**Tekla Open API – Создание Стадий.**

Давайте попробуем создать и удалить стадии в моделе Tekla:

/\*  
 \* Created by Vadim Semenov  
 \* vad.s.semenov@gmail.com  
 \* Date: 19.04.2021  
 \*   
 \*/  
**using** System;  
**using** System.Collections.Generic;  
**using** Tekla.Structures.Model;  
  
**namespace** CreatePhase  
{  
    class Program  
    {  
        **public** static void **Main**(string[] args)  
        {  
            //Создаем одну Стадию  
            **CreatePhase**(101100);  
             
            **int**[] phaseNumber = { 101101, 101102, 101103, 101104 };  
            string[] phaseName = { "1", "2", "3", "4" };  
            string[] phaseComment = { "1Com", "2Com", "3Com", "4Com" };  
            **int**[] isCurrentPhase = { 0, 0, 0, 1 };  
             
            //Создаем несколько Стадий  
            **for** (**int** i = 0; i < phaseNumber.Length; i++) {  
                **CreatePhase**(phaseNumber[i], phaseName[i], phaseComment[i], isCurrentPhase[i]);  
            }  
           
            **int**[] phaseNumber1 = { 101105, 101106, 101107, 101108 };  
             
            //Создаем еще несколько стадий  
            **CreatePhases**(phaseNumber1);  
  
            //Удаляем Стадию    
            **if** (**DeletePhase**(101106)) {  
                Console.**WriteLine**("Стадия 101106 удалена!");  
                Console.**Read**();  
            }  
             
        }  
          
          
          
        /// <**summary**>  
        /// Создание нескольких Стадий  
        /// <**/summary**>  
        /// <**param** **name="phaseNumber"**>Номер стадии<**/param**>  
        /// <**returns**>Коллекция созданных стадий<**/returns**>  
        **private** static List <Phase> **CreatePhases**(**int**[] phaseNumber)  
        {  
            List <Phase> phases = **new** List<Phase>();  
            Phase phase = **new** Phase();  
              
            //Создаем Стадии в цикле  
            **foreach** (var pN **in** phaseNumber) {  
                phase = **CreatePhase**(pN);  
                  
                **if** (phase != **null**) {  
                    phases.**Add**(phase);  
                }  
            }  
              
            //Возвращаем коллекцию созданных Стадий  
            return phases;  
        }  
          
      
        /// <**summary**>  
        /// Создание Стадии по номеру.  
        /// <**/summary**>  
        /// <**param** **name="phaseNumber"**>Номер Стадии<**/param**>  
        **private** static Phase **CreatePhase**(**int** phaseNumber)  
        {  
            Model model = **new** Model();  
              
            //Проверяем что Tekla открыта  
            **if** (model.**GetConnectionStatus**()) {  
              
                //Создаем стадию  
                Phase phase = **new** Phase(phaseNumber);  
                //Вставляем стадию  
                **if** (phase.**Insert**()) {  
                    //Если вставилась, возвращаем ее  
                    return phase;  
                }  
                model.**CommitChanges**();  
            }  
            return **null**;  
        }  
          
          
        /// <**summary**>  
        /// Создание Стадии по всем параметрам.  
        /// <**/summary**>  
        /// <**param** **name="phaseNumber"**>Номер Стадии<**/param**>  
        /// <**param** **name="phaseName"**>Имя Стадии<**/param**>  
        /// <**param** **name="phaseComment"**>Коментарий Стадии<**/param**>  
        /// <**param** **name="isCurrentPhase"**>Является ли текущей стадией<**/param**>  
        **private** static Phase **CreatePhase**(**int** phaseNumber, string phaseName, string phaseComment, **int** isCurrentPhase)  
        {  
          
            Model model = **new** Model();  
              
            //Проверяем что Tekla открыта  
            **if** (model.**GetConnectionStatus**()) {  
              
                //Создаем стадию  
                Phase phase = **new** Phase(phaseNumber, phaseName, phaseComment, isCurrentPhase);  
                //Вставляем стадию  
                **if** (phase.**Insert**()) {  
                    //Если вставилась, возвращаем ее  
                    return phase;  
                }  
              
                model.**CommitChanges**();  
            }  
              
            return **null**;  
        }  
          
  
        /// <**summary**>  
        /// Удаление Стадии  
        /// <**/summary**>  
        /// <**param** **name="phaseNumber"**>Номер стадии<**/param**>  
        /// <**returns**>True -если Стадия удалена<**/returns**>  
        **private** static **bool** **DeletePhase**(**int** phaseNumber)  
        {  
          
            Model model = **new** Model();  
            Phase ph = **new** Phase(phaseNumber);  
              
            //Проверяем что Tekla открыта  
            **if** (model.**GetConnectionStatus**()) {  
              
                //Проверяем что Стадия существующая и не является текущей  
                **if** (**PhaseIsExist**(phaseNumber) && ph.IsCurrentPhase != 1) {  
                      
                    //Удаляем Стадию  
                    ph.**Delete**();  
                    ph.**Modify**();  
                    return **true**;  
                }  
                               
                model.**CommitChanges**();  
            }  
            return **false**;  
        }  
          
  
        /// <**summary**>  
        /// Проверка существования стадии.  
        /// <**/summary**>  
        /// <**param** **name="phaseNumber"**>Номер Стадии<**/param**>  
        **private** static **bool** **PhaseIsExist**(**int** phaseNumber)  
        {  
          
            Model model = **new** Model();  
              
            //Проверяем что Tekla открыта  
            **if** (model.**GetConnectionStatus**()) {  
                  
                //Получаем коллекцию Стадий модели   
                PhaseCollection phaseCollection = model.**GetPhases**();  
                  
                //Перебираем все стадии по номеру  
                **foreach** (Phase phase **in** phaseCollection) {  
                    **if** (phase.PhaseNumber == phaseNumber) {  
                        return **true**;  
                    }  
                }  
          
                model.**CommitChanges**();  
            }  
            return **false**;  
        }  
    }  
}