Оглавление

[Этап 1) Постановка задачи 2](#_Toc378199833)

[Этап 2) Концепции и анализ предметной области 6](#_Toc378199834)

[Этап 3) Объектная модель 8](#_Toc378199835)

[Этап 4) Реализация модели на языке Smalltalk 14](#_Toc378199836)

[Список литературы 43](#_Toc378199837)

# Этап 1) Постановка задачи

**Учебно-методический комплекс (УМК)** дисциплины — стандартное название для совокупности учебно-методической документации, средств обучения и контроля, разрабатываемых в высшей школе Российской Федерации для каждой дисциплины. УМК должен включать полную информацию, достаточную для прохождения дисциплины.

УМК по отдельным дисциплинам составляют основную образовательную программу (ООП) специальности (направления подготовки).

Структура УМКД в высших учебных заведениях составляется по стандарту, написанному и утверждённому в данном заведении.

УМКД содержит:

* Титульный лист
* Аннотацию
* Рабочую программу дисциплины
* Методические указания для преподавателей
* Методические указания для студентов
* Лист изменения УМК
* Приложения

*Титульный лист УМКД*

Титульный лист содержит:

* Полное и краткое название Университета
* Индекс и наименование дисциплины в соответствии с ФГОС и учебным планом
* Индекс и наименование направления подготовки в соответствии с Перечнем направлений подготовки и специальностей высшего профессионального образования.
* Полное и краткое наименование факультета
* Форма обучения
* Полное и краткое название кафедры, составившей УМКД и реализующей дисциплину
* Выписку из учебного плана по направлению

*Аннотация*

Аннотация содержит:

* Индекс и наименование дисциплины
* Индекс и наименование направления подготовки или специальности
* Название документов, на основе которых разработан УМКД (уч. План и ФГОС)
* Содержание УМКД с кратким описанием разделов:
* Описание нагрузки преподавателя м студентов (в часах)
* Цели изучения дисциплины
* Задачи изучения дисциплины
* Навыки, необходимые для успешного освоения дисциплины

*Рабочая программа дисциплины*

Рабочая программа содержит:

* Титульный лист
  + Полное и краткое название Университета
  + Индекс и наименование дисциплины
  + Индекс и наименование направления подготовки или специальности
  + Квалификация (степень) выпускника
  + Форма обучения
* Цели и задачи изучения дисциплины
* Место дисциплины в структуре ООП бакалавриата
* Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля)
* Содержание дисциплины
  + Общая трудоёмкость дисциплины (в часах)
  + Наименование и содержание разделов дисциплины (модуля)
  + Разделы дисциплины (модуля), виды учебной работы и аттестации
  + Лабораторные работы (темы)
* Образовательные технологии
* Учебно – методическое обеспечение самостоятельной работы обучающихся. Оценочные средства текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины
  + Порядок оформления отчетов к лабораторным работам
  + Вопросы к коллоквиуму
  + Тематика курсовых работ (проектов)
* Учебно – методические материалы по дисциплине
  + Основная литература
  + Дополнительная литература
* Материально – техническое обеспечение учебного процесса

*Методические указания по дисциплине для преподавателей*

Методические материалы по дисциплине для преподавателя содержат:

* Индекс и наименование дисциплины
* Индекс и наименование направления подготовки (специальности)
* Методические рекомендации по чтению лекций
* Методические рекомендации по проведению лабораторных работ
* Методические рекомендации по руководству курсовым проектом (или курсовой работой)
* Билеты к экзамену
* Вопросы к зачету

*Методические указания по дисциплине для студентов*

Методические материалы по дисциплине для студентов содержат:

* Индекс и наименование дисциплины
* Индекс и наименование направления подготовки (специальности)
* Методические рекомендации по подготовке к лекциям
* Методические рекомендации по выполнению лабораторных работ
* Методические рекомендации по выполнению курсового проекта (или курсовой работы)

*Лист изменения УМК*

Лист изменения УМК содержит:

* Индекс и наименование дисциплины
* Таблицу, содержащую
  + Дату внесения изменений
  + Номер пункта рабочей программы УМКД, в который внесены изменения
  + Содержание изменений

# Этап 2) Концепции и анализ предметной области

1. Множество D – множество понятий, отражающих различные действия объектов предметной области (разные решения, принимаемые в данной области)
2. Множество Q – множество понятий, образующих объекты предметной области. Туда входят и абстрактные и конкретные понятия.
3. Множество P – множество понятий, отражающих свойства, признаки или различные состояния объектов предметной области.
4. Множество R – множество понятий, отражающих отношения между объектами предметной области.
5. Множество V – множество понятий, соответствующих значениям свойств и отношений.

Q = {Преподаватель, УМКД, титульный лист УМКД, аннотацию, рабочую программу дисциплины, методические указания для преподавателей, методические указания для студентов, лист изменения УМК, приложения, выписка из учебного плана, компетенции, ФГОС}

D(преподаватель) =

{

подготовить(УМКД)

запросить(выписка из учебного плана)

заполнить (титульный лист)

написать (аннотация)

определить(цели)

определить(задачи)

написать (рабочая программа)

выбрать (компетенции)

придумать (лабораторные работы)

придумать (вопросы к коллоквиуму)

придумать (темы курсовых)

составить (список литературы)

написать (методические указания для преподавателей)

написать (методические указания по дисциплине для студентов)

придумать (вопросы к зачету)

придумать (билеты к экзамену)

}

P(преподаватель) = {Работает}

P(УМКД) = {Готов}

P(Титульный лист) = {заполнен}

P(Аннотация) = {Написана}

P(Рабочая программа) = {Написана}

P(Методические указания для преподавателей) = {Написаны}

P(Методические указания для студентов) = {Написаны}

P(выписка из учебного плана) = {Получена}

P(цели изучения дисциплины) = {Определены}

P(задачи изучения дисциплины) = {Определены}

P(компетенции) = {Выбраны}

P(лабораторные работы) = {Придуманы}

P(вопросы к коллоквиуму) = {Придуманы}

P(вопросы к зачету) = {Придуманы}

P(темы курсовых) = {Придуманы}

P(билеты к экзамену) = {Придуманы}

P(список литературы) = {Составлен}

R(Титульный лист) = {в составе (УМКД)}

R(Аннотация) = { в составе (УМКД)}

R(Рабочая программа) = { в составе (УМКД)}

R(Методические указания для преподавателей) = { в составе (УМКД)}

R(Методические указания для студентов) = { в составе (УМКД)}

R(выписка из учебного плана) = {есть у (Преподаватель)}

R(цели изучения дисциплины) = {в составе (рабочая программа)}

R(задачи изучения дисциплины) = {в составе (рабочая программа)}

R(компетенции) = { в составе (рабочая программа)}

R(лабораторные работы) = { в составе (рабочая программа)}

R(вопросы к коллоквиуму) = { в составе (рабочая программа)}

R(вопросы к зачету) = { в составе (Методические указания для преподавателей)}

R(темы курсовых) = { в составе (рабочая программа)}

R(билеты к экзамену) = { в составе (Методические указания для преподавателей)}

R(список литературы) = { в составе (рабочая программа)}

# Этап 3) Объектная модель

Для реализации модели были выбраны следующие классы:

|  |  |
| --- | --- |
| Человек | |
| Торт  Тесто  Ингредиенты | Приготовить торт  Приготовить тесто  Перемешать ингредиенты  Опустить миксер  Включить миксер  Вынуть миксер  Выключить миксер  Выпекать тесто  Включить духовку  Положить тесто  Подождать духовку  Выключить духовку  Вытащить тесто  Нарезать украшения  Украсить торт |
| Нож  Миска  Доска  Духовка  Миксер |

|  |  |
| --- | --- |
| Торт | |
| Готов |  |
|  |

|  |  |
| --- | --- |
| Тесто | |
| Готово |  |
| Для торта  Положено в духовку |

|  |  |
| --- | --- |
| Украшения | |
|  |  |
| На доске |

|  |  |
| --- | --- |
| Рабочая программа | |
| Готова  Цели изучения дисциплины  Задачи изучения дисциплины  Компетенции  Лабораторные работы  Вопросы к коллоквиуму  Темы курсовых  Список литературы |  |
| В составе УМКД |

*Концептуальные структуры действия:*







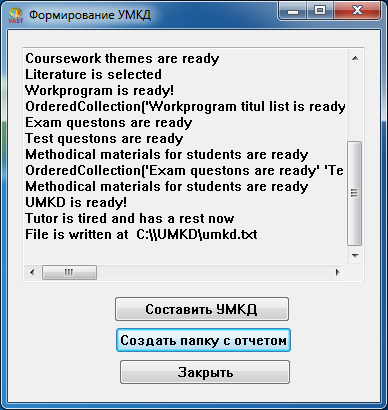


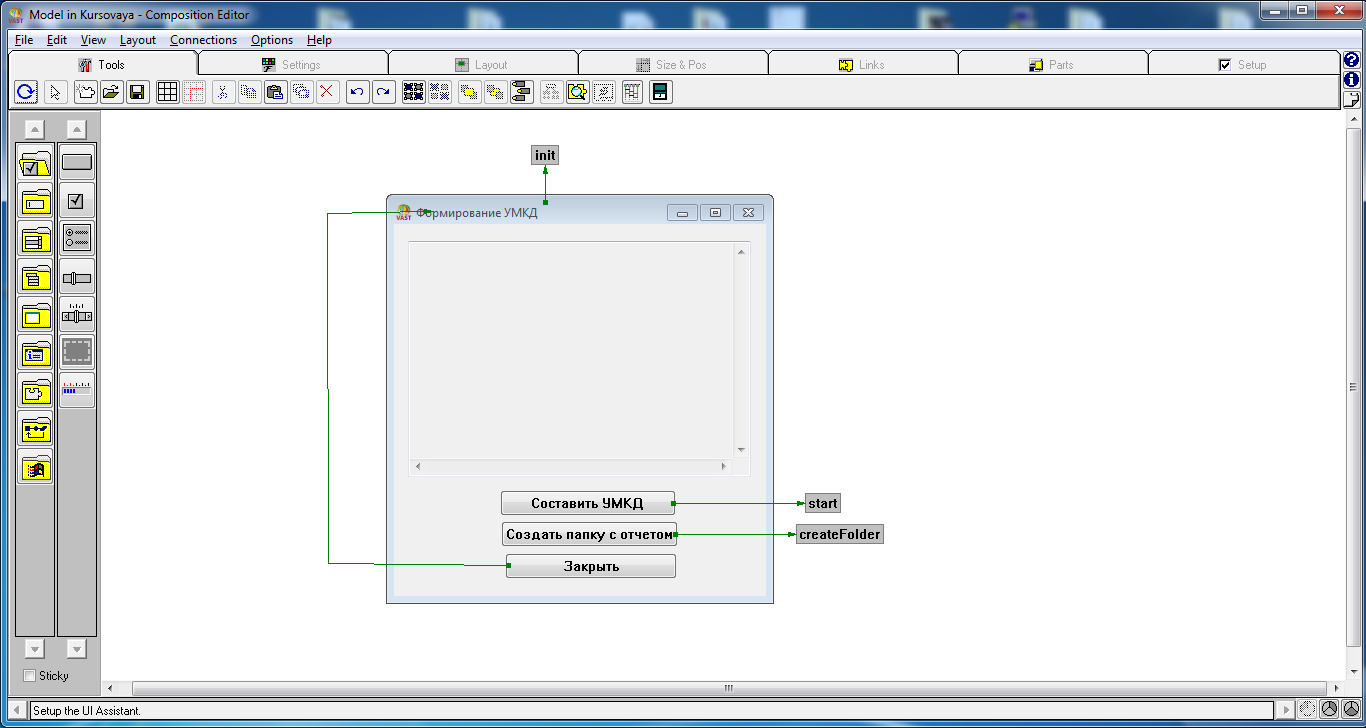


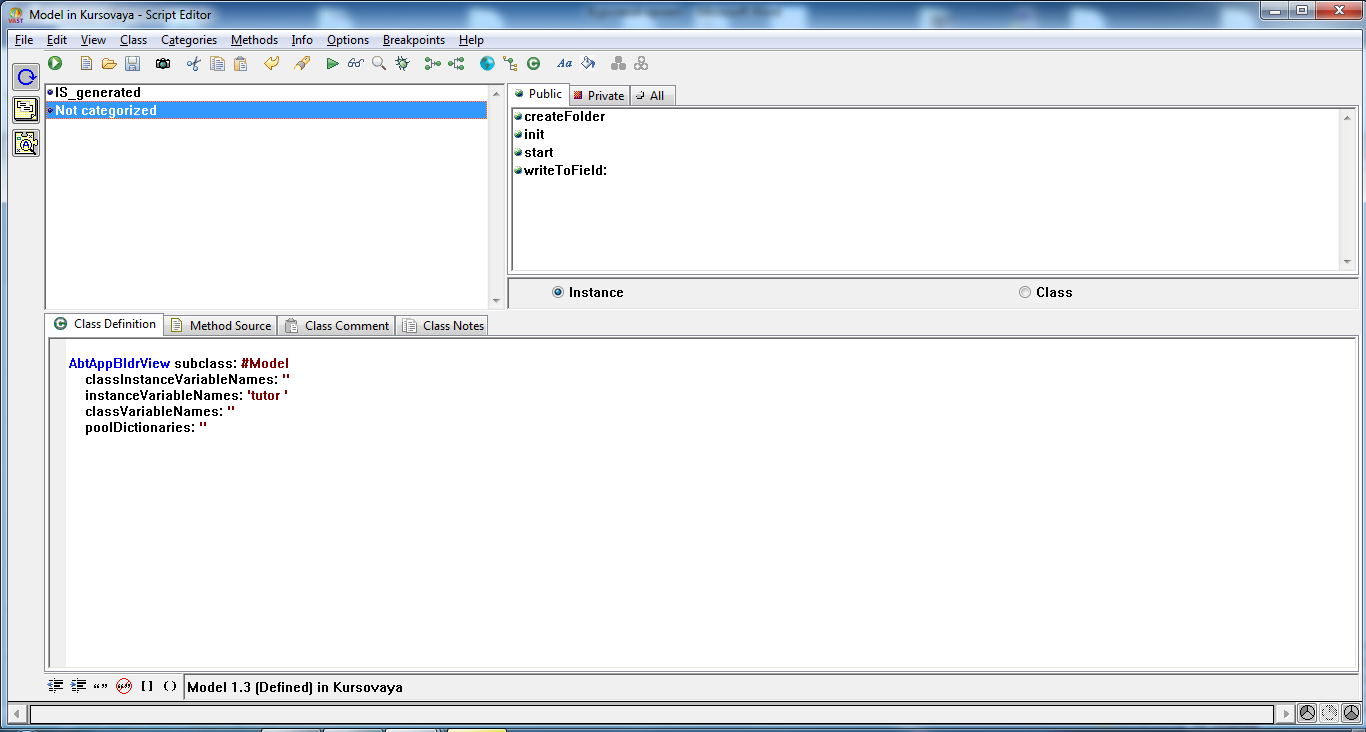


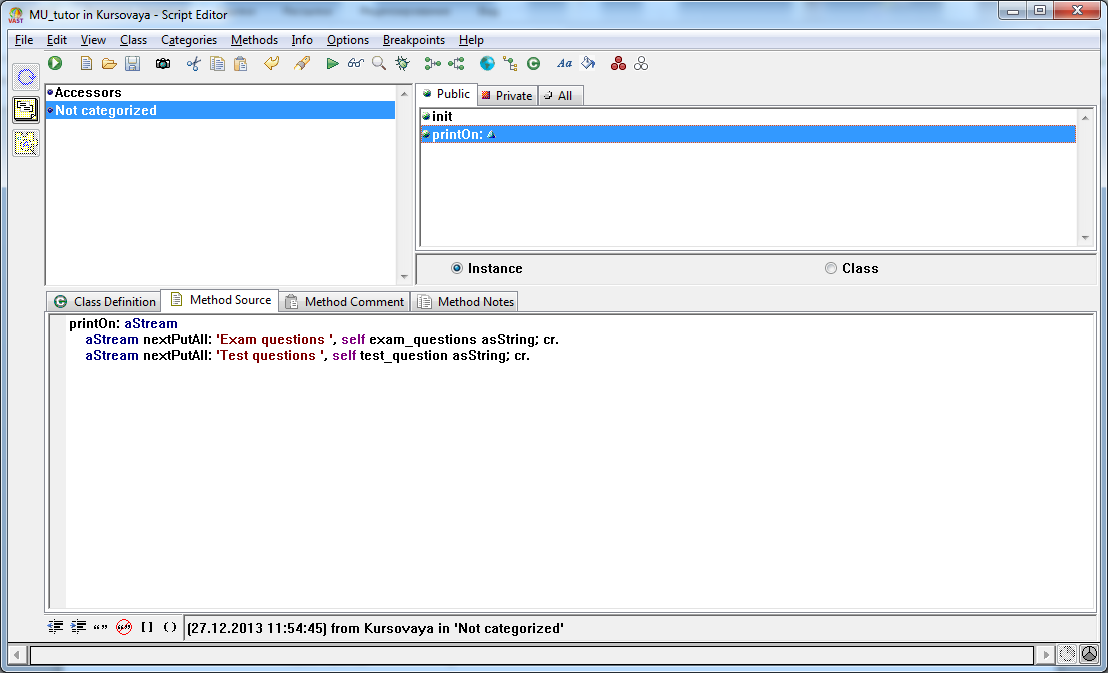


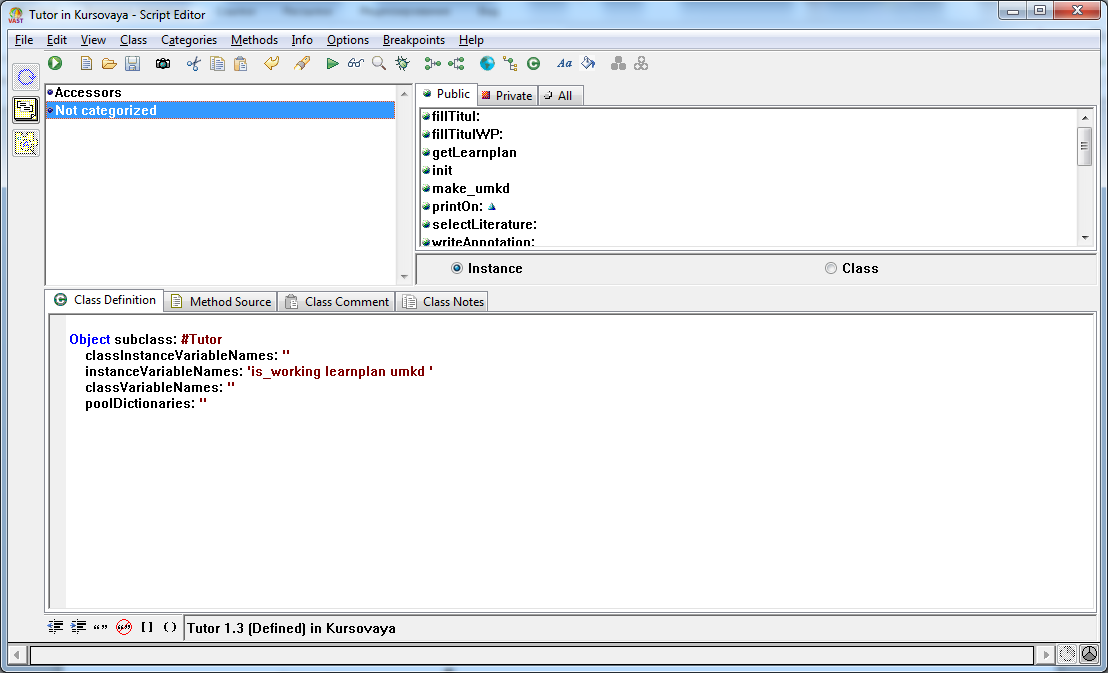
# Этап 4) Реализация модели на языке Smalltalk

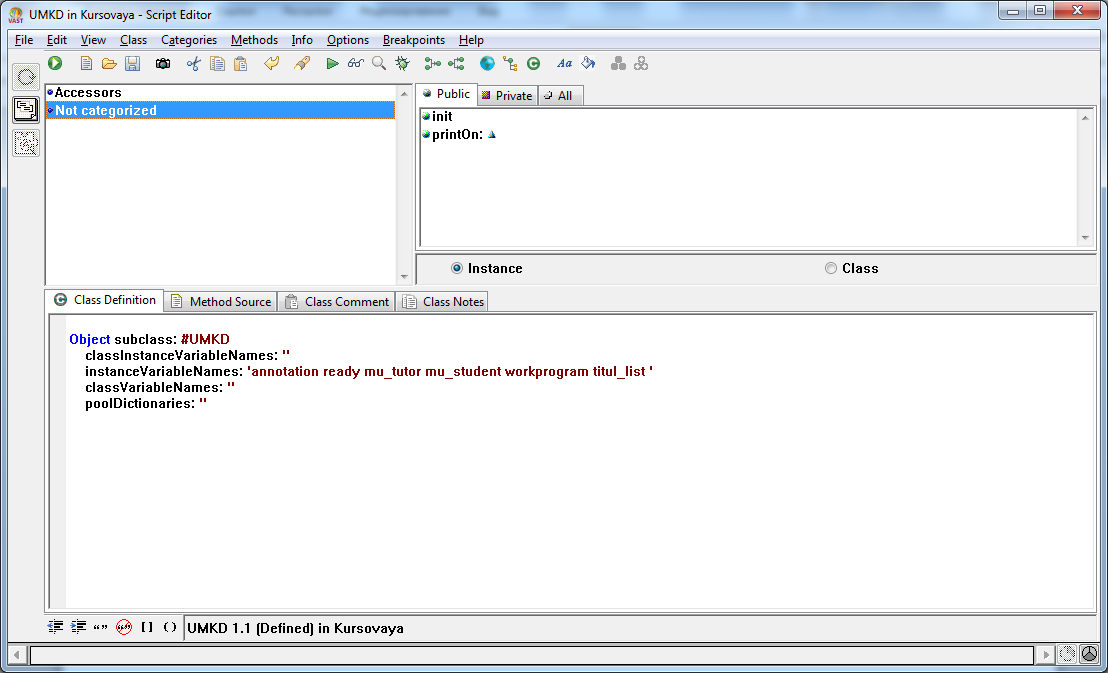


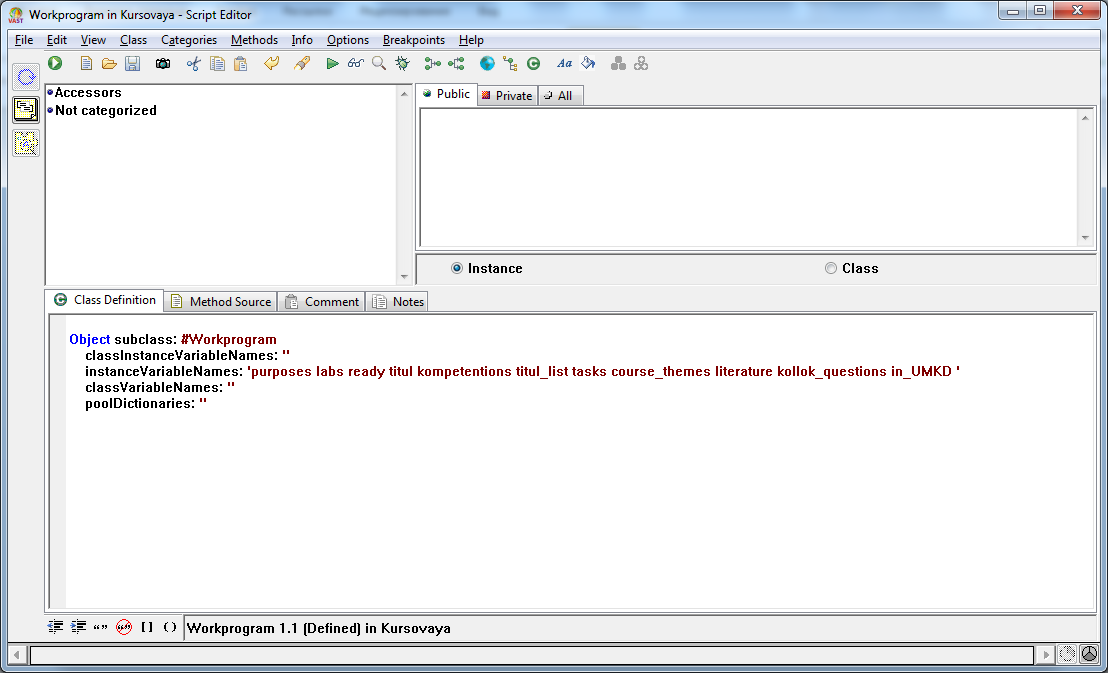












**Код программы:**

"$COMPATIBLE-DECLARATIONS-START$"

Smalltalk declarePoolDictionary: 'IS\_Model'.

Smalltalk declareVariable: 'IS\_instanceInterfaceSpec' poolName: 'IS\_Model'.

Smalltalk declarePoolDictionary: 'IS\_MU\_tutor'.

Smalltalk declareVariable: 'IS\_ready' poolName: 'IS\_MU\_tutor'.

Smalltalk declareVariable: 'IS\_exam\_questions' poolName: 'IS\_MU\_tutor'.

Smalltalk declareVariable: 'IS\_test\_questions' poolName: 'IS\_MU\_tutor'.

Smalltalk declareVariable: 'IS\_init' poolName: 'IS\_MU\_tutor'.

Smalltalk declareVariable: 'IS\_instanceInterfaceSpec' poolName: 'IS\_MU\_tutor'.

Smalltalk declarePoolDictionary: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_set' poolName: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_is\_working' poolName: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_init' poolName: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_fill' poolName: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_umkd' poolName: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_write' poolName: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_learnplan' poolName: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_get' poolName: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_make' poolName: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_select' poolName: 'IS\_Tutor'.

Smalltalk declareVariable: 'IS\_instanceInterfaceSpec' poolName: 'IS\_Tutor'.

Smalltalk declarePoolDictionary: 'IS\_UMKD'.

Smalltalk declareVariable: 'IS\_ready' poolName: 'IS\_UMKD'.

Smalltalk declareVariable: 'IS\_titul\_list' poolName: 'IS\_UMKD'.

Smalltalk declareVariable: 'IS\_mu\_student' poolName: 'IS\_UMKD'.

Smalltalk declareVariable: 'IS\_init' poolName: 'IS\_UMKD'.

Smalltalk declareVariable: 'IS\_annotation' poolName: 'IS\_UMKD'.

Smalltalk declareVariable: 'IS\_mu\_tutor' poolName: 'IS\_UMKD'.

Smalltalk declareVariable: 'IS\_workprogram' poolName: 'IS\_UMKD'.

Smalltalk declareVariable: 'IS\_instanceInterfaceSpec' poolName: 'IS\_UMKD'.

Smalltalk declarePoolDictionary: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_ready' poolName: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_kollok\_questions' poolName: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_labs' poolName: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_course\_themes' poolName: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_titul\_list' poolName: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_init' poolName: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_tasks' poolName: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_purposes' poolName: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_kompetentions' poolName: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_literature' poolName: 'IS\_Workprogram'.

Smalltalk declareVariable: 'IS\_instanceInterfaceSpec' poolName: 'IS\_Workprogram'.

"$COMPATIBLE-DECLARATIONS-END$"!

Application create: #Kursovaya with:

(#( AbtViewApplication)

collect: [:each | Smalltalk at: each ifAbsent: [

Application errorPrerequisite: #Kursovaya missing: each]])!

Kursovaya becomeDefault!

AbtAppBldrView subclass: #Model

classInstanceVariableNames: ''

instanceVariableNames: 'tutor '

classVariableNames: ''

poolDictionaries: ''!

Kursovaya becomeDefault!

Object subclass: #MU\_tutor

classInstanceVariableNames: ''

instanceVariableNames: 'ready exam\_questions in\_UMKD test\_questions '

classVariableNames: ''

poolDictionaries: ''!

Kursovaya becomeDefault!

Application subclass: #Kursovaya

classInstanceVariableNames: ''

instanceVariableNames: ''

classVariableNames: ''

poolDictionaries: ''!

Kursovaya becomeDefault!

Object subclass: #Tutor

classInstanceVariableNames: ''

instanceVariableNames: 'is\_working learnplan umkd '

classVariableNames: ''

poolDictionaries: ''!

Kursovaya becomeDefault!

Object subclass: #UMKD

classInstanceVariableNames: ''

instanceVariableNames: 'annotation ready mu\_tutor mu\_student workprogram titul\_list '

classVariableNames: ''

poolDictionaries: ''!

Kursovaya becomeDefault!

Object subclass: #Workprogram

classInstanceVariableNames: ''

instanceVariableNames: 'purposes labs ready titul kompetentions titul\_list tasks course\_themes literature kollok\_questions in\_UMKD '

classVariableNames: ''

poolDictionaries: ''!

Kursovaya becomeDefault!

!Kursovaya class privateMethods !

abtIsViewApplication

^true! !

!Model class privateMethods !

\_PRAGMA\_IS\_

"%%PRAGMA DECLARE

(name: IS\_Model isPool: true isConstant: false)

(pool: IS\_Model declarations: (

(name: IS\_instanceInterfaceSpec isConstant: false)

))"!

abtPrimFlushInterfaceSpecCache

IS\_Model associationsDo: [: poolDictionaryAssoc | poolDictionaryAssoc value: nil].

super abtPrimFlushInterfaceSpecCache!

abtUntranslatedConstants

"\*\* Do not modify or delete \*\* See: AbtAppBldrPart class>>#about\_abtUntranslatedConstants"

^#(

'Формирование УМКД'

'Закрыть'

'Создать папку с отчетом'

'Составить УМКД'

)!

IS\_instanceInterfaceSpec

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Model::IS\_instanceInterfaceSpec notNil

ifTrue: [IS\_Model::IS\_instanceInterfaceSpec]

ifFalse: [

IS\_Model::IS\_instanceInterfaceSpec := AbtInterfaceSpec new]! !

!Model publicMethods !

createFolder

|file|

CfsDirectoryDescriptor mkdir: 'C:\\UMKD'.

CfsDirectoryDescriptor chdir: 'C:\\UMKD'.

file := CfsReadWriteFileStream openEmpty: 'umkd.txt'.

tutor printOn: file.

file close.

self writeToField: 'File is written at C:\\UMKD\umkd.txt'.

!

init

"initialization"

tutor := Tutor new.

tutor init.

!

start

tutor is\_working = true ifTrue: [

self writeToField: (tutor make\_umkd).

tutor is\_working: false.

] ifFalse: [self writeToField: ('Tutor is tired and has a rest now').]

!

writeToField: str

"write"

|printFunc|

printFunc := [:st | (self subpartNamed: 'mainEditor') insertText: (st asString, CwText lineDelimiter) position: (self subpartNamed: 'mainEditor') cursorPosition].

str class = False ifTrue: [ ^false ].

str class = OrderedCollection ifTrue: [

str do: [ :val |

val class = OrderedCollection ifTrue: [self writeToField: val.].

printFunc value: (val asString)

]

] ifFalse: [

printFunc value: str.

].

! !

!Model privateMethods !

abtBuildInternals

"\*\* Do not modify or delete \*\* See: AbtAppBldrPart class>>#about\_abtBuildInternals"

| gui window mainEditor pushButton1 pushButton2 pushButton3 conn0 conn1 conn2 conn3 |

gui := self class abtSeparatedConstants.

window := AbtShellView abtCreatePart: 'Window' parent: nil owner: self .

mainEditor := AbtMultiLineEditView abtCreatePart: 'mainEditor' parent: window.

pushButton1 := AbtPushButtonView abtCreatePart: 'Push Button1' parent: window.

pushButton2 := AbtPushButtonView abtCreatePart: 'Push Button2' parent: window.

pushButton3 := AbtPushButtonView abtCreatePart: 'Push Button3' parent: window.

self

primaryPart: window.

window

framingSpec: (AbtViewAttachmentConstraint new

leftEdge: (AbtEdgeConstant new offset: 296);

rightEdge: (AbtEdgeConstant new offset: 372);

topEdge: (AbtEdgeConstant new offset: 84);

bottomEdge: (AbtEdgeConstant new offset: 372));

title: (gui at: 1) " 'Формирование УМКД' " .

mainEditor

framingSpec: (AbtViewAttachmentConstraint new

leftEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHFORM; offset: 14);

rightEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHSELFOPPOSITE; offset: 343);

topEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHFORM; offset: 17);

bottomEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHSELFOPPOSITE; offset: 236));

editable: false.

pushButton1

object: (gui at: 2) " 'Закрыть' " ;

framingSpec: (AbtViewAttachmentConstraint new

leftEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHFORM; offset: 111);

rightEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHSELFOPPOSITE; offset: 172);

topEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHFORM; offset: 329);

bottomEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHNONE)).

pushButton2

object: (gui at: 3) " 'Создать папку с отчетом' " ;

framingSpec: (AbtViewAttachmentConstraint new

leftEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHFORM; offset: 107);

rightEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHNONE);

topEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHFORM; offset: 297);

bottomEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHNONE)).

pushButton3

object: (gui at: 4) " 'Составить УМКД' " ;

framingSpec: (AbtViewAttachmentConstraint new

leftEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHFORM; offset: 106);

rightEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHSELFOPPOSITE; offset: 176);

topEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHFORM; offset: 266);

bottomEdge: (AbtRunEdgeAttachmentConstraint new attachment: XmATTACHNONE)).

pushButton1

abtWhenPrimitive: #clicked

perform:

(DirectedMessage new

receiver: window;

selector: #closeWidgetCommand;

arguments: #()).

window

abtWhenPrimitive: #openedWidget

perform:

(DirectedMessage new

receiver: self;

selector: #init;

arguments: #()).

pushButton3

abtWhenPrimitive: #clicked

perform:

(DirectedMessage new

receiver: self;

selector: #start;

arguments: #()).

pushButton2

abtWhenPrimitive: #clicked

perform:

(DirectedMessage new

receiver: self;

selector: #createFolder;

arguments: #()).

self finalInitialize.

! !

!MU\_tutor class privateMethods !

\_PRAGMA\_IS\_

"%%PRAGMA DECLARE

(name: IS\_MU\_tutor isPool: true isConstant: false)

(pool: IS\_MU\_tutor declarations: (

(name: IS\_ready isConstant: false)

(name: IS\_exam\_questions isConstant: false)

(name: IS\_test\_questions isConstant: false)

(name: IS\_init isConstant: false)

(name: IS\_instanceInterfaceSpec isConstant: false)

))"!

abtPrimFlushInterfaceSpecCache

IS\_MU\_tutor associationsDo: [: poolDictionaryAssoc | poolDictionaryAssoc value: nil].

super abtPrimFlushInterfaceSpecCache!

IS\_exam\_questions

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_MU\_tutor::IS\_exam\_questions notNil

ifTrue: [IS\_MU\_tutor::IS\_exam\_questions]

ifFalse: [

IS\_MU\_tutor::IS\_exam\_questions := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #exam\_questions:;

getSelector: #exam\_questions;

changeSymbol: #exam\_questions)]!

IS\_init

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_MU\_tutor::IS\_init notNil

ifTrue: [IS\_MU\_tutor::IS\_init]

ifFalse: [

IS\_MU\_tutor::IS\_init := (AbtActionSpec new

selector: #init)]!

IS\_instanceInterfaceSpec

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_MU\_tutor::IS\_instanceInterfaceSpec notNil

ifTrue: [IS\_MU\_tutor::IS\_instanceInterfaceSpec]

ifFalse: [

IS\_MU\_tutor::IS\_instanceInterfaceSpec := AbtInterfaceSpec new

featureNamed: #ready put: self IS\_ready;

featureNamed: #exam\_questions put: self IS\_exam\_questions;

featureNamed: #test\_questions put: self IS\_test\_questions;

featureNamed: #init put: self IS\_init]!

IS\_ready

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_MU\_tutor::IS\_ready notNil

ifTrue: [IS\_MU\_tutor::IS\_ready]

ifFalse: [

IS\_MU\_tutor::IS\_ready := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #ready:;

getSelector: #ready;

changeSymbol: #ready)]!

IS\_test\_questions

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_MU\_tutor::IS\_test\_questions notNil

ifTrue: [IS\_MU\_tutor::IS\_test\_questions]

ifFalse: [

IS\_MU\_tutor::IS\_test\_questions := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #test\_question:;

getSelector: #test\_question;

changeSymbol: #test\_question)]! !

!MU\_tutor publicMethods !

exam\_questions

"Return the value of exam\_questions."

^exam\_questions!

exam\_questions: aBoolean

"Save the value of exam\_questions."

exam\_questions := aBoolean.

self signalEvent: #exam\_questions

with: aBoolean.!

init

"Perform the init action."

self exam\_questions: false.

self test\_question: false.

self ready: false.!

printOn: aStream

aStream nextPutAll: 'Exam questions ', self exam\_questions asString; cr.

aStream nextPutAll: 'Test questions ', self test\_question asString; cr. !

ready

"Return the value of ready."

^ready!

ready: aBoolean

"Save the value of ready."

ready := aBoolean.

self signalEvent: #ready

with: aBoolean.!

test\_question

"Return the value of test\_questions."

^test\_questions!

test\_question: aBoolean

"Save the value of test\_questions."

test\_questions := aBoolean.

self signalEvent: #test\_question

with: aBoolean.! !

!Tutor class privateMethods !

\_PRAGMA\_IS\_

"%%PRAGMA DECLARE

(name: IS\_Tutor isPool: true isConstant: false)

(pool: IS\_Tutor declarations: (

(name: IS\_set isConstant: false)

(name: IS\_is\_working isConstant: false)

(name: IS\_init isConstant: false)

(name: IS\_fill isConstant: false)

(name: IS\_umkd isConstant: false)

(name: IS\_write isConstant: false)

(name: IS\_learnplan isConstant: false)

(name: IS\_get isConstant: false)

(name: IS\_make isConstant: false)

(name: IS\_select isConstant: false)

(name: IS\_instanceInterfaceSpec isConstant: false)

))"!

abtPrimFlushInterfaceSpecCache

IS\_Tutor associationsDo: [: poolDictionaryAssoc | poolDictionaryAssoc value: nil].

super abtPrimFlushInterfaceSpecCache!

IS\_fill

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_fill notNil

ifTrue: [IS\_Tutor::IS\_fill]

ifFalse: [

IS\_Tutor::IS\_fill := (AbtActionSpec new

selector: #fill)]!

IS\_get

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_get notNil

ifTrue: [IS\_Tutor::IS\_get]

ifFalse: [

IS\_Tutor::IS\_get := (AbtActionSpec new

selector: #get)]!

IS\_init

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_init notNil

ifTrue: [IS\_Tutor::IS\_init]

ifFalse: [

IS\_Tutor::IS\_init := (AbtActionSpec new

selector: #init)]!

IS\_instanceInterfaceSpec

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_instanceInterfaceSpec notNil

ifTrue: [IS\_Tutor::IS\_instanceInterfaceSpec]

ifFalse: [

IS\_Tutor::IS\_instanceInterfaceSpec := AbtInterfaceSpec new

featureNamed: #set put: self IS\_set;

featureNamed: #is\_working put: self IS\_is\_working;

featureNamed: #init put: self IS\_init;

featureNamed: #fill put: self IS\_fill;

featureNamed: #umkd put: self IS\_umkd;

featureNamed: #write put: self IS\_write;

featureNamed: #learnplan put: self IS\_learnplan;

featureNamed: #get put: self IS\_get;

featureNamed: #make put: self IS\_make;

featureNamed: #select put: self IS\_select]!

IS\_is\_working

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_is\_working notNil

ifTrue: [IS\_Tutor::IS\_is\_working]

ifFalse: [

IS\_Tutor::IS\_is\_working := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #is\_working:;

getSelector: #is\_working;

changeSymbol: #is\_working)]!

IS\_learnplan

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_learnplan notNil

ifTrue: [IS\_Tutor::IS\_learnplan]

ifFalse: [

IS\_Tutor::IS\_learnplan := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #learnplan:;

getSelector: #learnplan;

changeSymbol: #learnplan)]!

IS\_make

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_make notNil

ifTrue: [IS\_Tutor::IS\_make]

ifFalse: [

IS\_Tutor::IS\_make := (AbtActionSpec new

selector: #make)]!

IS\_select

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_select notNil

ifTrue: [IS\_Tutor::IS\_select]

ifFalse: [

IS\_Tutor::IS\_select := (AbtActionSpec new

selector: #select)]!

IS\_set

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_set notNil

ifTrue: [IS\_Tutor::IS\_set]

ifFalse: [

IS\_Tutor::IS\_set := (AbtActionSpec new

selector: #set)]!

IS\_umkd

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_umkd notNil

ifTrue: [IS\_Tutor::IS\_umkd]

ifFalse: [

IS\_Tutor::IS\_umkd := (AbtAttributeSpec new

attributeType: UMKD;

setSelector: #umkd:;

getSelector: #umkd;

changeSymbol: #umkd)]!

IS\_write

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Tutor::IS\_write notNil

ifTrue: [IS\_Tutor::IS\_write]

ifFalse: [

IS\_Tutor::IS\_write := (AbtActionSpec new

selector: #write)]! !

!Tutor publicMethods !

fillTitul: umkd

"Perform the titul page filling action."

umkd titul\_list = false ifTrue: [

umkd titul\_list: true.

^'UMKD titul list is ready'

].

^false!

fillTitulWP: workprogram

"Perform the titul page filling action."

workprogram titul\_list = false ifTrue: [

workprogram titul\_list: true.

^'Workprogram titul list is ready'

].

^false!

getLearnplan

"Perform the getting learning plan action."

self learnplan = false ifTrue: [

self learnplan: true.

^'List from learning plan is ready'

].

^false!

init

"Perform the init action."

self is\_working: true.

self learnplan: false.

self.umkd := UMKD new.

umkd init.

!

is\_working

"Return the value of is\_working."

^is\_working!

is\_working: aBoolean

"Save the value of is\_working."

is\_working := aBoolean.

self signalEvent: #is\_working

with: aBoolean.!

learnplan

"Return the value of learnplan."

^learnplan!

learnplan: aBoolean

"Save the value of learnplan."

learnplan := aBoolean.

self signalEvent: #learnplan

with: aBoolean.!

make\_umkd

"Perform the UMKD creation action."

|msg|

msg := OrderedCollection new.

self.umkd ready = false ifTrue: [

msg add: (self getLearnplan).

msg add: (self fillTitul: umkd).

msg add: (self writeAnnotation: umkd ).

msg add: (self writeWorkprogram: umkd workprogram).

msg add: (self writeMuTutor: umkd mu\_tutor).

msg add: (self writeMuStudent: umkd).

umkd ready: true.

msg add: 'UMKD is ready!!'.

^msg.

].

^false !

printOn: aStream

aStream nextPutAll: 'Workprogram ', umkd ready asString; cr.

umkd printOn: aStream.

aStream cr.

!

selectLiterature: workprogram

"Perform the selection literature in workprogram write action."

workprogram literature = false ifTrue: [

workprogram literature: true.

^'Literature is selected'

].

^false!

writeAnnotation: umkd

"Perform the Annotation write action."

umkd annotation = false ifTrue: [

umkd annotation: true.

^'Annotation is ready'

].

^false!

writeCourseThemes: workprogram

"Perform the coursework themes in workprogram write action."

workprogram course\_themes = false ifTrue: [

workprogram course\_themes: true.

^'Coursework themes are ready'

].

^false!

writeExamQuestons: mu\_tutor

"Perform the exam questons in materials for tutors write action."

mu\_tutor exam\_questions = false ifTrue: [

mu\_tutor exam\_questions: true.

^'Exam questons are ready'

].

^false!

writeKollokQuestons: workprogram

"Perform the kollokvium questons in workprogram write action."

workprogram kollok\_questions = false ifTrue: [

workprogram kollok\_questions: true.

^'Kollokvium questons are ready'

].

^false!

writeKompetentions: workprogram

"Perform the kompetentions in workprogram write action."

workprogram kompetentions = false ifTrue: [

workprogram kompetentions: true.

^'Kompetentions are ready'

].

^false!

writeLabs: workprogram

"Perform the labs in workprogram write action."

workprogram labs = false ifTrue: [

workprogram labs: true.

^'Labs are ready'

].

^false!

writeMuStudent: umkd

"Perform writing methodiacal materials for students action."

umkd mu\_student = false ifTrue: [

umkd mu\_student: true.

^'Methodical materials for students are ready'

].

^false!

writeMuTutor: mu\_tutor

"Perform the write action."

|msg|

msg := OrderedCollection new.

mu\_tutor ready = false ifTrue: [

msg add: (self writeExamQuestons: mu\_tutor).

msg add: (self writeTestQuestons: mu\_tutor ).

mu\_tutor ready: true.

msg add: 'Methodical materials for students are ready'.

^msg.

].

^false !

writePurposes: workprogram

"Perform the purposes in workprogram write action."

workprogram purposes = false ifTrue: [

workprogram purposes: true.

^'Purposes are ready'

].

^false!

writeTasks: workprogram

"Perform the tasks in workprogram write action."

workprogram tasks = false ifTrue: [

workprogram tasks: true.

^'Tasks are ready'

].

^false!

writeTestQuestons: mu\_tutor

"Perform the test questons in materials for tutors write action."

mu\_tutor test\_question = false ifTrue: [

mu\_tutor test\_question: true.

^'Test questons are ready'

].

^false!

writeWorkprogram: workprogram

"Perform the workprogram write action."

|msg|

msg := OrderedCollection new.

workprogram ready = false ifTrue: [

msg add: (self fillTitulWP: workprogram).

msg add: (self writeTasks: workprogram ).

msg add: (self writePurposes: workprogram ).

msg add: (self writeKompetentions: workprogram ).

msg add: (self writeLabs: workprogram ).

msg add: (self writeKollokQuestons: workprogram ).

msg add: (self writeCourseThemes: workprogram ).

msg add: (self selectLiterature: workprogram ).

workprogram ready: true.

msg add: 'Workprogram is ready!!'.

^msg.

].

^false ! !

!UMKD class privateMethods !

\_PRAGMA\_IS\_

"%%PRAGMA DECLARE

(name: IS\_UMKD isPool: true isConstant: false)

(pool: IS\_UMKD declarations: (

(name: IS\_ready isConstant: false)

(name: IS\_titul\_list isConstant: false)

(name: IS\_mu\_student isConstant: false)

(name: IS\_init isConstant: false)

(name: IS\_annotation isConstant: false)

(name: IS\_mu\_tutor isConstant: false)

(name: IS\_workprogram isConstant: false)

(name: IS\_instanceInterfaceSpec isConstant: false)

))"!

abtPrimFlushInterfaceSpecCache

IS\_UMKD associationsDo: [: poolDictionaryAssoc | poolDictionaryAssoc value: nil].

super abtPrimFlushInterfaceSpecCache!

IS\_annotation

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_UMKD::IS\_annotation notNil

ifTrue: [IS\_UMKD::IS\_annotation]

ifFalse: [

IS\_UMKD::IS\_annotation := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #annotation:;

getSelector: #annotation;

changeSymbol: #annotation)]!

IS\_init

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_UMKD::IS\_init notNil

ifTrue: [IS\_UMKD::IS\_init]

ifFalse: [

IS\_UMKD::IS\_init := (AbtActionSpec new

selector: #init)]!

IS\_instanceInterfaceSpec

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_UMKD::IS\_instanceInterfaceSpec notNil

ifTrue: [IS\_UMKD::IS\_instanceInterfaceSpec]

ifFalse: [

IS\_UMKD::IS\_instanceInterfaceSpec := AbtInterfaceSpec new

featureNamed: #ready put: self IS\_ready;

featureNamed: #titul\_list put: self IS\_titul\_list;

featureNamed: #mu\_student put: self IS\_mu\_student;

featureNamed: #init put: self IS\_init;

featureNamed: #annotation put: self IS\_annotation;

featureNamed: #mu\_tutor put: self IS\_mu\_tutor;

featureNamed: #workprogram put: self IS\_workprogram]!

IS\_mu\_student

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_UMKD::IS\_mu\_student notNil

ifTrue: [IS\_UMKD::IS\_mu\_student]

ifFalse: [

IS\_UMKD::IS\_mu\_student := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #mu\_student:;

getSelector: #mu\_student;

changeSymbol: #mu\_student)]!

IS\_mu\_tutor

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_UMKD::IS\_mu\_tutor notNil

ifTrue: [IS\_UMKD::IS\_mu\_tutor]

ifFalse: [

IS\_UMKD::IS\_mu\_tutor := (AbtAttributeSpec new

attributeType: MU\_tutor;

setSelector: #mu\_tutor:;

getSelector: #mu\_tutor;

changeSymbol: #mu\_tutor)]!

IS\_ready

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_UMKD::IS\_ready notNil

ifTrue: [IS\_UMKD::IS\_ready]

ifFalse: [

IS\_UMKD::IS\_ready := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #ready:;

getSelector: #ready;

changeSymbol: #ready)]!

IS\_titul\_list

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_UMKD::IS\_titul\_list notNil

ifTrue: [IS\_UMKD::IS\_titul\_list]

ifFalse: [

IS\_UMKD::IS\_titul\_list := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #titul\_list:;

getSelector: #titul\_list;

changeSymbol: #titul\_list)]!

IS\_workprogram

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_UMKD::IS\_workprogram notNil

ifTrue: [IS\_UMKD::IS\_workprogram]

ifFalse: [

IS\_UMKD::IS\_workprogram := (AbtAttributeSpec new

attributeType: Workprogram;

setSelector: #workprogram:;

getSelector: #workprogram;

changeSymbol: #workprogram)]! !

!UMKD publicMethods !

annotation

"Return the value of annotation."

^annotation!

annotation: aBoolean

"Save the value of annotation."

annotation := aBoolean.

self signalEvent: #annotation

with: aBoolean.!

init

"Perform the init action."

|tmp|

self annotation: false.

self mu\_student: false.

self ready: false.

self titul\_list: false.

tmp:= Workprogram new.

self workprogram: tmp.

self workprogram init.

tmp:=MU\_tutor new.

self mu\_tutor: tmp.

self mu\_tutor init.!

mu\_student

"Return the value of mu\_student."

^mu\_student!

mu\_student: aBoolean

"Save the value of mu\_student."

mu\_student := aBoolean.

self signalEvent: #mu\_student

with: aBoolean.!

mu\_tutor

"Return the value of mu\_tutor."

^mu\_tutor!

mu\_tutor: aMU\_tutor

"Save the value of mu\_tutor."

mu\_tutor := aMU\_tutor.

self signalEvent: #mu\_tutor

with: aMU\_tutor.!

printOn: aStream

aStream nextPutAll: 'Titul page ', self titul\_list asString; cr; cr.

aStream nextPutAll: 'Annotation ', self annotation asString; cr; cr.

aStream nextPutAll: 'Workprogram ', self workprogram ready asString; cr.

self workprogram printOn: aStream.

aStream cr.

aStream nextPutAll: 'Metodical materials for studens ', self mu\_student asString; cr;cr.

aStream nextPutAll: 'Metodical materials for tutors ', self mu\_tutor ready asString; cr.

self mu\_tutor printOn: aStream.

aStream cr.!

ready

"Return the value of ready."

^ready!

ready: aBoolean

"Save the value of ready."

ready := aBoolean.

self signalEvent: #ready

with: aBoolean.!

titul\_list

"Return the value of titul\_list."

^titul\_list!

titul\_list: aBoolean

"Save the value of titul\_list."

titul\_list := aBoolean.

self signalEvent: #titul\_list

with: aBoolean.!

workprogram

"Return the value of workprogram."

^workprogram!

workprogram: aWorkprogram

"Save the value of workprogram."

workprogram := aWorkprogram.

self signalEvent: #workprogram

with: aWorkprogram.! !

!Workprogram class privateMethods !

\_PRAGMA\_IS\_

"%%PRAGMA DECLARE

(name: IS\_Workprogram isPool: true isConstant: false)

(pool: IS\_Workprogram declarations: (

(name: IS\_ready isConstant: false)

(name: IS\_kollok\_questions isConstant: false)

(name: IS\_labs isConstant: false)

(name: IS\_course\_themes isConstant: false)

(name: IS\_titul\_list isConstant: false)

(name: IS\_init isConstant: false)

(name: IS\_tasks isConstant: false)

(name: IS\_purposes isConstant: false)

(name: IS\_kompetentions isConstant: false)

(name: IS\_literature isConstant: false)

(name: IS\_instanceInterfaceSpec isConstant: false)

))"!

abtPrimFlushInterfaceSpecCache

IS\_Workprogram associationsDo: [: poolDictionaryAssoc | poolDictionaryAssoc value: nil].

super abtPrimFlushInterfaceSpecCache!

IS\_course\_themes

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_course\_themes notNil

ifTrue: [IS\_Workprogram::IS\_course\_themes]

ifFalse: [

IS\_Workprogram::IS\_course\_themes := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #course\_themes:;

getSelector: #course\_themes;

changeSymbol: #course\_themes)]!

IS\_init

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_init notNil

ifTrue: [IS\_Workprogram::IS\_init]

ifFalse: [

IS\_Workprogram::IS\_init := (AbtActionSpec new

selector: #init)]!

IS\_instanceInterfaceSpec

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_instanceInterfaceSpec notNil

ifTrue: [IS\_Workprogram::IS\_instanceInterfaceSpec]

ifFalse: [

IS\_Workprogram::IS\_instanceInterfaceSpec := AbtInterfaceSpec new

featureNamed: #ready put: self IS\_ready;

featureNamed: #kollok\_questions put: self IS\_kollok\_questions;

featureNamed: #labs put: self IS\_labs;

featureNamed: #course\_themes put: self IS\_course\_themes;

featureNamed: #titul\_list put: self IS\_titul\_list;

featureNamed: #init put: self IS\_init;

featureNamed: #tasks put: self IS\_tasks;

featureNamed: #purposes put: self IS\_purposes;

featureNamed: #kompetentions put: self IS\_kompetentions;

featureNamed: #literature put: self IS\_literature]!

IS\_kollok\_questions

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_kollok\_questions notNil

ifTrue: [IS\_Workprogram::IS\_kollok\_questions]

ifFalse: [

IS\_Workprogram::IS\_kollok\_questions := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #kollok\_questions:;

getSelector: #kollok\_questions;

changeSymbol: #kollok\_questions)]!

IS\_kompetentions

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_kompetentions notNil

ifTrue: [IS\_Workprogram::IS\_kompetentions]

ifFalse: [

IS\_Workprogram::IS\_kompetentions := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #kompetentions:;

getSelector: #kompetentions;

changeSymbol: #kompetentions)]!

IS\_labs

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_labs notNil

ifTrue: [IS\_Workprogram::IS\_labs]

ifFalse: [

IS\_Workprogram::IS\_labs := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #labs:;

getSelector: #labs;

changeSymbol: #labs)]!

IS\_literature

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_literature notNil

ifTrue: [IS\_Workprogram::IS\_literature]

ifFalse: [

IS\_Workprogram::IS\_literature := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #literature:;

getSelector: #literature;

changeSymbol: #literature)]!

IS\_purposes

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_purposes notNil

ifTrue: [IS\_Workprogram::IS\_purposes]

ifFalse: [

IS\_Workprogram::IS\_purposes := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #purposes:;

getSelector: #purposes;

changeSymbol: #purposes)]!

IS\_ready

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_ready notNil

ifTrue: [IS\_Workprogram::IS\_ready]

ifFalse: [

IS\_Workprogram::IS\_ready := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #ready:;

getSelector: #ready;

changeSymbol: #ready)]!

IS\_tasks

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_tasks notNil

ifTrue: [IS\_Workprogram::IS\_tasks]

ifFalse: [

IS\_Workprogram::IS\_tasks := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #tasks:;

getSelector: #tasks;

changeSymbol: #tasks)]!

IS\_titul\_list

"Private - \*\* Warning \*\* This method is generated by VisualAge and should not

be modified or deleted. This method is responsible for returning a featureSpec

that describes the implementation of a particular feature of the receiver"

^IS\_Workprogram::IS\_titul\_list notNil

ifTrue: [IS\_Workprogram::IS\_titul\_list]

ifFalse: [

IS\_Workprogram::IS\_titul\_list := (AbtAttributeSpec new

attributeType: Boolean;

setSelector: #titul\_list:;

getSelector: #titul\_list;

changeSymbol: #titul\_list)]! !

!Workprogram publicMethods !

course\_themes

"Return the value of course\_themes."

^course\_themes!

course\_themes: aBoolean

"Save the value of course\_themes."

course\_themes := aBoolean.

self signalEvent: #course\_themes

with: aBoolean.!

in\_UMKD

"Return the value of in\_UMKD."

^in\_UMKD!

in\_UMKD: anObject

"Save the value of in\_UMKD."

in\_UMKD := anObject.

self signalEvent: #in\_UMKD

with: anObject.!

init

"Perform the init action."

self course\_themes: false.

self kollok\_questions: false.

self kompetentions: false.

self labs: false.

self literature: false.

self purposes: false.

self ready: false.

self tasks: false.

self titul\_list: false.!

kollok\_questions

"Return the value of kollok\_questions."

^kollok\_questions!

kollok\_questions: aBoolean

"Save the value of kollok\_questions."

kollok\_questions := aBoolean.

self signalEvent: #kollok\_questions

with: aBoolean.!

kompetentions

"Return the value of kompetentions."

^kompetentions!

kompetentions: aBoolean

"Save the value of kompetentions."

kompetentions := aBoolean.

self signalEvent: #kompetentions

with: aBoolean.!

labs

"Return the value of labs."

^labs!

labs: aBoolean

"Save the value of labs."

labs := aBoolean.

self signalEvent: #labs

with: aBoolean.!

literature

"Return the value of literature."

^literature!

literature: aBoolean

"Save the value of literature."

literature := aBoolean.

self signalEvent: #literature

with: aBoolean.!

printOn: aStream

aStream nextPutAll: 'Titul page ', self titul\_list asString; cr.

aStream nextPutAll: 'Purposes ', self purposes asString; cr.

aStream nextPutAll: 'Tasks ', self tasks asString; cr.

aStream nextPutAll: 'Kompetentions ', self kompetentions asString; cr.

aStream nextPutAll: 'Kollokvium questions ', self kollok\_questions asString; cr.

aStream nextPutAll: 'Labs ', self labs asString; cr.

aStream nextPutAll: 'Coursework themes ', self course\_themes asString; cr.

aStream nextPutAll: 'List of literature ', self literature asString; cr.

!

purposes

"Return the value of purposes."

^purposes!

purposes: aBoolean

"Save the value of purposes."

purposes := aBoolean.

self signalEvent: #purposes

with: aBoolean.!

ready

"Return the value of ready."

^ready!

ready: aBoolean

"Save the value of ready."

ready := aBoolean.

self signalEvent: #ready

with: aBoolean.!

tasks

"Return the value of tasks."

^tasks!

tasks: aBoolean

"Save the value of tasks."

tasks := aBoolean.

self signalEvent: #tasks

with: aBoolean.!

titul

"Return the value of titul."

^titul!

titul: aBoolean

"Save the value of titul."

titul := aBoolean.

self signalEvent: #titul

with: aBoolean.!

titul\_list

"Return the value of titul\_list."

^titul\_list!

titul\_list: aBoolean

"Save the value of titul\_list."

titul\_list := aBoolean.

self signalEvent: #titul\_list

with: aBoolean.! !

Kursovaya toBeLoadedCode: '"$COMPATIBLE-DECLARATIONS-START$"

Smalltalk declarePoolDictionary: ''IS\_Model''.

Smalltalk declareVariable: ''IS\_instanceInterfaceSpec'' poolName: ''IS\_Model''.

Smalltalk declarePoolDictionary: ''IS\_MU\_tutor''.

Smalltalk declareVariable: ''IS\_ready'' poolName: ''IS\_MU\_tutor''.

Smalltalk declareVariable: ''IS\_exam\_questions'' poolName: ''IS\_MU\_tutor''.

Smalltalk declareVariable: ''IS\_test\_questions'' poolName: ''IS\_MU\_tutor''.

Smalltalk declareVariable: ''IS\_init'' poolName: ''IS\_MU\_tutor''.

Smalltalk declareVariable: ''IS\_instanceInterfaceSpec'' poolName: ''IS\_MU\_tutor''.

Smalltalk declarePoolDictionary: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_set'' poolName: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_is\_working'' poolName: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_init'' poolName: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_fill'' poolName: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_umkd'' poolName: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_write'' poolName: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_learnplan'' poolName: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_get'' poolName: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_make'' poolName: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_select'' poolName: ''IS\_Tutor''.

Smalltalk declareVariable: ''IS\_instanceInterfaceSpec'' poolName: ''IS\_Tutor''.

Smalltalk declarePoolDictionary: ''IS\_UMKD''.

Smalltalk declareVariable: ''IS\_ready'' poolName: ''IS\_UMKD''.

Smalltalk declareVariable: ''IS\_titul\_list'' poolName: ''IS\_UMKD''.

Smalltalk declareVariable: ''IS\_mu\_student'' poolName: ''IS\_UMKD''.

Smalltalk declareVariable: ''IS\_init'' poolName: ''IS\_UMKD''.

Smalltalk declareVariable: ''IS\_annotation'' poolName: ''IS\_UMKD''.

Smalltalk declareVariable: ''IS\_mu\_tutor'' poolName: ''IS\_UMKD''.

Smalltalk declareVariable: ''IS\_workprogram'' poolName: ''IS\_UMKD''.

Smalltalk declareVariable: ''IS\_instanceInterfaceSpec'' poolName: ''IS\_UMKD''.

Smalltalk declarePoolDictionary: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_ready'' poolName: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_kollok\_questions'' poolName: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_labs'' poolName: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_course\_themes'' poolName: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_titul\_list'' poolName: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_init'' poolName: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_tasks'' poolName: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_purposes'' poolName: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_kompetentions'' poolName: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_literature'' poolName: ''IS\_Workprogram''.

Smalltalk declareVariable: ''IS\_instanceInterfaceSpec'' poolName: ''IS\_Workprogram''.

"$COMPATIBLE-DECLARATIONS-END$"'!

Kursovaya wasRemovedCode: '"$COMPATIBLE-DECLARATIONS-START$"

Smalltalk undeclare: ''IS\_Model''.

Smalltalk undeclare: ''IS\_MU\_tutor''.

Smalltalk undeclare: ''IS\_Tutor''.

Smalltalk undeclare: ''IS\_UMKD''.

Smalltalk undeclare: ''IS\_Workprogram''.

"$COMPATIBLE-DECLARATIONS-END$"'!

Model initializeAfterLoad!

MU\_tutor initializeAfterLoad!

Kursovaya initializeAfterLoad!

Tutor initializeAfterLoad!

UMKD initializeAfterLoad!

Workprogram initializeAfterLoad!

Kursovaya loaded!

# Список литературы

Программирование на языке Smalltalk. Методические указания. 0020 - 0026, В.А. Смольянинова, С.С. Смирнов, И.А. Акимов, Москва, МИРЭА, 2010