1.
Jaki będzie wynik kompilacji poniższego kodu?

```
public class A {
    public static void m(Object o) {
      if(o instanceof Integer)
        System.out.println("OK");
    }

  public static void main(String[] args) {
    m(12);
  }
}
```

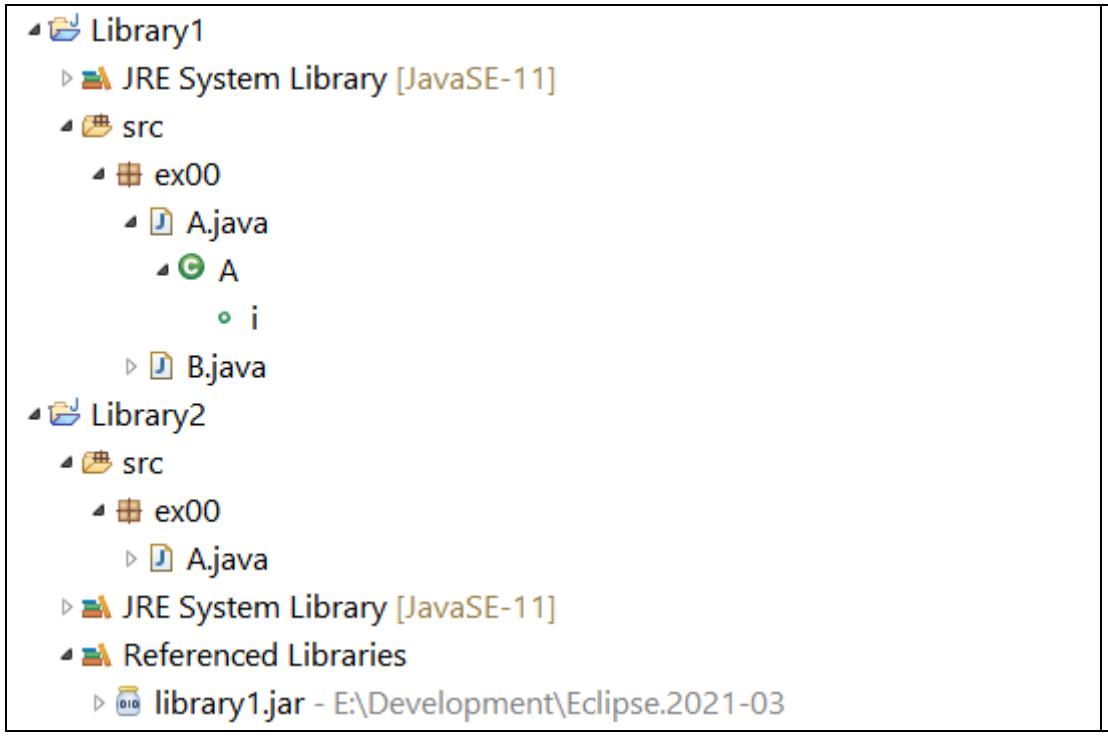
a) Program nie skompiluje się
b) Po kompilacji i uruchomieniu na ekranie pojawi się `OK`
c) Po kompilacji i uruchomieniu wyrzucony zostanie wyjątek: `Exception in thread "main" java.lang.ClassCastException:`
d) Po kompilacji i uruchomieniu wyrzucony zostanie wyjątek: `Exception in thread "main" java.lang.RuntimeException:`

2.
Które ze zdań są prawdziwe?

a) W projekcie modułowym (z `modules-info.java`) można skorzystać z niemodułowej biblioteki klas (nie zawierającej `modules-info.java`) dostarczonej w pliku jar. Wystarczy umieścić ten plik jar w ścieżce klas.
b) W projekcie modułowym (z `modules-info.java`) można skorzystać z niemodułowej biblioteki klas (nie zawierającej `modules-info.java`) dostarczonej w pliku jar. Wystarczy umieścić ten plik jar w ścieżce modułów.
c) W projekcie modułowym (z `modules-info.java`) można skorzystać z niemodułowej biblioteki klas (nie zawierającej `modules-info.java`) dostarczonej w pliku jar. Wystarczy umieścić ten plik jar w ścieżce modułów albo w ścieżce klas oraz zrobić odpowiedni wpis deklarujący zależność w pliku `modules-info.java` projektu.
d) Żadna odpowiedź nie jest prawdziwa.

3.

Załóżmy, że istnieją dwa niemodułowe projekty bibliotek `Library1` oraz `Library2` i że w projekcie `Library2` do ścieżki klas wstawiono jar zawierający klasy biblioteki `Library1` (struktury projektu widać po lewej). Co można powiedzieć o kodzie klasy `A` z projektu `Library2` (widocznym po prawej)?




```
package ex00;

public class A {
   {
     B b = new B();  // 1
     A a = new A();  // 2
     a.i = 10;       // 3
   }
}
```

a) Kod ten nie skompiluje się poprawnie z uwagi na konflikt nazw pakietów
b) Kod ten nie skompiluje się poprawnie z powodu braku deklaracji importu klasy `B` (linia `1`)
c) Kod ten nie skompiluje się poprawnie z powodu powtórnego wystąpienia klasy `A` w pakiecie `ex00` (linia `2`)
d) Kod ten nie skompiluje się poprawnie z powodu braku pola `i` w klasie `A` (linia `3`)

4.
Co można powiedzieć o poniższym kodzie?

```
package ex00;

public class A {
  private static void m(Integer... ti) { // 1
    System.out.println(ti[0]);           // 2
  }

  public static void main(String[] args) {
    m();                                 // 3
  }
}
```

a) Kod ten nie skompiluje się poprawnie. Błąd wystąpi w linii `1`
b) Kod ten nie skompiluje się poprawnie. Błąd wystąpi w linii `2`
c) Kod ten nie skompiluje się poprawnie. Błąd wystąpi w linii `3`
d) Kod ten skompiluje się poprawnie, jednak podczas uruchomienia wyrzucony zostanie wyjątek: `Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException:`

5.
Co można powiedzieć o poniższym kodzie?

```java
package ex00;

public class A {
    enum E { A, B, C };

  public static void main(String[] args) {
        new B().m(E.C);
  }
}
```

```java
package ex00;

public class B {
   void m(E e) {
           System.out.println(e);
   }
}
```

a) Jego kompilacja zakończy się błędem
b) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` na ekranie pojawi się `E.C`
c) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` na ekranie pojawi się `C`
d) Jego kompilacja powiedzie się. Uruchomieniu klasy `A` zakończy się wyrzuceniem wyjątku

6.
Co można powiedzieć o poniższym kodzie?

```java
package ex00;

public class A {
  void m() {
    notify();
  }

  public static void main(String[] args)
throws InterruptedException {
    W w = new W();
    w.start();
    Thread.sleep(100);
    A a = new A();
    a.m();
  }
}
```

```java
package ex00;

public class W extends Thread {
  public void run() {
    try {
      wait();
    } catch (InterruptedException e) {
      e.printStackTrace();
    }
    System.out.println("End");
  }
}
```

a) Jego kompilacja zakończy się błędem
b) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` na ekranie pojawi się `End` i program zakończy działanie
c) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` wyrzucone zostaną dwa wyjątki: `Exception in thread "Thread-0" java.lang.IllegalMonitorStateException` oraz `Exception in thread "main" java.lang.IllegalMonitorStateException`
d) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` na ekran pozostanie pusty, program nie zakończy działania

7.
Co można powiedzieć o poniższym kodzie?

```
package ex00;

class E extends Exception{
  private static final long serialVersionUID = 1L;
}

interface I {
  void m() throws E;                          // 1
}

public class A implements I{
  public static void m() throws Exception { // 2
  }

  public static void main(String[] args) throws Exception {
        m();                                  // 3
  }
}
```

a) Jest to niepoprawny kod (błąd występuje w linii 1)
b) Jest to niepoprawny kod (błąd występuje w linii 2)
c) Jest to niepoprawny kod (błąd występuje w linii 3)
d) Jest to poprawny kod

8.
Co można powiedzieć o poniższym kodzie?

```
import java.util.stream.IntStream;

public class A {
  static int m(int i) {
    int sum=0;
    for(int j=3; j>i; j--)
      sum +=j;
    return sum;

  }
  static void n(int i) {
    System.out.print(i+" ");
  }
  public static void main(String[] args) {
    IntStream.iterate(0, i -> m(i)).limit(3).forEach(A::n);
  }
}
```

a) Jego kompilacja zakończy się błędem
b) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` zostanie wyrzucony wyjątek
c) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` na ekranie pojawi `0 6 0`
d) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` na ekranie pojawi `0 6 12`

9.
Co można powiedzieć o poniższym kodzie?

```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class A {
  static List<Integer> ldi = new ArrayList<Integer>();
  static List<Integer> lda = new LinkedList<Integer>();
  public static void main(String[] args) {
    for(int i=0; i<3; i++) {
      ldi.add(i);
      lda.add(i);
    }
    System.out.println(ldi + " " + lda);

  }
}
```

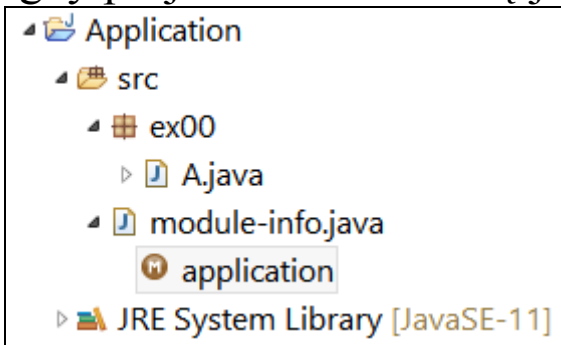
a) Jego kompilacja zakończy się błędem
b) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` na ekranie pojawi się `[0, 1, 2, 0, 1, 2]`
c) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` na ekranie pojawi się `[0, 1, 2] [0, 1, 2]`
d) Jego kompilacja powiedzie się. Po uruchomieniu klasy `A` na ekranie pojawi się `[0, 1, 2] [2, 1, 0]`

10.

Co można powiedzieć o wyniku wywołania komendy głównym katalogu projektu

```
> java -cp ./bin ex00.A 123
```

gdy projekt miał strukturę jak niżej?

<table>
<tr>
<td>


</td>
<td>

```
package ex00;

public class A {
  public static void main(String[] args) {
    System.out.println("Hello "+args[0]);
  }
}
```

</td>
</tr>
<tr>
<td>

```
.
├──bin
│   module-info.class
│   └──ex00
│           A.class
└──src
    module-info.java
    └──ex00
            A.java
```

</td>
<td>

```
module application {
}
```

</td>
</tr>
</table>

a) Jeśli komenda zostanie wydana w systemie Windows to pojawi się komunikat o wystąpieniu wyjątku: `java.lang.ClassNotFoundException: ex00.A`

b) Pojawi się komunikat o wystąpieniu wyjątku: `java.lang.NullPointerException`

c) Pojawi się komunikat o wystąpieniu wyjątku: `java.lang.ArrayIndexOutOfBoundsException`

d) Na ekranie zostanie wypisane: `Hello 123`