

## **ВЕРИФИКАЦИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА ИНФОРМАЦИОННОЙ СИСТЕМЫ**

**Зиннатулин Ф.Ф. (Университет ИТМО)**

**Научный руководитель – к.т.н., доцент Поляков Владимир Иванович  
(Университет ИТМО)**

В работе предлагается метод верификации вычислительного процесса информационной системы на основе метода верификации вычислительного процесса корпоративного портала.

В настоящее время растущее влияние информационных технологий в обществе требует интеграции программных комплексов, их данных и метаданных, созданных на основе различных концепций, методологий, моделей и подходов, унификация которых не была получена ни ведущими компаниями-разработчиками ПО (Microsoft, IBM, Oracle, SAP, BEA и др.), ни крупными научными коллективами [1].

Проблема анализа качества аппаратного и программного обеспечения становится сегодня все более острой, особенно по мере расширения использования информационных технологий и нанотехнологий в приборостроении [2, 3].

Как известно, информационные системы - это и программное обеспечение, которое выполняет задачи компании, и среда реализации бизнес-функций, и инструмент обеспечения взаимодействия с хранимым объемом данных в компании. Информационные системы (ИС) составляют важную часть программного обеспечения, интегрированы в контур программной инфраструктуры, реализованной в компании.

Структура ИС представляет собой главные функциональные элементы, которые решают задачи ИС для обеспечения выполнения требований к самой ИС. Функциональные элементы ИС состоят из:

1. Блока обеспечения обработки входных данных пользователя.
2. Блока обеспечения вывода информации пользователю.
3. Блока получения входных данных от пользователя.
4. Блока предобработки данных для хранения в базе данных.
5. Блока хранения данных в базе данных.
6. Блока получения и сохранения пользовательских данных в базе данных.
7. Блока резервного копирования данных.

Графическая схема блоков представлена на рис. 1.

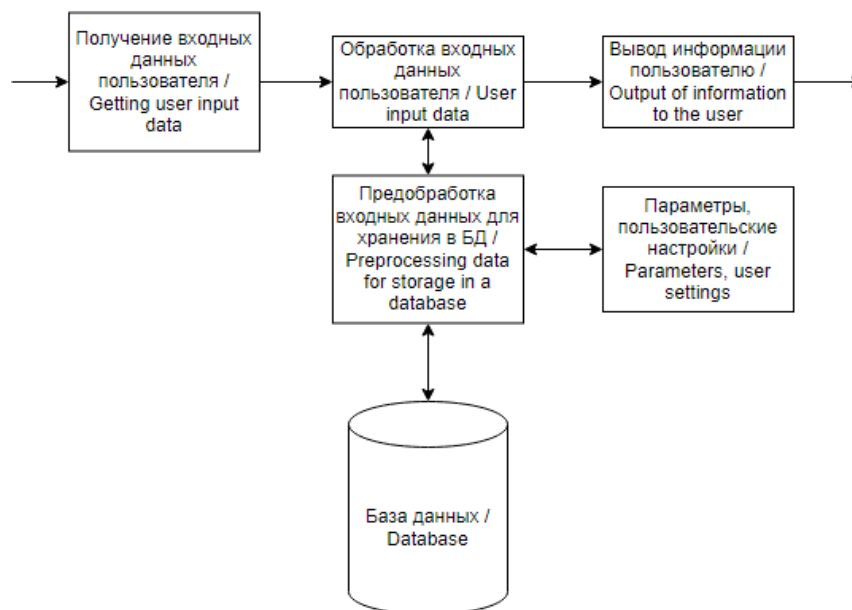


Рисунок 1 - Схема перемещения данных вычислительных процессов информационной системы.

Решение, принимаемое блоком на этапе обработки, зависит от реализации методов функционирования блока. Блок получения данных обеспечивается следующими методами:

1. Проверка корректного наименования.
2. Проверка значения по регулярному выражению.
3. Проверка идентификатора по корректности значения.
4. Проверка номера по диапазону.
5. Проверка дополнительного значения по вычислению типа его значения.

Верификация вычислительного процесса информационной системы рассматривается на основе метода верификации вычислительного процесса корпоративного портала, который рассматривался ранее. Данный метод может применяться в отношении информационных систем (например, 1С:Предприятие), которые могут использоваться в компаниях ЛеруаМерлен или ИКЕА. Вычислительный процесс корпоративного портала состоял в получении общего доступа к информационным ресурсам компании. В случае информационных систем рассматривается уже вычислительный процесс получения доступа к значению на складе (блока обработки входных данных/Userinputdata).

**Требования к верификации.** Для успешной верификации вычислительного процесса получения значения со склада необходимо выполнение следующих требований вычислительного процесса:

1. Наличие численного количества товара на складе.
2. Численное количество товара на складе должно быть ненулевым.
3. Возможность получения численного количества товара со склада.
4. Возможность вывода численного количества товара на пользовательский вывод.

В случае выполнения перечисленных выше требований верификация вычислительного процесса получения значения со склада считается пройденной успешно.

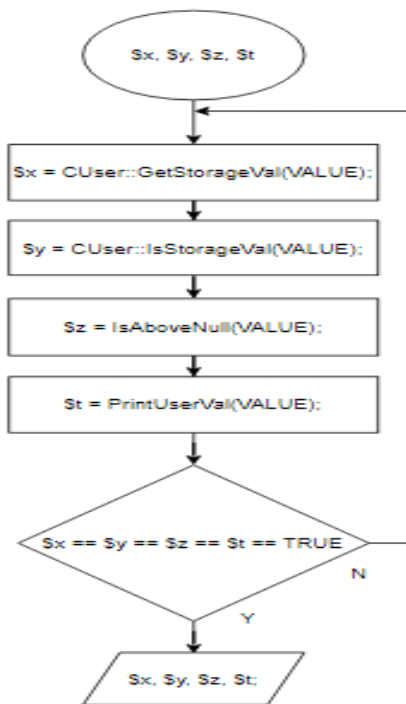


Рисунок 2 - Графоаналитическая модель вычислительного процесса получения значения со склада информационной системы.

Формальные методы все шире используются для обеспечения качества программного обеспечения [6-12]. Графоаналитическая модель вычислительного процесса изображена на рис. 2. Программная модель изображена на рис. 3.

```

1  <?php
2  a:
3  function initValues(){
4      $x = 0, $y = 0, $z = 0, $t = 0;
5  }
6
7  function setValues(VALUE){
8      $x = CUser::GetStorageVal(VALUE);
9      $y = Cuser::IsStorageVal(VALUE);
10     $z = IsAboveNull(VALUE);
11     $t = PrintUserVal(VALUE);
12 }
13
14 initValues();
15 setValues(VALUE);
16
17 if($x == true){
18     if($y == true){
19         if($z == true){
20             if($t == true){
21                 echo $x." ", $y." ", $z." ", $t." ";
22             }
23             else goto a;
24         }
25         else goto a;
26     }
27     else goto a;
28 }
29 else goto a;
  
```

Рисунок 3 - Программная модель верификации вычислительного процесса получения значения со склада информационной системы.

Графоаналитическая модель по сравнению с программной моделью обладает блоком инициализации переменных, 4 блоками со значением переменных, подлежащих дальнейшему тестированию, а также блоком-условием с проверкой переменных. Программная модель по сравнению с графоаналитической моделью обладает функциями setValues() для установки значений проверяемым переменным, а также функции initValues() для инициализации проверяемых переменных. Функция setValues() содержит проверку с помощью функции GetStorageVal() для проверки возможности получения значения со склада, а также IsStorageVal() для проверки наличия численного значения на

складе; функции `IsAboveNull()` для проверки того, что значение на складе больше 0; функции `PrintUserVal()` для проверки того, что значение со склада можно вывести в пользовательский вывод. Далее располагается блок проверяющий значения переменных.

Результатом верификации с помощью графоаналитической модели и программной модели вычислительного процесса является набор успешно выполненных тестов на основе графоаналитической и программной модели. Набор тестов с помощью обеих моделей показал одинаковый истинный результат. Данные результаты логировались в отдельном текстовом файле.

Для графоаналитической модели были следующие входные данные: количество складского запаса равно 3. В ходе верификации на 1 блоке  $\$x = \text{CUser}::\text{GetStorageVal}(\text{VALUE})$  переменной  $\$x$  было присвоено значение 3 (int). Далее на 2 блоке  $\$y = \text{CUser}::\text{IsStorageVal}(\text{VALUE})$  переменной  $\$y$  было присвоено значение TRUE. Далее на 3 шаге  $\$z = \text{IsAboveNull}(\text{VALUE})$  переменной  $\$z$  было присвоено значение TRUE. Далее на 4 шаге  $\$t = \text{PrintUserVal}(\text{VALUE})$  переменной  $\$t$  было присвоено значение 3. Далее на 5 шаге  $\$x == \$y == \$z == \$t == \text{TRUE}$  происходит положительная проверка на равенство TRUE. Далее на 6 шаге  $\$x, \$y, \$z, \$t$  происходит вывод значений 3 (int), TRUE, TRUE, 3. Таким образом верификация была пройдена успешно, т.к. были получены значения переменных в выводе. Ниже представлены 3 других набора входных данных.

**Входные данные: 10. Выходные данные: 10 (int), TRUE, TRUE, 10.**

**Входные данные: 15. Выходные данные: 15 (int), TRUE, TRUE, 15.**

**Входные данные: 20. Выходные данные: 20 (int), TRUE, TRUE, 20.**

Для программной модели были выбраны такие же входные данные как у графоаналитической модели. Количество складского запаса равно 3. После интерпретации функций `initValues()` и `setValues()` запустилась функция `initValues()`, инициализировав нулевым значением переменные  $\$x, \$y, \$z, \$t$ . Значение переменной  $\$x$  стало 0; переменной  $\$y$  стало 0; переменной  $\$z$  стало 0; переменной  $\$t$  стало 0. Далее была запущена функция `setValues()`, которая использует функции `CUser::GetStorageVal()`, `CUser::IsStorageVal()`, `IsAboveNull()`, `PrintUserVal()`. На данном шаге переменная  $\$x$  получила значение 3 (int); переменная  $\$y$  получила значение TRUE; переменная  $\$z$  получила значение TRUE; переменная  $\$t$  получила значение 3. Далее выполнялся блок условий для проверки истинности значений переменных  $\$x, \$y, \$z, \$t$ . Далее выполнялся вывод значений переменных  $\$x, \$y, \$z, \$t$ : 3 (int), TRUE, TRUE, 3. Таким образом верификация была пройдена успешно, т.к. были получены значения переменных в выводе. Ниже представлены 3 других набора входных данных.

**Входные данные: 10. Выходные данные: 10 (int), TRUE, TRUE, 10.**

**Входные данные: 15. Выходные данные: 15 (int), TRUE, TRUE, 15.**

**Входные данные: 20. Выходные данные: 20 (int), TRUE, TRUE, 20.**

### **Заключение**

В статье была исследована верификация вычислительного процесса ИС. В качестве рассмотрения была взята ИС 1С:Предприятие. Вычислительный процесс содержал обращение к складским запасам. Верификация была проведена с помощью графоаналитической модели и программной модели. Данные модели пригодны для верификации вычислительных процессов программных систем. Верификация вычислительного процесса получения складского значения через ИС 1С:Предприятие показала, что все обязательные требования были выполнены.

Поляков В. И. (научный руководитель)

Подпись \_\_\_\_\_

Зиннатулин Ф.Ф. (студент)

Подпись \_\_\_\_\_