Binder

Project Brief for Principles of Programming Languages 2018-2019

Razvan Gabriel Apostu - II A5

Binder

Preview

Hello John! I am writing to you regarding our last conversation. Upon your

request, here is our estimate for the entire platform, including strategy,

development and design.

The estimate for the implementation of the requested features is **12000$**.

Over a period of **24** months the operational cost would be **345$**. This

should cover every common aquisitions such as the Google Play and the

Apple Dev accounts or the server.

Editor

▷ Bind

define_defaults

my_duration 24

save as predefined?

define_content

"Hello John! I am writing to you regarding our last conversation. Upon your

request, here is our estimate for the entire platform, including strategy,

development and design.

The estimate for the implementation of the requested features is"

do sum on features.price

Over a period of " my_duration " months, the operational cost would be"

do sum on (do times between common.price, (do times between at_least

common.frequency, my_duration))

"This should cover every common aquisition such as the Google Play and the

Apple Dev accounts or the server."

Predefined Actions and Defaults

action times (a,b)
↳ do * between [[a]], [[b]]

action sum (a,b)
↳ do + between [[a]], [[b]]

default common_aquisition_price
↳ do sum on (do times between common.price, (do times between at_least common.frequency, 24))

Binder

Preview

Hello John! I am writing to you regarding our last conversation. Upon your request, here is our estimate for the entire platform, including strategy, development and design.

The estimate for the implementation of the requested features is **12000$**.

Over a period of **24** months the operational cost would be **345$**. This should cover every common aquisitions such as the Google Play and the Apple Dev accounts or the server.

Tables

name: **common**

#	name	frequency	price
1	Apple Developer	12	100$
2	Android Developer	999999999999	25$
3	Shared Hosting	1	25$

name: **support**

#	name	frequency	price
1	Bug Fixing	1	100$
2	Feature Ideation	1	200$

name: **features**

#	name	price
1	Web Design	3000$
2	Web Development	4000$
3	React Native App	5000$

Binder

Data should be stored inside tables on a certain instance of a DB.

Preview

Hello John! I am writing to you

request, here is our estimate for the entire platform, including strategy,

development and design.

The estimate for the implementation of the requested features is **12000$**.

Over a period of **24** months the operational cost would be **345$**. This

n aquisitions such as the Google Play and the

server.

Simple strings, identified between quotation marks " ..." will be directly moved into the edited preview. The dynamic values will be calculated and the result moved into the preview.

Some default values can be declared inline. Those won't be available on multiple projects.

Editor

Bind

define_defaults

my_duration 24

save as predefined?

define_content

"Hello John! I am writing to you re

request, here is our estimate for t

development and design.

The estimate for the implementation of the requested features is"

do *sum* on *features.price*

Over a period of " my_duration " months, the operational cost would be"

do *sum* on (do *times* between *common.price*, (do *times* between *at_least*

***common.frequency,* my_duration))**

"This should cover every common aquisition such as the Google Play and the

Apple Dev accounts or the server."

Keywords:

do - defines the start of an action set
between, on - defines the data-set used by the action
at_least - special keyword *at_least a |* (a > result ? a : result)

2 types of high-level components
a) actions
b) default values

a) an action is a function that will implement a JS algorithm based on the keywords and parameters used. For example, the user can define "times" as being another way to call * = multiplication (* being defined as a default platform sign for multiplication). For **between, on** check the Keywords section.

An action should contain parameters, keywords and **one operation** than needs to be a binary action in order to happen between 2 items at a time.
+ inside (e.g. ((a+b)+c)

b) a default value will implement some actions over predefined columns and will return a static result. While actions can return arrays(see *between*), **default** will always return static single values. *//for now*

Predefined Actions and Defaults

action times (a,b)
↳ do * between [[a]], [[b]]

action sum (a,b)
↳ do + between [[a]], [[b]]

default common_aquisition_price
↳ do sum on (do times between common.price, (do times between at_least common.frequency, 24))

Keywords $\boxed{\text{between}}$

words with restricted access, used only by the "Binder" language

Data Set: Database Row

Extra Data: A **list** of *selected* columns and an ***n*-binary** action/operation (before) to take place between the items
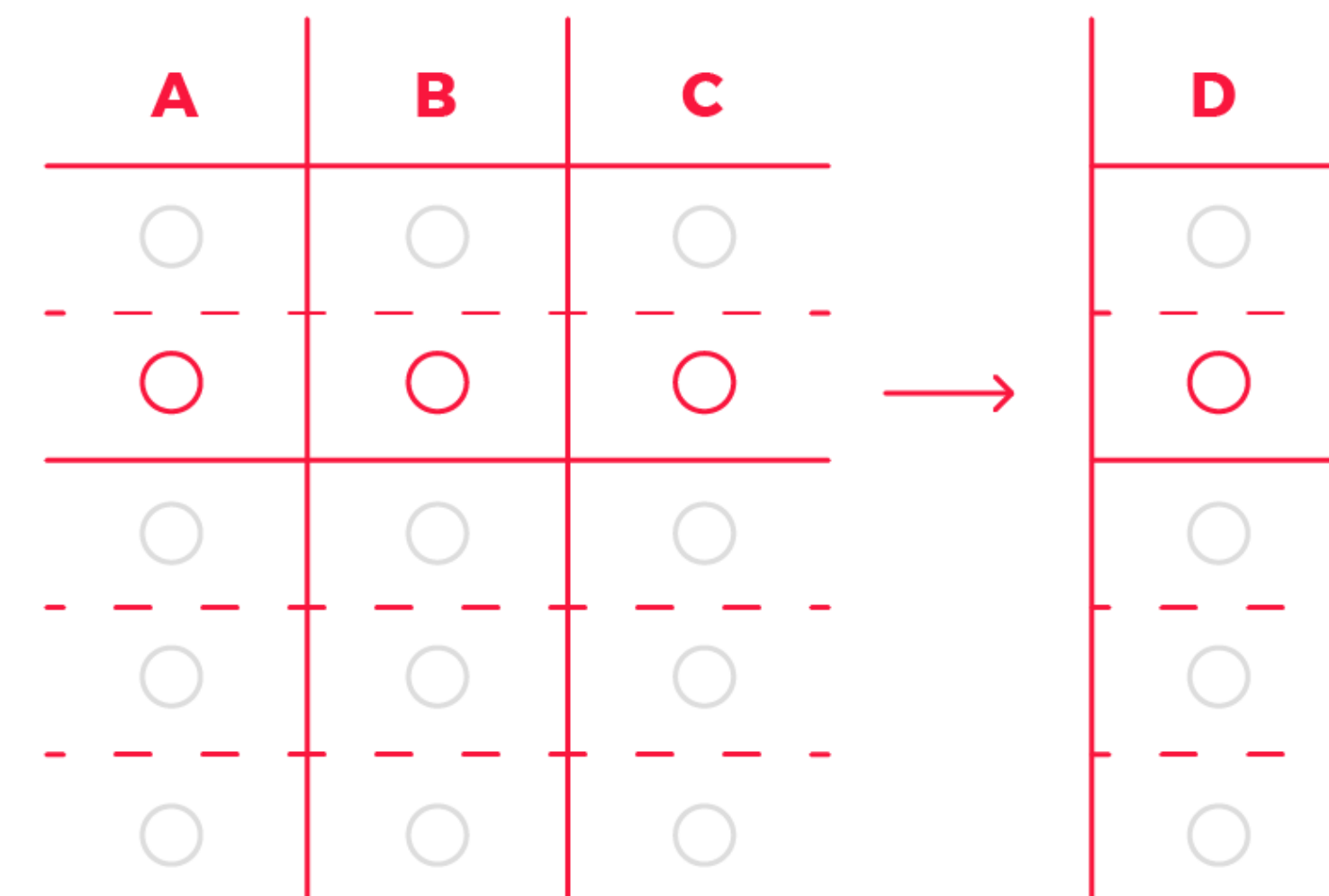
Result: Delivers a new data set consisting of a single value **array**

Syntax: **do *action* between** column_A , column_B , ... , column_Z

Effect (~JS):

```
/* @param {Array.<Array>} columns
   @returns {Array}
*/
function(columns){
    for(let i = 0; i < nr_rows; i++){ //for each row
        row_resultⱼ = columnsᵢ₀; //safety measure
        for(let j = 1; j < nr_cols; j++){ //for each col-item, apply action gradually
            row_resultⱼ = action(row_resultⱼ , columnsᵢⱼ);
        }
        resultᵢ = row_resultⱼ;
    }
    return result;
}
```

Visual Representation (DB)

Example

do *times* between support.price, my_duration

Keywords `on`

words with restricted access, used only by the "Binder" language

Data Set: Database Column / Array

Extra Data: A **list** that can either be a column or a single-valued array and an **n-ary** action/operation (before) to take place between each item/row

Result: Delivers a new data set consisting of a single **value**

Syntax: **do** *action* **on** column$_A$

Effect (~JS):

```
/* @param {Array} column
   @returns {number|string}
*/
function(column){
    result = column₀ ; //safety measure
    for(let i = 1; i < nr_cols; i++){ //for each col-item, apply action gradually
        result = action(row_result_j , columns_ij);
    }

    return result;
}
```

Visual Representation (DB)

Example

do sum *on* features.price

Keywords

do   at_least   tableify

words with restricted access, used only by the "Binder" language

---

do
Info: Defines the start of a new action

at_least
Info: special keyword **at_least a** | (a > result ? a : result)

tableify
Info: Given a group of columns, it should print them.
Case: The columns might be default ones (directly from DB) or might be afected by a **between** action

Development Plan

development order for features (may change)

1. Environment and structure

2. Database support

3. Editor strings and inline defaults (define_defaults)

4. Predefined actions

    4.1. Binary actions (sum, times)

    4.2. Default actions with static columns (pre-work for 5)

5. Defining actions and important keywords (between, on, ...)

Binder

Project Brief for Principles of Programming Languages 2018-2019

Razvan Gabriel Apostu - II A5