

Тестирање софтвера и обезбеђивање квалитета, група А

Директоријум који је корен пројекта (STExam) је потребно назвати у формату индекс_име_презиме. Пројекти који не буду потписани неће бити уважени.

Тестирати само наведено!

1. Метод *groupTicketOrder(int numOfTickets)* се користи за израчунавање процента попушта на основу броја наручених карата. Метод враћа израчунати проценат попушта. Ако је број наручених карата мањи од 5, попуст се не даје (0%). Уколико је број наручених карата већи или једнак 5, а мањи од 10, даје се 10% попушта. Ако је број карата 10 или више, попуст је 20%. Ако је унет недозвољен број (негативан број), метода треба да избаци изузетак *IllegalArgumentException*. Морате да:
 - а) Дефинишите класе еквиваленције и напишите тестове са репрезентативним вредностима
 - б) Дефинишите вредности граница и напишите тестове за све вредности граница
2. Метод *calculateTicektPrice()* – основна цена улазнице је 1000 динара. Ова метода узима као параметре старост купца, боолеан ако купац има ваучер и посебан статус купца (могуће вредности *NORMAL* или *BENEFICIAL*). Претпостављамо да ће старост бити унета исправно (не очекујемо негативну вредност). Карта се затим формира коришћењем датих параметара дефинисаних у табели испод. Потребно је написати следеће тестове који ће гарантовати:
 - а. 100% statement coverage
 - б. 100% decision coverage
 - с. 100% path coverage

Age	Status	Ticket price reduction
<=18	Not important	-400
<=18	BENEFICIAL	Additional -100
>18 i <=65	Not important	0
>18 i <=65	BENEFICIAL	-200
>65	Not important	-300

Уколико купац има ваучер, обрачуната цена се додатно умањује за 50%.

3. *CFG (control flow graph)* који за претходни задатак треба послати заједно са задатком.
4. Параметризовани тест – класа *WordConcatenator* има једну методу, *concatenateWords()*, који узима два стринга као улазе, спаја их и враћа спојени

стринг као резултат (пример, позив са низовима „Hello“ и „World“ ће вратити „HelloWorld“). Направите параметризовани тест за овај метод.

5. У класи *ProductService*, потребно је тестирати две методе, *findProduct* (који ће претражити базу података и вратити производ који одговара ИД-у претраге у облику објекта *Movie*), и *countProducts* (који ће вратити целобројну вредност која представља количину филмова у бази података). *ProductDatabase* класа још увек није спроведена. Направите све неопходне ставке да бисте могли да креирате тестове. Предлог – користити *JMock* радни оквир.
6. Направити *TestSuite*, и конфигуришите га да покреће све тестне класе креиране у претходним корацима. *WordConcatenator* тестови би требало да се извршавају први.