Lösung **Aufgabe 1 (Alles eine Frage der Logik)** Blatt 12

```
(a)   x OR (y OR x) OR ~y
    = x OR y OR x OR ~y
    = (x OR x) OR (y OR ~y)
    =     x    OR   TRUE
    = TRUE

(b)   (x OR y) & (x OR ~y)
    = x OR (y & ~y)
    = x OR  FALSE
    = x

(c)   x OR y OR ~x
    = (x OR ~x) OR y
    =   TRUE    OR y
    = TRUE

(d)   (x OR y) & (x OR ~y) & (~x OR y) & (~x OR ~y)
    = (x OR (y & ~y)) & (~x OR (y & ~y))
    = (x OR   FALSE ) & (~x OR   FALSE )
    = x & ~x
    = FALSE

(e)   (x & y) OR (x & ~y) OR (~x & y) OR (~x & ~y)
    = (x & (y OR ~y)) OR (~x & (y OR ~y))
    = (x &   TRUE   ) OR (~x &   TRUE   )
    = x OR ~x
    = TRUE

(f)   (x & y) OR ( (x & ~x) & z )
    = (x & y) OR (  FALSE   & z )
    = (x & y) OR       FALSE
    = x & y
```

Lösung **HitTheMiddle.om** Blatt 12

```
MODULE HitTheMiddle;

        (**************************
         * Allgemeine Informatik I *
         *      WS 2001/2002       *
         *     Uebungsblatt 12     *
         *     Beispielloesung     *
         **************************)

IMPORT Print, Read, RandomGenerators, Conclusions;

CONST MaxGameLength = 10;               (* How long do you want to play? *)
      FieldSize = 7;                    (* This number must be odd! *)

TYPE FieldPosition = RECORD
                        x,y : INTEGER;
                      END;
     SavedMoves = ARRAY MaxGameLength OF FieldPosition;

VAR position       : FieldPosition;
    moves          : SavedMoves;
    capital, round : INTEGER;


  PROCEDURE GameOver(p : FieldPosition; money : INTEGER) : BOOLEAN;
    (* Tells the user if the game is already over
     * Either the player hits the middle of the field
     * or his balance gets negative
     *)
  BEGIN
  RETURN ((money = 0) OR ((p.x = p.y) & (p.x = FieldSize DIV 2)));
  END GameOver;

  PROCEDURE MoveInField(VAR p : FieldPosition; x,y : INTEGER);
    (* Moves the player on the field according to the values of x and y *)
  BEGIN
  p.x:=x;
  p.y:=y;
  END MoveInField;

  PROCEDURE GetRandomNumber(max : INTEGER) : INTEGER;
    (* Returns a random number between 0 and max-1 *)
  BEGIN
  RETURN (SHORT(RandomGenerators.Int32Val()) MOD max);
  END GetRandomNumber;

  PROCEDURE PrintField(p : FieldPosition);
    (* prints the current field; the player's current position is 'O' *)
  VAR i,j : INTEGER;
  BEGIN
  i:=0;
  WHILE (i < FieldSize) DO
    j:=0;
    WHILE (j < FieldSize) DO
      IF ((i = p.x) & (j = p.y)) THEN
                                   Print.F(" X");
                                 ELSE
                                   Print.F(" O");
                                 END;
      INC(j);
      END;
    Print.F("\n");;
```

```
    INC(i);
    END;
  END PrintField;

  PROCEDURE GainLoss(p : FieldPosition) : INTEGER;
    (* Returns the win resp. loss for the actual position
     * $2,500 on diagonals
     * -$4,000 on the border line (except on edges)
     * $50,000 for the middle
     *)
  BEGIN
  IF ((p.x = p.y) OR (p.x = (FieldSize-1)-p.y))
        THEN
          IF (p.x = FieldSize DIV 2)
              THEN
                RETURN 50000;
              ELSE
                RETURN 2500;
              END;
        ELSIF ((p.x = 0) OR (p.x = FieldSize-1) OR
               (p.y = 0) OR (p.y = FieldSize-1))
              THEN
                RETURN -4000;
        ELSE
          RETURN 0;
        END;
  END GainLoss;

  PROCEDURE SaveMove(VAR m : SavedMoves; round : INTEGER; p : FieldPosition);
    (* Saves the current move in the given ARRAY *)
  BEGIN
  m[round]:=p;
  END SaveMove;

  PROCEDURE PrintMoves(m : SavedMoves; lastround : INTEGER);
    (* Prints all previous moves of the game *)
  VAR i : INTEGER;
  BEGIN
  i:=0;
  WHILE (i < lastround) DO
    Print.F3("  Runde %2d : Feld (%d,%d)\n",i+1,m[i].x+1,m[i].y+1);
    INC(i);
    END;
  END PrintMoves;


BEGIN
  (* Set initial values *)
capital:=10000;                  (* begin game with $10,000 *)
position.x:=0; position.y:=0;    (* start at initial position (0,0) *)
round:=1;

  (* Play *)
WHILE ((round <= MaxGameLength) & (~GameOver(position,capital))) DO
  MoveInField(position,GetRandomNumber(FieldSize),GetRandomNumber(FieldSize));
  capital:=capital+GainLoss(position);
  IF (capital < 0) THEN capital:=0 END;
  SaveMove(moves,round-1,position);
  Print.F1("Runde %d:\n",round);
  PrintField(position);
  Print.F1("Kontostand : $%d\n", capital);
  Read.Ln;
```



```
  INC(round);
  END;

  (* Print final result *)
Print.F2("Nach %d Runden ist Dein Endguthaben : $%d\n",round-1,capital);
Print.F("Hier ist eine Aufzeichnung des Spiels:\n");
PrintMoves(moves,round-1);

END HitTheMiddle.
```