

## Exercise 6 – Procedural programming an OOP Modeling

### Task 1 – Procedural problem solving approach

You should write a programme that solves the following problem:

The travel insurance policies of three insurance companies should be compared for a travel insurance portal. The premiums of the three insurance policies are calculated as follows:

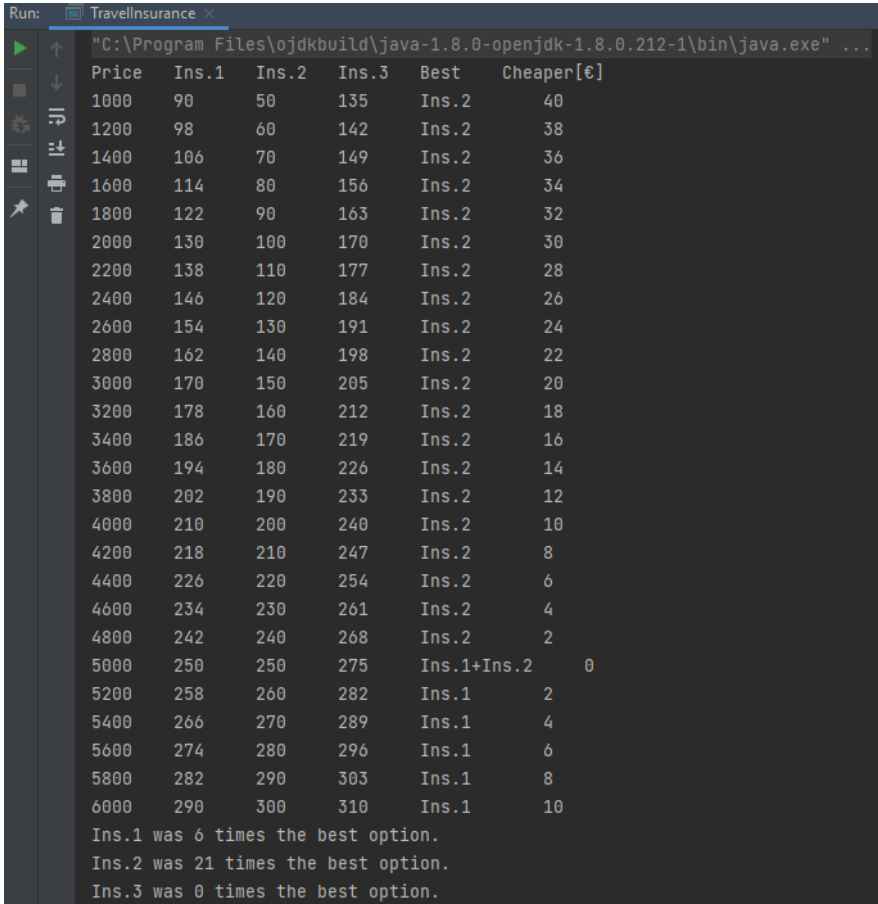
- Insurance policy 1: Premium = €50 + 4% of the travel price
- Insurance policy 2: Premium = 5% of the travel price
- Insurance policy 3: Premium = €100 + 3.5% of the travel price

The premiums for travel prices from €1,000 to €6,000 should be calculated in increments of €200.

For each travel price, it should be indicated which insurance policy is the cheapest. The price difference between the cheapest premium and the second cheapest premium must also be indicated. The premiums must be rounded off to whole number (integer) amounts.

In addition, the number of cases in which each insurance policy is the cheapest must be indicated.

The programme should output the results of the comparison in a table as follows:



Price	Ins.1	Ins.2	Ins.3	Best	Cheaper[€]
1000	90	50	135	Ins.2	40
1200	98	60	142	Ins.2	38
1400	106	70	149	Ins.2	36
1600	114	80	156	Ins.2	34
1800	122	90	163	Ins.2	32
2000	130	100	170	Ins.2	30
2200	138	110	177	Ins.2	28
2400	146	120	184	Ins.2	26
2600	154	130	191	Ins.2	24
2800	162	140	198	Ins.2	22
3000	170	150	205	Ins.2	20
3200	178	160	212	Ins.2	18
3400	186	170	219	Ins.2	16
3600	194	180	226	Ins.2	14
3800	202	190	233	Ins.2	12
4000	210	200	240	Ins.2	10
4200	218	210	247	Ins.2	8
4400	226	220	254	Ins.2	6
4600	234	230	261	Ins.2	4
4800	242	240	268	Ins.2	2
5000	250	250	275	Ins.1+Ins.2	0
5200	258	260	282	Ins.1	2
5400	266	270	289	Ins.1	4
5600	274	280	296	Ins.1	6
5800	282	290	303	Ins.1	8
6000	290	300	310	Ins.1	10

Ins.1 was 6 times the best option.  
Ins.2 was 21 times the best option.  
Ins.3 was 0 times the best option.

## Task 2 – Object-oriented modelling

You have decided to open a coffee bar in a student hall of residence. At the beginning, you would like to offer the following product range:

- Cappuccino, consisting of an espresso and foamed milk,
- Latte macchiato, consisting of an espresso and twice as much foamed milk as the cappuccino,
- Espresso.

This is the price list of the individual ingredients:

Price List	
Espresso	1,50 €
Cappuchino	2,00 €
Latte Macchiato	2,50 €
Milk	0,50 €
Sweetened	+0,20 €
To go	+0,10 €

The strength of the espresso is adjusted to the customer's wishes for each of these beverages. The options are "weak", "medium" and "strong".

At the customer's request, the selected coffee can have an optional extra of sugar, sweetener or vanilla syrup.

You serve your coffees either as a takeaway (to go) in a paper cup of the appropriate size, or in suitable stylish porcelain cups (cappuccino and espresso) or glasses (latte macchiato).

Your idea is very well received, so that you can make many customers happy with coffees on the very first day. In particular, the following orders are received:

Peter's Order  
cappuccino  
cup = paper

- Peter orders a strong cappuccino in a paper cup.
- Annabell orders a medium-strength latte macchiato with vanilla syrup in a glass.
- Kai orders a weak espresso with sugar in a porcelain cup.
- Connie orders a medium-strength cappuccino with sweetener in a paper cup.

Annabell's Order  
Latte macchiato  
milk = 1  
vanilla syrup  
cup = glass

Kai's Order  
espresso  
milk = 2  
extra sugar  
cup = porcelain

Connie's Order  
Cappuccino  
milk = 1  
sweetener  
cup = paper

Model these orders in an object-oriented manner. To do so, please complete the following key questions / work steps.

- What are the things (*nouns, objects*) that are involved in the individual orders?
- How are these things related?
- What are the relevant properties of the things being considered?
- Model each of these orders graphically in a UML object diagram.
- How many portions of foamed milk were served?
- How many paper cups were used?
- (Additional task) Create the appropriate classes in Java and display the instances previously represented.