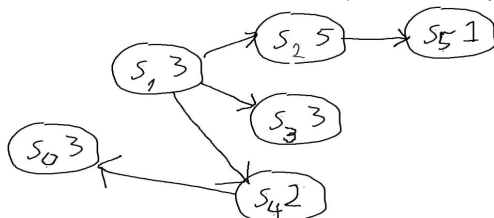


**Proponowany sposób rozwiązania****Faza pisania:**

1. Zadania rozwiązać samemu w ciągu 3 godzin na kartkach, włącznie z zadaniem programistycznym. (Nie dotyczy ostatniego podzadania ZZW 1 (zaznaczone innym fontem), które wymaga zajrzenia do literatury.)
2. Po rozwiązaniu, zadania sprawdzić dokładnie czytając i analizując rozwiązania oraz poprawić ewentualne błędy.

**Faza ustalania poprawności odpowiedzi oraz samoocenia (nie liczy się do czasu pisania):**

3. Zapisany na kartce program przepisać używając Eclipse. Następnie program przetestować i poprawić ewentualne błędy. Uzupełnienia i poprawki nanieść na rozwiązanie na kartce czerwonym kolorem.
4. Podobnie, przepisać program zadań ZZL 2 oraz ZZL 3 i przetestować za pomocą odpluskwiacza. Można też użyć odpowiednich poleceń `println` w celu ustalenia wartości zmiennych i atrybutów. Jak poprzednio, uzupełnienia i poprawki nanieść czerwonym długopisem na rozwiązanie na kartce.
5. Rozwiązania zadań wykładowych porównać z materiałem na slajdach oraz krytycznie przedyskutować w grupie. Podobnie jak poprzednio, nanieść uzupełnienia i poprawki czerwonym długopisem.
6. Za rozwiązanie każdego zadanie jest maksimum 20 punktów. Za bardziej istotne braki i błędy odejmować jakieś 10% - 20% punktów, za mniej istotne 5%.

**Zaliczenie laboratorium, Zadanie 1, Poprawna odpowiedź 20 pkt. (oznaczenie (ZZL 1))**

**Zaimplementuj** grafy skierowane z wierzchołkami etykietowanymi liczbami całkowitymi.

**Zaimplementuj** metodę, która będzie znajdowała sumę liczb etykietujących sąsiada danego wierzchołka. Dla wierzchołka `s1`, sąsiadami są `s2`, `s3`, `s4`.

**Zaimplementuj** metodę, która będzie znajdowała w danym grafie taki wierzchołek, dla którego suma etykiet sąsiadów jest

maksymalna, na diagramie jest to `s1`. Jeśli jest takich wierzchołków więcej, to metoda może zwracać którykolwiek z nich. Dla wierzchołka `s1` ta suma wynosi 10.

**Zilustruj** powstałą strukturę klas odpowiednim diagramem klas języka UML.

**Zaliczenie laboratorium, Zadanie 2, Poprawna odpowiedź 20 pkt. (oznaczenie (ZZL 2))**

Wykonanie programu wymaga użycia stosu, na którym zapisywane są zmienne oraz ich wartości. Stany stosu zapisujemy jako ciąg list par odpowiadający metodom na stosie i zmiennym zadeklarowanym i modyfikowanym w czasie ich wykonania. Lista par odpowiada metodzie na stosie, jest zapisana w nawiasie kwadratowym, a każda para składa się z nazwy zmiennej oraz jej wartości. Np.  $([... , (x, 7), (y, \text{true})])_{\text{main}}$  to stan, w którym wykonywana jest metoda `main`, a wartości zmiennych `x` i `y` są odpowiednio równe 7 i `true`. Rozważ następujący kod:

```

public class Fibonacci {
    public static int fibFailure(int n) {
        int z = 1;
        if (n == 0) //A
            throw new IllegalArgumentException("argument = " + n);
        if (n == 1) //B
            return z;
        int x = fibFailure(n-1); //C
    }
}

```

```

        int y = fibFailure(n-2); //B
        System.out.println("n = " + n);
        z = x + y; //D
        return z;
    }

    public static void main(String[] args)
    {
        int z = 0; //I0
        {
            int x = 3; //I1
            int y = 2; //I2
            z = x + y; //I3
        }
        z = ++z; //I4
        try {
            fibFailure(++z); //I5
        } catch (IllegalArgumentException e) {
            System.out.println("e = " + e); //I4
        }
    }
}

```

Wypisz stany stosu kiedy wykonanie programu znajdzie się w punktach A, ..., D, I0, ... I5, czyli po zakończeniu wykonania odpowiednich linii. Wypisz je w **kolejności w jakich występują** podczas wykonaniu programu.

### Zaliczenie laboratorium, Zadanie 3, Poprawna odpowiedź 10 pkt.(oznaczenie (ZZL 3)

Co będzie wynikiem wykonania poniższej metody `main()`?

Odpowiedź szczegółowo uzasadnij odwołując się do zasad wiązania atrybutów i metod, oraz przekazywania wartości parametrów. Tj. dla każdego polecenia `println` wyjaśnij dokładnie, jakie metody lub atrybuty będą związane, z której klasy i kiedy będą związane, oraz jakie wartości będą miały, zmienne, atrybuty i parametry. Podobnie dla parametrów podaj, jak będą przekazywane.

```

1. class A {
2.     static int cA = 0;
3.     int a = 0;
4.     int b = 0;
5.     A(){
6.         ++cA;
7.         a = cA;
8.     }
9.     public String toString() {
10.        return "a = " + a + ", b = " + b;
11.    }
12.    public int mt(int x) {
13.        b = cA;
14.        return ++b + ++x;
15.    }
16. }
17.
18. class B extends A {
19.     static int cB = 0;
20.     int a = 0;
21.     int b = 0;
22.     B(){cB++;}
23.     public String toString() {
24.        return "a = " + a + ", b = " + b;
25.    }

```

```

26.         public int mt(int z){
27.             b = cB;
28.             return ++a + z + b;
29.         }
30.     }
31.     public class C extends B{
32.         public int a = 5;
33.         static int cC = 1;
34.         public C() {
35.             ++cC;
36.         }
37.         public String toString() {
38.             return "a = " + a + ", b = " + b;
39.         }
40.         public int mt(int y){
41.             y = 3 * y;
42.             a = 5*a;
43.             cC = 0;
44.             return (++a)*(y--);
45.         }
46.         public static void main(String[] args) {
47.             A o = new A();
48.             System.out.println("cA = " + A.cA);
49.             A o1 = new B();
50.             System.out.println("cB = " + B.cB);
51.             System.out.println(o.toString());
52.             A o2 = new C();
53.             System.out.println("cC = " + C.cC);
54.             System.out.println("o1 = " + o1.toString());
55.
56.             int x = 5;
57.             System.out.println("cA = " + A.cA +
58.                 ", cB = " + B.cB + ", cC = " + C.cC);
59.             System.out.println("o.mt =" + o.mt(--x));
60.             System.out.println("o1.mt =" + o1.mt(--x));
61.             System.out.println("o2.mt =" + o2.mt(x--));
62.             System.out.println("x = " + x);
63.             System.out.println("o = " + o.toString());
64.             System.out.println("o1 = " + o1.toString());
65.             System.out.println("o2 = " + o2.toString());
66.             System.out.println("cA = " + A.cA +
67.                 ", cB = " + B.cB + ", cC = " + C.cC);
68.         }
69.     }

```

---

**Zaliczenie wykładu, Zadanie 1, Poprawna odpowiedź 20 pkt.**

1. Wyjaśnij, co to jest sygnatura i jakie ma elementy. Opisz role elementów sygnatury oraz ich typy, jeśli te elementy posiadają różne typy.
2. Jakie są dwa zasadnicze rodzaje parametrów metody? Wyjaśnij jaka jest rola tych dwu rodzajów i jak je rozpoznać?

3. Jakie są dwa zasadnicze rodzaje atrybutów? Wyjaśnij jaka jest rola tych dwu rodzajów i jak je rozpoznać?
4. Jakie są dwa zasadnicze rodzaje metod. Wyjaśnij jaka jest rola tych dwu rodzajów i jak je rozpoznać?
5. Wyjaśnij, czym jest wiązanie statyczne, jaki jest jego mechanizm i jakich elementów dotyczy? Wyjaśnij, czym jest wiązanie dynamiczne, jaki jest jego mechanizm i jakich elementów dotyczy? Wyjaśnij, dlaczego sposób wiązania odgrywa rolę. Odwołaj się do przykładu z zadania na zaliczenie laboratorium ZZL nr 3.
6. Porównaj i omów główne podklasy klasy `Throwable`. Wyjaśnij metodykę ich użycia. Wyjaśnij mechanizmy z nimi związane, w tym sposób ich deklaracji w metodzie oraz znaczenie deklaracji oraz mechanizm ich obsługi.
7. Jaka jest różnica pomiędzy przeładowaniem metody a polimorfizmem? (wymaga doczytania w podanej na wykładzie literaturze).

**Zaliczenie wykładu, Zadanie 2, Poprawna odpowiedź 20 pkt.**

Narysuj stany stosu oraz, używając diagramu obiektowego UML (zobacz slajdy), narysuj stany składu po zakończeniu linii 47, 49, 52.

Wyjaśnij, jak wygląda krok po kroku działanie stosu podczas wykonania programu. Wyjaśnij dokładnie, co się wtedy dzieje ze zmiennymi, atrybutami oraz metodami, a także konstruktorami. Wyjaśnij, jak wtedy działała mechanizm wiązania atrybutów, a jak działa mechanizm wiązania metod. Zilustruj to odwołując się do zadania ZZL3.

Co się stanie, gdy zostanie rzucony wyjątek?