

государственный институт электроники и математики. - М.: МГИЭМ, 1998. - 27 с.

3. Путилов Г.П. Концепция построения информационно-образовательной среды технического вуза / М.: МГИЭМ, 1999. – 28 с.
4. Путилов Г.П., Тарасов И.А., Тумковский С.Р. Технология построения виртуальных лабораторных практикумов// Интернет в образовании и технических приложениях. Сб. науч. трудов СИТМО и каф. РТУиС, под общ. ред. Г.П. Путилова,- М.: МГИЭМ, 2000. – с. 42-46.

Реализация математического ядра подсистемы «АСОНИКА-К»

Жаднов В. В. (МГИЭМ), Измайлов А.С. (МГИЭМ)

Подсистема АСОНИКА-К предназначена для расчета показателей надежности и качества РЭС и, следовательно, в этой подсистеме основным блоком является расчетное ядро. На этапе проектирования ядра я столкнулся с некоторыми проблемами. Главной из них было то, что существует много (порядка 30) классов электрорадиоизделий (ЭРИ). Такое разделение на классы обусловлено тем, что разным группам ЭРИ сопоставляются разные математические модели для расчетов показателей их надежности и качества. В целом для расчета этих показателей по внезапным отказам в подсистеме используются 2 модели – это EXP(экспоненциальное) распределение и DN(диффузионное) распределение. EXP-модель – однопараметрическое распределение, параметр которого λ_3 – эксплуатационная интенсивность отказов, а DN-модель – двухпараметрическая модель, параметрами которой являются α_3 – средняя скорость изменения определяющего параметра и v_3 – коэффициент вариации средней скорости изменения определяющего параметра. Но параметры λ_3 , α_3 и v_3 , которые используются для расчетов по EXP и DN - моделям – это параметры, которые должны учитывать все условия эксплуатации элемента (режима его работы, температуры окружающей среды, техническое средство, на борту которого установлен элемент и т.д.). Моделями для расчетов λ_3 и α_3 являются выражения:

$$\lambda_3 = \lambda_0 \cdot \prod_{i=1}^N K_i, \quad \alpha_3 = \alpha_0 \cdot \prod_{i=1}^N K_i, \quad v_3 = v_0, \quad (1)$$

где

λ_0 – базовая интенсивность отказов, α_0 – базовая средняя скорость изменения определяющего параметра, v_0 – базовый коэффициент вариации средней скорости изменения определяющего параметра, K_i – составляющие модели, N – Количество составляющих модели.

Если элемент резервируется, то после данных вычислений производится расчет $\lambda_{эр}$, $\alpha_{эр}$ и $v_{эр}$ – параметров моделей с учетом резервирования. Для каждого типа резервирования формулы не зависят от класса ЭРИ, поэтому они не заслуживают особого внимания. Подробно остановимся на формулах (1) для

расчета λ_3 и α_3 . Несмотря на свою простоту именно они вызывают наибольшее затруднение. λ_0 и α_0 – это базовые параметры ЭРИ (при его работе в нормальных условиях эксплуатации, при номинальном режиме и т.д.). Но обычно условия эксплуатации не совпадают с номинальными, поэтому модели содержат произведение составляющих модели K_i .

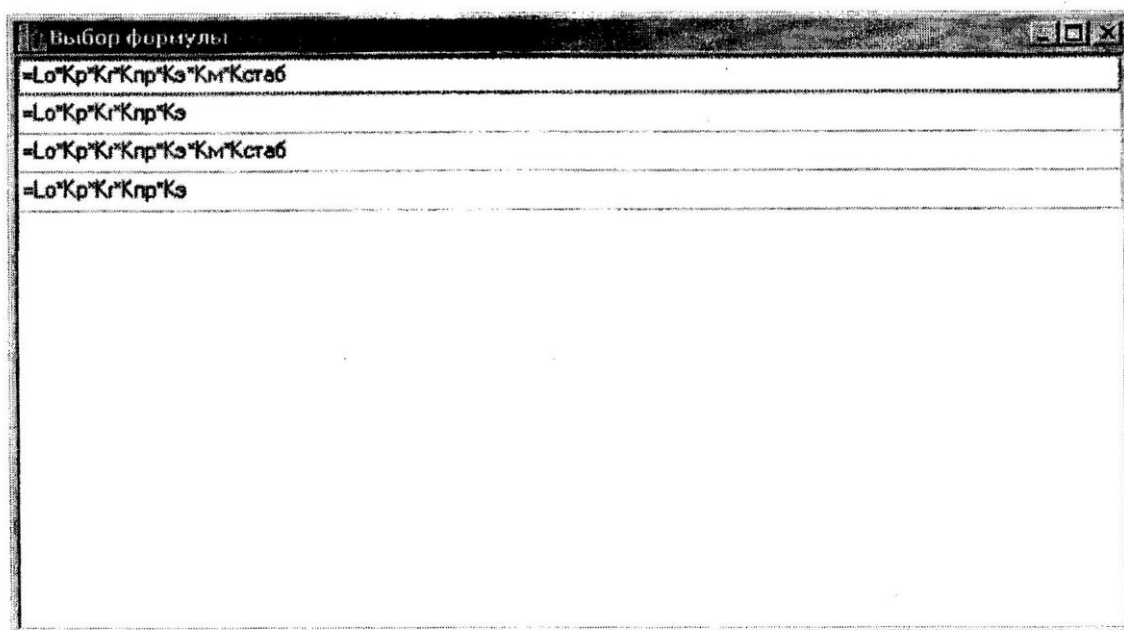
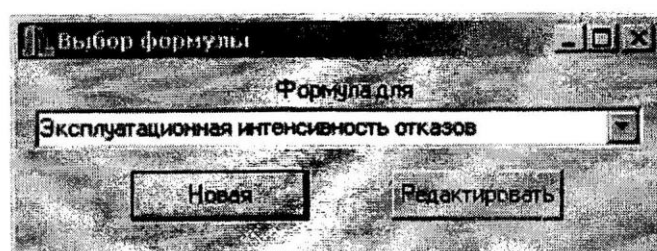
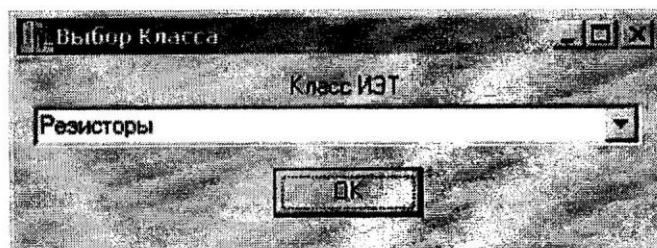
Для каждого класса ЭРИ набор составляющих различен. Каждый класс, в свою очередь, может разделяться на группы, и количество составляющих для каждой группы различно, а их набор берется из набора составляющих группы. Мало того, некоторые составляющие представляются в виде формулы (например такие как K_p – составляющая, зависящая от режима эксплуатации, которая представляется в виде достаточно сложной зависимости от температуры и электрического режима работы). Одно объединяет все эти формулы: использование элементарных арифметических действий и алгебраических функций.

Если эти формулы программировать в теле расчетного ядра, то при выходе нового справочника (они выходят с периодичностью раз в четыре года), придется исправлять исходный код ядра, добавлять или удалять формулы, что само по себе не очень удобно и иногда сложно осуществимо. Поэтому, проанализировав ситуацию, я пришел к выводу, что нужно написать что-то универсальное, расчетный модуль, который рассчитывает любую такую модель без вмешательства в исходный код программы. Подумав, я решил, что можно написать программу, в которой не содержится никаких формул, а содержатся лишь инструкции для чтения формул и расчета по ним, а сами формулы для расчетов содержатся в справочной части базы данных в виде какого-то кода, понятного только расчетному ядру и занимающего небольшую область в таблице, поскольку столбцы таблицы PARADOXa могут содержать не более чем 256 символов. Далее оставалось только придумать систему кодировки и раскодировки формул.

Сам код формул был разработан в достаточно короткий срок, поскольку кодировать пришлось достаточно простые формулы. В этом коде каждый элемент (будь то переменная, знак, функция и т.п.) занимает один байт. В старших трех битах содержится код элемента, в младших пяти – его идентификационный номер, отсюда берутся ограничения на количество элементов одного и того же типа, например, в формуле не может использоваться более 32 констант, 32 переменных и т.д. Также каждой формуле должен сопоставляться массив констант, который занимает 256 байт и также вмещается в столбцы таблиц PARADOXa. Массив переменных не может сопоставляться формуле, так как в нем хранятся значения переменных, которые изменяются для каждого нового расчета по одной и той же формуле. Исходя из этих соображений в справочной части базы данных были созданы еще несколько таблиц, в которых содержится весь список используемых переменных для каждой формулы в определенном порядке, а также для каждой переменной местонахождение ее значения в массиве переменных, содержащемся в теле подсистемы, а также таблицы с запросами к справочной части базы данных для того, чтобы поместить в массив эти значения, полученные из базы. Далее нужно

было написать две программы: кодировщик формул и программу расчета формулы.

Кодировщик формул – программа, содержащаяся в интерфейсе администратора подсистемы АСОНИКА-К и предназначенная для создания кода формул по ее обычному математическому представлению. Эта программа была реализована в среде программирования Borland C++ Builder. Ее вид приведен на рис. 1.



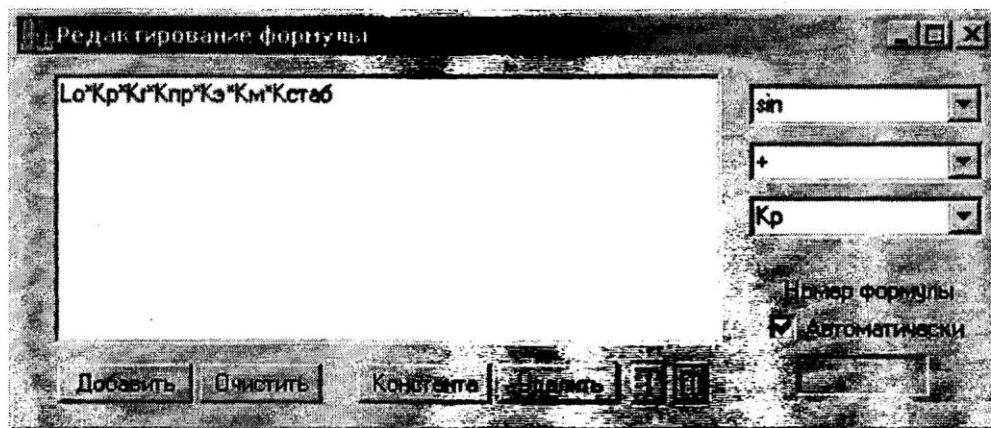
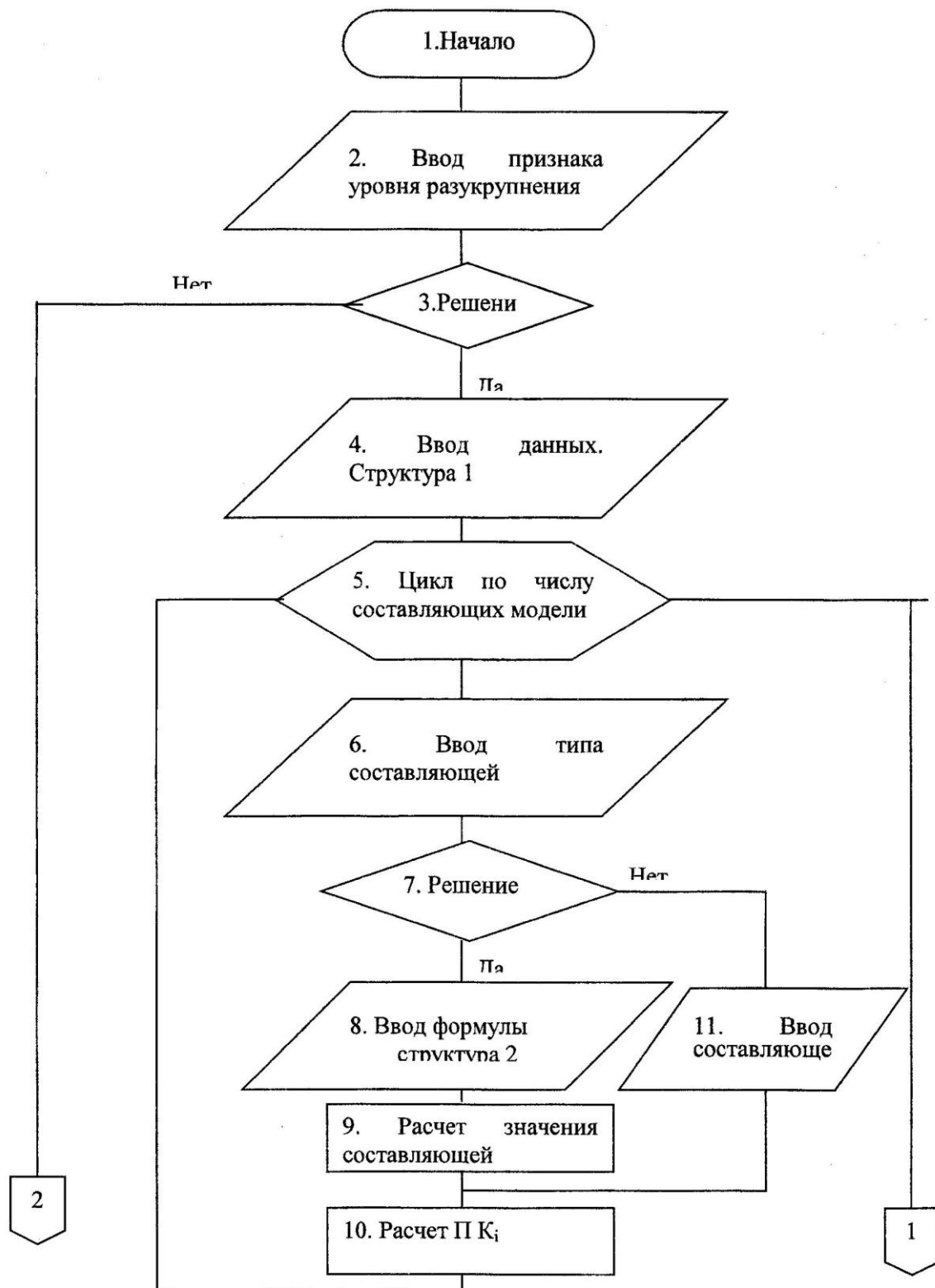


Рис. 1.

Программа расчета формулы – программа, которая предназначена для получения числового значения математического выражения, представленного в виде разработанного кода с использованием переданного ей списка значений переменных. Эта программа была реализована на языке программирования Borland C++ и подключена к расчетному ядру в виде процедуры, так как расчетный модуль использует ее непосредственно и иногда несколько раз для расчета одного показателя безотказности.

Также была написана небольшая процедура, предназначенная для отображения кода формулы в понятном для пользователя виде, так как в некоторых случаях требуется показать формулу для расчета пользователю.

Весь комплекс программ и процедур был подключен к подсистеме АСОНИКА-К, а расчетное ядро протестировано с помощью предыдущей версии подсистемы. Блок-схема расчетного ядра приведена на рис. 2. и рис 3.



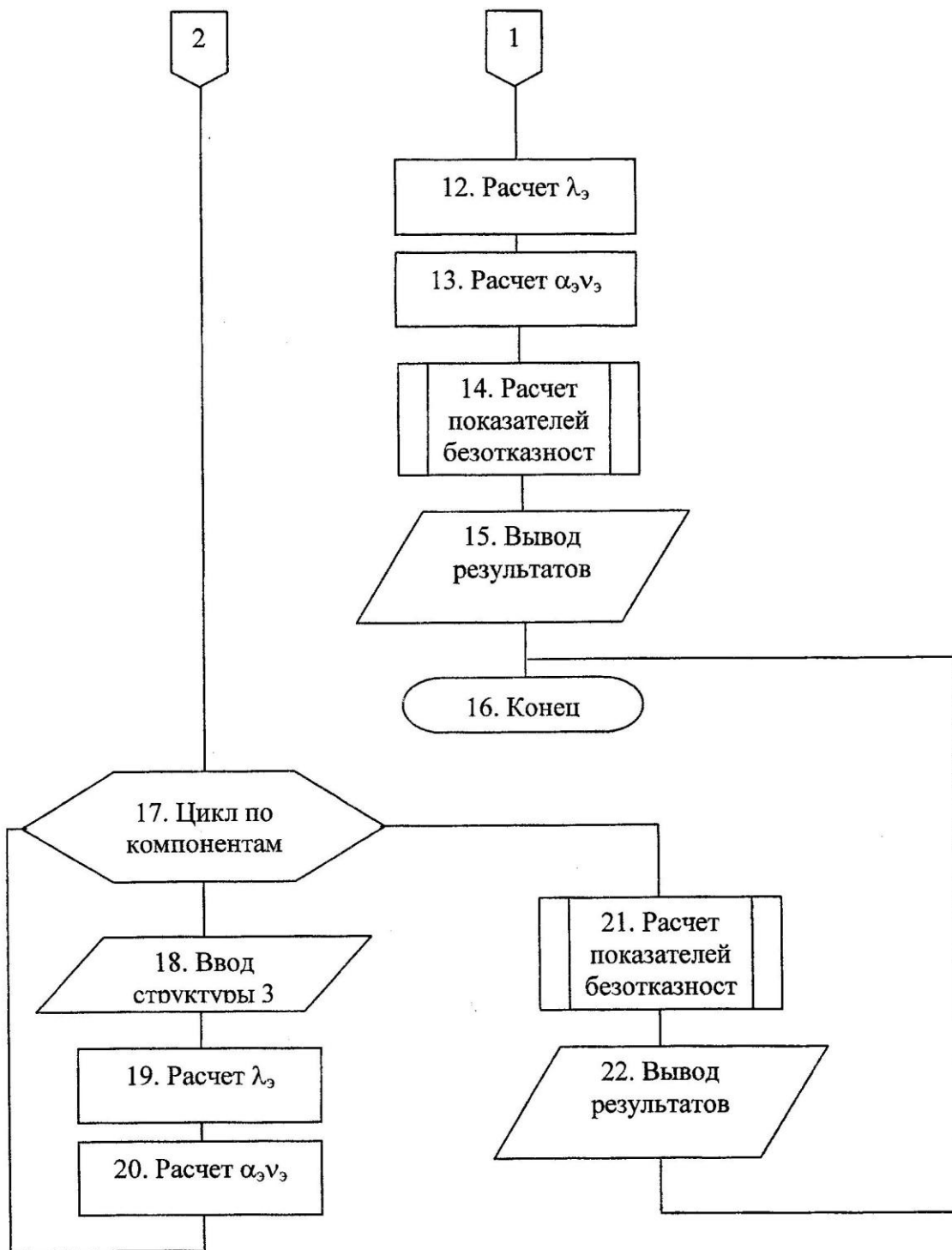


Рис. 2. Блок схема математического модуля по внезапным отказам.

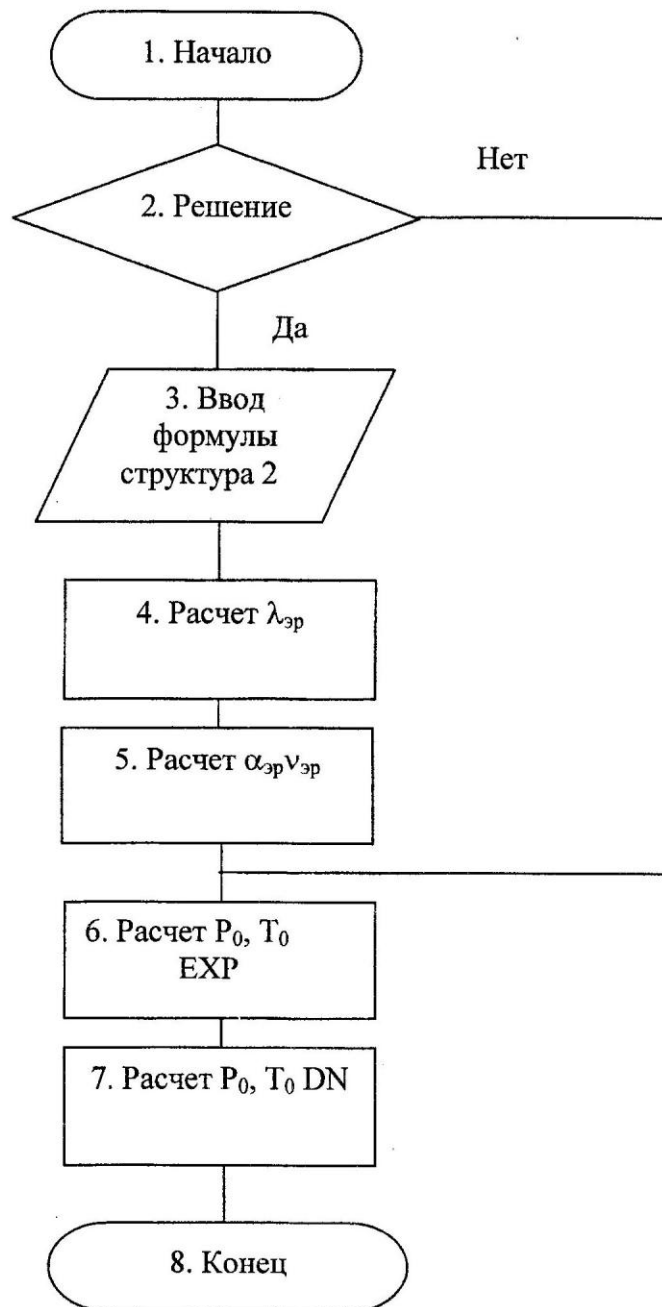


Рис. 3. Блок схема блока расчета показателей безотказности (блок 14).