Abstract class. A class that is never instantiated. Used to factor out properties shared by its subclasses. As an example, class Boolean is an abstract class with two concrete subclasses True and False.
Abstract data type.
Abstraction.
Accelerator.
Action button. Component of *user interface* looking like a button. Its activation by the mouse in Smalltalk sends a message.
Algorithm. An orderly sequence of steps describing the solution of a problem such as sorting a collection of names alphabetically or finding the largest of a set of numbers.
Application.
Application model. A class implementing the connection between the user interface (buttons, checkboxes, input fields, sliders, etc.) and the domain model consisting of objects that represent components of the problem domain. The user interface sends messages to the application model, which converts them to messages to specific objects in the domain model. The communication also works that way in the opposite direction. Example: *User interface* - buttons, sliders, and other widgets on the screen triggering and displaying deposits, withdrawals, account balances; *domain model* - objects representing accounts, transactions, and bank clients; application model - the object converting signals from the interface to messages to domain objects and vice versa.
Argument. An object sent as a parameter of a message. As an example, in message 3 + 17 object 3 is the receiver and object 17 is the argument. Message 5 between: 2 and: 17 has two arguments, objects 2 and 17.
ASCII. American Standard Code for Information Interchange. An internationally used code for representing printable and control characters as one-byte code. The most widely used representation of textual data.
ASCII file. File whose contents are to be interpreted as ASCII codes and thus displayable on a screen as text. Compare with *binary file*.
Association. A pair consisting of a 'keyword' and its 'value' as in a dictionary. The basic component of a *dictionary* structure.
Binary file. File whose contents are not to be interpreted as ASCII codes but rather as encoded data or instruction codes of an executable program. Consequently not displayable on a screen as text unless interpreted by a special program. Compare with *ASCII file*.
Binary message. A message composed of one or two special characters including ,+-@/\ with a single argument as in 3 + 4 or in 3 @ 17.
Binding
Block. Short for *Block closure*.
Block argument. Variable declared inside a block as in [:index| array at: index]. Its scope is restricted to the block.
Block closure. One or more statements possibly preceded by block arguments, surrounded by square brackets as in [x sin], [x := rad sin. y := rad cos], [:x :y| Transcript show: x, y].
BOSS.
Browser. A multi-window Smalltalk tool providing access to source code of all declarations in the library. Used to examine existing code and to create and edit new code. Several browsers are provided ranging from a system wide system browser to browsers restricted to a specific class or method.
Canvas.
Cascading. A shorthand allowing sending a sequence of messages to the same receiver without repeating the receiver as in Transcript cr; show: 'abc'; cr instead of Transcript cr. Transcript show: 'abc'. Transcript cr.
Category. Grouping of related classes enforced by the *browser*. Helps to reduce the multitude of classes in the library into a more manageable set of categories, makes it possible to *file out* and *file in* a group of classes by a single command. Does not have any semantic meaning and moving a class from one category to another thus has no effect on its operation.
Chaining. Using the result of a statement as an argument or as the receiver of a message as in Transcript show: (3 factorial) printString where 3 factorial is the receiver of printString, and 3 factorial printString

is an argument of show: Makes code more compact, may save temporary variables, but makes code more difficult to read and should not be overused.

Changes file.

Character. Usually refers to a symbol that can be displayed on the screen although some characters have only 'control' functions such as starting a new line.

Check box.

Class. A mold from which objects are created - a formal specification of an abstract template of which individual objects are instances. Contains the name of the class and its superclass, its variables, and methods. All instances of a class (objects created from its specification) share all variables and can execute all methods declared in the class and its superclasses.

Class instance variable. A variable declared with the 'class' button of the browser active, inherited by all subclasses, and accessible to all of their class and instance methods. Unlike a *class variable*, a class instance variable is distinct for each subclass that inherits it. Unlike class methods which are accessible to classes but not their instances, all types of class variables are accessible both to classes in which they are declared and their instances.

Class method. A method whose receiver must be the class in which the method is declared or its subclass. Compare with *instance method*. Used mainly for creation of class instances (as in Array new or Array new: 3), for initialization of class variables (as in Random initialize), and for generally useful tasks that do not require the creation of an instance (as in Dialog warn: 'This operation is illegal').

Class variable. A variable declared with the 'class' button of the browser active, inherited by all subclasses, and accessible to all of their class and instance methods. Unlike a *class instance variable*, a class instance variable is shared by the class in which it is declared and its subclasses; consequently, a change of its value by one class affects all other classes that share it. Unlike class methods which are accessible to classes but not their instances, all types of class variables are accessible both to classes in which they are declared and their instances.

Clean block. A block that does not depend on the context in which it is used. As an example, [Dialog warn" 'Illegal operation'] is clean while [Dialog warn: warningMessage] is not because it depends on the value of variable warning block declared and manipulated outside the block.

Collection. *Abstract class* holding all properties shared by concrete collections such as Array, OrderedCollection, and Dictionary.

Compilation. The process of converting the human readable source code to internal representation executable by the computer. In Smalltalk compilation occurs, for example, upon activation of the 'accept' command in the browser or the debugger or upon activation of 'do it' in the workspace. In the latter case, compilation is immediately followed by execution of the code, in the former case, the compiled code is only internally stored.

Concatenation.

Concrete class. A class declared with the intent to be used to create instances. As an example, class Character is a concrete class whose instances are widely used, mainly as elements of strings. See also *Abstract class*.

Control statement. Usually a composite statement that controls how its components are executed. Examples: ifTrue:, whileFalse:, do: which select how many times their components are executed or which of their components are executed.

Controller. In the Smalltalk *MVC* (model-view-controller) paradigm, that part of the user interface which monitors users input (keyboard and mouse) and communicates it to the model which then decides whether the action should cause a change in the displayed view, and notifies the view to change itself if necessary. Each active widget (button, input field, etc.) has a controller monitoring and processing its activation, a model to which it communicates all activity, and a view which takes care of redisplaying itself if requested to do so by the model.

Data abstraction.

Data structure. An established view and organization of data with attached access mechanisms such as an array, a set, or a dictionary. A composite object containing several components is also a form of a data structure.

Dataset widget. Similar to a table but used to display rows of objects with identical structure consisting of multiple components, such as rows of course marks for individual students.

Decompilation. The reverse of *compilation*. Converts compiled code to *source code*. Since compiled code does not contain information about variable names, decompiled code contains artificially created variable names instead of the original ones. Decompiled code may differ from source code because the compiler performs certain optimizations to improve the efficiency of execution.

Debugger. A multi-window Smalltalk tool allowing step-by-step execution, inspection of variables, and modification of source code and variable values. Used for debugging new code and for examining operation of existing code.

Debugging. Step-by-step execution performed to locate and correct mistakes (bugs) in a program.

Declaration. Formal specification of the name of a class and its parameters, its comment, or its methods. Must satisfy the syntactic and semantic rules of Smalltalk and is performed using a browser.

Default. Value to which an object is initialized when it is created. Normally nil, unless the initiallization method explicitly initializes the object to a different value. As an example, button are initialized to display themselves with black text on gray background.

Dependency. A mechanism built into Smalltalk that allows mutually dependent objects to notify one another automatically when they change. Used by the *MVC paradigm*: When the *model* changes, it automatically notifies all its dependent *views* which then change their appearance if the change of the model requires it.

Dictionary. A *set* of *associations* - a collection of associations similar to a real life dictionary (but not sorted).

Domain model. That part of an application that represents the problem domain objects such as bank accounts, clients, deposits, withdrawals, and etc., in a banking application. The usually graphical appearance that the application presents to the user in the form of buttons, labels, etc. (the *user interface*), is connected to the domain model through the *application model*. Note that an application could represent the same domain model using several different user interfaces each requiring its own application model. Similarly, a given user interface could be used with several different domain models, each again requiring its own application model. In a narrower sense, each class representing a part of a problem domain, for example a class representing a bank account, is also called a domain model.

Double dispatching. Executing a message by sending it to the argument object, with the original receiver being used as argument. Used in arithmetic messages to simplify decision making and to make declaration of arithmetic messages more general. As an example, execution of 3 + 5.4 (message + with an integer receiver and a float argument) consists of sending message 5.4 sumFromInteger: 3 (message sumFromInteger: with a float receiver and an integer argument. Whereas the first form would require determining the type of the argument and executing appropriate form of addition, the second form does not require any tests because method sumFromInteger: is declared to deal with integer arguments.

Distinguished instance. Instance of a class that is intended to be unique. As an example, classes True, False, and UndefinedObject each have a single instance.

Dynamic binding. Languages using dynamic binding, such as Smalltalk, do not require the specification of the class to which an identifier denoting a variable or an argument must belong. This binding is performed when an object is assigned to the identifier during program execution. Main advantage: code may be very general because it may be executable with instances of many different classes. Disadvantage: Does not provide the extra dgree of security provided in languages with *static typing* where the compiler can recognize improper use of variables even before the code is executed.

Editor widget. A widget taht can display text on multiple lines and automatically adjust the contents of the lines to its current size. A writeable editor widget allows the user to cut, paste, edit, and perform similar operations, in the case of a code view also to execute code. When the execution ability is missing, the widget is usually referred to as a text editor.

Encapsulation. Bundling of the information held by an object and of its functionality into a single entity. Compare to procedural languages in which data is separated from functionality.

Expression.

File. Binary representation of data or code or a combination of both stored on a secondary storage device such as a disk or a CD ROM, under a user selected file name.

File in. The procedure of reading and automatically compiling and inserting into the library of a file containing the code of a method, a protocol, a class, or a category. The file must be in a special format produced by the *file out* command. The 'file in' command is available in file browsing tools.

File out. Command available in file browsing tools and used to store a method, a protocol, a class, or a category in a file, using a special form expected by the *file in* command. Used for transporting source code from one computer to another. The command is available in various Smalltalk browsers.
Finalization. Actions executed by the program when a window is being closed.
Floating point number. Internal computer representation of numbers with fractional components such as 3.14 or -0.00143. Compare to *integer*. In Smalltalk, represented by several different classes depending on magnitude and number of significant digits.
Framework.
Global variable. A variable accessible to any object and at any time, automatically saved when the Smalltalk environment is saved and automatically restored when Smalltalk is started up. Global variables are stored in the Smalltalk dictionary along with all other globally accessible objects including classes and pool dictionaries. This makes global variables dangerous because naming a global variable with the name of an existing class replaces the class with the global variable and destroys it.
Graphical user interface (GUI) - the collection of windows, buttons, labels, and other widgets seen by the user of a computer application
Icon. Visual representation of a collapsed window. In Smalltalk, the user can assign any image to the icon of a window.
Image file.
Immediate object. Object whose internal representation is not accessed by a pointer. In fact, the 'pointer' is the value of the object. Includes SmallInteger and Character.
Indexable variable. An instance varible or an object whose elements are accessed by index, as in an array.
Information hiding. The rule that internal variables of an object can only be accessed through methods explicitly declared for this purpose in the object's class. As an example, a point's x and y coordinates can be obtained and changed only because class Point contain declaration of messages to do this. Without these methods, the values of x and y would not be accessible.
Inheritance. The mechanism through which all variables and methods declared in a class are valid for all its subclasses as well. As an example, class Collection declares method contents and all its subclasses (Array, Set, Dictionary, etc.) thus also understand this message. Similarly, class ControllerWithMenu declares variable menuHolder and all its subclasses thus automatically inherit it too.
Input field widget. A one line text widget with editing capabilities such as cut, copy, and paste. May be writeable or read only.
Inspector. A Smalltalk tool opening a two-view window listing the components of the object being inspected in one subview, and the value of the selected coonent in the adjacent view.
Instance. Synonym of *object*. A realization of a class with specific values for each instance variable declared or inherited by the defining class. As an example, 3 and 5.1 are instances of class SmallInteger while individual labeled action buttons on a window are instances of class ActionButton.
Instance method. A method declared with the 'instance' button of the browser on. Instance methods can only be sent to instances of a class. Compare with *class methods* which can only be sent to the class itself but are not understood by instances.
Instance variable. Variable holding an object defining one of the parameters of an *instance* of a class.
Each instance of the same class has the same instance variables but each may have a different value.
Compare with *class variable* and *class instance variable*.
Integer. Internal representation of the mathematical concept of an integer, a number without a fractional point. In Smalltalk implemented by several classes for different ranges of magnitude.
Interval.
Iteration.
Keyword
Keyword message
Label widget.
Launcher.
List widget.
Literal.
Loop.
Menu bar widget.
Menu button.

Message.
Metaclass.
Method.
Model.
Mouse buttons.
Multiple inheritance.
MVC paradigm.
Object.
Object file.
Optimization.
Order of evaluation.
Ordered collection.
Overloading.
Palette.
Pane.
Parameter.
Persostent object.
Pointer.
Pool dictionary.
Po-up menu.
Polymorphism.
Private method.
Primitive method.
Private method.
Protocol.
Pseudovariable.
Radio button.
Receiver.
Return object.
Reusability.
Selector.
self.
Semantics.
Set.
Shortcut key.
Signature.
Simulation.
Single inheritance.
Slider widget.
Source code.
Standard protocol names.
State.
State variable.
Statement.
Static binding.
Stream.
String.
Symbol.
Syntax.
Subclass.
Superclass.
super.
Temporary variable.
Text.
Text editor widget.

Text file.
Transcript.
UI.
User interface.
Unary message.
Undeclared variable.
Value holder.
Variable.
View.
Widget.
Wildcard.
Window.
Workspace.