uruchomieniu programu?”

```
class Auto {
    String marka;

    public Auto(String marka) { this.marka = marka; }

    public void jedzie() {
        System.out.println("Auto jedzie");
    }
}
```

```
class Toyota extends Auto {
    public Toyota() { super("Toyota"); }

    public void jedzie() {
        System.out.println("Jedzie Toyota");
    }
}
```

```

public void jedzie(String model){
    System.out.println("Jedzie Toyota " + model);
}
}

```

```

public class Program {
    public static void main(String[] args) {
        Auto a1 = new Toyota();
        Toyota a2 = new Toyota();
        a1.jedzie();
        a2.jedzie("Aigo");
    }
}

```

Możliwe odpowiedzi:

- program wypisze "Auto jedzie" i "Jedzie Toyota Aigo"
- program nie skompiluje się
- program wypisze "Jedzie Toyota" i "Jedzie Toyota Aigo"

Ponieważ a1 to new Toyota(), wywołuje się metoda nadpisana ("Jedzie Toyota"), potem a2.jedzie("Aigo") wypisuje "Jedzie Toyota Aigo".

Poprawna odpowiedź: **program wypisze "Jedzie Toyota" i "Jedzie Toyota Aigo".**

11. „Modyfikatory dostępu to: 1 default, 2 public, 3 private, 4 protected. Uszereguj modyfikatory dostępu od najobszerniejszego do najmniejszego.”

Możliwe odpowiedzi:

- 2 4 1 3
- 1 2 3 4
- 2 4 3 1
- 1 2 4 3

Poprawna odpowiedź: **2 4 1 3** (public > protected > default > private).

12. „Dany jest fragment (obrazek). Jaki będzie wynik działania programu?”

```

abstract class Figura {

```

```
    abstract void rysuj();  
    abstract double obliczPole();  
}
```

```
class Kwadrat extends Figura {  
    double a;  
    public Kwadrat(double a) {  
        this.a = a;  
    }  
    double obliczPole() {  
        return a * a;  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
        Kwadrat k = new Kwadrat(2);  
        double P = k.obliczPole();  
        System.out.println("Pole kwadratu = " + P);  
    }  
}
```

W tym kodzie Klasa Kwadrat nie implementuje void rysuj(), a jest klasą nieabstrakcyjną dziedziczącą po Figura. Zatem kompilacja zakończy się błędem: "Kwadrat is not abstract and does not override abstract method rysuj()..."

Możliwe odpowiedzi:

- Program uruchomi się, ale rzuci wyjątek
- Program uruchomi się i wypisze 2
- Program nie skompiluje się
- Program uruchomi się i wypisze 4

Poprawna odpowiedź: **Program nie skompiluje się** (brak implementacji metody rysuj).

Możliwe odpowiedzi:

- definiując pola jako składowe publiczne
- definiując publiczne metody, które mają dostęp do pól
- należy zastosować wszystkie rzeczy wymienione we wszystkich odpowiedziach
- definiując pola z modyfikatorem dostępu private

Poprawna odpowiedź: **definiując pola z modyfikatorem dostępu private.**

---

14. „Typ podstawowy języka Java do reprezentacji liczb całkowitych o największym zakresie to:”

Poprawna odpowiedź: **long**

---

15. „Dany jest fragment (obrazek). Które deklaracje są poprawne?”

Mamy:

1. `class A {}`
2. `class B implements A {}` // niepoprawne, bo implementuje klasę, a nie interfejs
3. `interface L1 {}`
4. `interface L2 {}`
5. `interface l3 implements L1 {}` // niepoprawne, interface rozszerza, nie implementuje
6. `interface l4 implements A {}` // niepoprawne
7. `interface l5 extends A {}` // niepoprawne, A to klasa
8. `interface l6 extends L2 {}` // poprawne
9. `class C extends A, B {}` // niepoprawne, Java nie ma wielokrotnego dziedziczenia klas
10. `class D implements l2, l1 {}` // poprawne (l2 i l1 to interfejsy)
11. `interface l7 extends l2, l1 {}` // poprawne (interfejs może rozszerzać wiele interfejsów)
12. `class E extends A implements l2 {}` // poprawne
13. `class F extends A implements l2, l1 {}` // poprawne

Zatem poprawne numery to: **1, 3, 4, 8, 10, 11, 12, 13.**

---

16. „Elementami kodu źródłowego mogą być: 1 komentarze; 2 deklaracje import; 3 deklaracje package; 4 metody; 5 deklaracje klasy; 6 zmienne. Która z poniższych kolejności występowania jest dopuszczalna?”

Możliwe odpowiedzi:



- 3 1 2 5 4 6
- 1 3 2 5 6 4
- 3 2 1 5 6 4
- 3 2 1 4 5 6

Poprawna odpowiedź: **3 2 1 5 6 4**, czyli package -> import -> komentarze -> deklaracja klasy -> zmienne -> metody.

---

17. „Jaka jest minimalna wartość typu danych byte?”

Możliwe odpowiedzi: -128, -2<sup>8</sup>, -127, -32768, -256

Poprawna odpowiedź: **-128**

---

18. „Metody przeciążone (overloaded) mogą mieć różne:”

Możliwe odpowiedzi: typ zwracanej wartości, modyfikatory dostępu, listę argumentów.

Najważniejsze jest **różna lista argumentów**.

---

19. „Dany jest fragment (obrazek). Co się stanie?”

```
class Wyjatek1 extends Exception {}
```

```
class Wyjatek2 extends Exception {}
```

```
class Program {
```

```
    void zlaMetoda() throws Wyjatek1, Wyjatek2 {
```

```
        throw new Wyjatek2();
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Program a = new Program();
```

```
        try {
```

```
            a.zlaMetoda();
```

```
            System.out.println("a");
```

```
        } catch (Wyjatek1 e) {
```

```
            System.out.println("b");
```

```
        } catch (Wyjatek2 e) {
```

```
        System.out.println("c");
    } finally {
        System.out.println("d");
    }
}
}
```

W `zlaMetoda()` rzucamy `Wyjatek2`, który przechwyci `catch (Wyjatek2 e)` => wypisze "c". Następnie `finally` zawsze się wykona => wypisze "d".

Możliwe odpowiedzi:

- Program nie skompiluje się
- Program uruchomi się i wypisze ad
- Program uruchomi się i wypisze cd
- Program uruchomi się i wypisze d
- Program uruchomi się i wypisze acd

Poprawna odpowiedź: **Program uruchomi się i wypisze "cd".**

---

## Podsumowanie

Powyżej znajdują się wszystkie pytania z pliku wraz z poprawnymi odpowiedziami i krótkimi wyjaśnieniami.