**Student propunător: Teofana Enăchioiu**

**Grupa:223**

**Email: teofanaenachioiu@gmail.com**

* Care este diferenta intre a declara o lista de tipul List<? extends T> si List<? super T> ?

a. Secventele sunt echivalente;

b. "? extends T" defineste limita inferioara si se foloseste cand adaugam valori intr-o lista (ex: lista.add(t));

c. "? super T" defineste limita superioara si se foloseste cand prelucram valorile unei liste (ex: lista.get(0));

d. "? super T" defineste limita inferioara si se foloseste cand adaugam valori intr-o lista (ex: lista.add(t));

**Raspuns corect:** d.

Exista o diferenta remarcabila intre a folosi List<? extends T> si List<? super T>.

List<? extends T> defineste limita superioara si este utilizat atunci cand se doareste extragerea valorilor listei si prelucrarea acestora.

Exemplu:

public static double suma(Collection<? extends Number> numere) {

double s = 0.0;

for (Number n : numere)

s += n.doubleValue();

return s;

}

!!! Exceptie: Se poate pune valoarea nul intr-o lista declarata ca List<? extends T>

List<? extends Number> numere = Arrays.<Number>asList(1,2,2.78,3.14);

numere.add(null); // ok

List<? super> defineste limita inferioara si este folosit cand se construieste lista.

Exemplu:

public static void numara(Collection<? super Integer> ints, int n) {

for (int i = 0; i < n; i++) ints.add(i);

}

!!! Exceptie: Se pot prelucra datele din lista doar data lista are elemente de tip Object.

String str="";

for (Object obj : ints) str += obj.toString();