

**UNIVERZITA KOMENSKÉHO V BRATISLAVE**  
**FAKULTA MATEMATIKY, FYZIKY A**  
**INFORMATIKY**

**Technická dokumentácia**  
**SPEKTROSKOPICKÉ DÁTA**

**Michal Chamula,**  
**Tomáš Bordáč,**  
**Martina Bodišová,**  
**Patrik Fašang**

**2018**

# Obsah

<b>Obsah .....</b>	<b>2</b>
<b>1 Úvod.....</b>	<b>4</b>
1.1 Podstata dokumentu .....	4
1.2 Určenie dokumentu .....	4
1.3 Rozsah Systému .....	4
1.4 Slovník cudzích pojmov .....	4
1.5 Referencie.....	5
<b>2 Všeobecný popis .....</b>	<b>6</b>
2.1 Perspektíva projektu.....	6
2.2 Funkcie produktu .....	6
2.3 Charakteristika používateľov .....	6
<b>3 Špecifikácia požiadaviek.....</b>	<b>7</b>
3.1 Načítanie vstupného súboru .....	7
3.2 Zobrazenie načítaných vstupov.....	7
3.3 Výstupy do LabView .....	8
<b>4 Komunikácia s DLL.....</b>	<b>8</b>
<b>5 Funkcie .....</b>	<b>9</b>
<b>6 Popis jednotlivých funkcií .....</b>	<b>10</b>
6.1 Gauss .....	10
6.2 Lorentz .....	10
6.3 Voigt.....	10
6.4 Hartmann-tran .....	11
<b>7 Návrh.....</b>	<b>12</b>
7.1 Špecifikácia vonkajších interfejsov.....	12
7.2 Vstupný interface .....	12
7.3 Výstupný interface .....	14

---

<b>8</b>	<b>Implementácia .....</b>	<b>15</b>
<b>9</b>	<b>Diagramy.....</b>	<b>16</b>
<b>9.1</b>	<b>Sekvenčný diagram .....</b>	<b>16</b>
<b>9.2</b>	<b>Triedne diagramy .....</b>	<b>17</b>
<b>10</b>	<b>Testovací scenár.....</b>	<b>19</b>
<b>11</b>	<b>Používateľská príručka .....</b>	<b>20</b>
<b>11.1</b>	<b>Ovládanie aplikácie .....</b>	<b>20</b>
<b>11.2</b>	<b>Načítanie DLL knižnice .....</b>	<b>20</b>
<b>11.3</b>	<b>Zadanie vstupných parametrov .....</b>	<b>20</b>
<b>11.4</b>	<b>Analýza výstupu.....</b>	<b>21</b>
<b>12</b>	<b>Záver.....</b>	<b>22</b>

# 1 Úvod

## 1.1 Podstata dokumentu

Tento dokument popisuje požiadavky zadávateľa na softvér vyvíjaný v projekte Spracovanie spektroskopických dát, návrh systému, popis implementácie, testovacie scenáre a používateľskú príručku.

## 1.2 Určenie dokumentu

Tento dokument je určený stakeholderom, ktorí na základe popísaných požiadaviek budú softvér vyvíjať. Finálna verzia dokumentu špecifikácia požiadaviek je odsúhlasená zadávateľom.

## 1.3 Rozsah Systému

Projekt je dynamicky linkovaná knižnica (DLL), ktorá slúži na rátanie komplexných, neanalytických funkcií. Neobsahuje grafické používateľské rozhranie.

## 1.4 Slovník cudzích pojmov

- **DLL** -(angl. Dynamic Link Library) je množina malých programov, ktorá môže byť použitá viac ako jedným programom v tom istom čase. Táto množina je zväčša uložená v súboroch so suffixom “.dll“
- **Stakeholder**- osoba alebo skupina osôb, ktorá sa podieľa na rovnakom projekte, napr. podnikaní, programovaní, vede a pod.
- **LabVIEW** – (angl. Laboratory Virtual Instrument Engineering Workbench) je vývojové prostredie určené na vizuálne programovanie s podporou čítať DLL knižnice napísané v jazyku c++.
- **Konvolúcia** – matematický operátor spracovávajúci dve funkcie. Je definovaný vzťahom:

$$\int_{-\infty}^{\infty} f(x)g(x-\alpha)dx$$

---

## 1.5 Referencie

- [1]        *"Decay time integrals in neutral meson mixing and their efficient evaluation"* - Till Moritz Karbach, Gerhard Raven, Manuel Schiller (CERN - Switzerland, NIKHEF - The Netherlands)
  
- [2]        *"An isolated line-shape model to go beyond the Voigt profile in spectroscopic databases and radiative transfer codes"* - N.H. Ngo, D. Lisak, H. Tran, J.-M. Hartmann
  
- [3]        *"Efficient computation of some speed-dependent isolated line profiles"* - H. Tran, N.H. Ngo, J.-M. Hartmann

---

## **2 Všeobecný popis**

### ***2.1 Perspektíva projektu***

Projekt bude súčasťou väčšieho celku, ktorý má za úlohu analyzovať spektroskopické dáta. Optická Spektroskopia je oblasť fyziky, zaoberajúca sa štúdiom elektromagnetického žiarenia emitovaného alebo pohlteneho vzorkou. Získané informácie sa dajú použiť buď kvalitatívne (charakteristika vnútornej štruktúry vzorky, poprípade prostredia kde sa nachádza), alebo kvantitatívne (určenie koncentrácie známej vzorky).

### ***2.2 Funkcie produktu***

Softvér by mal byť schopný v optimálnom čase aplikovať rôzne transformácie na vstupné hodnoty – spektrá a modelovať ich tvar použitím funkcií opisujúcich žiarenie (absorpciu) vzoriek. Medzi tieto funkcie patria: Lorentzova, Gaussova, Voigtova alebo Hartmann–Tran.

### ***2.3 Charakteristika používateľov***

Finálny produkt bude využívať oddelenie experimentálnej Fyziky FMFI UK. Z používateľského hľadiska bude produkt použiteľný iba pod vývojovým prostredím LabVIEW.

---

## 3 Špecifikácia požiadaviek

### 3.1 Načítanie vstupného súboru

Systém prečíta vstupný súbor a dáta poukladá do objektov v DLL a pripraví tak namerané dáta na ďalšie spracovanie. Načítavanie dát je už implementované v knižnici DLL, ktorú sme zdedili. LabView načítané dáta zobrazí vo formulároch a nijak ich nemení. Umožňuje tieto dáta len prezerat', prípadne zvoliť transformačnú funkciu, ktorou sa tieto hodnoty prepočítajú a zobrazí počiatočný stav hodnôt, ktoré sme načítali so súboru v jednom formulári a zmenené dáta zvolenou funkciou v druhom formulári.

### 3.2 Zobrazenie načítaných vstupov

3.2.1 Prostredie LabView ponúka dva formuláre súvisiace s načítanými dátami.

3.2.2 Názov prvého formulára je **Data IN** (takto je nazvaný v LabView), ktorý ponúka používateľovi zadať:

3.2.2.1 **Name** - názov datasetu (merania) napr. pondelok, utorok , ...

3.2.2.2 Hodnotu **X-in**, kde si zvolíme z ktorého indexu chceme zobrazit' X-ovú súradnicu

3.2.2.3 Hodnotu **Y-in**, kde si zvolíme z ktorého indexu chceme zobrazit' Y-ovú súradnicu

3.2.2.4 Hodnotu **W-in**, kde si zvolíme z ktorého indexu chceme zobrazit' W (chybu merania)

3.2.3 Názov druhého formulára je **Data PAR In** (názov v LabView), ktorý ponúka používateľovi zadať:

3.2.3.1 **Name** - názov datasetu (merania) napr. pondelok, utorok , ...

3.2.3.2 **Func.names** - umožní používateľovi zvoliť funkciu, ktorou sa majú vstupné hodnoty upraviť.

---

### 3.3 Výstupy do LabView

- 3.3.1 Používateľ má k dispozícii v LabView ďalší formulár, v ktorom sa zobrazia transformované dáta. Keďže dáta sú uložené v poliach, tak sa zobrazujú iba z jedného indexu, ktorý môže používateľ meniť a tak si prezrieť postupne všetky hodnoty. Až keď používateľ zvolí funkciu a stlačí tlačidlo na transformovanie dát, tieto dáta vo formulári prepočíta podľa zvolenej funkcie a vypíše ich znovu do tohto formulára. Nasledujúci popis počíta s tým, že už bola zvolená funkcia a stlačené tlačidlo na transformáciu dát.
- 3.3.2 Názov formulára v LabView je **Data\_OUT\_Fast**, ktorý obsahuje kolónky:
- 3.3.2.1 **Name** - názov datasetu
- 3.3.2.2 Hodnotu **X\_Out**, kde si zvolíme z ktorého indexu chceme zobrazit' X-ovú súradnicu. Táto súradnica je už zmenená zvolenou funkciou.
- 3.3.2.3 Hodnotu **Y\_Out**, kde si zvolíme z ktorého indexu chceme zobrazit' Y-ovú súradnicu. Táto súradnica je už zmenená zvolenou funkciou.
- 3.3.2.4 Hodnotu **W\_Out**, kde si zvolíme z ktorého indexu chceme zobrazit' W (chybu merania). Táto chyba merania je stále rovnaká. Teda ju DLL nebude nijako meniť.

## 4 Komunikácia s DLL

DLL bude priamo komunikovať s LabView len cez funkciu `fast()`, ktorá načíta vstupný súbor. Dáta poukladá do štruktúr a nezmenené ich zobrazí v LabView. Používateľ bude môcť zvolit' funkciu, ktorou bude chcieť dáta transformovať/zmeniť. Ďalej sa budú dáta spracovávať podľa zvolenej funkcie.



---

## 5 Funkcie

Funkcie, ktoré si bude môcť používateľ zvoliť sú:

- transformácia x-ovej osi **xt()**,
- transformácia y-ovej osi **yt()**,
- **gauss()**,
- **lorentz()**,
- **voigt()**
- a hartmann-tran **htp()**.

Tieto všetky funkcie budú vždy počítať len s jednou hodnotou x, y, w, ktoré máme načítané v štruktúre. Ďalšie parametre, ktoré sú potrebné na výpočet funkcie sú načítané zo súboru a uložené do poľa, obsahujúceho štruktúry Parameters. Štruktúra obsahuje parametre pre všetky funkcie, ale vyplnené sú len tie, ktoré daná funkcia potrebuje (potrebné parametre sú popísané v časti 6. pre jednotlivé funkcie). Výsledné hodnoty X a Y sa zapíšu do nového dvojrozmerného poľa. Výpočet sa vykoná pre každú X-ovú a Y-ovú hodnotu, ktoré vstupujú do funkcií ako vektor.

---

## 6 Popis jednotlivých funkcí

Popis funkcí, které si může uživatel zvolit. Nie sú tu popísané funkcie **xt()** a **yt()**, pretože už sú implementované v zdedenom DLL.

### 6.1 Gauss

Gauss() bude počítat postupne s každou hodnotou X, Y. Tieto hodnoty tvoria vektor **v**, ktorý vstupuje do funkcie. Parametre pre  $\Gamma_D$ ,  $v_0$ ,  $\Delta_0$  načíta so svojej sady parametrov. Výsledkom bude vektor so zmenenými hodnotami X a Y, ktoré budú uložené do poľa výsledkov na rovnakom indexe, ako boli pôvodné hodnoty X, Y. Definičný obor je od najmenšie načítaného X, po najvyššie načítané X.

$$P(v) = \frac{1}{|\Gamma_D|} \sqrt{\frac{\ln(2)}{\pi}} \exp\{-\ln 2 * \frac{[v-(v_0+\Delta_0)]^2}{\Gamma_D^2}\}$$

### 6.2 Lorentz

Lorentz() bude počítat postupne s každou hodnotou X, Y. Tieto hodnoty tvoria vektor **v**, ktorý vstupuje do funkcie. Parametre pre  $\Gamma_0$ ,  $v_0$ ,  $\Delta_0$  načíta so svojej sady parametrov. Výsledkom bude vektor so zmenenými hodnotami X a Y, ktoré budú uložené do poľa výsledkov na rovnakom indexe, ako boli pôvodné hodnoty X, Y. Definičný obor je od najmenšie načítaného X, po najvyššie načítané X.

$$P(v - v_0) = \frac{1}{\pi} \frac{|\Gamma_0|}{[(v - (v_0 + \Delta_0))]^2 + \Gamma_0^2}$$

### 6.3 Voigt

Voigt() bude počítat postupne s každou hodnotou X, Y. Tieto hodnoty tvoria vektor **v**, ktorý vstupuje do funkcie. Parametre pre  $\Gamma_D$ ,  $v_0$ ,  $w_0$ ,  $\Delta_0$  načíta so svojej sady parametrov. Výsledkom bude vektor so zmenenými hodnotami X a Y, ktoré budú uložené do poľa výsledkov na rovnakom indexe, ako boli pôvodné hodnoty X, Y. Definičný obor je od najmenšie načítaného X, po najvyššie načítané X.

$$P(v) = \frac{1}{|\Gamma_D|} \sqrt{\frac{\ln(2)}{\pi}} * \text{Re}[w(z)], \quad z = \ln 2 \frac{w_0 + i(v_0 + \Delta_0 - v)}{\Gamma_D}$$

$$w(z) = e^{-z^2} \text{erfc}(-z)$$

#### 6.4 Hartmann-tran

Hartmann-tran bude počítať postupne s každou hodnotou  $X$ ,  $Y$ . Tieto hodnoty tvoria vektor  $v$ , ktorý vstupuje do funkcie. Parametre pre  $v_0$ ,  $C_0$ ,  $C_2$ ,  $\Delta_0$ ,  $Z$  načíta so svojej sady parametrov, pričom  $Z$  je komplexné číslo a preto do výsledku prepočítaných hodnôt vektora vezmeme do úvahy iba reálnu časť. Výsledkom bude vektor so zmenenými hodnotami  $X$  a  $Y$ , ktoré budú uložené do poľa výsledkov na rovnakom indexe, ako boli pôvodné hodnoty  $X$ ,  $Y$ . Definičný obor je od najmenšieho načítaného  $X$ , po najvyššieho načítaného  $X$ .

$$F_{\text{HTP}}(v) = \frac{1}{\pi} \text{Re} \left\{ \frac{A(v)}{1 - [v_{vc} - \eta \left( C_0 - \frac{3C_2}{2} \right)] A(v) + \left( \frac{\eta C_2}{V_{a0}^2} \right) B(v)} \right\}$$

$$w(z) = e^{-z^2} \text{erfc}(-z)$$

$$A(v) = \frac{\sqrt{\pi}c}{V_0 V_{a0}} [w(iZ_-) - w(iZ_+)]$$

$$B(v) = \frac{V_{a0}^2}{C_2} \left[ -1 + \frac{\sqrt{\pi}}{2\sqrt{Y}} (1 - Z_-^2) w(iZ_-) - \frac{\sqrt{\pi}}{2\sqrt{Y}} (1 - Z_+^2) w(iZ_+) \right]$$

$$Z_{\pm} = \sqrt{X + Y} \pm \sqrt{Y}$$

$$X = \frac{-i(V_0 - v) + C'_0}{C'_2}$$

$$Y = \left( \frac{V_0 V_{a0}}{2cC'_2} \right)^2$$

$$C'_0 = (1 - \eta) \left( C_0 - \frac{3C_2}{2} \right) + V_{vc}$$

$$C'_2 = (1 - \eta)C_2$$

$$V_{a0} = \frac{c}{\sqrt{\ln 2 V_0}}$$

---

## 7 Návrh

### 7.1 Špecifikácia vonkajších interfejsov

DLL knižnica vyvíjaná v tomto projekte bude komunikovať s prostredím LabView. Používateľ načíta dáta z grafického rozhrania v programe LabView a nastaví vstupné parametre v tabuľke, ktoré bude vedieť DLL prečítať, spracovať a výsledok ponúknuť používateľovi.

### 7.2 Vstupný interface

1. Popis vstupného formuláru, Spektrum (obr.1):

- 1.1. Polia:

**Name:** meno dát

**X-in:** x-ová súradnica

**Y-in:** y-ová súradnica

**W-in:** neistota

Obrázok 1: Vstupný formulár

2. Popis vstupného formuláru, Parametre (obr. 2)

- 2.1. Polia:

**Param.strings:** charakterizujúce mená parametrov a ich vlastnosti (meno, model, nezávislé parametre)

**Param.values:** charakterizujúce hodnoty parametrov (hodnota, neistota, škála pre GUI)

**Func.names:** (L je počet transformácií a funkcií modelu – max

---

6: XT, YT, BL, PK, RF, MC, ktoré treba vypočítať), prvý stĺpec obsahuje meno funkcie (XT, YT, BL, PK, RF, MC), druhý mená „skupín=groups“ v rámci danej funkcie zoradených do jedného stringu: Meno1@Meno2@...@MenoN

**Func.par.adresses:** 3xLxQ (Q je max počet skupín nachádzajúci sa niektorej z funkcií), prvý inde(page) definuje funkciu podľa poradia v poli Func.names následne každá skupina má jeden riadok, kde prvá hodnota hovorí koľko hodnôt je v danom riadku (adries – poradie v Param.strings a Param.values), poradie adries parametrov je pevne definované pre každú funkciu-skupinu

**Data.names:** Mená vektorov pred simulovaných dát pre funkciu REF

**Data.length:** Zodpovedajúca dĺžka(počet bodov) pre vektory pred simulovaných dát (uložené v binárnom súbore na disku)

Data PAR in

Name	
Param.strings	
▲▼ 0	
▲▼ 0	
Param.numbers	
▲▼ 0	0
▲▼ 0	
Func.names	
▲▼ 0	
▲▼ 0	
Func.Par.adresses	
▲▼ 0	0
▲▼ 0	
▲▼ 0	
Data.names	▲▼ 0
Data_length	▲▼ 0

Obrázok 2: Štruktúra vstupných dát

---

### 7.3 Výstupný interface

1. Popis výstupného formuláru, Spektrum (obr. 3):

1. Polia:

**Name:** meno dát

**X-in:** x-ová súradnica

**Y-in:** y-ová súradnica

**W-in:** neistota

Data\_OUT\_Fast

Name			
X_out	0	0	W_out
Y_out	11	0	F
			100
			0

Obrázok 3: Štruktúra výstupných dát

---

## 8 Implementácia

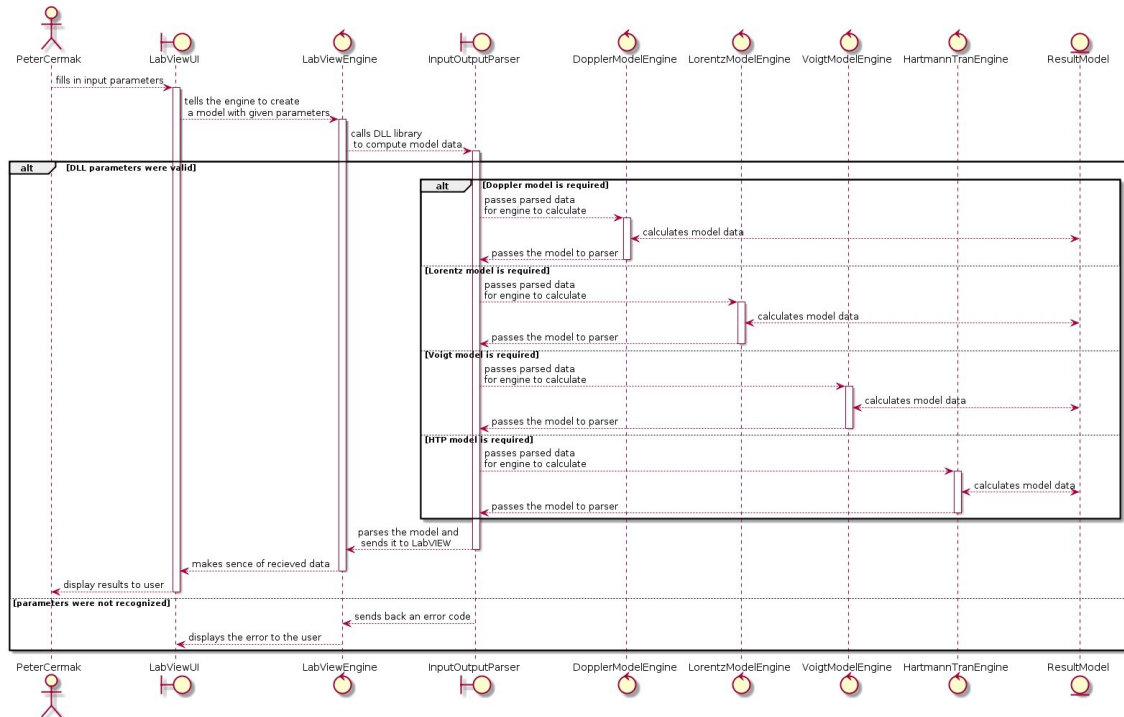
Z LabView je z DLL volaná funkcia `fdata_fast()`, ktorú začneme upravovať ako prvú. Do jej tela zimplementujeme volanie funkcií na výpočet dát, ktoré sú popísané v katalógu požiadaviek. Volaná metóda má vstupné parametre pointe na 4 štruktúry (`TD1 *DataPARin`, `TD7 *DataIN`, `TDFast *DataOUT_F`, `TD10 *Error`) a výstupom je typ `<int32_t>` s ktorým ale nič nerobíme ani neupravujeme, takže ho ani neriešime. Všetky hodnoty s ktorými pracujeme sú iba v štruktúrach. Štruktúry sú pevne definované a nemožno ich meniť. Zavolaním tejto funkcie sa dáta vypočítajú, upraví a pošlú do štruktúry `TDFast *DataOUT_F` v ktorej zostanú uložené. Po zavolaní metódy sa vypočítané dáta vykreslia v tabuľke `Data_OUT_Fast`. Funkcie a metódy pre výpočet dát budú dopísané do triedy `Transform`. Funkcia `fdata_fast()` už od staršej verzii programu vytvára inštanciu tohto objektu kde sú statické funkcie a metódy takže do jej tela treba implementovať volanie našich metód pre výpočet dát.

Každá funkcia bude mať vlastný template. Tie vytvoríme v súbore `PeakFunkcions.h`. Každá funkcia v tomto súbore bude obsahovať inicializáciu. Inicializácia bude funkcia, ktorá do členskej premennej uloží vstupný parameter typu `double`. Najdôležitejšia funkcia v rámci template bude funkcia `Apply`, ktorej vstupné parametre budú **in** a **out** a budú typu `double`. Táto funkcia bude modifikovať vstupné dáta **in**, podľa matematického vzorca popísaného pre každú matematickú funkciu. Prepočítané dáta uloží do **out**.

## 9 Diagramy

### 9.1 Sekvenčný diagram

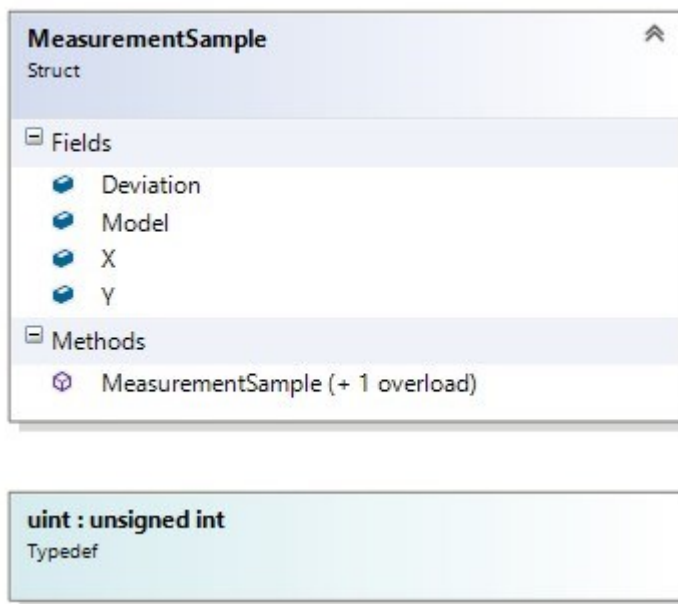
Tento sekvenčný diagram detailnejšie popisuje fungovanie vyššie uvedeného použitia. Zamieriava sa na interakciu medzi LabVIEW programom a DLL knižnicou, ktorá obsahuje moduly na spracovanie vstupu, transformáciu funkciami a vyprodukovanie modelov.

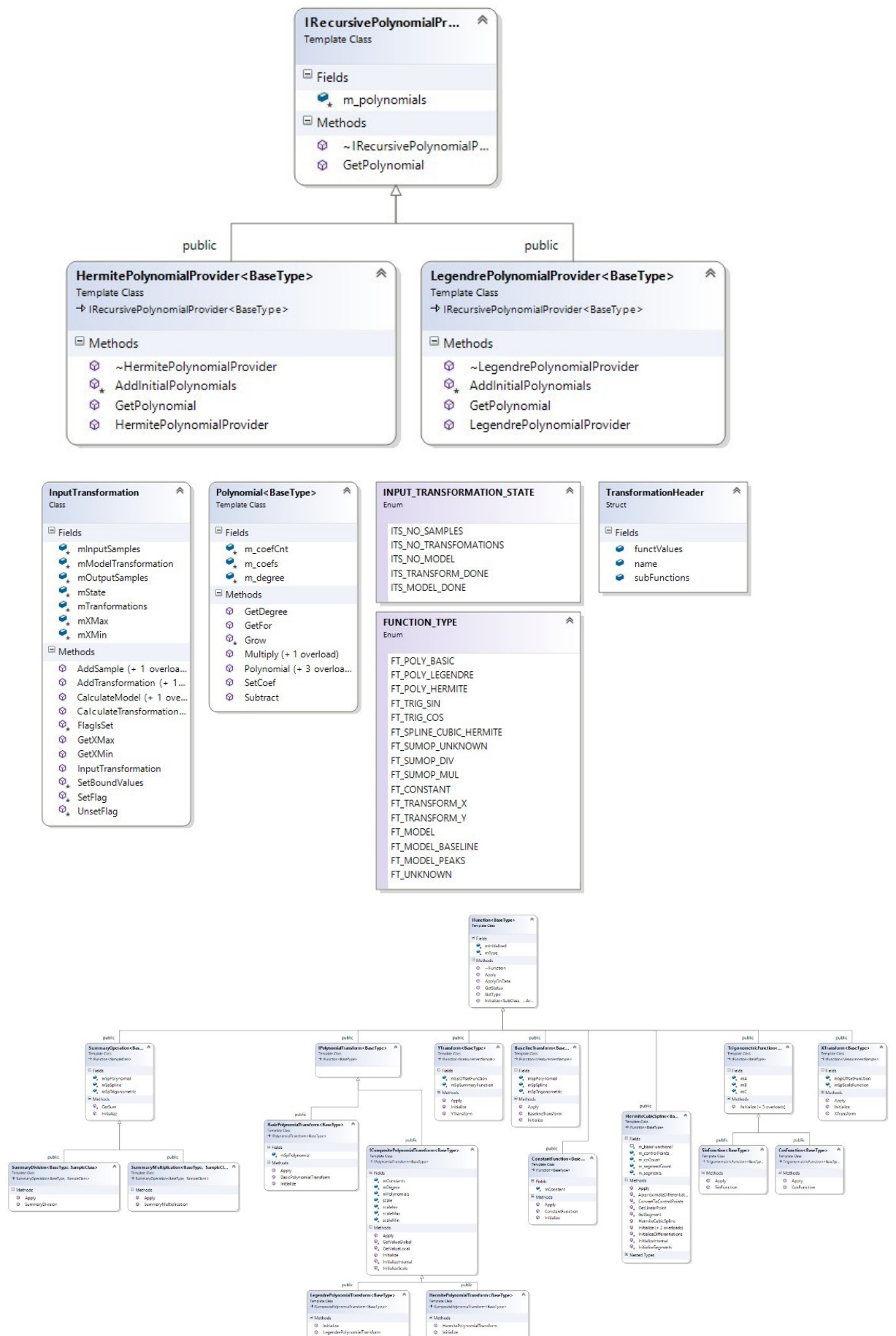




---

## 9.2 Triedne diagramy





---

## 10 Testovací scénár

o Vstup:

1. Vyplnenie vstupných parametrov v grafickom prostredí

LabView

2. Zvolí sa funkcia, ktorou majú byť vstupné dáta spracované
3. Spustíme proces vypočítania dát z prostredia LabView stlačením tlačidla

o Výstup:

1. Kontrola výstupných parametrov naplnených v tabuľke data\_out\_fast
2. Kontrola požadovaného grafického modelu
3. Kontrola grafického modelu po zvolení údajov do transformačných funkcií

---

## 11 Používateľská príručka

### 11.1 Ovládanie aplikácie

Naša aplikácia je DLL knižnica, ktorá poskytuje výpočtovú funkcionálnosť. Je úzko spätá so softvérom LabVIEW, ktorý poskytuje aj užívateľské rozhranie na používanie tejto knižnice. Samozrejme, ako každú inú knižnicu, aj túto knižnicu je možné použiť mimo LabVIEW. Takéto využitie nie je očakávané a vyžadovalo by vytvorenie špecifických vstupných parametrov, ktoré LabVIEW generuje z užívateľského vstupu. Preto je z takéhoto hľadiska pre používateľa nepraktické.

### 11.2 Načítanie DLL knižnice

Závisí od konkrétnej konfigurácie LabVIEW aplikácie. V našom konkrétnom prípade sa musí DLL knižnica volať **Win32Project\_AIstart** a musí byť umiestnená v adresári **data**.

### 11.3 Zadanie vstupných parametrov

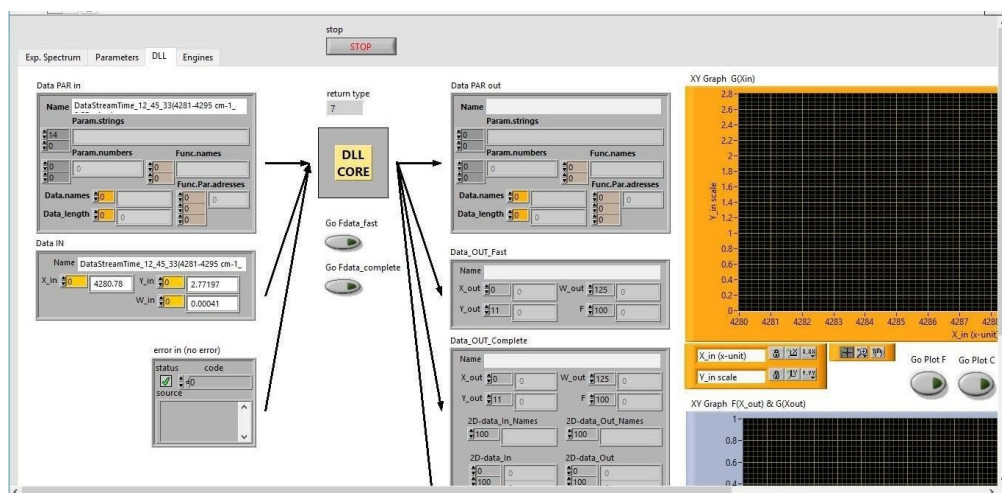
Formát vstupu taktiež závisí od konkrétnej aplikácie. Na ukážke nášho prípadu sú k dispozícii dva spôsoby a to vyplnenie formulára, alebo import zo súboru.

Param.strings		
XT@XOff@p0	C	
YT@YTyp@function	C_YT	
YT@YSpl@p0-0	C	
YT@YSpl@p0-1	C	
YT@YSpl@p1-0	C	
YT@YSpl@p1-1	C	
YT@YSpl@p2-0	C	
YT@YSpl@p2-1	C	
PK@C1@sample	C	
PK@C1@A/S	C	
PK@C1@shape	C_PT	
PK@C1@center	C	
PK@C1@surface	C	
PK@C1@amplitude	C	
PK@C1@w0	C	
PK@C1@wD	C	
PK@C1@d0	C	

Param.numbers		
0.1	0	1
2	0	0
4280	0	0
2.8	0	0
4288	0	0
2.6	0	0
4295	0	0
2.8	0	0
11	0	0
0	0	0
4290	0	0
0.5	0	0
0.1	0	0
0.01	0	0
0.001	0	0
0.001	0	0
0.1	0	0
0	0	0

## 11.4 Analýza výstupu

Tlačidlami “Go\_fdata\_fast” a “Go\_fdata\_complete” sa púšťa výpočet so zadanými vstupnými parametrami. V poli “return” type sa zobrazuje výstupný kód a dáta, ktoré poskytne DLL knižnica sa použijú na vykreslenie grafu v pravej časti.



---

## 12 Záver

Knižnica DLL obsahuje ďalšie možnosti transformácie dát, podľa požiadaviek zadávateľa projektu. Umožňuje používateľovi zvoliť v prostredí LabView transformáciu vstupných dát pomocou funkcií: Gauss, Lorentz, Voigt, Harttman trann. Používateľ môže zvoliť jednu z týchto funkcií. Prepočítané dáta sa mu zobrazia na grafe v LabView. Tieto zmenené dáta sú výstupom DLL knižnice, ktorá komunikuje z LabView.