

Laboratorium 8



CEL: IO, NIO.2.



Zadanie#1. Przeanalizuj poniższy fragment kodu.

```
InputStream is = new BufferedInputStream(new FileInputStream("zoo.txt"));
InputStream wrapper = new _____(is);
```

Wstawienie której z poniższych klas zapewni poprawną jego kompilację (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	BufferedInputStream
b)	FileInputStream
c)	BufferedWriter
d)	ObjectInputStream
e)	ObjectOutputStream
f)	BufferedReader



Zadanie#2. Które z poniższych stwierdzeń jest/są prawdziwe?

a)	Gdy wywoływany jest <code>System.console()</code> za każdym razem tworzony jest nowy obiekt klasy <code>Console</code>
b)	<code>Console</code> może być tylko wykorzystany do odczytu wejścia, lecz nie do zapisu wyjścia
c)	<code>Console</code> jest uzyskiwany przy użyciu wzorca projektowego Singleton
d)	Uzyskany obiekt <code>Console</code> może mieć wartość <code>null</code>
e)	Uzyskany obiekt <code>Console</code> nie może mieć wartości <code>null</code>



Zadanie#3. Przeanalizuj poniższy fragment kodu.

```
Console c = System.console();
String s = _____;
```

Które z poniższych instrukcji wstawione w brakujący fragment kodu zapewni jego poprawną kompilację?

a)	<code>c.input()</code>
b)	<code>c.read()</code>
c)	<code>c.readLine()</code>
d)	<code>c.readPassword()</code>
e)	<code>c.readString()</code>
f)	Żadne z powyższych





Zadanie#4. Przeanalizuj poniższy fragment kodu.

```
String line;
Console c = System.console();
Writer w = c.writer();
if ((line = c.readLine()) != null)
    w.append(line);
w.flush();
```

Jaki będzie wynik wykonania (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	Kod wykona się bezbłędnie lecz nic nie wyświetli
b)	Kod wyświetli tekst wpisany przez użytkownika
c)	Może zostać wyrzucony wyjątek <code>ArrayIndexOutOfBoundsException</code>
d)	Może zostać wyrzucony wyjątek <code>NullPointerException</code>
e)	Może zostać wyrzucony wyjątek <code>IOException</code>
f)	Błąd kompilacji



Zadanie#5. Jakie są wymagania wobec klasy, która ma mieć możliwość serializacji przy użyciu `ObjectOutputStream` (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	Klasa musi implementować interfejs <code>Serializable</code>
b)	Klasa musi dziedziczyć z klasy <code>Serializable</code>
c)	Klasa musi deklarować zmienną statyczną <code>serialVersionUID</code>
d)	Klasy wszystkich pól instancyjnych muszą być <code>Serializable</code>
e)	Klasy wszystkich pól instancyjnych muszą być oznaczone jako <code>transient</code>
f)	Każda klasa może być serializowana przy użyciu <code>ObjectOutputStream</code>



Zadanie#6. Poniższa metoda ma być przeznaczona do usuwania rekursywnie drzewa wskazanego katalogu.

```
1: public static void deleteTree(File file) {
2:     if(!file.isFile())
3:         for(File entry: file.listFiles())
4:             deleteTree(entry);
5:     else file.delete();
6: }
```

Które z poniższych stwierdzeń odzwierciedla właściwie zachowanie metody (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	Może usunąć katalog, który zawiera tylko pliki
b)	Może usunąć katalog o dowolnej głębokości
c)	Może usunąć pojedynczy plik
d)	Błąd kompilacji spowodowany w linii 2
e)	Błąd kompilacji spowodowany w linii 3
f)	Kompiluje się, jednak może wyrzucić wyjątek w trakcie działania





Zadanie#7. Które z poniższych są metodami klasy `java.io.File` (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	<code>mv()</code>
b)	<code>createDirectory()</code>
c)	<code>mkdirs()</code>
d)	<code>move()</code>
e)	<code>renameTo()</code>
f)	<code>copy()</code>
g)	<code>mkdir()</code>



Zadanie#8. Przy założeniu, że istnieje plik `C:\book\java`, które z poniższych instrukcji tworzą obiekt reprezentujący ten plik (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	<code>new File("c:\book\java");</code>
b)	<code>new File("c:\\book\\java");</code>
c)	<code>new File("c:/book/java");</code>
d)	<code>new File("c://book//java");</code>
e)	Żadne z powyższych



Zadanie#9. Które z poniższych są strumieniami wbudowanymi w Java (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	<code>System.err</code>
b)	<code>System.error</code>
c)	<code>System.in</code>
d)	<code>System.input</code>
e)	<code>System.out</code>
f)	<code>System.output</code>



Zadanie#10. Które z poniższych nie są klasami pakietu `java.io`?

a)	<code>BufferedReader</code>
b)	<code>BufferedWriter</code>
c)	<code>FileReader</code>
d)	<code>FileWriter</code>
e)	<code>PrintReader</code>
f)	<code>PrintWriter</code>



Zadanie#11. Przy założeniu, że `zoo-data.txt` jest plikiem wieloliniowym, które ze stwierdzeń jest prawdziwe dla następującej metody (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

```
private void echo() throws IOException {
    try (FileReader fileReader = new FileReader("zoo-data.txt");
        BufferedReader bufferedReader = new BufferedReader(fileReader)) {
        System.out.println(bufferedReader.readLine());
    }
}
```

a)	Wyświetla na konsoli pierwszą linię pliku
b)	Wyświetla całą zawartość pliku
c)	Kod się nie kompiluje ponieważ <code>BufferedReader</code> nie jest zamykany
d)	Kod się kompiluje, ale <code>BufferedReader</code> nie jest zamykany
e)	Kod się nie kompiluje z innej przyczyny



Zadanie#12. Przeanalizuj poniższy fragment kodu.

```
Console console = System.console();
String color = console.readLine("What is your favorite color? ");
console. _____("Your favorite color is "+color);
```

Które z poniższych instrukcji wstawione w brakujący fragment kodu zapewni jego poprawną kompilację (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	<code>print</code>
b)	<code>printf</code>
c)	<code>println</code>
d)	<code>format</code>
e)	<code>writer().println</code>
f)	<code>out</code>



Zadanie#13. Jakie są powody stosowania strumieni znakowych, takich jak `Reader/Writer`, zamiast strumieni bajtowych, takich jak `InputStream/OutputStream` (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	Bardziej wygodna składnia przy pracy z danymi typu <code>String</code>
b)	Lepsza wydajność
c)	Automatyczne kodowanie znaków
d)	Wbudowana serializacja/deserializacja
e)	Strumienie znakowe są strumieniami wysokiego poziomu
f)	Wsparcie dla wielowątkowości



Zadanie#14. Przy założeniu, że poniższa klasa posiada poprawne metody getter/seter dla wszystkich swoich prywatnych pól, które z jej pól zawsze będą posiadały wartość `null`, jeśli instancja tej klasy zostanie najpierw zserializowana a następnie zdeserializowana (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

```
public class Zebra implements Serializable {
    private static final long serialVersionUID = 1L;
    private transient String name = "George";
    private static String birthPlace = "Africa";
    private transient Integer age;
    private java.util.List<Zebra> friends = new java.util.ArrayList<>();
    private Object tail = null;

    { age = 10; }

    public Zebra() {
        this.name = "Sophia";
    }
}
```

a)	name
b)	tail
c)	age
d)	friends
e)	birthPlace
f)	Błąd kompilacji
g)	Kompiluje się jednak wyrzuca wyjątek w trakcie działania



Zadanie#15. Jaka jest wartość pola name po tym, jak instancja Eagle zostanie najpierw zserializowana a następnie zdeserializowana?

```
public class Bird implements Serializable {
    protected transient String name = "Bridget";
    public void setName(String name) { this.name = name; }
    public String getName() { return name; }

    public Bird() {
        this.name = "Matt";
    }
}

public class Eagle extends Bird implements Serializable {
    { this.name = "Janette"; }

    public Eagle() {
        this.name = "Daniel";
    }
}
```

a)	Bridget
b)	Matt
c)	Janette
d)	Daniel
e)	null
f)	Błąd kompilacji
g)	Kod kompiluje się, jednak może wyrzucić wyjątek w trakcie działania
h)	Wartość może nie być znana aż do momentu uruchomienia kodu



Zadanie#16. Przy założeniu, że kolejne bajty `InputStream is` tworzą łańcuch XYZABC, jaki będzie wynik wywołania poniższej metody dla wartości `count` równej 3?

```
public static String pullBytes(InputStream is, int count) throws IOException {
    is.mark(count);
    final StringBuilder sb = new StringBuilder();
    for(int i=0; i<count; i++)
        sb.append((char)is.read());
    is.reset();
    is.skip(1);
    sb.append((char)is.read());
    return sb.toString();
}
```

a)	Zwróci String o wartości XYZ
b)	Zwróci String o wartości XYZA
c)	Zwróci String o wartości XYZX
d)	Zwróci String o wartości XYZB
e)	Zwróci String o wartości XYZY
f)	Błąd kompilacji
g)	Kod kompiluje się, jednak może wyrzucić wyjątek w trakcie działania
h)	Wynik jest niemożliwy do przewidzenia na podstawie podanych informacji



Zadanie#17. Przeanalizuj poniższy fragment kodu.

```
Path path1 = Paths.get("/pets/../cat.txt");
Path path2 = Paths.get("./dog.txt");
System.out.println(path1.resolve(path2));
System.out.println(path2.resolve(path1));
```

Jaki będzie wynik wykonania?

a)	/pets/../cat.txt/./dog.txt /pets/../cat.txt
b)	/pets/../cat.txt/./dog.txt ./dog.txt/pets/../cat.txt
c)	/cats.txt /dog.txt
d)	/cats.txt/dog.txt /cat.txt
e)	Kod kompiluje się, jednak wyrzuca wyjątek w trakcie działania



Zadanie#18. Przeanalizuj poniższy fragment kodu.

```
Path path = Path.get("/user/../../root", "../kodiabear.txt");
path.normalize().relativize("/lion");
System.out.println(path);
```

Jaki będzie wynik wykonania (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	/user/../../root/../../kodiabear.txt
b)	/user/./root/kodiabear.txt/lion
c)	/kodiabear.txt
d)	kodiabear.txt
e)	../lion
f)	Błąd kompilacji



Zadanie#19. Przeanalizuj poniższy fragment kodu.

```
Path path1 = Paths.get("./goat.txt").normalize(); // k1
Path path2 = Paths.get("mule.png");
Files.copy(path1, path2, StandardCopyOption.COPY_ATTRIBUTES); //k2
System.out.println(Files.isSameFile(path1, path2)); //k3
```

Jaki będzie wynik wykonania powyższego fragmentu kodu przy założeniu, że dla metody `copy()` plik źródłowy istnieje jako zwykły plik w systemie plików natomiast plik docelowy nie istnieje?

a)	Zostanie wyświetlone <code>false</code>
b)	Zostanie wyświetlone <code>true</code>
c)	Błąd kompilacji z powodu linii <code>k1</code>
d)	Błąd kompilacji z powodu linii <code>k2</code>
e)	Błąd kompilacji z powodu linii <code>k3</code>
f)	Kod kompiluje się, jednak wyrzuca wyjątek w trakcie działania



Zadanie#20. Przeanalizuj poniższy fragment kodu.

```
final Path path = Paths.get(".").normalize(); // h1
int count = 0;
for(int i=0; i<path.getNameCount(); ++i) {
    count++;
}
System.out.println(count);
```

Jaki będzie wynik wykonania powyższego fragmentu kodu przy założeniu, że bieżący katalog to `/seals/harp/food`?

a)	0
b)	1
c)	2
d)	3
e)	4
f)	Kod wyrzuca wyjątek w trakcie działania z powodu linii <code>h1</code>



Zadanie#21. Jakie są trzy korzyści korzystania z API NIO.2 w stosunku do `java.io.File` podczas pracy z plikami?

a)	NIO.2 wspiera obsługę atrybutów zależną od systemu operacyjnego
b)	NIO.2 może zostać użyte do wylistowania wszystkich plików wewnątrz pojedynczego katalogu
c)	NIO.2 umożliwia bezpośrednie przechodzenie przez drzewo katalogów
d)	NIO.2 może zostać użyte do usuwania plików i niepustych katalogów
e)	NIO.2 wspiera linki symboliczne
f)	NIO.2 może zostać użyte do odczytu czasu ostatniej modyfikacji



Zadanie#22. Która z metod NIO.2 jest najbardziej podobna do metody `java.io.File.listFiles()`?

a)	<code>Path.listFiles()</code>
b)	<code>Files.walk()</code>
c)	<code>Files.find()</code>
d)	<code>Files.files()</code>
e)	<code>Files.list()</code>
f)	<code>Files.lines()</code>



Zadanie#23. Przeanalizuj poniższy fragment kodu.

```
Path path = _____;
if(Files.isDirectory(path))
    System.out.println(Files.deleteIfExists(path) ? "Success": "Try Again");
```

Jakie wartości ścieżki wstawione w pustą linię umożliwią poprawne wykonanie kodu i spowodują wyświetlenie na konsoli komunikatu `Success` (proszę zaznaczyć wszystkie poprawne odpowiedzi)?

a)	Ścieżka odnosząca się do zwykłego pliku w systemie plików
b)	Ścieżka odnosząca się do symbolicznego linku w systemie plików
c)	Ścieżka odnosząca się do pustego katalogu w systemie plików
d)	Ścieżka odnosząca się do katalogu z zawartością w systemie plików
e)	Ścieżka odnosząca się do rekordu istniejącego w systemie plików
f)	Bez względu na ścieżkę wystąpi błąd kompilacji



Zadanie#24. Napisz implementację poniższej metody, w której strumienie `STDOUT` oraz `STDERR` zostają przekierowane do strumienia pliku o nazwie przekazanej parametrem wywołania metody oraz rozszerzeniu „.log”.

```
public static void setStdOut2File( String filename ) {
    .....
    .....
    .....
    .....
    .....
    .....
}
```




Poprawnych odpowiedzi	Ocena
<0>	n/k
<1, 14>	2.0
<15, 16>	3.0
<17, 18>	3.5
<19, 20>	4.0
<21, 22>	4.5
<23, 24>	5.0