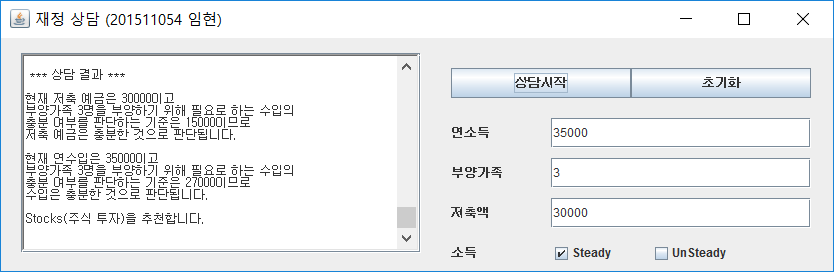
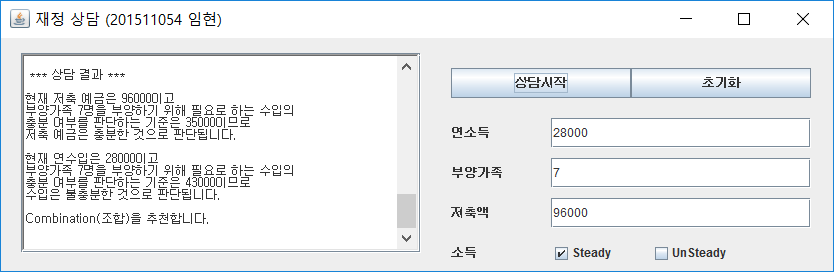
재정 상담 프로그램

● 출력 결과





● 소스 코드

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; @title : Consult.clp

; @author : 임현 (201511054@sangmyung.kr)

; @since : 2017 - 05 - 01

; @brief : 재정상담 프로그램 제스 과제

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Savings 판단

(deffunction investment\_savings ()

(printout t "Savings(저축 예금)을 추천합니다." crlf crlf))

; Stocks 판단

(deffunction investment\_stocks ()

(printout t "Stocks(주식 투자)을 추천합니다." crlf crlf))

; Combination 판단

(deffunction investment\_combination ()

(printout t "Combination(조합)을 추천합니다." crlf crlf))

; 수입 충분 판단

(deffunction incomead (?x, ?y)

(printout t "현재 연수입은 " ?x "이고" crlf)

(printout t "부양가족 " ?y "명을 부양하기 위해 필요로 하는 수입의" crlf)

(printout t "충분 여부를 판단하는 기준은 " (+ 15000 (\* 4000 ?y))

"이므로" crlf)

(printout t "수입은 충분한 것으로 판단됩니다." crlf crlf))

; 수입 불충분 판단

(deffunction incomein (?x, ?y)

(printout t "현재 연수입은 " ?x "이고" crlf)

(printout t "부양가족 " ?y "명을 부양하기 위해 필요로 하는 수입의" crlf)

(printout t "충분 여부를 판단하는 기준은 " (+ 15000 (\* 4000 ?y))

"이므로" crlf)

(printout t "수입은 불충분한 것으로 판단됩니다." crlf crlf))

; 저축 예금 충분 판단

(deffunction savingsad (?x, ?y)

(printout t "현재 저축 예금은 " ?x "이고" crlf)

(printout t "부양가족 " ?y "명을 부양하기 위해 필요로 하는 수입의" crlf)

(printout t "충분 여부를 판단하는 기준은 " (\* 5000 ?y)

"이므로" crlf)

(printout t "저축 예금은 충분한 것으로 판단됩니다." crlf crlf))

; 저축 예금 불충분 판단

(deffunction savingsin (?x, ?y)

(printout t "현재 저축 예금은 " ?x "이고" crlf)

(printout t "부양가족 " ?y "명을 부양하기 위해 필요로 하는 수입의" crlf)

(printout t "충분 여부를 판단하는 기준은 " (\* 5000 ?y)

"이므로" crlf)

(printout t "저축 예금은 불충분한 것으로 판단됩니다." crlf crlf))

(defrule R1

(savings\_account ?x)

(savings\_account inadequate)

(income ?y)

=>

(investment\_savings)

(assert (investment savings)))

(defrule R2

(savings\_account ?x)

(savings\_account adequate)

(income ?y)

(income adequate)

=>

(investment\_stocks)

(assert (investment stocks)))

(defrule R3

(savings\_account ?x)

(savings\_account adequate)

(income ?y)

(income inadequate)

=>

(investment\_combination)

(assert (investment combination)))

(defrule R4

(amount\_saved ?x)

(dependents ?y&: (>= ?x (\* 5000 ?y))) ; greater(X, minsavings(Y))

=>

(savingsad ?x ?y)

(assert (savings\_account adequate)))

(defrule R5

(amount\_saved ?x)

(dependents ?y&: (< ?x (\* 5000 ?y))) ; ㄱgreater(X, minsavings(Y))

=>

(savingsin ?x ?y)

(assert (savings\_account inadequate)))

(defrule R6

(earnings ?x steady)

(dependents ?y&: (>= ?x (+ 15000 (\* 4000 ?y)))) ; greater(X, minincome(Y))

=>

(incomead ?x ?y)

(assert (income adequate)))

(defrule R7

(earnings ?x steady)

(dependents ?y&: (< ?x (+ 15000 (\* 4000 ?y)))) ; ㄱgreater(X, minincome(Y))

=>

(incomein ?x ?y)

(assert (income inadequate)))

(defrule R8

(earnings ?x unsteady)

=>

(incomein ?x ?y)

(assert (income inadequate)))

(run)