第3回
関数を使ったプログラムの実行と作成

情報処理演習II

Literacy

例題

Literacy

例題1．実行結果に至る過程

- 次の Scheme 式について、実行結果「48」に至る過程を見る

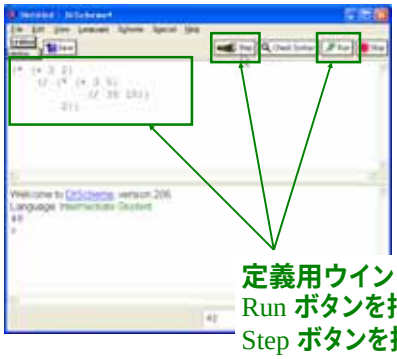
```
(* (+ 2 2)
   (/ (* (+ 3 5)
         (/ 30 10))
      2))
```

「*」とあるのは、「乗算」の意味

```
(* (+ 2 2) (/ (* (+ 3 5) (/ 30 10)) 2))
  = (* 4 (/ (* (+ 3 5) (/ 30 10)) 2))
  = (* 4 (/ (* 8 (/ 30 10)) 2))
  = (* 4 (/ (* 8 3) 2))
  = (* 4 (/ 24 2))
  = (* 4 12)
  = 48
```

Literacy

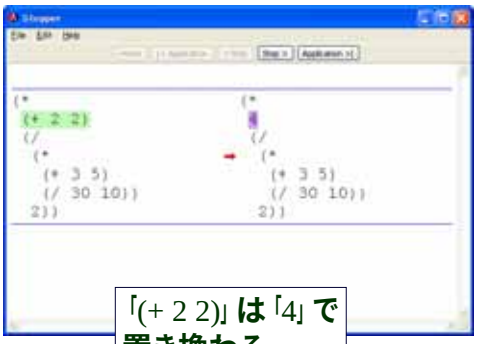
例題1．式のステップ実行



定義用ウインドウに入力して、Run ボタンを押した後、Step ボタンを押すと

Literacy

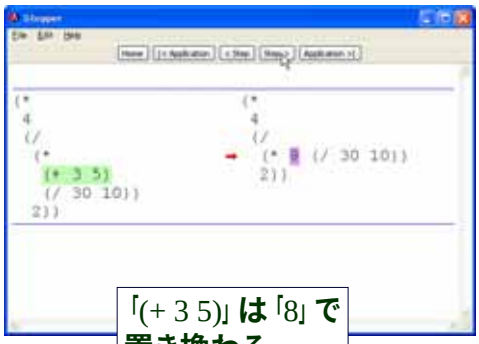
例題1のステップ実行(1/6)



「(+ 2 2)」は「4」で置き換わる

Literacy

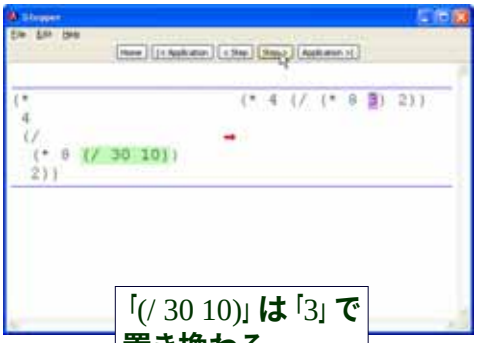
例題1のステップ実行(2/6)



「(+ 3 5)」は「8」で置き換わる

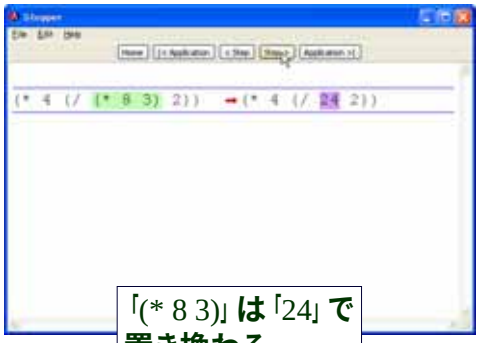
Literacy

例題1のステップ実行(3/6)



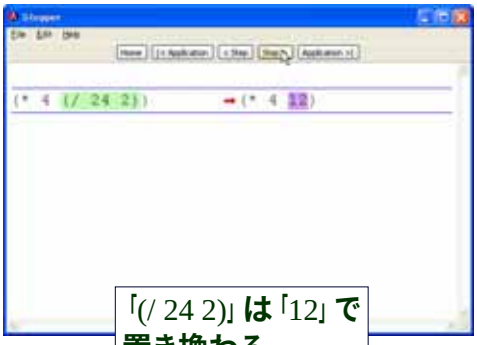
「(/ 30 10)」は「3」で置き換わる

例題1のステップ実行(4/6)



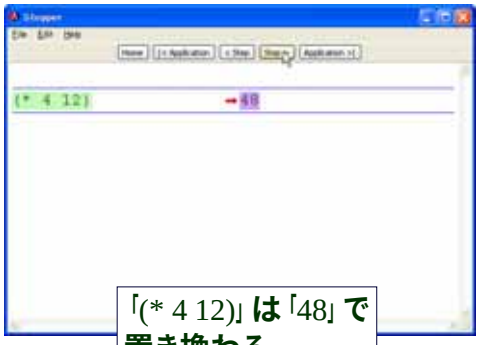
「(* 8 3)」は「24」で置き換わる

例題1のステップ実行(5/6)



「(/ 24 2)」は「12」で置き換わる

例題1のステップ実行(6/6)



「(* 4 12)」は「48」で置き換わる

実行結果「48」に至る過程

```
(* (+ 2 2) (/ (* (+ 3 5) (/ 30 10)) 2))   最初の式
= (* 4 (/ (* (+ 3 5) (/ 30 10)) 2))   (+ 2 2) → 4
= (* 4 (/ (* 8 (/ 30 10)) 2))   (+ 3 5) → 8
= (* 4 (/ (* 8 3) 2))   (/30 10) → 3
= (* 4 (/ 24 2))   (* 8 3) → 2
= (* 4 12)   (/24 2) → 12   コンピュータ内部での計算
= 48   実行結果
```

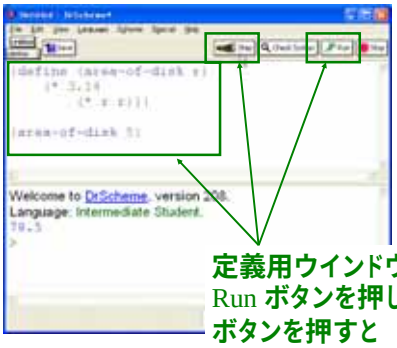
例題2．関数のステップ実行

- 円の半径 r から面積を求める関数 area-of-disk について、実行結果に至る過程を見る
  - (area-of-disk 5) から「78.5」に至る過程を見る

```
(define (area-of-disk r)
    (* 3.14
       (* r r)))
```

```
(area-of-disk 5)
= (* 3.14 (* 5 5))
= (* 3.14 25)
= 78.5
```

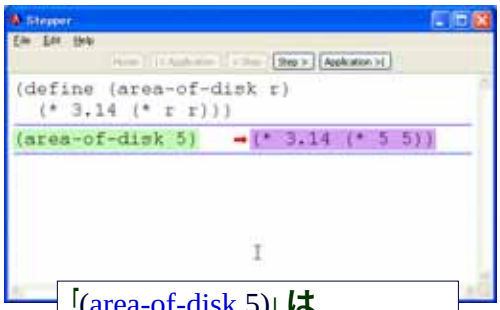
例題2．関数のステップ実行



定義用ウインドウに入力して、Run ボタンを押した後、Step ボタンを押すと

Literacy

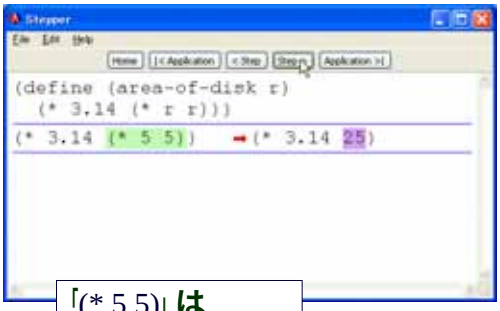
例題2のステップ実行(1/3)



「(area-of-disk 5)」は「(* 3.14 (* 5 5))」で置き換わる

Literacy

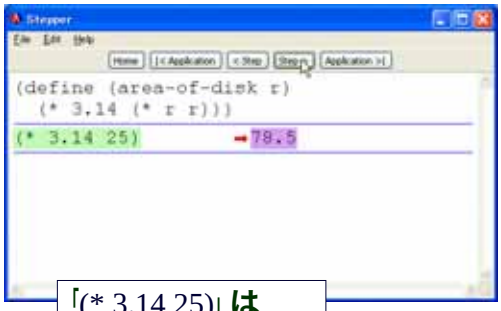
例題2のステップ実行(2/3)



「(* 5 5)」は「25」で置き換わる

Literacy

例題2のステップ実行(3/3)



「(* 3.14 25)」は「78.5」で置き換わる

Literacy

(area-of-disk 5) から 78.5 が得られる過程

(area-of-disk 5) 最初の式

= (* 3.14 (* 5 5)) (* 3.14 (* r r)) に r = 5 が代入される

= (* 3.14 25) (* 5 5) → 25 コンピュータ内部での計算

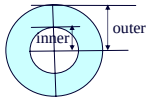
= 78.5 実行結果

Literacy

例題3．リングの面積

- 外径 outer、内径 inner からリングの面積を求める関数 area-of-ring を定義し、実行する
  - 円の面積を求める関数 area-of-disk（前回の演習）と組み合わせる

真ん中に穴のあいた円の面積と考える

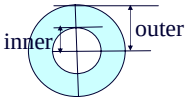


まず問題をよく分析して理解する

Literacy

リングの面積

外径: outer
内径: inner

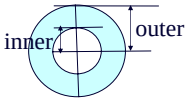


リングの面積
＝ 外側の円の面積 － 内側の円の面積
半径 outer の円　半径 inner の円

Literacy

リングの面積のプログラム

外径: outer
内径: inner




```
(define (area-of-disk r)
  (* 3.14
     (* r r)))
(define (area-of-ring outer inner)
  (- (area-of-disk outer)
     (area-of-disk inner)))
```

area-of-disk の部分

area-of-ring の部分

Literacy

関数を分割する理由

分割する場合

```
(define (area-of-disk r)
   (* 3.14
      (* r r)))
(define (area-of-ring outer
inner)
   (- (area-of-disk outer)
      (area-of-disk inner)))
```

分割しない場合

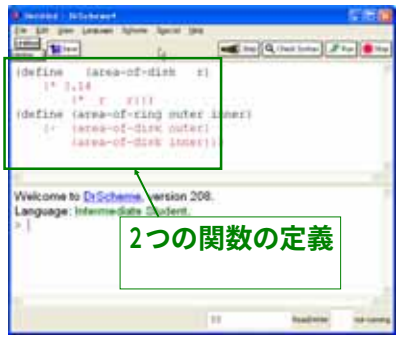
```
(define (area-of-ring outer inner)
   (- (* 3.14 (* outer outer))
      (* 3.14 (* inner inner))))
```

「働き」は同じ ⟺

リングの面積は、「外側の円の面積」－「内側の円の面積」であることがひと目で分かる

Literacy

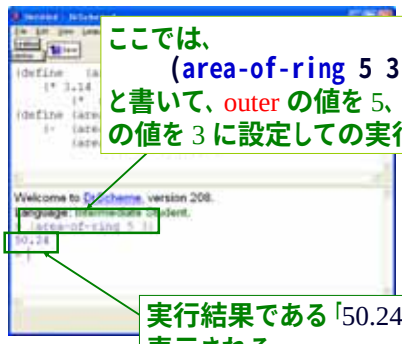
例題3の実行例(1/3)



2つの関数の定義

Literacy

例題3の実行例(2/3)

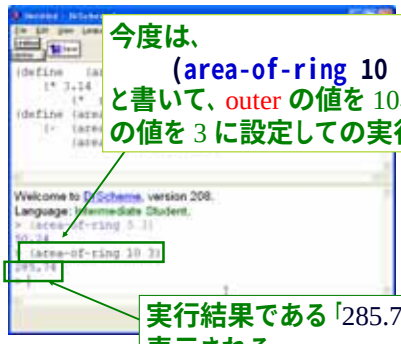


ここでは、
(area-of-ring 5 3)
と書いて、outer の値を 5、inner の値を 3 に設定しての実行

実行結果である「50.24」が表示される

Literacy

例題3の実行例(2/3)



今度は、
(area-of-ring 10 3)
と書いて、outer の値を 10、inner の値を 3 に設定しての実行

実行結果である「285.74」が表示される

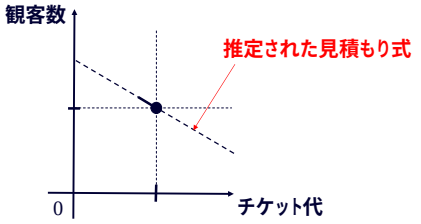
Literacy

例題4．利益の計算

- 公演での利益を求める関数を定義し、実行する
- チケット代 ticket-price から利益、収入、支出、観客数を求める関数 profit、revenue、cost、attendees
  - profit: 利益 ＝ 収入(revenue) － 費用(cost)
  - revenue: 収入 ＝ 観客数(attendees) × チケット代(ticket-price)
  - cost: 支出 ＝ 固定費 ＋ 観客数(attendees) × 費用
    → 「固定費」と「費用」は公演ごとに異なる
  - attendees: チケット代(ticket-price)と観客数には関係がある
    → この「関係」は公演ごとに異なる

まず問題をよく分析して理解する

観客数の見積もり式



■チケット代と観客数には関係がある

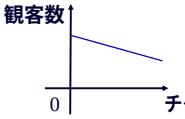
今回） チケット代: ＄5 のとき、観客数は120人だった

チケット代: ＄0.1 値下げすると15人増えた

⇒ 観客数 ＝ －(15／0.1)×(チケット代－＄5) ＋ 120 と見積もる

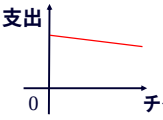
チケット代との関係

観客数のグラフ



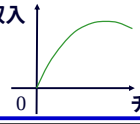
チケット代を上げると、観客数が減る

支出のグラフ



チケット代を上げると、観客数が減り、結果として支出も減る

収入のグラフ

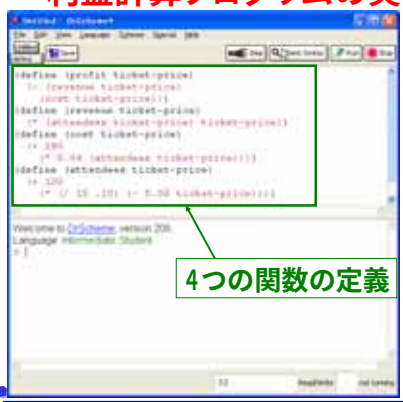


収入は、観客数×チケット代（これは2次曲線）

利益計算のプログラム

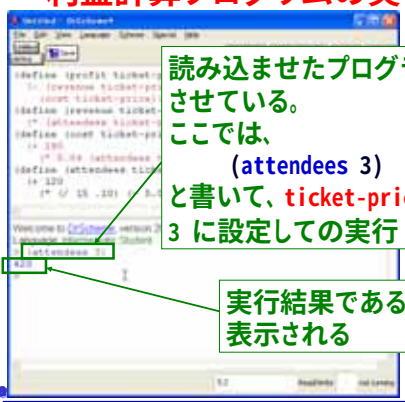
```
;; profit: number -> number
;; to compute the profit as the difference between
;; revenue and costs at some given ticket-price
(define (profit ticket-price)
  (- (revenue ticket-price)
     (cost ticket-price)))

;; revenue:number number -> number
;; to compute the revenue, given ticket-price
(define (revenue ticket-price)
  (* (attendees ticket-price) ticket-price))

;; cost : number -> number
;; to compute the cost, given ticket-price
(define (cost ticket-price)
  (+ 180
     (* .04 (attendees ticket-price))))

;; attendees:number -> number
;; to compute the number of attendees,
;; given ticket-price
(define (attendees ticket-price)
  (+ 120
     (* (/ 15 .10) (- 5.00 ticket-price))))
```

利益計算プログラムの実行例



4つの関数の定義

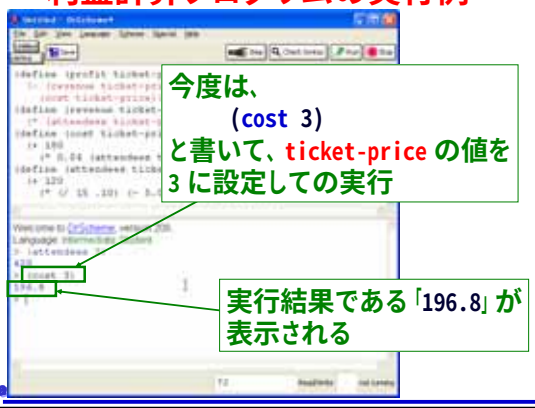
利益計算プログラムの実行例



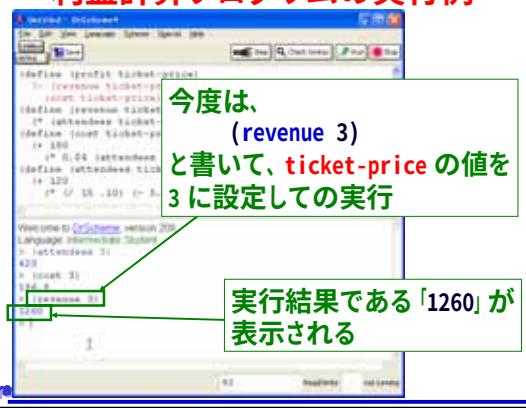
読み込ませたプログラムを実行させている。
ここでは、
(attendees 3)
と書いて、ticket-price の値を 3 に設定しての実行

実行結果である「420」が表示される

利益計算プログラムの実行例



利益計算プログラムの実行例



実習

実習の進め方

- 資料を見ながら、「実習」を行ってみる
- その後、各自、「課題」に挑戦する
  - 各自で自習 ＋ 巡回指導
  - 遠慮なく質問してください
- 自分のペースで先に進んで構いません

DrScheme の使用

- DrScheme の起動
  プログラム → PLT Scheme → DrScheme
- 今日の実習では「Intermediate Student」に設定
  Language
  → Choose Language
  → Intermediate Student
  → OK ボタン

DrScheme の使用 / ステップ実行

- プログラム実行の振る舞いを観察するツール
- 「定義用ウインドウ」のみを使用
  (通常の実行と異なる)
- 「Intermediate Student」に設定する必要あり
  Language → Choose Language
  → Intermediate Student
  → Run ボタン

実習1．2乗の和

- x と y とから、x^2+y^2 を求める関数 sum-of-squares について、実行結果に至る過程を見る
  - (sum-of-squares 20 30) から「1300」に至る過程を見る
  - DrScheme のステップ実行機能を使用

```
(define (square x)
    (* x x))
(define (sum-of-squares x y)
    (+ (square x) (square y)))
```

```
(sum-of-squares 20 30)
= (+ (square 20) (square 30))
= (+ (* 20 20) (square 30))
= (+ 400 (square 30))
= (+ 400 (* 30 30))
= (+ 400 900)
= 1300
```

Literacy

「実習1．2乗の和」の手順

1. 次を「定義用ウインドウ」で、実行しなさい
   - Intermediate Student で実行すること
   - 入力した後に、Run ボタンを押す

```
(define (square x)
    (* x x))
(define (sum-of-squares x y)
    (+ (square x) (square y)))
(sum-of-squares 20 30)
```

ステップ実行したいので、定義用ウインドウだけを使う

2. DrScheme を使って、ステップ実行の様子を確認しなさい （Step ボタン、Next ボタンを使用）
   - 理解しながら進むこと

Literacy

実習2．profit のステップ実行

- 例題4の関数 profit について、実行結果に至る過程を見る
  - (profit 3) から「1063.2」に至る過程を見る
  - DrScheme のステップ実行機能を使用

```
(profit 3)
= (- (revenue 3) (cost 3))
= (- (* (attendees 3) 3) (cost 3))
= (- (* (+ 120 (* (/ 15 0.10) (- 5.00 3))) 3) (cost 3))
= (- (* (+ 120 (* 150 (- 5.00 3))) 3) (cost 3))
= (- (* (+ 120 (* 150 2)) 3) (cost 3))
= (- (* (+ 120 300) 3) (cost 3))
= (- (* 420 3) (cost 3))
= (- 1260 (cost 3))
= (- 1260 (+ 180 (* 0.04 (attendees 3))))
= (- 1260 (+ 180 (* 0.04 (+ 120 (* (/ 15 0.10) (- 5.00 3))))))
= (- 1260 (+ 180 (* 0.04 (+ 120 (* 150 (- 5.00 3))))))
= (- 1260 (+ 180 (* 0.04 (+ 120 (* 150 2)))))
= (- 1260 (+ 180 (* 0.04 (+ 120 300))))
= (- 1260 (+ 180 (* 0.04 420)))
= (- 1260 (+ 180 16.8))
= (- 1260 196.8)
= 1063.2
```

「実習2．profit のステップ実行」の手順

1. 次を「定義用ウインドウ」で、実行しなさい
   - Intermediate Student で実行すること
   - 入力した後に、Run ボタンを押す

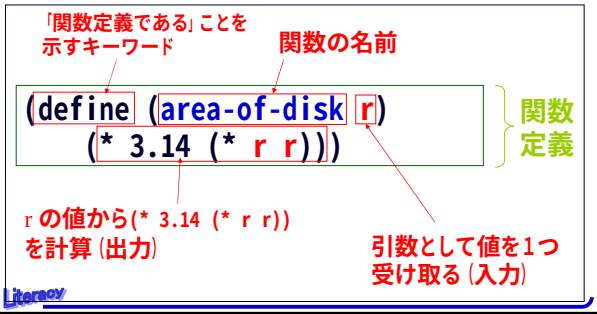
```
;; profit: number -> number
;; to compute the profit as the difference between
;; revenue and costs at some given ticket-price
(define (profit ticket-price)
  (- (revenue ticket-price) (cost ticket-price)))
;; revenue:number number -> number
;; to compute the revenue, given ticket-price
(define (revenue ticket-price)
  (* (attendees ticket-price) ticket-price))
;; cost : number -> number
;; to compute the cost, given ticket-price
(define (cost ticket-price)
  (+ 180 (* .04 (attendees ticket-price))))
;; attendees:number -> number
;; to compute the number of attendees,
;; given ticket-price
(define (attendees ticket-price)
  (+ 120 (* (/ 15 .10) (- 5.00 ticket-price))))
(profit 3)
```

ステップ実行したいので、定義用ウインドウだけを使う

2. DrScheme を使って、ステップ実行の様子を確認しなさい （Step ボタン、Next ボタンを使用

実習3．構文エラー

- 正しく無い構文で関数定義を行ったときの DrScheme の振る舞いを見る



Literacy

「実習3．構文エラー」の手順

1. 次を「定義用ウインドウ」で、実行しなさい
   - 入力した後に、Run ボタンを押す

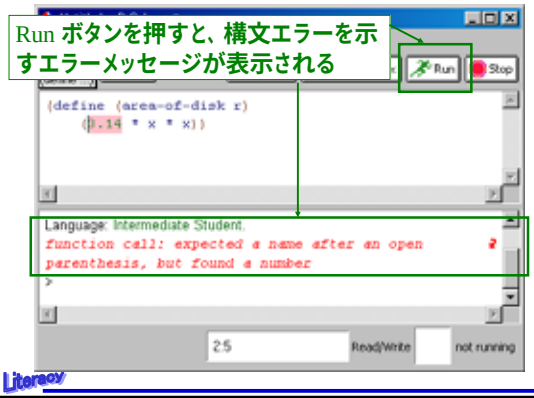
```
(define (area-of-disk r)
(3.14 * x * x))
```

意図的な間違い

Literacy

構文エラーを起こしてみる(1/2)

構文エラーを起こしてみる(2/2)



実習4．実行時エラー

- 意味的に間違っている関数定義での DrScheme の振る舞いを見る
  - 変数名の間違い
  - 関数名の間違い
  - パラメータ数の間違い
  - 0での除算　など

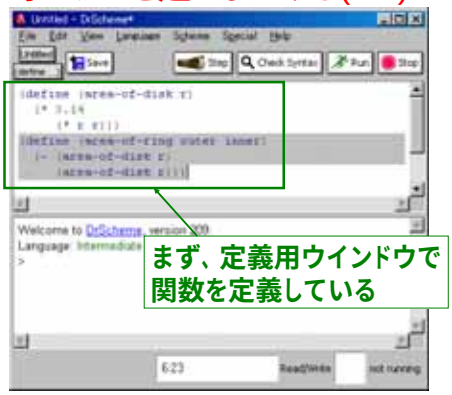
「実習4．実行時エラー」の手順

1. 次を「定義用ウインドウ」で、実行しなさい
   - 入力した後に、Run ボタンを押す

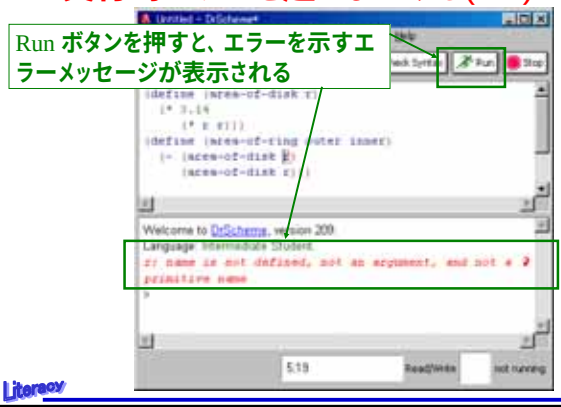
```
(define (area-of-disk r)
  (* 3.14
     (* r r)))
(define (area-of-ring outer inner)
  (- (area-of-disk r)
     (area-of-disk r)))
```

意図的な間違い

実行時エラーを起こしてみる(1/2)



実行時エラーを起こしてみる(1/2)



課題

Literacy

課題1

- 関数 profit (例題4)についての問題
  - 業務内容の変更で固定費が $0 、一人当たりの支出が $1.5 となった。対応するよう例題4の関数を変更しなさい。
  - チケット代を 3、4、5 としたときの関数 profit の実行結果を報告しなさい

Literacy

課題2．構文エラー

- 次の Scheme式のエラーの原因について、分かりやすく説明しなさい
  - 次のScheme式を DrScheme を使って実行すると、赤い文字で、エラーメッセージが表示される

```
1. (10 + 20)

2. (define (g x)
     + x 10)

3. (define h(x)
     (+ x 10))
```

Literacy

課題3．実行時エラー

- 次の Scheme式のエラーの原因について、分かりやすく説明しなさい

```
1. (define (f y) (+ x 10))
   のとき、(f 5)を実行

2. (define (f n) (+ (/ n 3) 2))
   のとき、(f 5 8)を実行
```

Literacy

課題4

1. x 円の硬貨 y 枚の金額を求める関数 coins を定義し、実行結果を報告しなさい

2. 1.で定義した関数 coins を用い、1円硬貨 a 枚、5円硬貨 b 枚、10円硬貨 c 枚、50円硬貨 d 枚、100円硬貨 e 枚、500円硬貨 f 枚の合計を求める関数 total を定義し、実行結果を報告しなさい

Literacy

課題5

- アメリカ合衆国は以下のように、世界標準とは異なる単位制度を使っている：

合衆国		世界標準
1 inch		= 2.54 cm
1 foot	= 12 in.	
1 yard	= 3 ft.	
1 rod	= 5.5 yd.	
1 furlong	= 40 rd.	
1 mile	= 8 fl.	

  - 以下の関数を定義し、実行結果を報告しなさい
    inch->cm, feet->inches, yards->feet, rods->yards, furlongs->rods, miles->furlongs
  - その後、以下の関数を定義し、実行結果を報告しなさい
    feet->cm, yards->cm, rods->inches, miles->feet

Literacy (ポイント：関数の再利用)