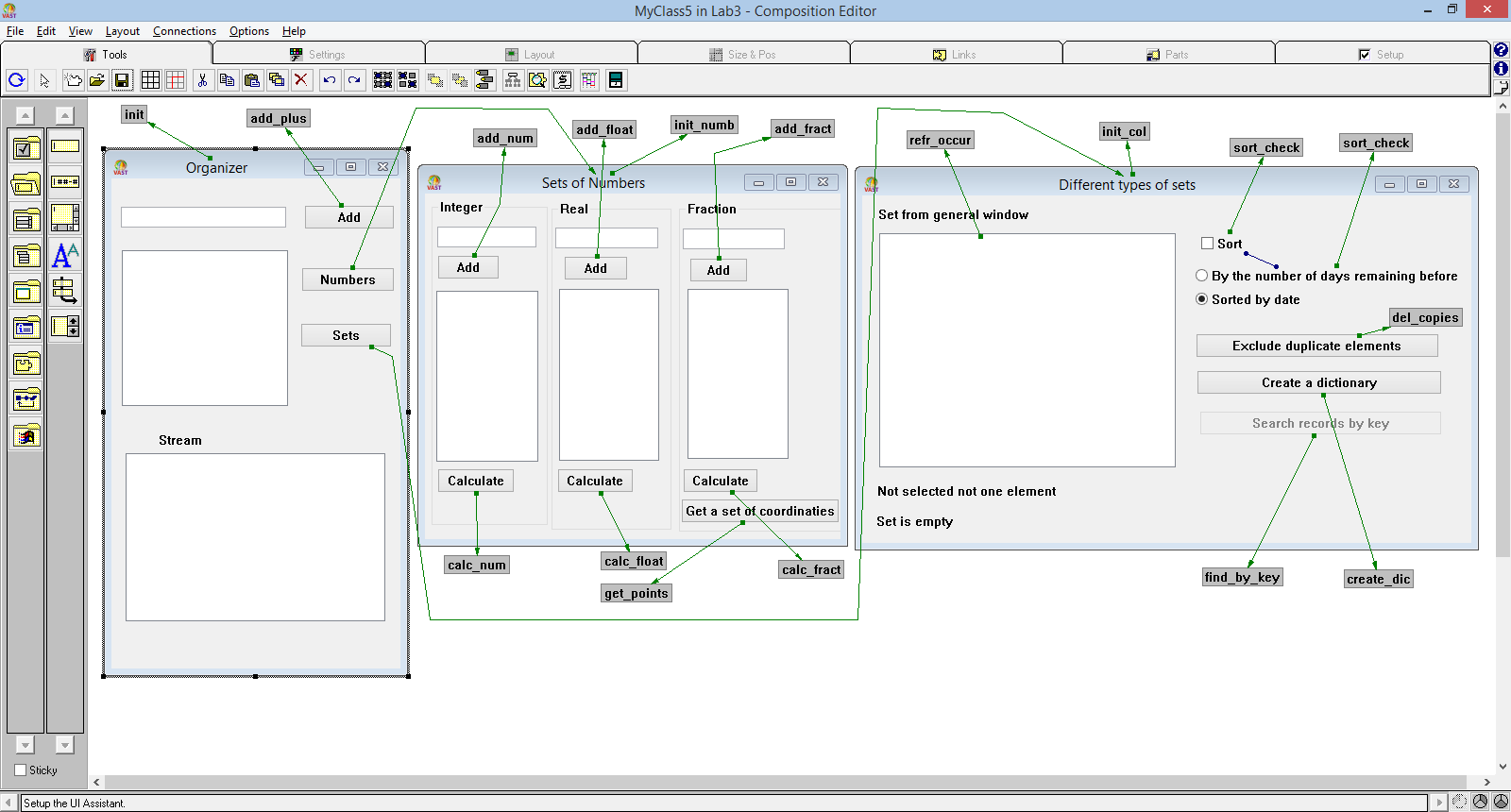
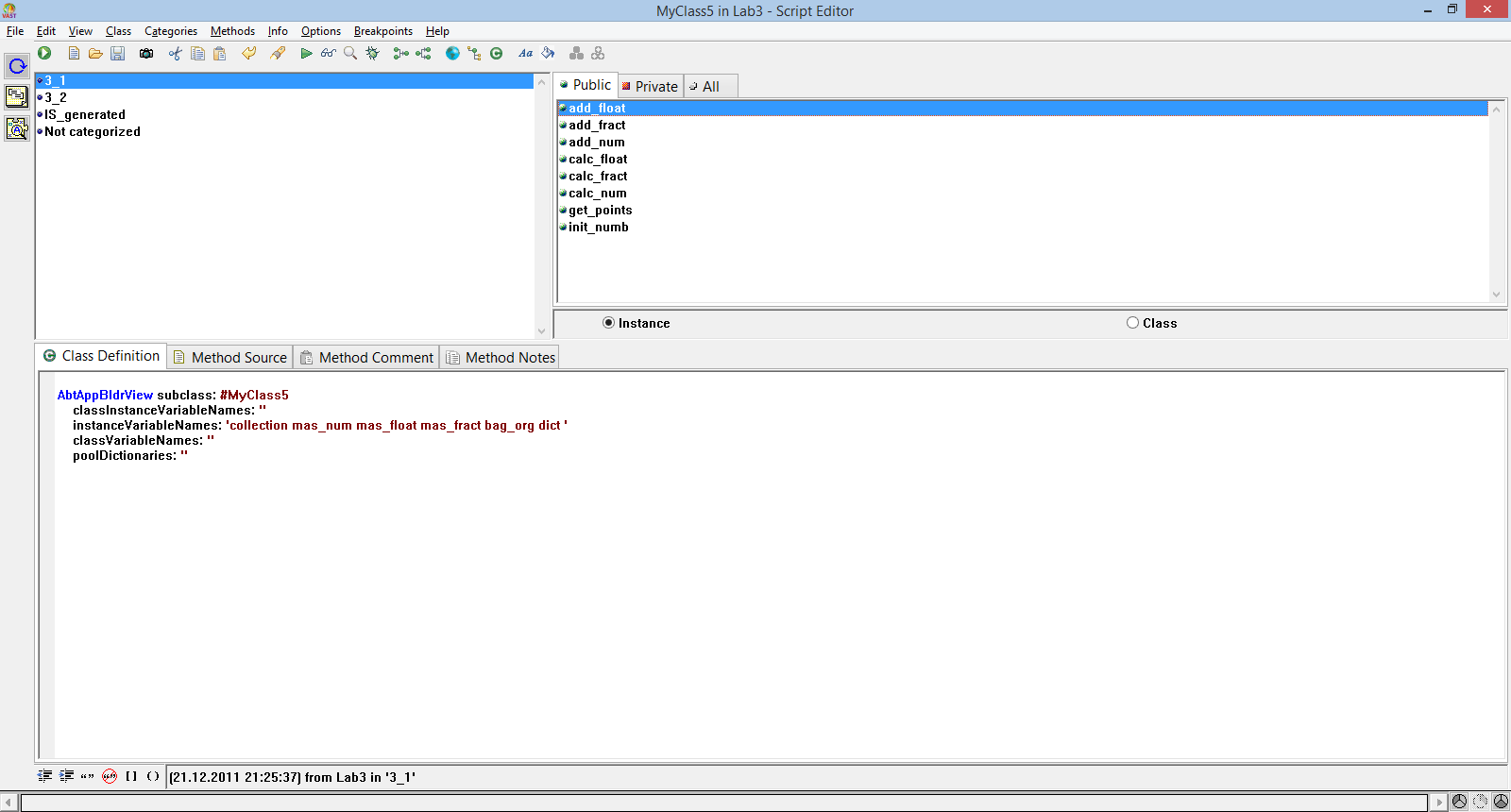
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | для прик эмбл | | | |  |
| МИНОБРНАУКИ РОССИИ | | | | | | |
| Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования"Московский государственный технический университет радиотехники,электроники и автоматики"МГТУ МИРЭА | | | | | | |
| Факультет Информационных Технологий Кафедра МОСИТ Лабораторная работа №3 | | | | | | |
|  | | | | | | |
| по дисциплине«Объектно-ориентированное программирование» | | | | | | |
| Выполнил: | | | | | | |
| Студент группы ИКБО-02-13 | | | Силаева А.Э. | | | |
|  | | |  | | | |
| Преподаватель | | | Смольянинова В.А, | | | |
|  | |  | |  | | |

# Москва 2015

Composite editor



Script editor



***Код программы***

AbtAppBldrView subclass: #MyClass5

classInstanceVariableNames: ''

instanceVariableNames: 'collection mas\_num mas\_float mas\_fract bag\_org dict '

classVariableNames: ''

poolDictionaries: ''

***init\_numb***

mas\_num := OrderedCollection new.

mas\_float := OrderedCollection new.

mas\_fract := OrderedCollection new.

***add\_float***

|element|

element:=(self subpartNamed: 'Text3') object .

mas\_float add: element asDecimal.

(self subpartNamed: 'List3') items: mas\_float;

selectedItem: element.

***add\_fract***

|element divi|

element:=(self subpartNamed: 'Text4') object .

(element indexOf: $/) == 0

ifTrue: [ element := Fraction numerator: element asInteger denominator: 1.]

ifFalse: [ element := Fraction numerator: ((element copyFrom:1 to: (element indexOf: $/)) asDecimal asInteger) denominator: ((element copyAfter: $/) asDecimal asInteger). ].

mas\_fract add: element.

(self subpartNamed: 'List4') items: mas\_fract;

selectedItem: element.

***add\_num***

|element|

element:=(self subpartNamed: 'Text2') object .

mas\_num add: (element asDecimal asInteger).

(self subpartNamed: 'List2') items: mas\_num;

selectedItem: element.

***calc\_float***

“отсечь дробную часть от всех чисел в наборе”

|mas2|

mas2 := OrderedCollection new.

mas\_float do: [ :element | mas2 add: element truncated. ].

mas\_float := mas2.

(self subpartNamed: 'List3') items: mas\_float.

***calc\_fract***

“оставить в наборе только отрицательные элементы”

mas\_fract := mas\_fract select: [ :element | element negative. ].

(self subpartNamed: 'List4') items: mas\_fract.

***calc\_num***

“сменить знак чисел в наборе на противоположный”

|mas2|

mas2 := OrderedCollection new.

mas\_num do: [ :element | mas2 add: element\*(-1). ].

mas\_num := mas2.

(self subpartNamed: 'List2') items: mas\_num.

***get\_points***

“получить набор точек, взяв в качестве координат числитель и знаменатель дробей из набора”

(self subpartNamed: 'Push Button8') enabled: false. “точки уже нельзя проверить на отрицательность”

mas\_fract := mas\_fract collect: [ :element | element numerator @ element denominator ].

(self subpartNamed: 'List4') items: mas\_fract.

***init***

collection:=OrderedCollection new.

collection1:=OrderedCollection new.

***add\_plus***

|str newstr f d dt stream stream1 stream2|

str:=(self subpartNamed: 'Text1') object .

newstr := ''.

stream:= ReadStream on: str.

stream1:= ReadStream on: newstr.

stream2:= WriteStream on: String new.

[stream1 atEnd]

whileFalse: [stream2 nextPut: stream1 next].

stream2 space;

nextPutAll: Date today asString, ' , ', Time now asString,' | '.

dt := Date stsDateArrayPrim: (CwTextPrompter prompt: 'Enter the date of the event(numbers or month in English) :') twoDigitYear: 40.

d := Date newDay: (dt at: 1) monthIndex: (dt at: 2) year: (dt at: 3) .

f := true.

str do: [

:ch |

(ch = 32 asCharacter)

ifTrue: [

f ifFalse: [

newstr := newstr, ch asString.

f := true.

]

]

ifFalse: [

newstr := newstr, ch asString.

f := false.

]

].

[stream1 atEnd]

whileFalse: [stream2 nextPut: stream1 next].

stream2 space;

nextPutAll: '(', d asString, ')' .

dt := 'Remaining days: ', (d daysFromBaseDay - Date today daysFromBaseDay) asString.

collection1 add: 'stream size => ' , stream size asString.

collection1 add: 'stream copyFrom: 2 to: 5 => ' , (stream copyFrom: 2 to: 5) asString.

collection1 add: 'stream isEmpty => ' , stream isEmpty asString.

collection1 add: 'stream position => ' , stream position asString.

collection1 add: 'stream peek => ' , stream peek asString.

collection1 add: 'stream contents => ' , stream contents asString.

collection add: stream2 contents, ' >> ', dt.

(self subpartNamed: 'List6') items: collection1;

selectedItem: str.

(self subpartNamed: 'List1') items: collection;

selectedItem:newstr.

***init\_col***

“инициализация окна работы с наборами”

bag\_org := collection asBag. “преобразовали коллекцию строк органайзера в набор с дублями”

bag\_org addAll: collection. “добавили те же элементы еще раз, чтобы протестировать сообщение добавления а заодно создать дубли элементов набора”

bag\_org addAll: (collection select: [ :element | (collection indexOf: element) odd.]). “и еще добавили элементы с нечетными индексами”

(self subpartNamed: 'List5') items: bag\_org asOrderedCollection. "связали набор со списковой рамкой, используя сообщение преобразования типа набора"

self col\_size.

(self subpartNamed: 'Push Button13') enabled: false.

***col\_size***

“вывести число элементов набора”

|str|

bag\_org isEmpty ifTrue: [ str := 'Set is empty.' ] ifFalse: [str := 'Set is not empty.'].

(self subpartNamed: 'Label2') object: 'The numbers of elements in the set ', bag\_org size asString, ' - ', str.

***create\_dic***

“создать словарь на основе имеющегося набора. ключ - дата, на которую было добавлено событие”

|key dat dict\_el|

dict := Dictionary new.

bag\_org asSet do: [ :element |

dat := element copyAfterLast: $(.

dat := dat copyFrom: 1 to: (dat indexOf: $)) - 1.

dat := Date stsDateArrayPrim: dat twoDigitYear: 40.

key := Date newDay: (dat at: 1) monthIndex: (dat at: 2) year: (dat at: 3).

dict\_el := Association new key: key value: element.

dict add: dict\_el.

].

CwMessagePrompter message: 'Dictionary created. Used as the key date of the event' title: 'Create a dictionary'.

(self subpartNamed: 'Push Button13') enabled: true.

***del\_copies***

“сохранить все записи в единственном экземпляре”

bag\_org := bag\_org asSet.

(self subpartNamed: 'List5') items: bag\_org asOrderedCollection.

self col\_size.

***find\_by\_key***

“поиск записи в словаре по ключу”

|key element|

key := CwTextPrompter prompt: 'Enter the date of the event(numbers or month in English):'.

key := Date stsDateArrayPrim: key twoDigitYear: 40.

key := Date newDay: (key at: 1) monthIndex: (key at: 2) year: (key at: 3).

element := dict at: key ifAbsent: 'No record with this key. You need to re-create the dictionary'.

CwMessagePrompter message: element title: 'Search record in the dictionary'.

***refr\_occur***

“обновить число вхождений выбранного элемента набора”

|str|

(self subpartNamed: 'List5') selectionIndex == 0 ifTrue: [ str := 'Not selected not one element.' ] ifFalse: [ str := 'The numbers of occurrences of the selected item in the set: ', (bag\_org occurrencesOf: (self subpartNamed: 'List5') selectedItem) asString].

(self subpartNamed: 'Label3') object: str.

***sort\_check***

"включение/выключение сортировки"

|sort\_block d1 d2 t1 t2|

(self subpartNamed: 'Toggle Button1') selection ifTrue: [

(self subpartNamed: 'Radio Button Set1') selectionIndex == 1

ifTrue: [ sort\_block := [ :a :b | (a copyAfterLast: $ ) asDecimal asInteger <= (b copyAfterLast: $ ) asDecimal asInteger]. ]

ifFalse: [ sort\_block := [ :a :b |

d1 := Date stsDateArrayPrim: (a copyFrom: 1 to: 8) twoDigitYear: 40.

d2 := Date stsDateArrayPrim: (b copyFrom: 1 to: 8) twoDigitYear: 40.

t1 := a copyFrom: (a indexOf: $,)+1 to: (a indexOf: $|)-1.

t2 := b copyFrom: (b indexOf: $,)+1 to: (b indexOf: $|)-1.

d1 := Date newDay: (d1 at: 1) monthIndex: (d1 at: 2) year: (d1 at: 3).

d2 := Date newDay: (d2 at: 1) monthIndex: (d2 at: 2) year: (d2 at: 3).

t1 := t1 allSubStrings: $:.

t2 := t2 allSubStrings: $:.

t1 := Time new hours: (t1 at: 1) asDecimal asInteger minutes: (t1 at: 2) asDecimal asInteger seconds: (t1 at: 3) asDecimal asInteger.

t2 := Time new hours: (t2 at: 1) asDecimal asInteger minutes: (t2 at: 2) asDecimal asInteger seconds: (t2 at: 3) asDecimal asInteger.

(d1 <= d2) & (t1 <= t2). ].

].

bag\_org := bag\_org asSortedCollection: sort\_block. ]

ifFalse: [

bag\_org := bag\_org asBag.

].

(self subpartNamed: 'List5') items: bag\_org asOrderedCollection.

Пример работы программы

