OOP HOMEWORK 5 (30 сентября 2015 г.)

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

Problem 1.

```
note
        description: "Creating new objects for Zurich."

class
        OBJECT_CREATION

inherit
        ZURICH_OBJECTS

feature -- Explore Zurich

        explore
                        -- Create new objects for Zurich.
                do
                        add_buildings
                        add_route
                end
        add_buildings
                local
                        first_corner, second_corner: VECTOR
                        eth, opera: BUILDING
                do
                    create first_corner.make(100, -100)
                    create second_corner.make(200, -200)
                    create eth.make ("Super Street", first_corner, second_corner)

                    create first_corner.make (400, -1000)
                    create second_corner.make (300, -1100)
                    create opera.make ("Super Street 2", first_corner, second_corner)

                    opera.set_name("Opera")
                    eth.set_name("ETH")
                    Zurich.add_building(eth)
                    Zurich.add_building(opera)
                end

                add_route
                        local
                                leg1, leg2, leg3: LEG
                                opera_route: ROUTE
                        do
```

```
                create leg1.make (Zurich.station("Polyterrasse"), Zurich.
                create leg2.make (Zurich.station("Central"), Zurich.stati
                create leg3.make (Zurich.station("Paradeplatz"), Zurich.s
                leg1.link(leg2)
                leg2.link(leg3)
                create opera_route.make(leg1)
                Zurich.add_route(opera_route)
            end
end
```

Problem 2.

```
note
        description: "Temperature."

class
        TEMPERATURE

create
        make_celsius, make_kelvin

feature -- Initialization

        make_celsius (v: INTEGER)
                        -- Create with Celsius value 'v'.
                require
                        above_absolute_zero: v >= -Celsius_zero
                do
                        celsius := v
                ensure
                    celsius_value_set: celsius = v
                        -- Create a temperature object encapsulating value 'v' intend
                        -- Your code here
                end

        make_kelvin (v: INTEGER)
                        -- Create with Kelvin value 'v'.
                require
                        above_absolute_zero: v >= 0
                do
                        celsius := v - Celsius_zero
                ensure
                    kelvin_value_set: kelvin = v
                        -- Your code here
                        -- Create a temperature object encapsulating value 'v' intend
                end

feature -- Access
```

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

```
        Celsius_zero: INTEGER = 273

        celsius: INTEGER
                        -- Value in Celsius scale.

        kelvin: INTEGER
                        -- Value in Kelvin scale.
                do
                        Result := celsius + Celsius_zero
                        -- Your code here
                        -- Compute the Kelvin temperature value from the Celsius valu
                end

feature -- Measurement

        average (other: TEMPERATURE): TEMPERATURE
                        -- Average temperature between 'Current' and 'other'.
                require
                    other_exists: other /= Void
                do
                        create Result.make_celsius((celsius + other.celsiuss) // 2)
                        ensure
                                between: (celsius <= Result.celsius and Result.celsiu
                                (other.celsius <= Result.celsius and Result.celsius <
                        -- Your code here.
                        -- Compute the average of two temperature. One is given by th
                        -- the other is passed as an argument.
                end
invariant
    above_absolute_zero: kelvin >= 0
end

note
        description : "project application root class"
class
        APPLICATION

inherit
        ARGUMENTS

create
        make

feature {NONE} -- Initialization

        make
                local
                    t1, t2, t3: TEMPERATURE
```

```
                -- Run application.
        do
                --| Add your code here
                -- print ("Hello Eiffel World!%N")
                print ("Enter t1 in Celsius: ")
                Io.read_integer
                create t1.make_celsius (Io.last_integer)
                print ("t1 in Kelvin is: ")
                print (t1.kelvin)
                print ("%N")

                print ("Enter t2 in Kelvin: ")
                Io.read_integer
                create t2.make_kelvin(Io.last_integer)
                print ("t2 in Celsius is: ")
                print (t2.celsius)
                print ("%N")

                t3 := t1.average(t2)
                print ("Average in Celsius is: ")
                print (t3.celsius)
                print ("%N")
                print ("Average in Kelvin is: ")
                print (t3.kelvin)
                print ("%N")
        end

end
```

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