OOP HOMEWORK 6 (3 октября 2015 г.)

Tropin Andrew
e-mail: andrewtropin@gmail.com
github: abcdw

Problem 1.

1. d
2. a
3. d
4. e
5. b
6. d, e
7. a
8. d, e
9. e, f

Problem 2.

```
local
  i: like Zurich.stations.new_cursor
do
  from
    i := Zurich.stations.new_cursor
  until
    i.after or else i.item.name ~ "Central"
  loop
    i.forth
  end
  if not i.after then
    i.item.set_position ([0.0, 0.0])
  end
end

local
  station: STATION
do
  across
    Zurich.stations as i
  loop
    if i.item.name ~ "Central" then
      station := i.item
    end
```

```
        end
        if station /= Void then
          station.set_position ([0.0, 0.0])
        end
      end
```

Problem 3.

```
class
  APPLICATION

inherit

  ARGUMENTS

create
  make

feature {NONE} -- Initialization
  make
    local
        i: INTEGER
        j: INTEGER
        w: INTEGER
        tail: INTEGER
        n: INTEGER
        width: INTEGER
    do
      Io.read_integer
      n := Io.last_integer

      width := (n // 2 + 1) * 4 - 1 - ((n + 1)\\ 2) * 2

      from
          i := 0
      variant
          n - i
      until
        i = n
      loop
        w := (i // 2 + 1) * 2

        from
          j := 0
          tail := width - w - 1
        variant
            width - j
        until
```

Tropin Andrew
e-mail: andrewtropin@gmail.com
github: abcdw

```
            j = width
          loop
            if (j < w or j > tail) and (i + j) \\ 2 = 0 then
              print ("*")
            else
                print (" ")
            end
            j := j + 1
          end

          Io.new_line
          i := i + 1
        end

    end
end
```

Problem 4. Board game

```
note
    description: "project application root class"

class
    APPLICATION

inherit

    ARGUMENTS

create
    make

feature {NONE} -- Initialization
    g: GAME
    make
        local
        do
            create g.make
            g.start
        end
end
```

game.e

```
note
    description: "Summary description for {GAME}."
    author: ""
    date: "$Date$"
```

Tropin Andrew
e-mail: andrewtropin@gmail.com
github: abcdw

```
    revision: "$Revision$"

class
    GAME

create
    make

feature

    players: ARRAY [PLAYER]

    dice: DIE

    n: INTEGER

    finished: BOOLEAN

    field_length: INTEGER

    make
        local
            i: INTEGER
        do
            create dice.make

            finished := False
            field_length := 40
            create players.make_empty

            print ("Input player count:%N")
            Io.read_integer
            n := Io.last_integer

            from
                i := 0
            variant
                n - i
            until
                i = n
            loop
                players.force(create {PLAYER}.make (Current), i)
                i := i + 1
            end

        end

    start
```

Tropin Andrew
e-mail: andrewtropin@gmail.com
github: abcdw

```
        local
            i: INTEGER
            p: PLAYER
        do
            from
                i := 0
            until
                finished
            loop
                i := (i + 1) \\ n
                print (i)
                Io.new_line
                p := players[i]
                p.do_turn
                if p.got_end then
                    print (p.name)
                    print (" won%N")
                    finished := True
                end
            end
        end

end
```

die.e

```
note
    description: "Summary description for {DIE}."
    author: ""
    date: "$Date$"
    revision: "$Revision$"

class
    DIE

create
    make

feature {NONE}
    rand: V_RANDOM
feature
    make
        do
            create rand
        end

    roll: INTEGER

        do
```

```
            Result := rand.bounded_item (1, 6)
            rand.forth
            print ("Die value: ")
            Io.put_integer (Result)
            Io.new_line
        end

end
```

player.e

```
note
    description: "Summary description for {PLAYER}."
    author: ""
    date: "$Date$"
    revision: "$Revision$"

class
    PLAYER
create
    make
feature
    name: STRING
    g: GAME
    position: INTEGER

    make(ga: GAME)
    do
        g := ga
        position := 1
        print ("Input player name:%N")
        Io.read_line
        create name.make_from_string (Io.last_string)

    end

    do_turn
    local
        d1: INTEGER
        d2: INTEGER
    do
        d1 := g.dice.roll
        d2 := g.dice.roll

        if d1 = d2 then
            if d1 < position then
                position := position - d1
            else
```

```
                position := 1
            end
        else
            position := position + d1 + d2
            if position > g.field_length then
                position := g.field_length
            end
        end
        print (name)
        print (" position is ")
        print (position)
        Io.new_line
    end

    got_end: BOOLEAN
    do
        Result := position >= g.field_length
    end

invariant
    in_field: 1 <= position
end
```

Tropin Andrew
e-mail: andrewtropin@gmail.com
github: abcdw