

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Линник Оксана Владимировна

Должность: Руководитель СФТИ НИЯУ МИФИ

Дата подписания: 06.04.2023 15:25:29

Уникальный программный ключ:

d85fa2f259a0913da9b08299985891756420181f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»

Снежинский физико-технический институт–

филиал федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

(СФТИ НИЯУ МИФИ)

УТВЕРЖДАЮ

Зам. руководителя по учебной
и научно-методической работе

П.О. Румянцев

« 04 » 04 2022 г



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.11 Компьютерное моделирование

Специальность 11.02.16 «Монтаж, техническое обслуживание и ремонт
электронных приборов и устройств»

Квалификация выпускника Специалист по электронным приборам и
устройствам

Форма обучения очная

Снежинск

2022

Фонд оценочных средств дисциплины ОП.11 «Компьютерное моделирование» разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования (СПО) по специальности 11.02.16 «Монтаж, техническое обслуживание и ремонт электронных приборов и устройств».

Организация-разработчик: Снежинский физико-технический институт – филиал Федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ».

Разработал: Орлов Сергей Григорьевич

Содержание

Общие положения	4
1 Формы промежуточной аттестации по дисциплине.....	5
2 Результаты освоения дисциплины, подлежащие проверке	5
2.1 Профессиональные и общие компетенции	5
3 Оценка освоения дисциплины	7
3.1. Примерный перечень заданий для проведения зачётного тестирования с целью оценки освоения дисциплины «ОП.11 Компьютерное моделирование».....	7
3.2. Вопросы для подготовки к экзамену по дисциплине «ОП.11 Компьютерное моделирование».....	13
3.3. Тематика курсовой работы по дисциплине «ОП.11 Компьютерное моделирование».....	14
4 Практические и лабораторные занятия.....	15
4.1. Критерии оценки практических и лабораторных занятий	15
4.2 Задания для практических и лабораторных занятий.....	17

Общие положения

В результате освоения учебной дисциплины ОП.11 «Компьютерное моделирование» Общепрофессиональный цикла обучающийся должен:

Знать:

- основные приёмы и методы автоматизированной обработки информации;
- базовые системные продукты и пакеты прикладных программ; в области применения имитационного моделирования;
- структуру SimInTech, состав и структуру главного меню;
- примеры непроизводственных и производственных систем.

Уметь:

- использовать базовые системные продукты и пакеты прикладных программ;
- составлять модель;
- осуществлять моделирование;
- запускать, сохранять, открывать файлы в SimInTech;
- моделировать задачи непроизводственных и производственных систем с применением SimInTech.

Владеть:

- методами реализации простых математических моделей на ЭВМ, создавая алгоритмы и программы на языке SimInTech;
- возможностями табличного процессора Excel для проведения несложных математических расчётов и иллюстрирования результатов математического моделирования графиками и столбчатыми диаграммами;
- средством «Поиск решения» табличного процессора Excel для решения задач линейного и нелинейного программирования;
- системой SimInTech для проведения несложных математических расчётов, графического иллюстрирования результатов моделирования.

1 Формы промежуточной аттестации по дисциплине

Учебный семестр	Формы промежуточной аттестации и текущего контроля
3	Выполнение и защита практических и лабораторных работ
4	Выполнение и защита практических и лабораторных работ Дифференцированный зачет
5	Экзамен Курсовая работа

2 Результаты освоения дисциплины, подлежащие проверке

2.1 Профессиональные и общие компетенции

а) общих (ОК):

– ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

Показатели оценки результата - проведение вычислительных экспериментов с использованием компьютерных моделей

б) профессиональных (ПК):

– ПК 3.1. Разрабатывать структурные, функциональные и принципиальные схемы простейших электронных приборов и устройств.

Показатели оценки результата:

- овладение навыками анализа рабочей ситуации, самоанализа и коррекции результатов собственной работы;

- демонстрация готовности нести ответственность за результаты своей работы.

Перечень основных показателей оценки результатов, элементов практического опыта, знаний и умений, подлежащих текущему контролю, промежуточной аттестации и формы контроля.

Наименование основных показателей оценки результатов (ОПОП) А	Наименование элемента практического опыта Б	Наименование элемента умение В	Наименование элемента знание Г	Форма контроля вид аттестации Д
Моделирование и его виды	Быстрое ориентирование в основных видах моделирования.	Различать виды моделирования и их классифицировать.	Основные понятия и виды моделирования. Различные классификации моделей.	Практические работы, экзамен, диф. зачет, курсовая работа
Имитационное моделирование	Опыт использования программных средств имитационного моделирования.	Использовать основные методы имитационного моделирования.	Основные понятия и принципы имитационного моделирования.	
Детерминированные модели. Системный подход в моделировании	Обладать навыками разработки и анализа моделей, навыком проведения вычислительного эксперимента.	Разрабатывать и анализировать математические модели в различных областях деятельности.	Примеры математических моделей в различных областях науки и практики. Основы системного подхода в моделировании.	
Программная среда SimInTech	Обладать опытом работы и опытом работы в программной среде SimInTech	Быстро и эффективно ориентироваться в программной среде.	Знать основные функциональные возможности программной среды и языка SimInTech.	

3 Оценка освоения дисциплины

3.1. Примерный перечень заданий для проведения зачётного тестирования с целью оценки освоения дисциплины «ОП.11 Компьютерное моделирование»

Тестовый контроль по дисциплине используется для промежуточной оценки и степени усвоения учебного материала студентами.

Основной задачей тестирования является получение достоверных и объективных результатов о качестве подготовки обучающихся в целях установления их соответствия требованиям государственного образовательного стандарта. При тестировании используется банк контрольных тестовых материалов, который содержит тестовые задания, сформированные в тестовый комплект по дисциплине. В комплекте материалов теста реализована закрытая форма тестового задания с выбором одного правильного ответа. В комплекте материалов теста реализована закрытая форма тестового задания с выбором правильного ответа. Комплект тестовых заданий по дисциплине представлен в Приложении А.

На проведение теста отводится 20 минут. В процессе тестирования студентам разрешается пользоваться тестовым материалом, ручкой, калькулятором. Использовать в качестве калькулятора сотовые телефоны не разрешается.

Комплект тестовых материалов для дифференцированного зачета по дисциплине :

- 1) Желаемая модель системы это ...
 - a) Задача;
 - b) Результат;
 - c) Цель;
 - d) Прогноз.
- 2) Изображение, представление объекта, системы, процесса в некоторой форме, отличной от реального существования называют

- a) системой;
- b) графиком;
- c) структурой;
- d) моделью.

3) Какие модели дают внешнее представление об оригинале и большей частью служат для демонстрационных целей?

- a) математические;
- b) аналитические;
- c) геометрические;
- d) физические.

4) Какие модели отражают подобие между оригиналом и моделью не только с точки зрения их формы и геометрических пропорций, но и точки зрения происходящих в них основных процессов?

- a) математические;
- b) аналитические;
- c) геометрические;
- d) физические.

5) Модель – это...

a) материальный или воображаемый объект, который в процессе познания замещает реальный объект, сохраняя при этом его существенные свойства;

b) реальный объект, который отражает существенные особенности (свойства) изучаемого объекта;

c) воображаемый объект, сохраняющий некоторые свойства;

6) Какого требования к моделям не хватает в списке: конечность; полнота (информативность); упрощённость; гибкость; трудоёмкость разработки модели

- a) тестируемость;
- b) изменяемость;
- c) адекватность;

d) непредсказуемость.

7) Выберите правильное определение полноты (информативности) модели.

a) понятная для любого исследователя;

b) предоставление исследователю всей необходимой информации об объекте в рамках гипотез, принятых при построении модели;

c) предоставление исследователю информации об объекте;

d) описывает объект большим объёмом информации.

8) Какой класс моделей по области их использования лишний в списке:

учебные модели; игровые модели; производственные модели; исследовательские модели; опытные модели; имитационные модели.

a) учебные модели;

b) имитационные модели;

c) игровые модели;

d) производственные модели.

9) Что не соответствует понятию учебной модели.

a) наглядные пособия;

b) тренажёры;

c) обучающие программы;

d) математические формулы.

10) Выберите правильное определение понятия игровой модели.

a) это игры исследователей на компьютере;

b) это экономические, военные, деловые игры. Они репетируют поведение объекта в различных ситуациях.

c) это использование игрушек при построение модели;

d) это игра актёров на сцене.

11) Какая из перечисленных моделей не является опытной моделью

a) уменьшенная копия;

b) увеличенная копия;

c) резиновая модель;

d) вербальная модель.

12) Выберите классы моделей способу представления.

a) натурную (материальную) модель;

b) информационные модели;

c) вербальная модель;

d) знаковая модель;

e) математическая модель.

13) Что из перечисленного не относится к информационной модели.

a) словесное описание;

b) чертёж;

c) рисунок;

d) формулу;

e) уменьшенная копия объекта.

14) Что из перечисленного не относится к вербальной модели.

a) модель в мысленной форме;

b) модель в математической форме;

c) модель в разговорной форме.

15) Что из перечисленного не относится к знаковой модели.

a) написанная на русском языке;

b) написанная на английском языке;

c) написанная на математическом языке;

e) написанная на языке C++.

16) Что из перечисленного не относится к геометрической модели.

a) диаграмма;

b) алгоритма решения;

c) блок-схема алгоритма решения задачи;

d) картинка;

e) граф.

17) Что из перечисленного не является моделями по характеру отображаемых свойств объекта моделирования

- a) структурные;
- b) multifunctional;
- c) functional.

18) Which of the models corresponds to the model, in which only input and output connections of the modeled object with the environment are specified.

- a) model «white box»;
- b) model «black box»;
- c) model «green box».

19) Which of the listed does not belong to models taking into account the time factor.

- a) time models;
- b) dynamic models;
- c) long modeled models;
- d) static models.

20) Which of the listed models does not belong to models by the character of change over time.

- a) discontinuous models;
- b) continuous models;
- c) discrete models;
- d) constant models.

21) Which stage of mathematical modeling is superfluous in the given list: 1-determination of modeling goals; 2-ranking of parameters; 3-choice of mathematical description. 4-choice of research method 5. preparation of the method for research 6. conducting research. 7. analysis of modeling results.

- a)1
- b)3
- c)5
- d)7

22) Какой из перечисленных типов математических моделей с точки зрения целей моделирования лишней. 1-описательные модели. 2-оптимизационные модели. 4-Игровые модели. 5-расчетные модели 6. имитационные модели.

- a)1
- b)3
- c)5
- d)2

Ключ к тесту

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.
C	D	C	D	A	C	D	D	D	B	D
12.	13.	14.	15.	16.	17.	18.	19.	20.	21.	22.
все	E	B	A,B	B,D	B	B	A,C	A,D	C	C

Каждое задание оценивается в 0,5 до 3 баллов. Весь тест оценивается в 14 баллов (100%) Перевод итогового балла в оценку осуществляется согласно таблице соответствия:

Баллы	Процент правильных ответов	Оценка
12-14	85% - 100%	«Отлично»
10-11	70% - 84%	«Хорошо»
8-9	60% - 69%	«Удовлетворительно»
0-8,5	0% - 59%	«Неудовлетворительно»

Итоги тестирования позволяют получить оценку усвоения отдельных тем, а также элементов знаний, умений, навыков, профессиональных компетенций. Преподаватель, анализируя результаты тестирования, организует коррекционную работу и внесение дополнений в конкретный учебный материал.

3.2. Вопросы для подготовки к экзамену по дисциплине «ОП.11 Компьютерное моделирование»

1. Дайте определение модели.
2. Перечислите требования, которым должна удовлетворять модель.
3. Дайте определение понятия адекватности модели.
4. Дайте определение понятия конечности модели.
5. Дайте определение понятия полнота (информативность) модели.
6. Дайте определение понятия упрощенности модели.
7. Дайте определение понятия гибкости модели.
8. Дайте определение понятия учебные модели.
9. Дайте определение понятия игровой модели.
10. Дайте определение понятия опытной модели.
11. Дайте определение понятия натурной (материальной) модели.
12. Дайте определение понятия информационной модели.
13. Дайте определение понятия вербальная модель.
14. Дайте определение понятия математическая модель.
15. Дайте определение понятия функциональной модели.
16. Дайте определение понятия модели «черный ящик».
17. Дайте определение понятия модели «белый ящик
18. Дайте определение понятия модели «серый ящик».
19. Дайте определение понятия непрерывной модели
20. Дайте определение понятия дискретной модели.
21. Перечислите классы моделей по способу реализации
22. Дайте определение понятия компьютерная модель.
23. Дайте определение понятию моделирования.
24. Дайте определение понятия математическое моделирование
25. Дайте определение понятию описательная математическая модель.
26. Дайте определение понятию оптимизационной модели.
27. Дайте определение понятию игровая модель.
28. Дайте определение понятию имитационной модели.

29. Дайте определение понятию случайная величина.

30. Дайте определение понятие компьютерного моделирования

3.3. Тематика курсовой работы по дисциплине «ОП.11 Компьютерное моделирование»

Тематика курсовой работы «Оптимизационная модель нагревателя воды накопительного типа». Каждому студенту персонально выдаются свои исходные данные.

4 Практические и лабораторные занятия

Лабораторные работы нацелены на закрепление и систематизацию полученных знаний студентами при выполнении лабораторных работ. Лабораторные работы содержат контрольные вопросы, расчётные задания, позволяющие подготовиться студентам к выполнению лабораторных работ и оформить отчет.

4.1. Критерии оценки практических и лабораторных занятий

При выставлении баллов за работу в рабочей тетради учитываются:

- уровень освоения студентом теоретического материала (ответы на контрольные вопросы);
- умения студента разбираться в схемах и собирать их;
- умения студента использовать теоретические знания при выполнении лабораторной работы.

Кол-во баллов	Шкала оценивания	Оценка
3	Даны полные, развёрнутые ответы на контрольные вопросы, раскрыты основные положения вопросов. Студент отлично разбирается в электрических схемах, правильно и быстро собирает их. Студент умеет увязывать теорию с практикой (правильно использует теоретический материал для оформления отчета).	«Зачтено»
2	Даны полные, развернутые ответы на контрольные вопросы могут быть допущены 2-3 неточности или незначительные ошибки, исправленные студентом после проверки ответов преподавателем. Студент хорошо разбирается в электрических схемах, правильно собирает их. Могут быть допущены 2-3 неточности или незначительные ошибки, исправленные студентом после проверки схемы преподавателем Студент умеет увязывать теорию с практикой (правильно использует теоретический материал для оформления отчёта, но содержание отчёта имеет некоторые неточности и требует уточнения и комментария со стороны преподавателя).	

Кол-во баллов	Шкала оценивания	Оценка
1	<p>Даны недостаточно полные и недостаточно развернутые ответы на контрольные вопросы. Допущены ошибки в раскрытии понятий, употреблении терминов.</p> <p>Студент неплохо разбирается в электрических схемах, но собирает их с множеством ошибок, которые исправляются студентом после проверки схемы преподавателем.</p> <p>Студент недостаточно умеет увязывать теорию с практикой (использует теоретический материал для оформления отчета с ошибками, которые исправляются после проверки отчета преподавателем).</p>	
0	<p>Отсутствуют ответы на контрольные вопросы.</p> <p>Студент плохо разбирается в электрических схемах, собирает их с множеством ошибок, которые исправляются студентом после проверки схемы преподавателем.</p> <p>Студент не умеет увязывать теорию с практикой</p>	«Не зачтено»

4.2 Задания для практических и лабораторных занятий

Лабораторная работа №1

Тема: Генерация псевдослучайных последовательностей. Метод Монте-Карло.

Задание: На лабораторных занятиях выполняется примерный вариант задания с помощью электронной таблицы (ЭТ). На самостоятельную работу планируется выполнение индивидуального варианта по образцу задания проделанного на занятиях.

Выполнение работы разбивается на 10 этапов:

1. На 3-х листах ЭТ производятся вычисления последовательностей псевдослучайных чисел с помощью методов Фон-Неймана, мультипликативного и стандартных генераторов.

2. В ЭТ производится расчет и построение гистограммы распределения и других статистических характеристик данных последовательностей.

3. Проводится анализ таблицы и сравнение качества полученных последовательностей псевдослучайных чисел. При наличии определяется период.

4. С помощью метода Монте-Карло определяется качество равномерности распределения полученных последовательностей псевдослучайных чисел.

5. На отдельных листах ЭТ несколько раз копируем результаты расчета полученных методами мультипликативного и стандартного генератора последовательностей псевдослучайных чисел. Эти данные используем как опорные для метода Монте-Карло.

6. С помощью метода Монте-Карло вычисляем значение определенного интеграла несколько раз (с увеличением числа точек) и сравниваем полученные значения с точным значением.

7. На отдельных листах ЭТ строим 2 раза последовательности с нормальным распределением, полученным на основе данных расчета

методами мультипликативного и стандартного генератора. Строим гистограмму распределения.

8. На отдельных листах ЭТ строим 2 раза последовательности с экспоненциальным распределением, полученным на основе данных расчета методами мультипликативного и стандартного генератора. Строим гистограмму распределения.

9. На отдельных листах ЭТ строим 2 раза последовательности с треугольным распределением, полученным на основе данных расчета методами мультипликативного и стандартного генератора. Строим гистограмму распределения. 10. Выполнение самостоятельного задания исследования с использованием индивидуального варианта.

Методические рекомендации

Суть метода Монте-Карло:

1. строится последовательность случайных чисел;
2. для каждого числа проверяется ряд условий, которые связывают его с некоторым событием (выполнение условий называют благоприятным исходом);
3. вычисляется отношение числа благоприятных исходов к общему числу шагов и полученное значение связывается со значением некоторой физической, математической или экономической величины

Генераторы случайных чисел разделяются на алгоритмические и аппаратные.

Алгоритмические генераторы работают быстро, но получается последовательность псевдослучайных чисел.

Аппаратные генераторы очень медленные, но "случайность" чисел последовательности хорошая.

Алгоритмические генераторы случайных чисел – это последовательные вычисления с помощью системы математических формул. Например, в методе чисел Фибоначчи с запаздыванием, разработанном еще в

1958, каждое следующее число вычисляется через два предыдущих по формуле:

$$X(n) = (X(n - 24) + X(n - 55)) \bmod 16,$$

для $n = 56, 57, \dots$, а первые 55 чисел выбираются специальным образом.

Простейшим аппаратным генератором случайных чисел в диапазоне от 1 до 6 является игральная кость, у которой рабочий период бесконечен.

Примером компьютерной аппаратной генерации случайных чисел может быть процессор, а схемой генерации – отслеживание случайных колебаний температуры процессора в установившемся режиме работы.

Во всех современных языках программирования есть датчики (операторы) случайных чисел. В языке Pascal это

`Random(x);`

– возвращает случайное число от 0 до x.

Например,

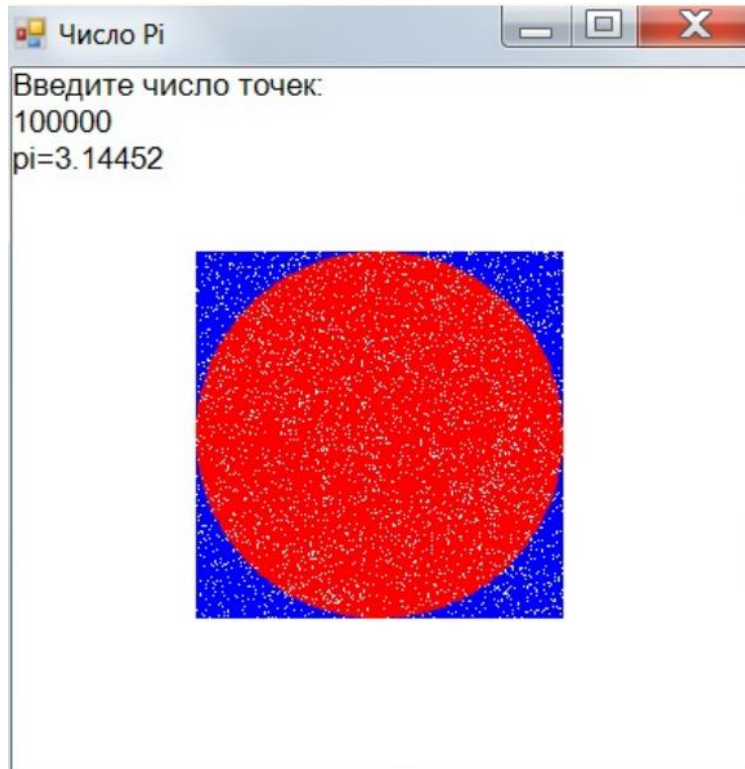
`Randomize;`

`For k := 0 to 100000 do`

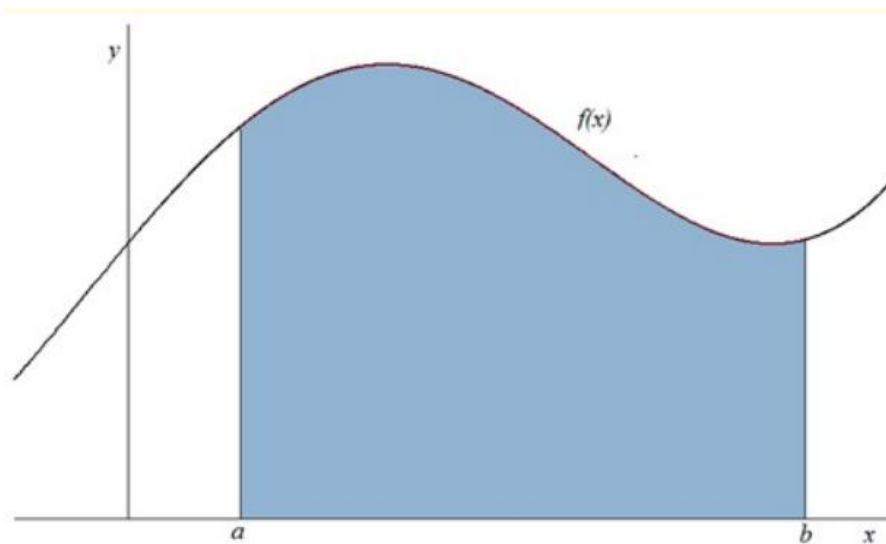
`Random(x);`

```
PascalABC.NET
Файл  Правка  Вид  Программа  Сервис  Модули  Помощь
•Program1.pas*
writeln('Введите число точек: ');
readln(N);
writeln(N);
setwindowsize (640,480);
randomize; // запускаем генератор случайных чисел
Nc := 0;
for k:=1 to N do
begin
x:=random; // задаем случайную точку (x,y)
y:=random; // в квадрате 0<x<1, 0<y<1
if ((x-0.5)*(x-0.5)+(y-0.5)*(y-0.5) <= 0.25)
then
begin
Nc := Nc + 1;
setwindowtitle ('Число Pi');
a:=trunc(x*100)+100;
b:=trunc(y*100)+100;
SetPixel(a,b,clRed);
end
else
begin
a:=trunc(x*100)+100;
b:=trunc(y*100)+100;
SetPixel(a,b,clBlue);
end
end
end
Окно вывода
```

Наилучший результат $\pi = 3.1415$ достигается при $N = 1000000$



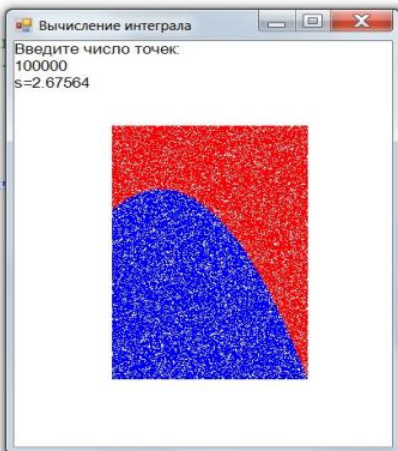
Вычисление определенного интеграла $\int_a^b f(x)dx$ сводится к вычислению площади фигуры, ограниченной осью x , вертикальными прямыми $x = a$, $x = b$, а также графиком функции $f(x)$.



Вычисление интеграла $\int_0^2 (x^2 - x + 1) dx$ Его точное значение равно 2.66(6), а значение, вычисленное методом Монте-Карло равно 2.664.

```

PascalABC.NET
Файл  Правка  Вид  Программа  Сервис  Модули  Помощь
Program1.pas*  •Program2.pas* [Запуцен]
Nc := 0;
for k:=1 to N do
begin
  x:=random*2; // задаем случайное число
  y:=random*3; // в квадрате 0
  if (y < x*x - x+1)
  then
  begin
    Nc := Nc + 1;
    setwindowtitle ('Вычисление интеграла');
    a:=trunc(x*100)+100;
    b:=trunc(y*100)+100;
    SetPixel(a,b,clRed);
  end
  else
  begin
    a:=trunc(x*100)+100;
    b:=trunc(y*100)+100;
    SetPixel(a,b,clBlue);
  end;
end;
s :=Nc/N*6;
writeln('s=',s);
end.
  
```



Метод Монте-Карло позволяет моделировать любой процесс, на протекание которого влияют случайные факторы. Используется для вычисления многомерных объемов областей сложной формы, а также задач, допускающих геометрическую формулировку.

Общим недостатком метода является невысокая точность получаемых результатов. Метод Монте-Карло эффективен при решении задач, где

результат нужен с небольшой точностью, например, в пределах от 10^{-2} до 10^{-5} .

Лабораторная работа №2

Целью данной лабораторной работы является:

1. Получить общее представление о системе SimInTech.
2. Познакомиться с основными методами и приемами работы в этой среде.
3. Изучить возможности встроенной в пакет справочной системы.
4. Рассмотреть ряд примеров проектов, включенных в состав пакета

Методические рекомендации

Среда динамического моделирования технических систем SimInTech является гибким и мощным средством для разработки сложных расчетных моделей. Создание таких моделей требует специального инструмента для подготовки модели и возможности ее разделения на более простые части для удобства коллективной работы. В комплексной модели могут быть объединены расчеты физических процессов, рассчитываемых по различным расчетным кодам (тепло-гидравлическим, электрическим и др.).

Концепция создания комплексной модели объекта в среде SimInTech

Одной из основных особенностей использования SimInTech для создания комплексных моделей является идеология использования «Базы данных сигналов» — структурированного списка переменных, обеспечивающих обмен расчетными значениями между расчетными схемами в единой модели.

База данных сигналов SimInTech является объектной и обеспечивает пользователю удобное решение следующих задач:

- объединение нескольких расчетных схем в единую модель;

- обеспечение возможности векторной обработки сигналов для типовых алгоритмов управления;
- объектно-ориентированное проектирование модели технических систем;
- автоматизацию создания и обработки переменных в комплексных моделях.

При создании модели с использованием базы данных сигналов возможно разделение сложной комплексной модели на произвольное количество.

Свойство:

- Количество портов – количество входных сигналов блока.

Примечания:

- Блок формирует один выходной сигнал.
- Именованные свойства могут быть заданы как локальные переменные модели (субмодели) во вкладке «Параметры» и как глобальные сигналы проекта при помощи пункта главного меню «Сервис → Сигналы», или как внешние сигналы проекта при помощи подключаемой базы сигналов.

Блок «Демультимплексор»

Этот блок векторизован, реализует “расщепление” векторного входного сигнала в отдельные выходные сигналы. Выходами могут быть как скалярные, так и векторные (“многожильные”) сигналы.

По умолчанию данный блок реализует демультимплексирование 2-х “жильного” входного векторного сигнала в скалярные сигналы:

$$y_1(t) = u_{1,1}(t),$$

$$y_2(t) = u_{1,2}(t),$$

где $y_1(t)$, $y_2(t)$ – первый и второй выходной скалярный сигнал блока, а $u_{1,1}(t)$, $u_{1,2}(t)$ – 1-й и 2-й элементы входного векторного сигнала.

Свойства:

- Массив размерностей выходов – массив целых чисел, указывающих размерности векторов каждого из выходных сигналов блока.

Примечания:

- Блок обрабатывает один входной сигнал.

Именованные свойства могут быть заданы как локальные переменные модели (субмодели) во вкладке «Параметры», как глобальные сигналы проекта при помощи пункта главного меню «Сервис → Сигналы», или как внешние сигналы проекта при помощи подключаемой базы сигналов.

Лабораторная работа №3

Целью данной лабораторной работы является:

1. Знакомство с взаимодействием проектов через общую базу данных сигналов.
2. Построение распределенной модели с обменом данными через общую базу.
3. Изучение методов сетевого расчета в ручном и автоматизированном режимах.
4. Получение навыков работы с векторной обработкой сигналов.

Задание на лабораторную работу

Если все рекомендации, изложенные в методических рекомендациях, выполнены, то в результате имеется отлаженная комплексная модель на базе пакета проектов (папка `tank_db_1`) и работоспособная распределенная модель (папка `server`).

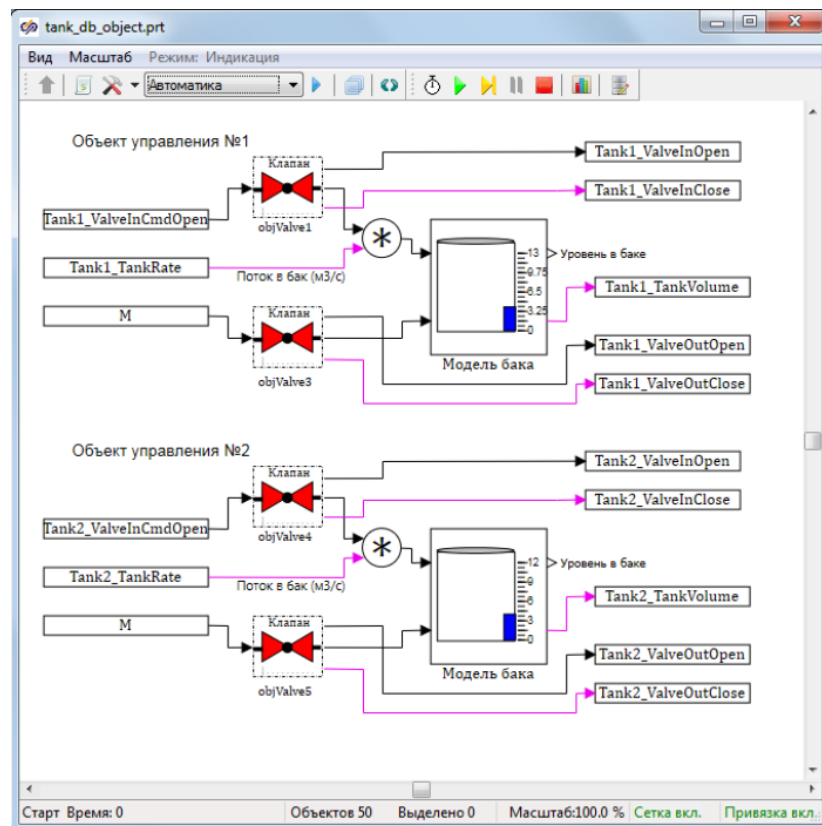
Задания, которые предстоит выполнить, потребуют модификации и изменения в каждой из этих моделей. При этом работа с пакетом проектов по

расширению его функциональности способствует получению более глубоких навыков работы с общесистемной базой данных, а задания по изменению распределенной модели должны дать представление об имеющихся в SimInTech более продвинутых способах запуска сетевого расчета. Исходя из этого подхода к изучению комплексных и распределенных моделей, в рамках данной лабораторной работы необходимо выполнить две группы заданий.

1. Задания по работе с пакетом проектов и его модификации

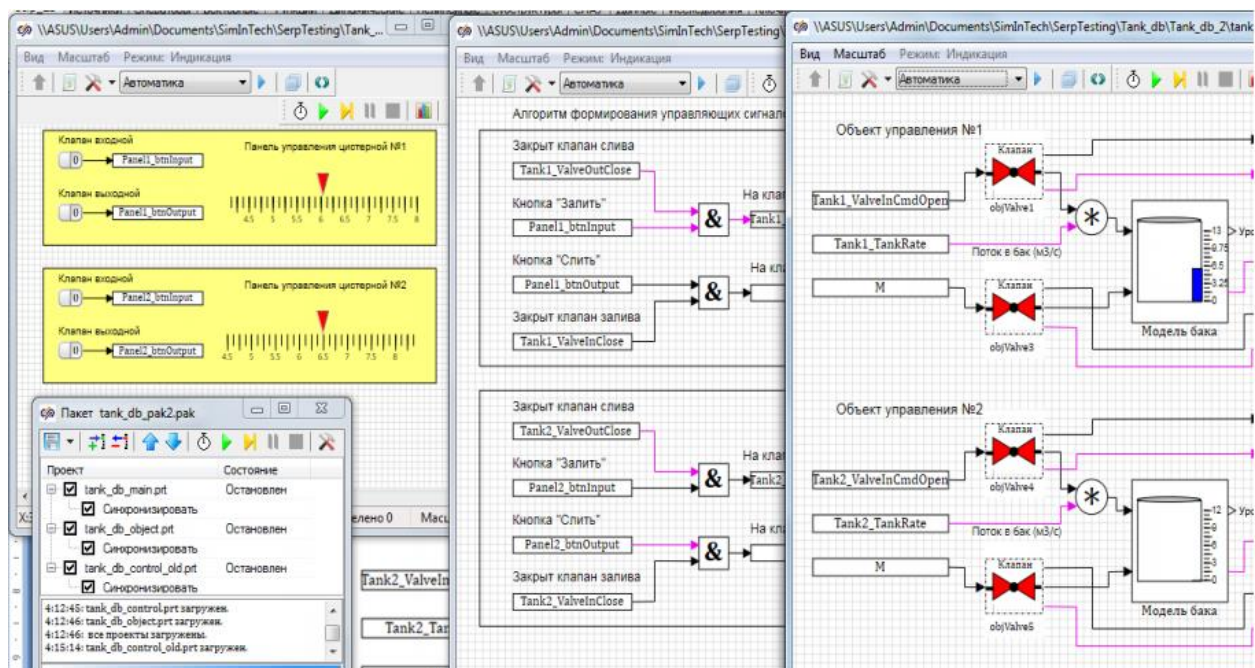
1. Скопировать папку tank_db_1 в папку tank_db_2, переименовать файл пакета в tank_db_pak2.pak и дальше работать с этим пакетом.

2. Расширить функциональность проекта, который теперь должен работать сразу с двумя объектами управления. Для этого в проекте tank_db_object.prt скопировать полностью фрагмент блоков "Объект управления №1" в "Объект управления №2" и изменить в нем привязку к сигналам базы банных (см. рисунок).



3. Аналогично изменить и другие проекты, сохранить сделанные изменения. После чего загрузить пакет tank_db_pak2.pak, запустить его на

выполнение и протестировать работу комплексной модели, подавая сигналы на клапаны обеих цистерн (см. рисунок).



4. Познакомиться с возможностями векторной обработки сигналов. Изменить проект `tank_db_control.prt` таким образом, чтобы в нем использовалась бы только одна версия алгоритма, который может работать с любым числом панелей и объектов управления, прописанных в базе данных. Выполнить тестирование измененного проекта.

5. Добавить в базу данных еще одну группу сигналов, например, `Tank3` и убедиться в том, что при запуске пакета проектов на выполнение появляется ошибка при загрузке проекта `tank_db_control.prt`.

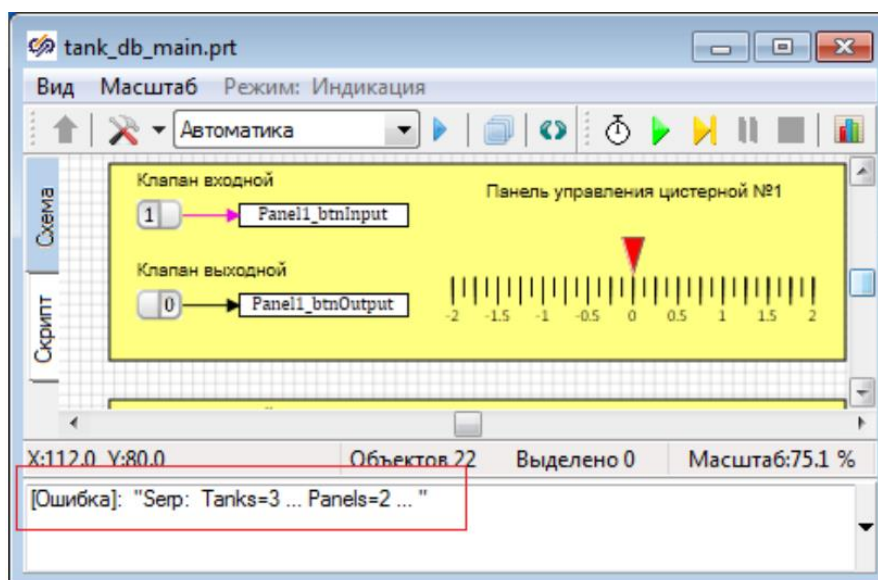
6. Для того, чтобы выполнялась проверка возможности работы комплексной модели при текущем состоянии базы данных, а также выводилось понятное пользователю сообщения о несогласованности базы данных, необходимо проект `tank_db_main.prt` дополнить следующим скриптом (см. рисунок).

```

C:\SimInTech\SerpTesting\Tank_db_2\tank_db_main.prt
Вид Масштаб Режим: Индикация
Автоматика
1 initialization
2 formattext
3 // количество объектов управления
4 nTanks = "{ query: category = "ObjectTank";
5             group = "*"; name = "TankVolume"; what = count}";
6 // количество панелей управления
7 nPanels = "{ query: category = "ObjectPanel";
8             group = "*"; name = "btnInput"; what = count}";
9
10 end;
11 if nTanks <> nPanels then
12 begin
13     seterrorstr("Serp: Tanks=" + nTanks + " ... Panels=" + nPanels+"
14             exit;
15 end
16

```

7. Проверить работу tank_db_main.prt при запуске пакета на выполнение (см. рисунок).



8. Привести базу данных в исходное состояние и подготовить пакет проектов для его преобразования в распределенную сетевую модель.

2. Задания по работе с распределенной моделью и ее сетевому тестированию

1. Выполнить тестирование новой распределенной модели в режиме сетевого расчёта в ручном режиме.

2. Используя среду виртуализации, настроить распределенную модель для запуска сетевого расчёта в автоматизированном режиме.

3. Используя сегмент реальной сети, выполнить настройку отдельных ее узлов, а также самой распределенной модели для возможности ее работы на отдельных сетевых станциях.

Три дополнительных задания:

1. Предусмотреть на панелях управления индикацию о достижении аварийно высоких и аварийно низких уровнях жидкости в цистернах.

2. Создать новый пакет проектов, который бы использовал расширенную базу данных, которая позволяла бы описывать все параметры модели, включая геометрию каждой из цистерн, индивидуальное время срабатывания каждого из клапанов и т.д.

3. Разработать пакет проектов, который объединил бы в себе возможности ручного и автоматического управления заливкой цистерн.

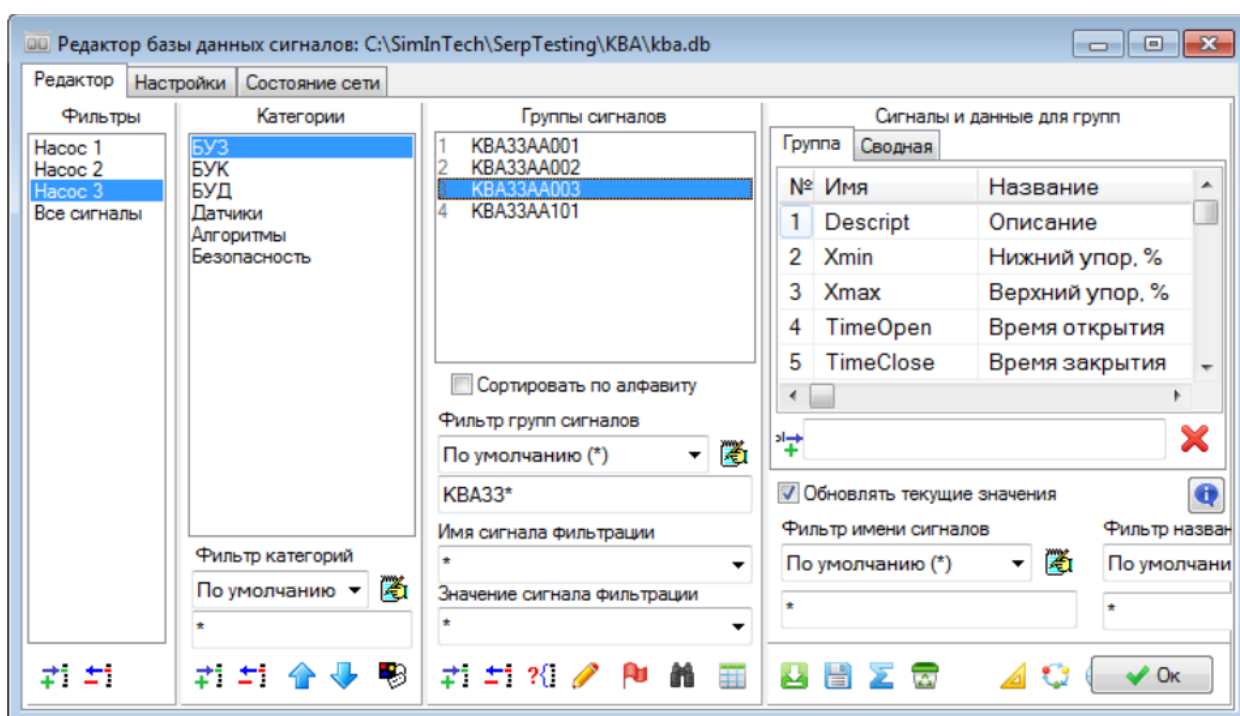
Методические рекомендации

Одной из основных особенностей использования SimInTech для комплексных моделей является идеология использования «Базы данных сигналов» - структурированную файловую базу данных, содержащую в себе константы и переменные (сигналы), обеспечивающих обмен расчетными значениями между расчетными схемами в единой модели. База данных SimInTech является объектной и позволяет:

- объединить нескольких расчетных схем в единую модель;
- обеспечить векторную обработку сигналов для типовых алгоритмов управления;
- выполнить объектно-ориентированное проектирование моделей систем;
- автоматизировать создание и обработку переменных в комплексных моделях.

Таким образом, в среде SimInTech, кроме сигналов проекта и глобальных переменных языка программирования, можно использовать внешнюю по отношению к файлу проекта базу данных (БД). Она подключается как динамическая библиотека. Также база данных может

содержать информационные поля (строковые константы), которые не используются для целей моделирования. Элементарной единицей базы сигналов является сигнал. Сигнал – переменная одного из доступных типов данных, несущая какую-либо информацию. В стандартную поставку SimInTech включена БД сигналов SDB, подключение которой к файлу проекта выполняется на вкладке «Настройки» в параметрах расчета проекта, а для работы с БД используется редактор, общий вид которого представлен на рисунке.



База данных сигналов SDB представляет собой структурированную базу данных, содержащую три уровня, а именно «Категории», «Группы сигналов» и «Сигналы групп».

- Категория – является прообразом, шаблоном или, в терминах программирования, классом для объектов базы данных.
- Группа сигналов – является логической единицей базы данных и всегда представляет собой объект какой-либо категории.
- Сигнал – является физически существующей в области памяти единицей: одной из переменных группы, именованных и типизированных.

Таким образом, категории создают архитектуру БД, группы сигналов – наполнение, а сигналы составляют список сигналов базы данных, который формируется и может быть использован в проекте.

В этом разделе основное внимание будет уделено использованию общесистемной базы данных для взаимодействия отдельных проектов в распределенных моделях, работающих как в рамках пакета проектов на одном компьютере, так и при их совместной работе на разных узлах сети. Поэтому тестовый пример будет максимально простым и посвящен реализации модели ручного управления клапанами для залива и осушения цистерн.

Формирование модели ручного управления клапанами

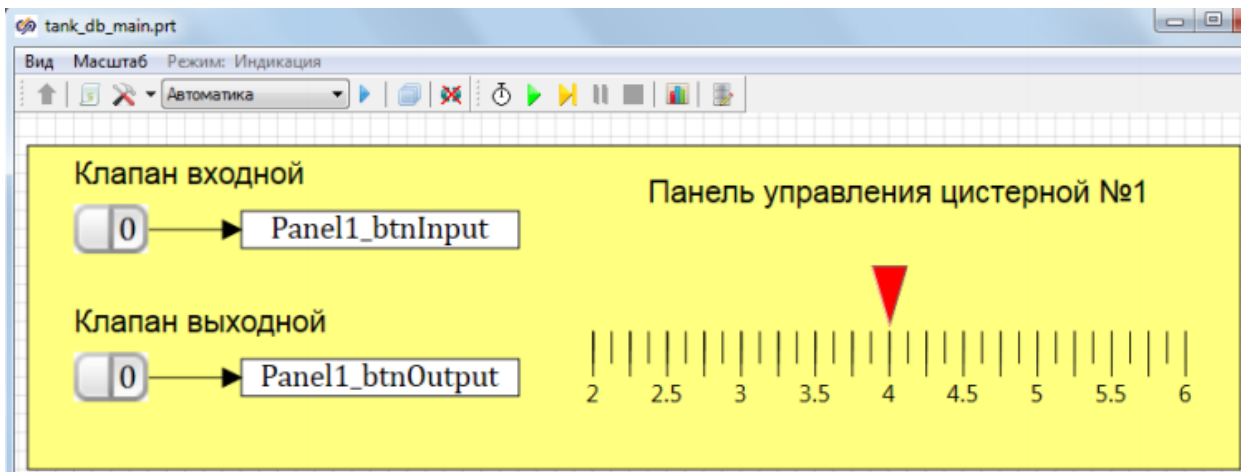
Ранее были рассмотрены достаточно сложные модели по управлению уровнем жидкости в цистерне. Здесь остановимся только на простейшей, но распределенной модели ручного управления входными и выходными клапанами цистерн, которая должна использовать базу данных SimInTech. Но прежде чем формировать общесистемную базу данных сигналов надо определиться со структурой модели.

На первом этапе построим комплексную модель, реализуемую на одном компьютере, которая представляет собой пакет проектов. Пакет включает в себя три проекта: модель панели управления, модель формирования управляющих воздействий на клапаны и модель объекта управления. С этой целью:

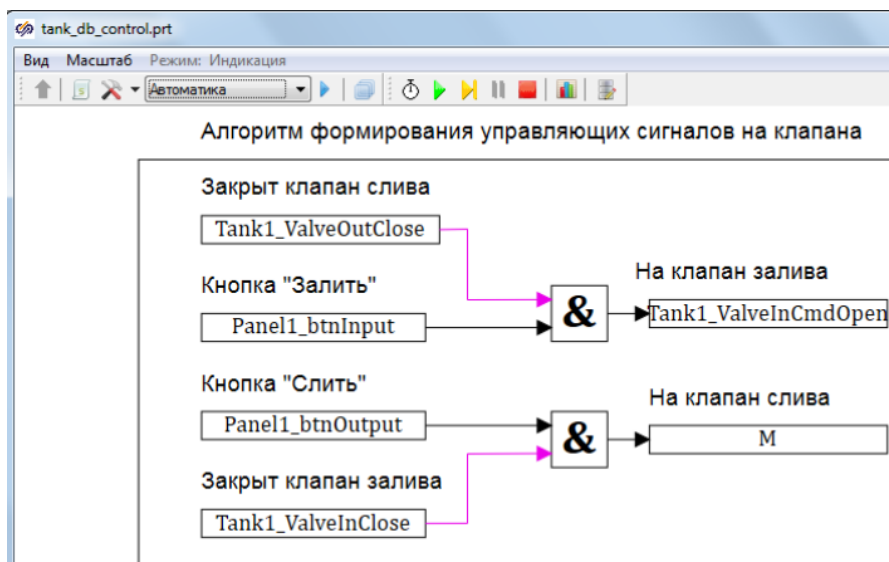
- создадим новую папку `tank_db_1`, а в ней пакет проектов `tank_db_pak.pak` из трех новых проектов – `tank_db_main.prt`, `tank_db_control.prt` и `tank_db_object.prt`.

- Особо отметим, что во всех проектах надо использовать блоки «Запись в список сигналов» и «Чтение из списка сигналов» для тех сигналов, которыми проекты обмениваются между собой. На первом этапе это будут пустые блоки, в которых отсутствует определение каких-либо свойств.

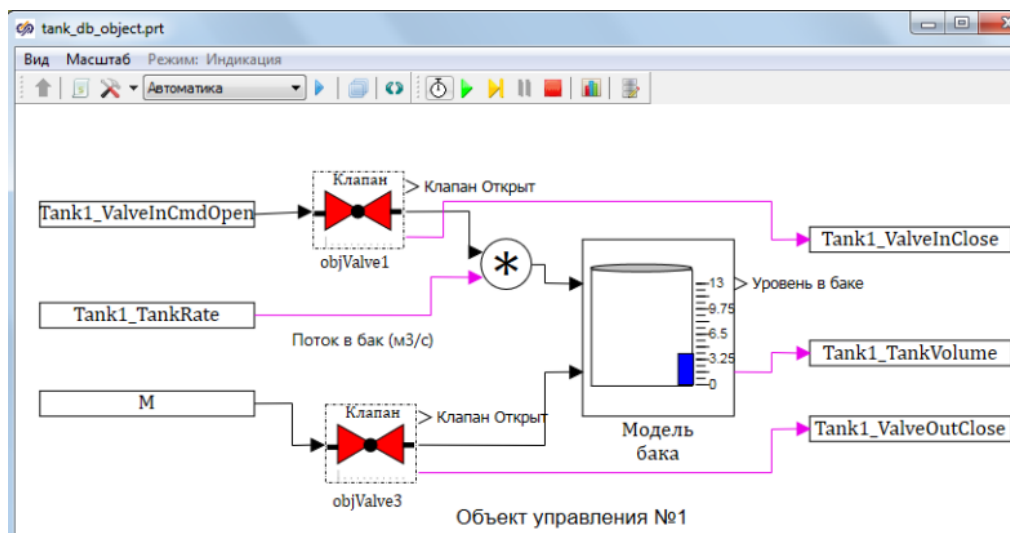
- Начнем с модели панели управления. Схема проекта содержит всего две кнопки, которые связаны с блоками записи сигналов в список (см. рисунок). Приведенную на рисунке шкалу на этом этапе можно не реализовывать. Здесь она использована как напоминание о том, что на панель управления должны приходить с сигналы с объекта управления.



- Сигналы от кнопок панели управления поступают в алгоритм формирования управляющих воздействий на клапаны, который реализует проект tank_db_control. (см. рисунок). В данном примере это простейшая блокировка одновременной работы двух клапанов.



- Модель объекта управления может быть получена путем копирования в новый проект набора блоков из любого ранее используемого проекта (см. рисунок).

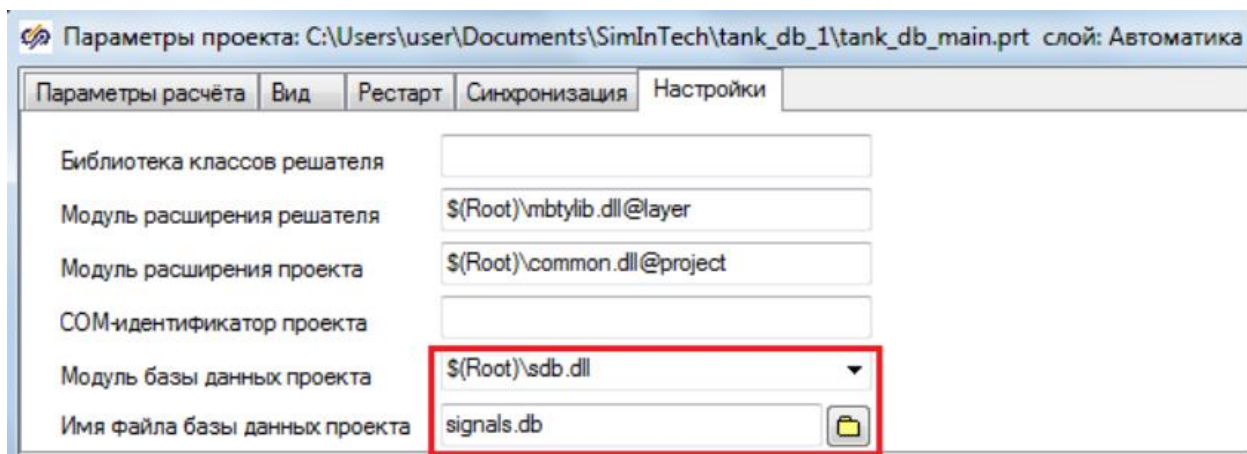


Данная комплексная модель может успешно работать только в том случае, если обеспечено взаимодействие между собой всех проектов. Одним из подходов к реализации этой задачи в SimInTech является возможность подключать к любой схеме базу данных сигналов. Она служит для структурного хранения глобальных констант и переменных, а также для организации взаимодействия между расчетными схемами.

Для совместной работы нескольких проектов, находящихся в разных файлах, надо, чтобы они использовали одну и ту же базу сигналов, структура которой учитывала бы особенность каждого из проектов.

Подключение базы данных к пакету проектов Будем считать, что базовым у нас является проект `tank_db_main.prt`, находясь в котором пользователь имеет полные права на создание и модификацию структуры общесистемной базы данных. Но прежде всего надо до обеспечить подключение базы данных к этому проекту и выполнить следующую последовательность действий.

- Проверить, что установлен режим разработчика, а затем, открыв окно параметров расчета проекта, выбрать вкладку «Настройки» (см. рисунок).

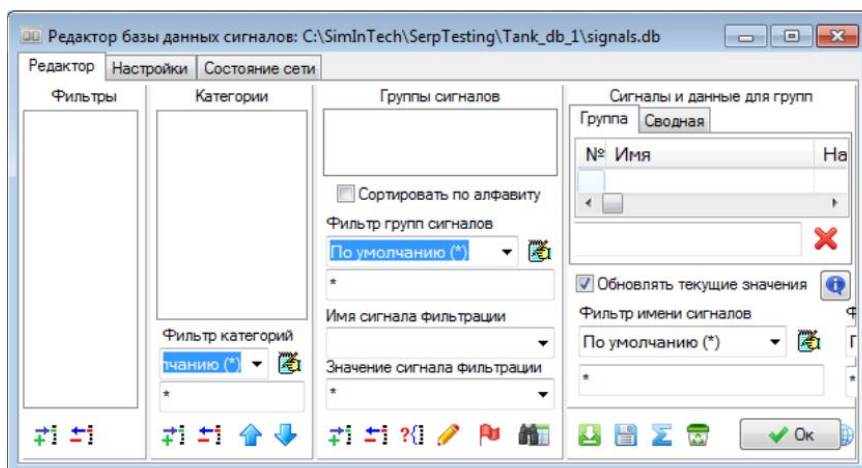


- В поле «Модуль базы данных проекта» ввести: $$(Root)\sdb.dll$, где $sdb.dll$ – имя динамической библиотеки программного модуля базы данных сигналов, поставляемой в составе стандартного дистрибутива; а значение $$(Root)$ – служебное имя, указывающее на папку $C:\SimInTech\bin$. В поле «Модуль базы данных проекта» можно ввести $C:\SimInTech\bin\sdb.dll$, что равнозначно $$(Root)\sdb.dll$.

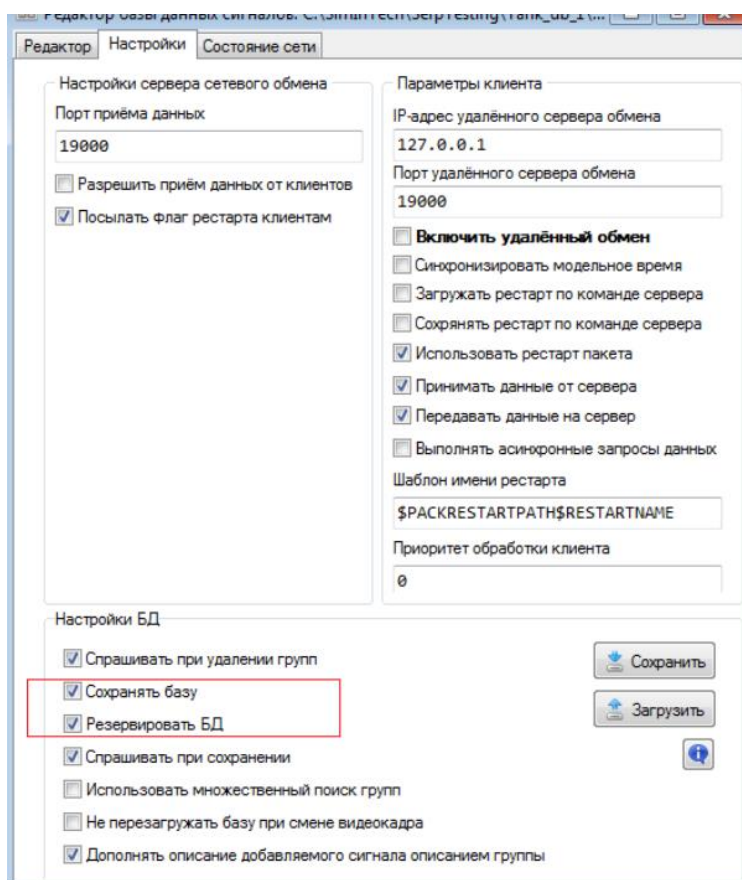
- В поле ввода «Имя базы данных проекта» ввести произвольное имя файла для сохранения базы данных. В рамках рассматриваемого примера присвоим имя базе сигналов $signals.db$, однако оно может быть любым. Если не указан полный путь к файлу, то он располагается в том же каталоге, что и файл проекта.

- Закрывать окно нажатием «Ок» и выполнить сохранение текущего проекта, нажав кнопку «Сохранить проект» в главном меню SimInTech.

После сохранения проекта с подключенной к нему базой, файл $signals.db$ появится на диске в каталоге проекта, что можно проверить просмотром этого каталога. Кроме того, для проекта с подключенной базой становится в главном меню доступна опция «Инструменты → База данных...». Выбрав ее, откроется окно редактора базы данных сигналов (см. рисунок).



Это окно пока пустое, его заполнять будем позднее. Но прежде чем закрывать редактор, следует убедиться, что настройки предполагают автоматическое сохранение базы данных при работе с этим проектом. Для этого надо перейти на вкладку «Настройки» и установить галочки в пунктах «Сохранять базу» и «Резервировать БД» (см. рисунок). Пункт «Сохранять базу» реализует алгоритм сохранения базы данных при сохранении проекта, к которому она подсоединена.



Пункт «Резервировать БД» реализует алгоритм сохранения базы данных в файл, обеспечивая при этом хранение ранее сохраненных версий базы данных с расширениями .1 .2 .3 и так далее.

- Установив нужные опции, надо вернуться на закладку «Редактор» и закрыть окно «Редактор базы данных» нажатием кнопки «Ок».

- После этого надо сохранить проект. Данные установки позволяют автоматически сохранять базу данных проекта при закрытии файла самого проекта. Отметим, что пользователь в любой момент времени может принудительно сохранить базу данных, не закрывая проект. Для этого достаточно нажать кнопку «Сохранить» на вкладке «Настройки». Кнопка «Сохранить в файл», которая находится на вкладке «Редактор», используется для сохранения базы сигналов в какой-либо файл с другим именем и/или форматом. Файл проекта при этом остается привязанным к той базе, которая прописана в параметрах проекта.

- Замечание 1. При новом подключении проекта к уже существующей базе, выполняя сохранение этого проекта надо ответить «Нет» на запрос об изменении базы. Иначе существующая база будет перезаписана пустым файлом.

- Замечание 2. База сигналов считывается из файла только при загрузке проекта, а не при прописывании файла базы сигналов в проекте.

После привязки проекта `tank_db_main` к базе `signals.db`, аналогичные действия надо выполнить и во всех других проектах пакета. Однако, так как базу использует ряд проектов, то нужна еще дополнительная настройка возможности ее сохранения в разных проектах.

Это требуется для того, чтобы изменения в базу сигналов вносились только из какого-то одного проекта, чтобы избежать конфликтов записи, некорректной перезаписи файла или потери информации, сохраненной в него из другого проекта. Поэтому, после подключения базы к проектам `tank_db_object.prt` и `tank_db_control.prt`, рекомендуется в этих проектах опции

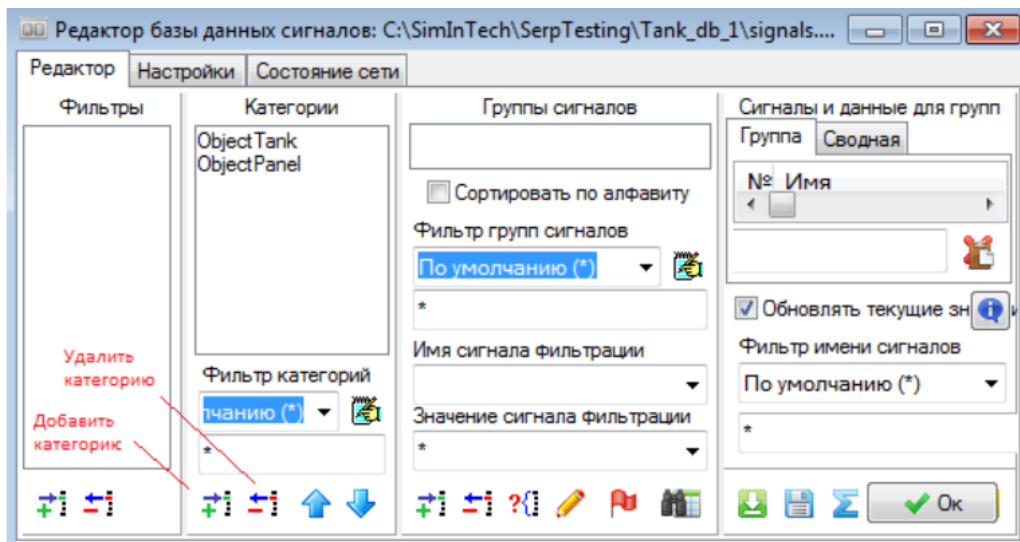
«Сохранять базу» и «Резервировать БД» отключить. То есть, все изменения в общесистемной базе данных будут выполняться только в основном проекте.

Заполнение базы данных и связь блоков модели с сигналами базы данных

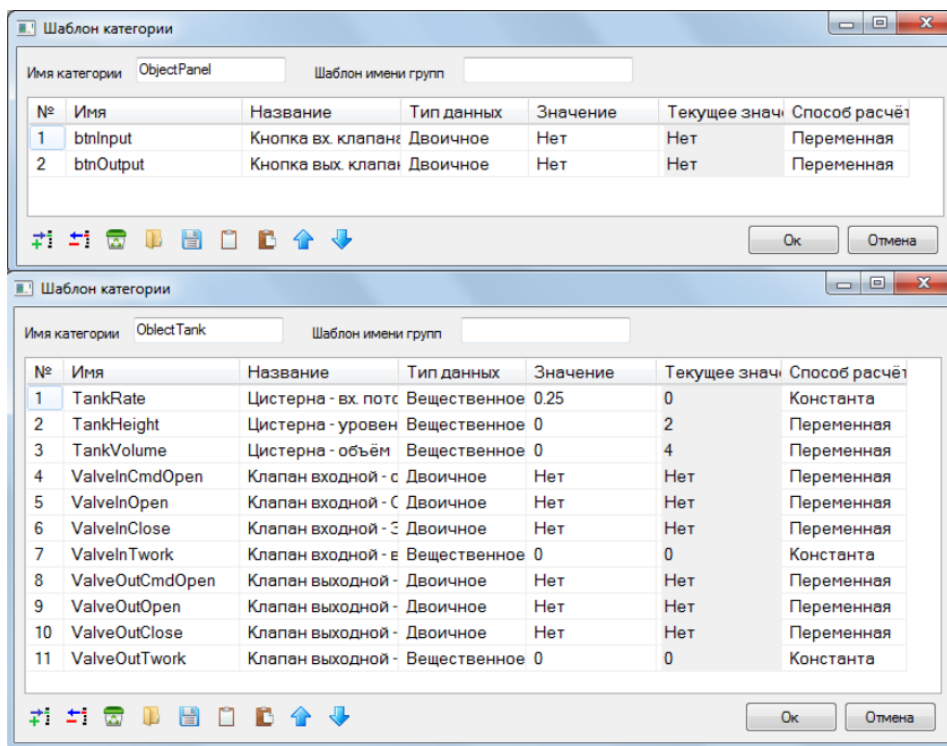
Что касается структуры самой базы данных, то обычно ее объектно-ориентированный характер описывает все элементы и узлы модели исследуемой системы с полным набором их параметров, характеристик и сигналов. Основное назначение рассматриваемого примера состоит в том, чтобы показать возможность взаимодействия разных проектов между собой через единую базу данных. Поэтому далее предлагается использовать упрощенную структуру БД, описывающую только те объекты, которые соответствуют основным структурным блокам проектов, взаимодействующих между собой в рамках распределенной модели.

Если обратить внимание на исследуемую модель и проанализировать все выделенные в ней блоки чтения и записи в память, то можно установить, что для описания всех взаимодействий между проектами достаточно всего двух категорий объектов – «Панели» (ObjectPanel) и «Объекты Управления» (ObjectTank). Третий проект не имеет собственных сигналов, а использует сигналы двух первых объектов.

Чтобы сформировать БД для исследуемой модели, надо открыть проект tank_db_main и активировать редактор базы данных. Подробную инструкцию по работе с ним можно найти в справочной системе SimInTech, в частности, в разделе 7 – «Общие сведения о базе данных сигналов SDB». Для редактирования списков категорий, групп сигналов, фильтров, а также сигналов и данных групп, под каждой панелью предусмотрены кнопки «Добавить» и «Удалить». Используя кнопку «Добавить», создадим две новые категории (см. рисунок).

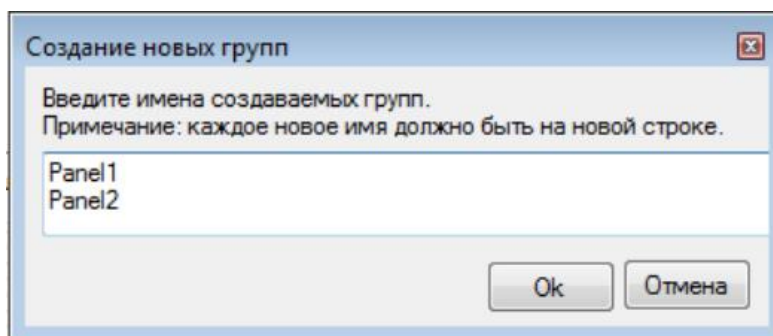


В процессе добавления и нажатия кнопки «Добавить категорию» в списке категорий появляется запись «Новая категория». Двойное нажатие на элементе «Новая категория» откроет окно «Шаблон категории», где надо задать имя категории и заполнить таблицу шаблонных сигналов. Расположенные в нижней части окна кнопки редактирования таблицы сигналов позволяют: добавить, удалить сигнал, очистить список сигналов, загрузить список сигналов из файла или сохранить в файл, копировать сигнал в буфер памяти, вставить сигнал из буфера памяти, переместить сигнал в таблице выше/ниже. При добавлении в проектируемую базу данных каждой из новых категорий окна «Шаблон категории» будут иметь вид, аналогичный тому, что представлен на рисунке.



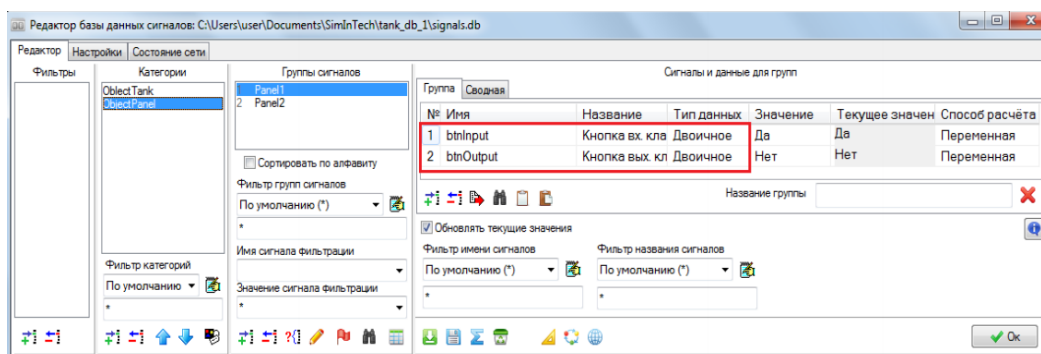
Для каждой категории пользователь может добавить в базу данных неограниченное количество групп сигналов. Например, создав категорию панелей, можно добавить в базу данных несколько панелей данного типа, хотя пока в расчетной модели используется только одна панель. Для этого необходимо:

- выделить название категории в панели категорий, например, ObjectPanel.
- Нажать кнопку «Добавить таблицу» в панели групп сигналов;
- откроется окно «Создание новых групп», в котором надо ввести имена групп сигналов данной категории, которые необходимо внести в базу данных (см. рисунок);

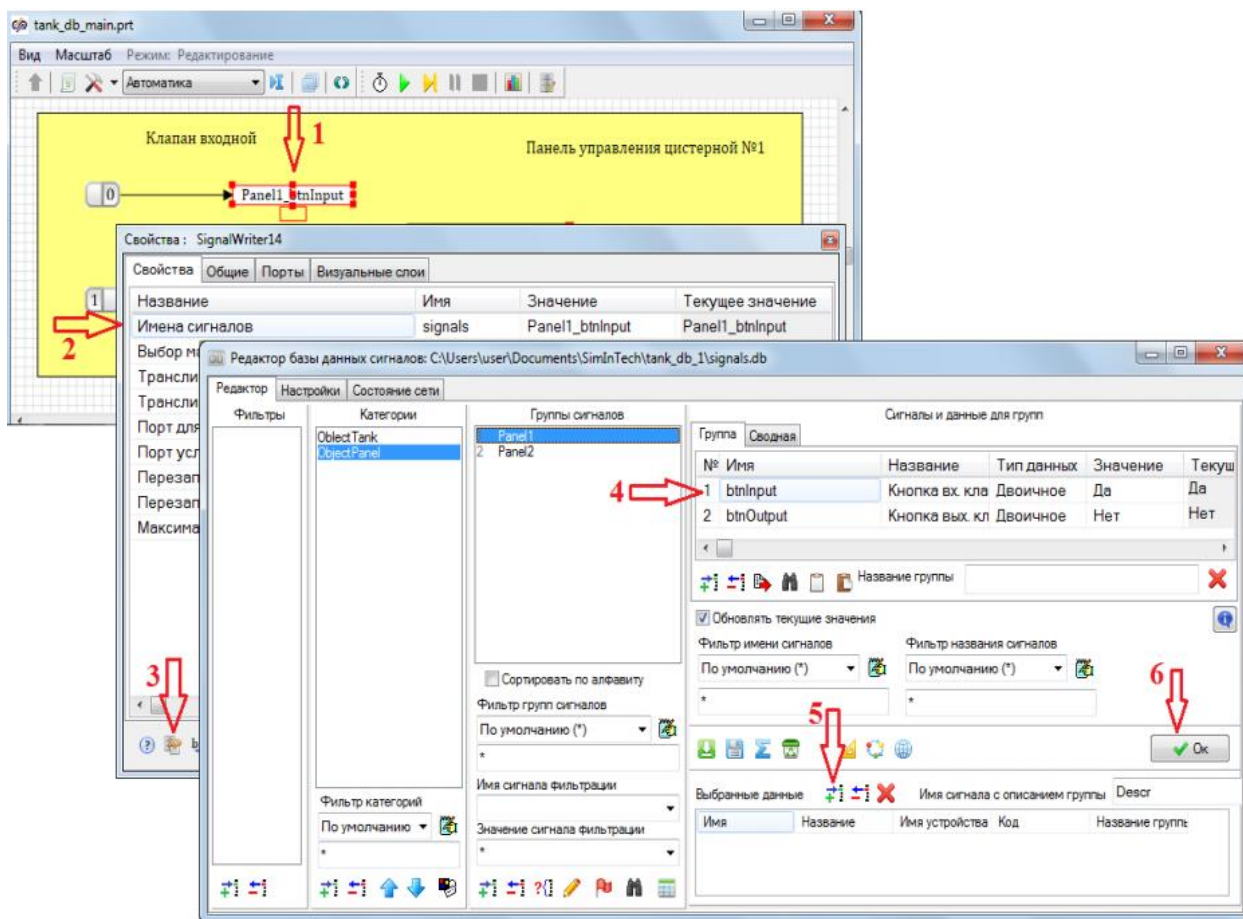


- закрыть диалоговое окно «Создание новых групп» нажатием кнопки «Ok».

Аналогично надо создать две группы сигналов, например Tank1 и Tank2, для категории ObjectTank. После заполнения БД категориями и группами станет доступным переключение между ними, путем выбора нужной категории и группы сигналов. При этом все сигналы выбранной группы будут отображаться в правой части редактора (см. рисунок).



После окончания заполнения базы данных сигналов, надо сохранить ее содержимое и приступить к связи блоков пакета проектов с сигналами базы данных. В качестве примера рассмотрим проект tank_db_main, где имеется два выходных сигнала от кнопок, которые подключены к блокам «Запись в список сигналов». Чтобы они могли взаимодействовать с блоками других проектов, надо некоторые их свойства подключить к БД. Так, для кнопки управления входным клапаном надо выполнить набор следующих операций (см. рисунок).



1. На схеме проекта надо выделить нужный блок «Запись в список сигналов» и открыть окно его свойств.

2. В окне свойств элемента выбрать свойство signals («Имена сигналов»).

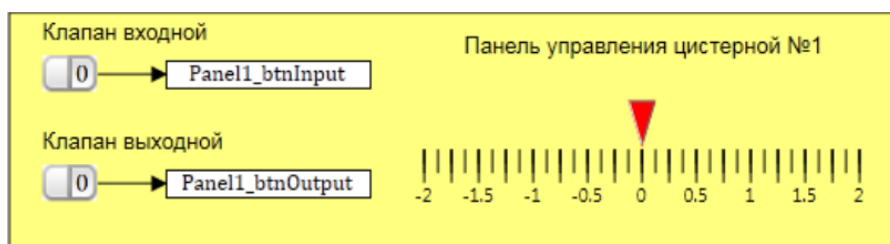
3. Нажать в нижней части окна кнопку «Выбрать сигнал из базы», при этом откроется окно редактора базы данных.

4. В этом окне надо выбрать соответствующий сигнал соответствующей группы из соответствующей категории. В нашем примере это сигнал btnInput из группы Panel1 категории ObjectPanel.

5. После этого необходимо добавить его в панель «Выбранные данные», нажав в этой панели кнопка «Добавить».

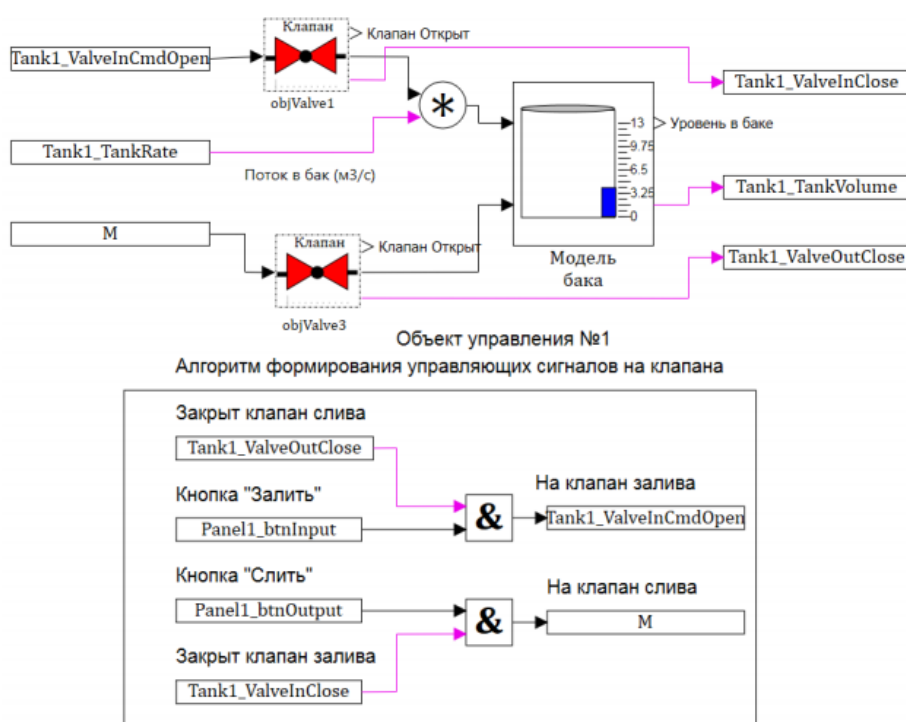
6. Завершить операцию нажатием кнопки «Ok» в этом, и всех открытых окнах. После этого вид блока «Запись в список сигналов» в схеме проекта будет иметь вид, аналогичный тому, что приведен на рисунке. Обратите

внимание, что сигнал базы данных, который подключен к блоку схемы, SimInTech назвал как <Имя группы>_<Имя сигнала>.



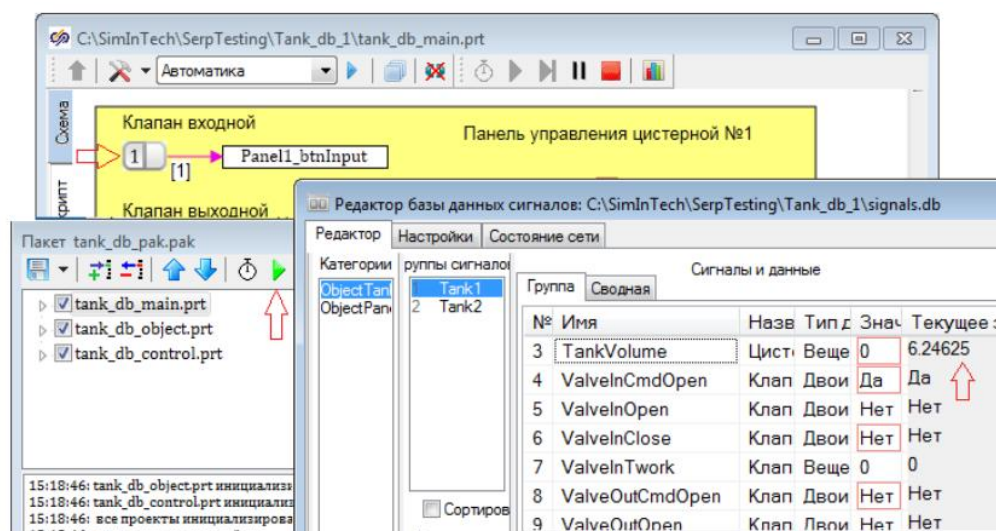
Принимая во внимание такое именование в SimInTech сигналов базы данных, можно для второго блока «Запись в список сигналов» открыть окно его свойств, и в свойстве signals сразу или с помощью редактора ввести текст Panel1_btnOutput.

Закончив с проектом tank_db_main, можно перейти к tank_db_object и tank_db_control, и выполнить подключение их блоков записи и чтения в список сигналов к сигналам базы данных (см. рисунок).



Обратите внимание, что сигнал Tank1_ValveOutCmdOpen на схеме не отобразился и блок не изменил свое обозначение. Это связано с тем, что SimInTech отображает в блоках только те сигналы, символические имена которых содержат не более 20 символов, а имя указанного выше сигнала состоит из 21 символа.

На текущий момент все необходимые преобразования комплексной модели по связи блоков с сигналами базы данных выполнены, и модель может быть запущена в работу. Для этого достаточно загрузить проект tank_db_pak.pak и нажать кнопку запуска проекта. Если все сделано было верно, то пакет должен работать, обмениваясь между проектами сигналами общесистемной базы данных. Причем об изменении всех сигналов в процессе работы пакета можно следить, если в любом из проектов дополнительно открыть окно редактора базы данных сигналов (см. рисунок).



Сигналы БД могут использоваться в задании свойств не только типовых блоков, но и в любых графических компонентах, в том числе и в примитиве типа «Bar» (см. рисунок). На рисунке видно, что нижний и верхний пределы шкалы, а также отображаемое значение через значение сигнала БД Tank1_TankVolume.

Свойства: Bar		
Общие		
Визуальные слои		
Название	Имя	Значение
Нижний предел	MinValue	Tank1_TankVolume - 2
Верхний предел	MaxValue	Tank1_TankVolume + 2
Шаг	Step	0.5
Число делений на шаг	DivCount	2
Высота штриха (% от размеров)	NumHeight	70
Высота деления (% от размеров)	DivHeight	30
Показывать деления	ShowDivs	Да
Показывать числа	ShowNums	Да
Шрифт	Font	Arial Narrow
Число больших делений на шаг	LargeDivCount	2
Толщина малого штриха	SmallWidth	1
Толщина большого штриха	LargeWidth	1
Штрихи наоборот	InvTicks	Нет
Высота среднего штриха (% от разм	MidHeight	50
Количество средних штрихов на шаг	MidDivCount	4
Коэффициент прозрачности	Opacity	1
Тип столбца	BarType	Нет
Цвет столбца	BarColor	16711680
Ширина столбца (% от высоты)	BarWidth	40
Тип ручки	ControlType	Треугольник
Трёхмерность ручки	Handle3d	Да
Цвет ручки	ControlColor	255
Толщина ручки	ControlWidth	15
Показывать шкалу	AxisVisible	Да
Режим управления	DirectMode	Нет
Дискретность управления	Discrete	0
Отображаемое значение	Value	Tank1_TankVolume

Данный примитив на схеме основного проекта – это подвижная шкала, отображающая текущий объем цистерны, вычисленный в проекте tank_db_object и переданный сюда через общесистемную базу данных. Одновременно с этим сигнал с кнопки этого проекта через БД передается в проект tank_db_control, а оттуда в проект tank_db_object.

Круг обмена данными между проектами замкнулся.

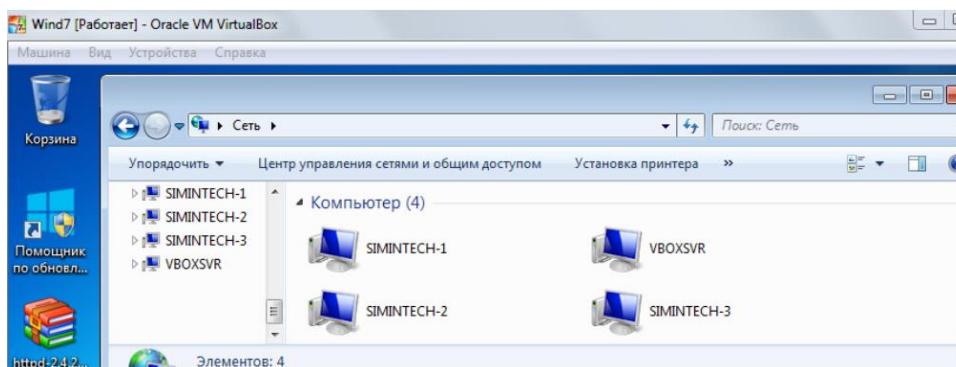
Реализация распределенной модели с общей базой данных

В предыдущих разделах описано создание отлаженной комплексной модели в виде пакета проектов, а сейчас преобразуем ее в распределенную сетевую модель, в которой каждый проект мог бы запускаться на своей рабочей станции. Обмен данными между проектами осуществляется через единую базу данных.

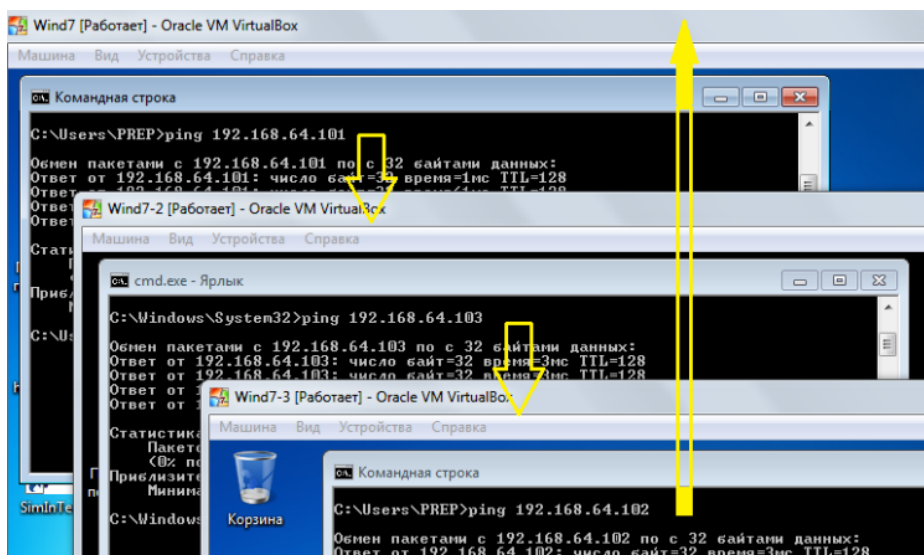
Для решения этой задачи сначала определим сетевую конфигурацию той среды, где будет реализована поставленная задача. Ее основу составляют следующие компоненты.

- Компьютер на базе Windows 10 с установленным на нем Oracle VM Virtual Box.
- Две виртуальные машины на базе Windows 7 с системой SimInTech. Они имеют сетевые имена SimInTech-1 (192.168.56.101) и SimInTech-2 (192.168.56.102). Эти имена могут не совпадать с именами машин в менеджере Oracle Virtual Box.
- Еще одна виртуальная машина на базе Windows XP с системой SimInTech. Она имеет сетевое имя SimInTech-3 (192.168.56.103).
- Внутри базового компьютера организована рабочая группа Workgroup, в которую объединены все виртуальные машины.
- Кроме того, на диске D: базового компьютера выделена папка, общедоступная со всех виртуальных машин через vboxsvr.

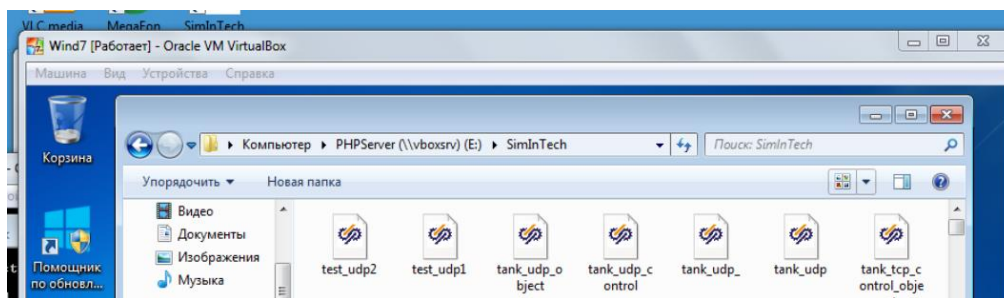
Проверить настройку сетевой среды можно, если в любой из виртуальных машин открыть проводник Windows и выбрать компонент Сеть (см. рисунок).



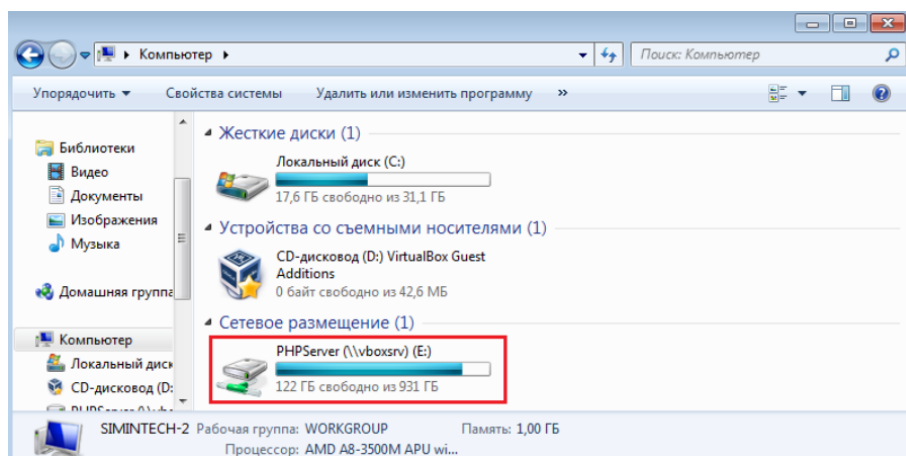
При построении распределенной модели планируется, что каждый проект будет запущен на своем компьютере. При этом в настройках каждого проекта будут использованы IP-адреса этих компьютеров. Поэтому желательно уже на начальной стадии убедиться во взаимной доступности всех виртуальных машин по их IP-адресам (см. рисунок).



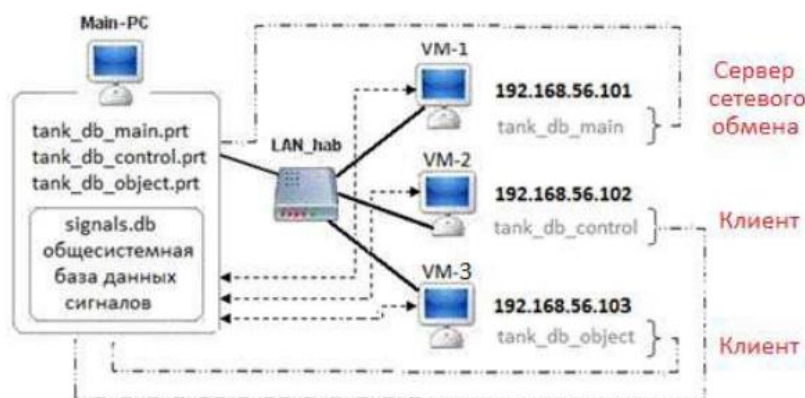
Для удобства дальнейшей работы имеет смысл настроить сетевой ресурс с разрешённым доступом к нему со всех трёх виртуальных машин, чтобы можно было настраивать всё на одном компьютере. Для этого используем папку основного компьютера, настроенную в качестве общей во всех виртуальных машинах. Дадим ей имя server и скопируем в нее все файлы работающих проектов, кроме файла пакета. Убедиться в сетевой доступности этого ресурса можно, используя в любой из виртуальных машин проводник Windows (см. рисунок).



Аналогично можно выяснить, что сетевое хранилище данных виртуальной машины, например SimInTech-2, воспринимается как сетевой диск E: (см. рисунок).

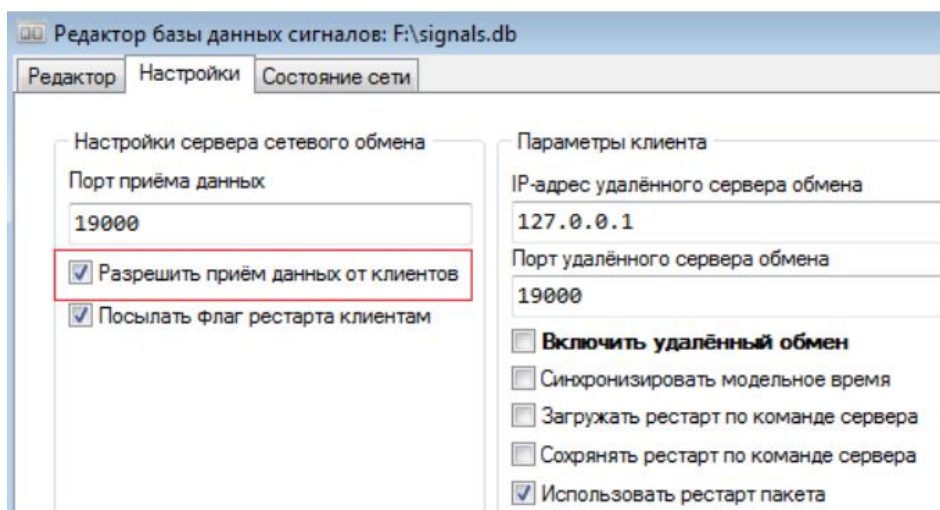


Простейшей сетевой организацией распределенной модели является вариант, когда все файлы хранятся в одном месте, а в процесс работы на каждом из узлов запускается свой проект, загружаемый с общесетевого хранилища данных. При этом один из проектов должен быть настроен как сервер сетевого обмена, а другие как его клиенты (см. рисунок).



Закончив настройку и тестирование сетевой среды, следует заняться настройкой распределенной модели для возможности ее работы в сети.

- Для настройки сервера сетевого обмена надо открыть проект `tank_db_main.prt`, зайти в настройки базы данных и включить опцию «Разрешить приём данных от клиентов». Порт приёма можно не менять (19000), а опция «Включить удалённый обмен» остается выключенной – это настройка для клиентов (см. рисунок). После этого надо сохранить проект `tank_db_main.prt`, так как настройки сервера сетевого обмена и параметры клиента хранятся в проекте, а не в файле базы данных.



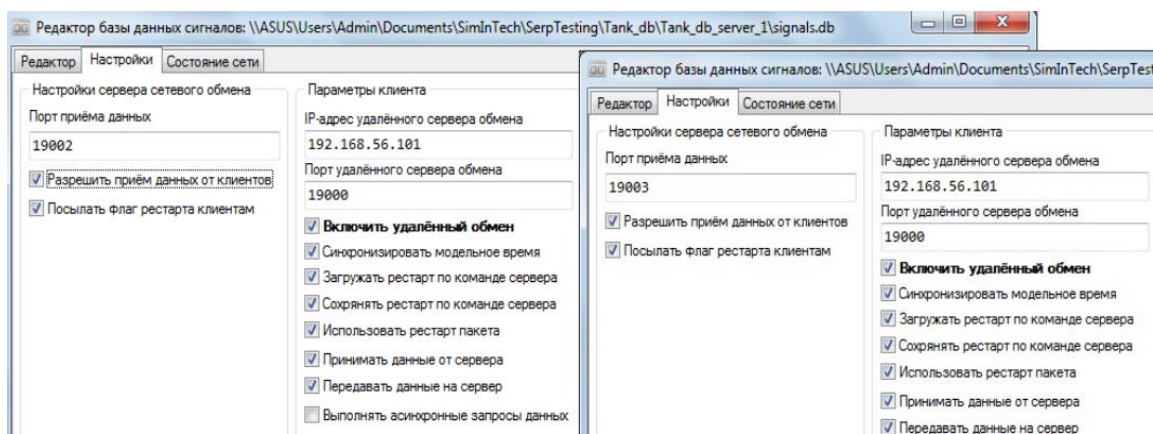
- Для настройки клиентов надо в проектах tank_db_control.prt и tank_db_object.prt в настройках БД включить опцию «Включить удалённый обмен», а в поле IP-адреса ввести значение 192.168.56.101 в соответствии с адресом сервера.

- Значение поля «Порт удалённого сервера обмена» должно быть таким же, какое ранее было установлено для сервера, то есть 19000. Порт приёма данных может быть любым. Укажем для первого клиента порт приёма данных –19002, а для второго – 19003.

- В данном примере это не критично, так как у каждого из клиентов открыто всего одному порту, и они могут быть любыми, в том числе и 19000, как у сервера.

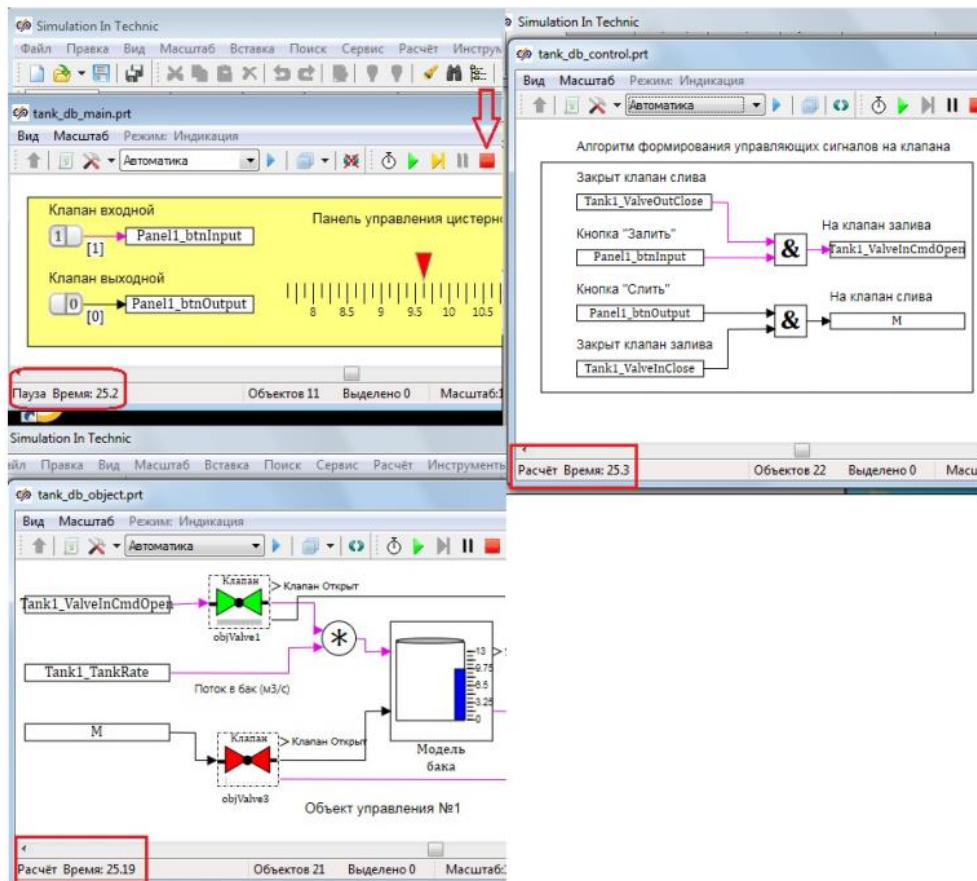
- У каждого из клиентов также надо включить опцию «Разрешить приём данных от клиентов» ввиду того, что сервер будет обратно присылать на клиенты данные, вычисляемые как на сервере, так и на других клиентах.

- Важно у каждого из клиентов установить опцию «Синхронизировать модельное время» для того, чтобы синхронизировать модельное время расчета сервера и всех клиентских проектов. Общие настройки клиентов приведены на рисунке.



После выполнения всех этих настроек можно переходить к проверке работоспособности распределенной сетевой модели. Для этого надо запустить виртуальные машины и открыть в каждой из них свой проект, используя для загрузки проекта папку `\\VBOXSVR\server\` или соответствующий сетевой диск. После того, как на каждом из узлов сети будет открыт свой проект, надо последовательно, начиная с основного проекта, выполнить их инициализацию.

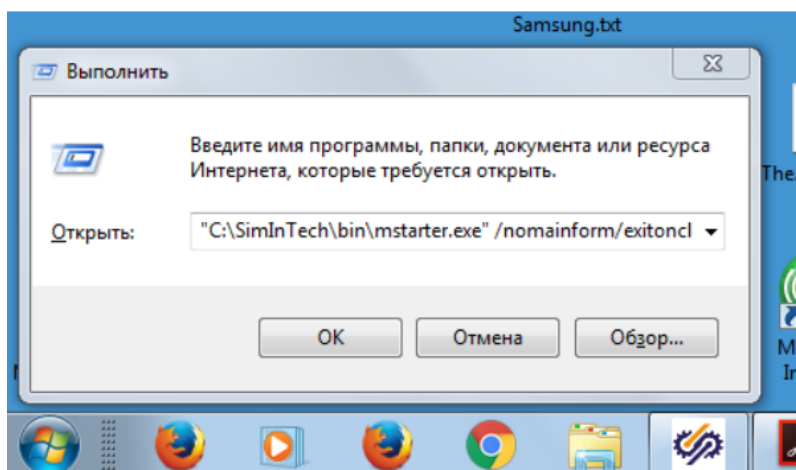
Отметим, что при этом в каждом из проектов расчетное время установится в 0, а статус перейдет в состояние "Старт" (рис. 6.24). Если в окне основного проекта нажать кнопку "Пуск", то начнется распределённый расчет с синхронизацией модельного времени и обменом данными из базы данных сигналов. Таким образом, распределенная модель работает на всех узлах сети. Для остановки ее работы надо в окне основного проекта нажать кнопку "Стоп" (см. рисунок).



Так как все проекты уже отлажены, и требуется исследование работы распределенной модели, то в этом случае загрузка основного меню среды разработки SimInTech становится ненужной. Ускорить работу пользователя по загрузке проектов распределенной модели можно, если после запуска виртуальной машины в ее командной строке ввести команду: "C:\SimInTech\bin\mstarter.exe"/nomainform/exitonclose /start где на разных узлах будет иметь разные значения:

- \\VBOXSVR\server\tank_db_main.prt
- \\VBOXSVR\server\tank_db_control.prt
- \\VBOXSVR\server\tank_db_object.prt

Выполнение этой команды (см. рисунок) обеспечит загрузку среды SimInTech, открытие нужного проекта и его автоматическую инициализацию, а также скрывание основного меню SimInTech. Кроме того, при закрытии окна проекта будет выполнена автоматическая выгрузка среды SimInTech из памяти компьютера.



Для удобства дальнейшей работы предпочтительнее создать ярлык, чтобы избавиться от необходимости многократного ввода этой команды при каждом запуске проекта. Можно еще более автоматизировать работу, если использовать эту команду в автозагрузке сетевого узла. Тогда при запуске виртуальной машины сразу откроется окно нужного проекта.

Подводя итог этому разделу, можно сделать вывод, что одним из способов реализации распределенных моделей является режим, так называемого, сетевого расчёта в ручном режиме. Для организации работы модели в этом режиме необходимо было выполнить следующие шаги.

- Настроить обмен данными через сеть и синхронизацию модельного времени, используя для этого возможности базы данных сигналов SDB.
- Инициализировать главный проект с включенным сервером сетевого обмена.
- Инициализировать всех клиентов – проекты с включенным удалённым обменом.
- Нажать кнопку "Пуск" в окне главного проекта. При этом все проекты на разных вычислительных узлах запустятся на расчёт.

Следует отметить, что это не единственный режим реализации распределенных моделей, который используется в SimInTech. В частности, есть еще режим запуска сетевого расчёта в автоматизированном режиме.