**Tekla Open API – Вставка компонента.**

Попробуем вставить компонент в модель Tekla.  
Для начала создадим перекрытие и конвертируем его в пользовательский компонент типа Деталь. Имя назначим – 101100. Вставку назначим по двум точкам.  
После этого запустим следующий код:

/\*  
 \* Created by Vadim Semenov  
 \* vad.s.semenov@gmail.com  
 \* Date: 14.04.2021  
 \*   
 \*/  
**using** System;  
**using** Tekla.Structures.Model;  
**using** Tekla.Structures.Geometry3d;  
**using** Tekla.Structures.Catalogs;  
  
**namespace** ComponentInsert  
{  
    class Program  
    {  
        **public** static void **Main**(string[] args)  
        {  
            string name ="101100";  
            **if** (**ComponentIsExist**(name)) {  
                **InsertComponenBy2Point**(name, **new** Point(0, 0, 0), **new** Point(0, 0, 50));  
            }  
        }  
          
          
        /// <**summary**>  
        /// Метод вставляющий компонент по двум точкам  
        /// <**/summary**>  
        /// <**param** **name="name"**>Имя компонента<**/param**>  
        /// <**param** **name="p1"**>Точка вставки 1<**/param**>  
        /// <**param** **name="p2"**>Точка вставки 2<**/param**>  
        /// <**returns**>True - вставка удалась<**/returns**>  
        **private** static **bool** **InsertComponenBy2Point**(string name, Point p1, Point p2)  
        {  
            **bool** insertOk = **false**;  
              
            Model model = **new** Model();  
              
            **if** (model.**GetConnectionStatus**()) {  
              
                //Создаем новый компонент  
                Component comp = **new** Component();  
                  
                //Создаем ComponentInput. ComponentInput нужен для того чтобы установить параметры вставки Component.  
                ComponentInput ci = **new** ComponentInput();  
                //Вставка Компонента в модели по двум точкам.  
                ci.**AddTwoInputPositions**(p1, p2);  
                  
                //Присваиваем ComponentInput.   
                comp.**SetComponentInput**(ci);  
                //Выбираем по имени какой компонент будем вставлять.  
                comp.Name = name;  
                  
                //Если вставка в модель удалась метод возвращает true.  
                **if** (comp.**Insert**()) {  
                    insertOk = **true**;  
                }  
                  
                model.**CommitChanges**();  
            }  
          
            return insertOk;  
        }  
          
          
          
        /// <**summary**>  
        /// Метод проверяет по имени существует ли компонент в открытом файле.  
        /// (Код взят с сайта Tekla)  
        /// <**/summary**>  
        /// <**param** **name="name"**>Имя компонента<**/param**>  
        /// <**returns**>True - компонент существует<**/returns**>  
        **public** static **bool** **ComponentIsExist**(string name)  
        {  
            **bool** Result = **false**;  
  
            //База данных файла  
            CatalogHandler CatalogHandler = **new** CatalogHandler();  
  
            **if** (CatalogHandler.**GetConnectionStatus**()) {  
                  
                // Набор существующих компонентов в каталоге  
                ComponentItemEnumerator ComponentItemEnumerator = CatalogHandler.**GetComponentItems**();  
                   
                //Перебираем все компоненты и сравнием имя у каждого.  
                **while** (ComponentItemEnumerator.**MoveNext**()) {  
                    ComponentItem ComponentItem = ComponentItemEnumerator.Current **as** ComponentItem;  
  
                    **if** (ComponentItem.Name == name) {   
                        Result = **true**;  
                        break;  
                    }  
                }  
            }  
            return Result;  
        }  
    }  
}