

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2021

Bc. Michal Gavenčíak





# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## VÝZKUM NOVÝCH PARAMETRŮ ONLINE PÍSMĀ U DĚTÍ S GRAFOMOTORICKÝMI OBTÍŽEMI

RESEARCH OF NEW ONLINE HANDWRITING FEATURES IN CHILDREN WITH GRAPHOMOTOR  
DIFFICULTIES

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Michal Gavenčiak

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vojtěch Zvončák

BRNO 2021





# Diplomová práce

magisterský navazující studijní program **Audio inženýrství**  
specializace Zvuková produkce a nahrávání  
Ústav telekomunikací

**Student:** Bc. Michal Gavenčiak

**ID:** 183430

**Ročník:** 2

**Akademický rok:** 2020/21

## NÁZEV TÉMATU:

**Výzkum nových parametrů online písma u dětí s grafomotorickými obtížemi**

## POKYNY PRO VYPRACOVÁNÍ:

Provedte rešerši odborné literatury a na jejím základě navrhnete nové parametry pro kvantifikaci potíží se psaním (např. tempo psaní, plynulost pohybu, změna velikosti písma apod.). Dílčími cíli diplomové práce budou: revize rozsáhlé databáze záznamů online písma u dětí s typickým vývojem psaní a u dětí s obtížemi se psaním; návrh a implementace programu pro zobrazení jednotlivých cvičení z databáze a jejich následné hodnocení (program bude přizpůsoben potřebám speciálních pedagogů). Vybrané parametry tzv. online písma následně implementujte v jazyce Matlab/Python/C# a proveďte explorativní analýzu, kde bude hodnoceno jejich využití při popisu různých grafomotorických obtíží. Dosáhnuté výsledky budou diskutovány ve vztahu k současnému stavu poznání.

## DOPORUČENÁ LITERATURA:

[1] ASSELBORN, Thibault, Mateo CHAPATTE a Pierre DILLENBOURG. Extending the Spectrum of Dysgraphia: A Data Driven Strategy to Estimate Handwriting Quality. Scientific Reports. 2020, 10(1). DOI: 10.1038/s41598-020-60011-8. ISSN 2045-2322.

[2] MEKYSKA, Jiri, Jirina BEDNAROVA, Marcos FAUNDEZ-ZANUY, et al. Computerised Assessment of Graphomotor Difficulties in a Cohort of School-aged Children. 2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). IEEE, 2019, 2019, 10(1), 1-6. DOI: 10.1109/ICUMT48472.2019.8970767. ISBN 978-1-7281-5764-1. ISSN 2045-2322.

**Termín zadání:** 1.2.2021

**Termín odevzdání:** 24.5.2021

**Vedoucí práce:** Ing. Vojtěch Zvončák

**doc. Ing. Jiří Schimmel, Ph.D.**  
předseda rady studijního programu

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.



## ABSTRAKT

V České republice doposud neexistují objektivní metody pro diagnostiku grafomotorických potíží u dětí. Probíhající výzkumy používají moderních digitizérů pro zachycení procesu psaní a následné kvantifikování parametrů písma. Mezi cíli této práce je vyvinout softwarové nástroje pro usnadnění práce se sbíranými daty, jako je třídění databáze a usnadnění práce odborníků při hodnocení jednotlivých cvičení. Dalším cílem práce je navrhnout a následně analyzovat nové parametry on-line písma na kohortě dětí ze 2. až 4. třídy ( $n=239$ ). V rámci práce byly vyvinuty dvě desktopové aplikace na platformě .NET a byly navrženy a naprogramovány tři nové kvantifikační parametry písma založené na principu Isochronie, dvourozměrné korelaci, a geometrickém centroidu. Všechny tři parametry vykázaly signifikantní korelaci ( $r = [0,2; 0,3]$ ) s hodnotami HPSQ-C u dětí druhé a čtvrté třídy a korelaci v rozsahu ( $\rho = [0,2; 0,5]$ ) se subjektivním hodnocením odborníka u celé kohorty. Analýza parametrů naznačuje potíže s regulací tempa písma a pracovní paměti u dětí s grafomotorickými potížemi.

## KLÍČOVÁ SLOVA

Grafomotorické potíže, dysgrafie, .NET, C#, On-line písmo, korelační analýza, isochronie, 2D korelace, centroid, databáze, HPSQ-C

## ABSTRACT

In the Czech Republic, there is currently no objective method to diagnose graphomotor difficulties in children. Ongoing research uses modern digitizers to capture the handwriting process and quantify its parameters. The first goal of this thesis is to develop software tools to facilitate work with the collected data, such as database validation and writing exercise rating, done by specialists. Another goal of this thesis is to design new on-line handwriting parameters which are then to be analysed on a cohort of school children from 2nd to 4th class of primary school ( $n=239$ ). The implementation of two desktop programs on the .NET platform is described, among three new quantifying parameters based on the principles of isochrony, two-dimensional cross-correlation, and geometrical centroid. All three parameters show significant correlation ( $r = [0,2; 0,3]$ ) with the HPSQ-C rating in 2nd- and 4th-graders and correlation ( $\rho = [0,2; 0,5]$ ) with specialist's subjective scores in all children from the cohort. The analysis suggests children with graphomotor difficulties struggle with regulating handwriting speed and working memory.

## KEYWORDS

Graphomotor difficulties, dysgraphia, .NET, C#, On-line handwriting, correlation analysis, isochrony, 2D cross-correlation, centroid, database, HPSQ-C





GAVENČIAK, Michal. *Výzkum nových parametrů online písma u dětí s grafomotorickými obtížemi*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2021, 91 s. Diplomová práce. Vedoucí práce: Ing. Vojtěch Zvončák



## Prohlášení autora o původnosti díla

<b>Jméno a příjmení autora:</b>	Bc. Michal Gavenčíak
<b>VUT ID autora:</b>	183430
<b>Typ práce:</b>	Diplomová práce
<b>Akademický rok:</b>	2020/21
<b>Téma závěrečné práce:</b>	Výzkum nových parametrů online písma u dětí s grafomotorickými obtížemi

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora\*

---

\*Autor podepisuje pouze v tištěné verzi.



## PODĚKOVÁNÍ

Rád bych poděkoval panu Ing. Vojtěchovi Zvončákovi, za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci a paní PhDr. Kataríně Šafárové, Ph.D. za konzultace a připomínky k analýze dat.



# Obsah

Úvod	21
<b>1 Databáze</b>	<b>23</b>
1.1 Charakter a struktura dat	23
1.2 Softwarové nástroje	24
1.3 Druhy chyb	25
1.3.1 Chyby napravitelné	25
1.3.2 Chyby nevratné	26
1.4 Složení databáze	28
<b>2 Sběr experimentálních dat</b>	<b>31</b>
2.1 Vývoj aplikace	31
2.1.1 Data management	32
2.1.2 Dataflow	33
2.1.3 GUI	36
<b>3 Návrh parametrů a statistická analýza</b>	<b>49</b>
3.0.1 Normalizovaný centroid	49
3.0.2 Normalizovaný korelační koeficient	50
3.0.3 Isochornie	54
3.1 Programové řešení extrakce parametrů	56
3.1.1 Objektová databáze	57
3.1.2 Implementace extrakce parametrů v kódu	59
3.2 Analýza dat	64
<b>4 Výsledky</b>	<b>69</b>
4.1 Statistická analýza	69
4.1.1 2. třída	69
4.1.2 3. třída	72
4.1.3 4. třída	74
<b>5 Diskuse</b>	<b>77</b>
5.1 Validace databáze	77
5.2 Aplikace pro sběr hodnocení	77
5.3 Statistická analýza nových parametrů	78
5.3.1 Limitace	79
<b>Závěr</b>	<b>81</b>

Literatura	83
Seznam symbolů a zkratk	87
Seznam příloh	89
A Obsah přiloženého adresáře	91



# Seznam obrázků

1.1	Záznam kreslicího cvičení . . . . .	23
1.2	Struktura Dat . . . . .	24
1.3	Aplikace pro ruční validaci dat . . . . .	25
1.4	Chyba segmentace . . . . .	26
1.5	Chyba labelování . . . . .	27
1.6	Chyba záznamu . . . . .	27
1.7	Chyba kreslení . . . . .	28
1.8	Neznámá chyba . . . . .	29
2.1	Proud dat mezi aplikací a soubory na disku počítače uchovávající metadata a nové informace. . . . .	32
2.2	Obecné schéma celé aplikace Skořápka. Aplikace čte data ze souborového systému do tří nezávislých zásobníků a zobrazuje je v GUI. Uživatel s GUI interaguje pomocí myši a klávesnice. . . . .	35
2.3	GUI aplikace Skořápka. V horní části ovládací panel aplikace. V pravé části legenda ovládaní a předloha cvičení. V bílé ploše uprostřed je zobrazeno právě hodnocené cvičení. . . . .	36
2.4	Ovládací panel aplikace obsahující možnosti otevírání souborů, přepínání režimů zobrazení, hodnocení cvičení a navigace. . . . .	37
2.5	Zobrazení při nízkém počtu cvičení. Aplikace přizpůsobí rozložení a velikost cvičení aby došlo k co největšímu zobrazení jednotlivých cvičení. . . . .	41
2.6	Zobrazení při vyšším počtu cvičení. Aplikace přizpůsobí rozložení a velikost cvičení aby došlo k co největšímu zobrazení jednotlivých cvičení. . . . .	41
2.7	Vlevo úsečka s viditelnými hranami a aliasingem. Vpravo stejná úsečka s náznakem jednoduchého antialiasingu. . . . .	42
2.8	Vlevo vykreslení části cvičení bez vyhlazování hran (antialiasing), vpravo část cvičení s vyhlazováním hran . . . . .	43
2.9	Ukázka vyskakovacího zoom okénka při zobrazení více cvičení najednou . . . . .	47
3.1	Max Pooling . . . . .	52
3.2	Závislost doby tahu na jeho délce pro psané cvičení 36 a všechny subjekty datasetu. Z grafu je patrný lineární vztah mezi oběma proměnnými, v souladu s předpokladem Isochronie. . . . .	55
3.3	Diagram databáze sloužící pro ukládání metadat cvičení a následně vypočtených parametrů a mezivýsledků výpočtu. . . . .	58
3.4	Zjednodušený flowchart výpočetního algoritmu pro výpočet nových parametrů on-line písma. . . . .	61

3.5	Diagram výpočtu korelační matice. $i$ a $j$ představují iterační proměnné vnořených for cyklů. Vypočtená hodnota je vložena do korelační matice na pozici $[j; i]$ . . . . .	63
3.6	Rozložení věku (a) a tříd (b) . . . . .	65
3.7	Histogramy použitých subskórů hodnocení HPSQ-C napříč celým použitým datasetem (a-e), včetně sumy subskórů 1, 2 a 10 (f). . . . .	67
3.8	Histogramy SHP napříč celým použitým datasetem. Celkového hodnocení (a) a jednotlivá hodnocení cvičení použitých pro analýzu (b-g). . . . .	68

# Seznam tabulek

1.1	Přehled složení databáze . . . . .	29
3.1	Věková statistika databáze . . . . .	64
3.2	Rozložení databáze do tříd . . . . .	64
3.3	Cvičení označená hvězdičkou jsou tzv. párová (viz kap. 3). Parametr je počítán ze cvičení uvedeného a cvičení následujícího, tedy <b>TSK8*</b> byl počítán z <b>TSK8</b> a <b>TSK9</b> . V závorkách jsou uvedeny zkratky značení parametrů použité při analýze. . . . .	65
4.1	Korelační analýza (Pearson) nových parametrů a hodnocení HPSQ-C u dětí 2. třídy . . . . .	70
4.2	Korelační analýza (Spearman) nových parametrů a SHP u dětí 2. třídy	71
4.3	Korelační analýza (Pearson) nových parametrů a hodnocení HPSQ-C u dětí 3. třídy . . . . .	72
4.4	Korelační analýza (Spearman) nových parametrů a SHP u dětí 3. třídy	73
4.5	Korelační analýza (Pearson) nových parametrů a hodnocení HPSQ-C u dětí 4. třídy . . . . .	74
4.6	Korelační analýza (Spearman) nových parametrů SHP u dětí 4. třídy	75



# Seznam výpisů

2.1	HitTestFilterBehaviour, který přeskočí vše, co není typu Image . . . .	44
2.2	Vytvoření vykrajovacího čtverce a následné vykrojení z původního obrázku . . . . .	46
3.1	Příklad načítání dat z SQL . . . . .	64



# Úvod

Přesto, že moderní doba přinesla nové způsoby psané komunikace, založené především na interakci s klávesnicí, nebo dotykovým displayem, psaní na papír zůstává stále důležitou dovedností, které musí dítě zvládnout pro úspěšný akademický život [4]. Psaním dítě tráví 30-60% času, který ve škole stráví [10], přesto potíže se psaním postihuje 10-30% z nich [8] [5].

Diagnóza grafomotorických potíží u dětí je přitom poměrně obtížná. Alespoň v České Republice prozatím neexistuje objektivní způsob, který by umožňoval určit přítomnost, či závažnost poruchy bez vlivu hodnotitele, který může výsledek ovlivnit svými zkušenostmi.

Zkoumání parametrů písma není disciplína nová. Avšak v nedávné době zveřejněné práce ukazují možnosti moderních výpočetních technologií a jejich uplatnění pro vyhodnocování grafomotorických potíží. v roce 2017 Palmis et al. [15] ukazují vývoj psaného projevu s věkem dítěte mezi pěti a dvanácti lety. V tom samém roce Pagliarini et al. uvádějí isochronii a homotetii jako principy, které řídí proces psaní [14]. O rok později Rosenblum a Dror využívají strojové učení na skupině 99 dětí pro vytvoření modelu s 90% sensitivitou i specificitou pro detekci vývojové dysgrafie [17]. Asselborn et al. vyvinuli diagnostickou metodu založenou na náhodných lesích a vzorcích online písma 298 dětí s 96% sensitivitou a 99% specificitou [3]. V roce 2020 navíc Asselborn et al. přicházejí s novým skórovacím systémem, který nenabízí jen celkové hodnocení, ale umožňuje hodnotit psaný projev ve čtyřech samostatných doménách [2].

Tato práce si klade tři cíle:

1. Validace rozsáhlého datasetu poskytnutého pro výzkum. Vzhledem k množství dat bude nutné vyvinout softwarový nástroj pro usnadnění třídění dat.
2. Návrh a implementace desktopové aplikace pro subjektivní hodnocení kreslicích a psacích cvičení odborníkem.
3. Návrh několika nových parametrů online písma, které by mohly být využity pro diagnostiku grafomotorických potíží u dětí.

V kapitole 1 jsou popsána předložená data, program vytvořený pro usnadnění validace a samotný proces třídění. Ve 2. kapitole je rozebrána problematika návrhu aplikace pro hodnocení cvičení. Kapitola 3 se pak věnuje návrhu nových parametrů. Ty jsou nejprve popsány teoreticky a následně implementovány v kódu. Poslední částí jsou výsledky a diskuse, kde jsou krátce rozebrány a rekapitulovány cíle práce, jak jich bylo dosaženo a jak bylo navrženo řešení účinné.





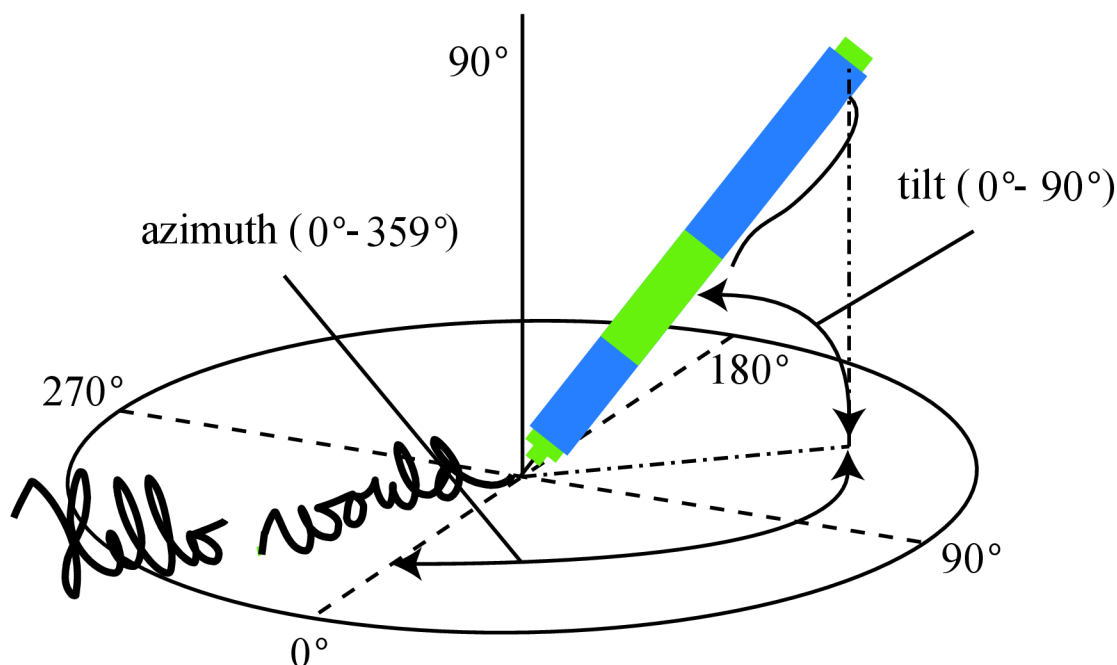
# 1 Databáze

Hlavním cílem práce je návrh nových parametrů pro posuzování grafomotorických potíží. Poskytnutá databáze však nebyla validována a bylo tedy v první řadě nutné projít všechny záznamy a ověřit, že všechny soubory jsou řádně segmentovány, nalabelovány a zároveň i jejich obsah netrpí některým z nedostatků, způsobeným např. chybou tabletu apod.

Data byla získána na několika školách a školkách v České Republice. Děti, v rozpětí od předškolních až po čtvrtou třídu, vyplnily protokol sestávající z 36 kreslicích a psacích cvičení na digitalizační tablet. Rodiče všech dětí podepsaly informovaný souhlas, schválený etickou komisí Masarykovy univerzity.

## 1.1 Charakter a struktura dat

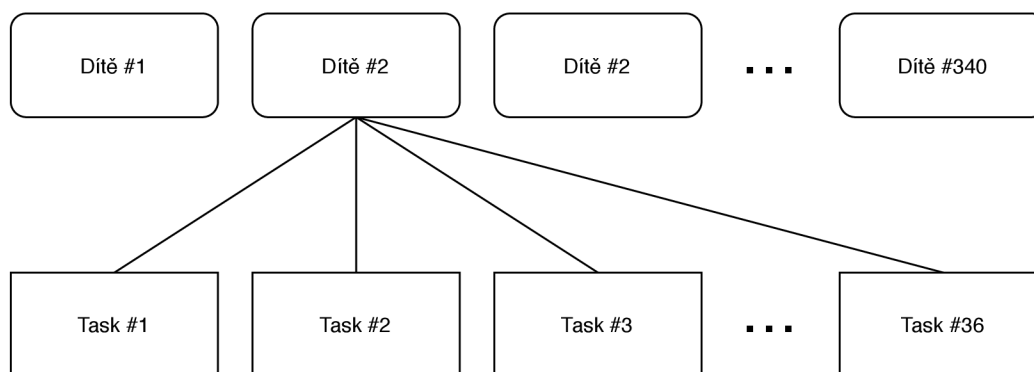
Obsahem dat jsou kreslicí a psací cvičení, vyplněné pomocí digitalizačních tabletů a stylusu (např. Wacom In-tuos Pro L a Wacom Inking pen). Tablet se vzorkovací periodou v řádech stovek Hertzů zaznamenává stavové údaje stylusu. Mezi těmito údaji jsou: čas, poloha, přítlak, sklon pera, natočení pera a informace, zda se pero dotýká plochy tabletu, viz obrázek 1.1. Tablet navíc dokáže zaznamenat polohu stylusu i při pohybu v krátké vzdálenosti nad tabletem.



Obr. 1.1: [11] Záznam kreslicího cvičení

Celkem bylo takto, za každé dítě, zaznamenáno až 36 cvičení, která však nebyla při exportu nijak oddělena a je nutno je před dalším použitím segmentovat. Toho bylo docíleno z části manuální cestou editace souboru, nebo s použitím specializovaného softwaru, který umožňuje alespoň z části proces automatizovat [6].

Data jsou tedy roztríděna do celkem 340 složek, kde každá představuje jedno dítě, které podstoupilo vyplnění kreslicího protokolu. Každá složka pak obsahuje až 36 souborů, kde každý z nich představuje právě jedno cvičení, tak jako na obrázku 1.2. Celkem bylo dodáno 9056 souborů, v průměru 26,3 cvičení na jedno dítě. Některé soubory byly pravděpodobně vyřazeny už při labelování, nebo dané dítě jedno, či více cvičení vůbec nekreslilo.

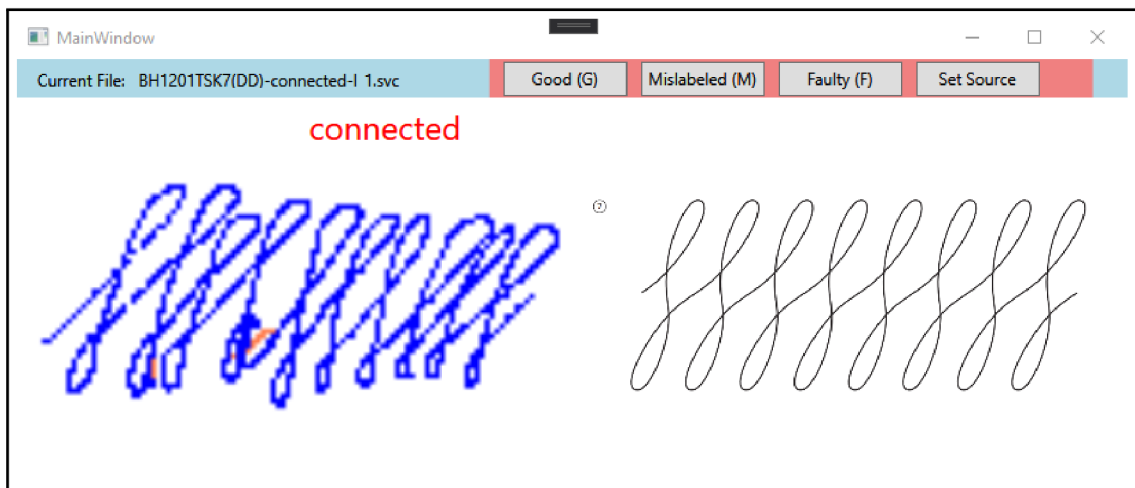


Obr. 1.2: Struktura dat

## 1.2 Softwarové nástroje

Vzhledem k obrovskému množství souborů bylo nutné vytvořit pro validaci databáze software, který umožní každé cvičení otevřít, vykreslit včetně předlohy a umožnit uživateli zvolit, zdali je soubor v pořádku, či nikoliv. Pro implementaci aplikace bylo zvoleno prostředí WPF (Windows Presentation Foundation), založené na .NET Core 3.1 (C#). WPF umožňuje velmi rychlou tvorbu aplikací s GUI (Graphical User Interface), zatímco funkce .NET Core, jako například LINQ (Language-Integrated Query) a Entity Framework jsou velice mocnými nástroji pro práci s velkými objemy dat a umožňují velice rychlé implementace datových přenosů, filtrací apod.

Aplikace přes svoji relativní jednoduchost umožňuje velice rychlé validování dat. Uživatel aplikace vidí v levé části obsah souboru, v pravé potom předlohu protokolu. Může tak snadno vyhodnotit, jestli přidělený label odpovídá obsahu souboru. Zároveň je zobrazen název souboru a název cvičení. Tlačítkem **Set Source** se otevře dialogové okno, které umožní zvolení souboru s cvičením (přípona .svc). Program



Obr. 1.3: Aplikace pro ruční validaci dat

pak projde všechny soubory složky, ve které se dané cvičení nachází a složky na stejné úrovni. Najde v nich všechny soubory SVC a ty tak není nutné jednotlivě otevírat. Tlačítka **Good (G)**, **Mislabeled (M)**, **Faulty (F)** se následně volí, co se má s aktuálně prohlíženým souborem stát. Po stisknutí tlačítka, nebo klávesy na klávesnici odpovídající znaku v závorce, program přesune soubor do složky s odpovídajícím názvem (Např. Good) a podsložky pojmenované po kódovém označení dítěte. Pokud složky neexistují, program je sám vytvoří. Následně se automaticky otevře další soubor a vyhodnocování tak může plynule pokračovat.

## 1.3 Druhy chyb

Obecně lze chyby dělit do dvou kategorií:

- Chyby napravitelné, které byly do souborů zaneseny většinou chybou v některém z předchozích kroků zpracování dat. Ty jsou však napravitelné, pokud jsou zachovány zdrojové soubory.
- Chyby nevratné, které znehodnocují použitelnost daného souboru, ale na rozdíl od předchozí kategorie vznikly už při samotném zaznamenání a pravděpodobně je tak nebude možné opravit.

### 1.3.1 Chyby napravitelné

V průběhu validace bylo u 20 různých dětí odhaleno celkem 26 souborů s tímto druhem chyby. V rámci programu je pro jejich označení používáno tlačítko **Mislabeled (M)**.

## Chyba segmentace

V tomto případě se jedná o chybu, která vznikla v důsledku špatné segmentace zdrojového souboru. Chybu lze většinou poznat tak, že soubor obsahuje body, které patří k jinému cvičení, nebo jsou velmi vzdálené od ostatních tahů, tvořících cvičení.



Obr. 1.4: Soubor obsahuje tah, který do cvičení nepatří

## Chyba labelování

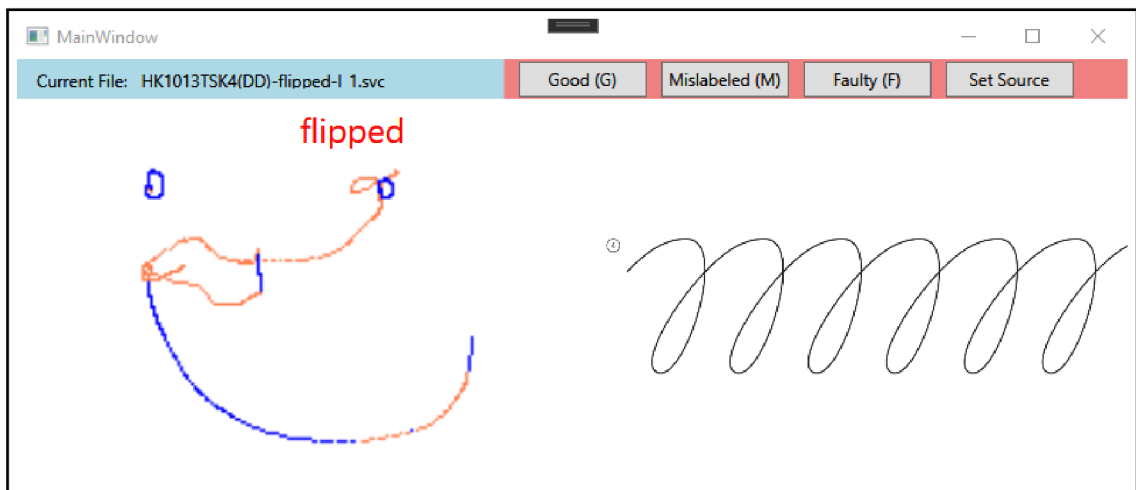
Jde o chyby, které se netýkají obsahu souborů, ale labelu, který jim byl přiřazen. Nejčastěji docházelo k záměně dvou podobných cvičení. Náprava této chyby je většinou velmi rychlá.

### 1.3.2 Chyby nevratné

Takto zařazené chyby byly odhaleny u celkem 122 dětí a 345 souborů. Pro jejich vytřížení je v programu použito tlačítko **Faulty (F)**.

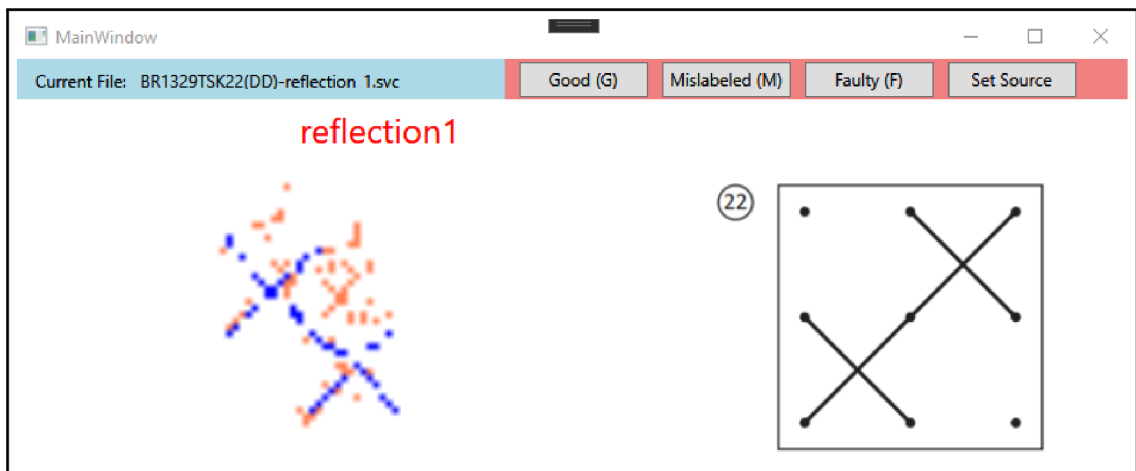
## Chyby záznamové techniky

U některých souborů se bohužel projevil chyby zařízení, případně softwaru na kterém byla cvičení zaznamenána. V naprosté většině případů šlo o situace, kdy se chod programu pravděpodobně zpomalil natolik, že došlo ke snížení vzorkovací frekvence, případně nebyly zaznamenány všechny body. Takto zaznamenané cvičení je však pro analýzu nepoužitelné, neboť chyba tohoto druhu by mohla změnit některé z parametrů písma, např. rychlost, délku tahu atd. Do dat by tak byly zaneseny nepřesnosti, které by mohly ovlivnit výsledky. Chybu navíc nelze dostatečně napravit.



Obr. 1.5: Soubor má obsahovat cvičení *Flipped*, ale pravděpodobně obsahuje jedno z volných cvičení.

Bylo by možné použít některé interpolační, či rekonstrukční metody z oblasti zpracování signálu. Takto dopočítanými informacemi však nelze nahradit originální data a soubory je proto nutné vyřadit.

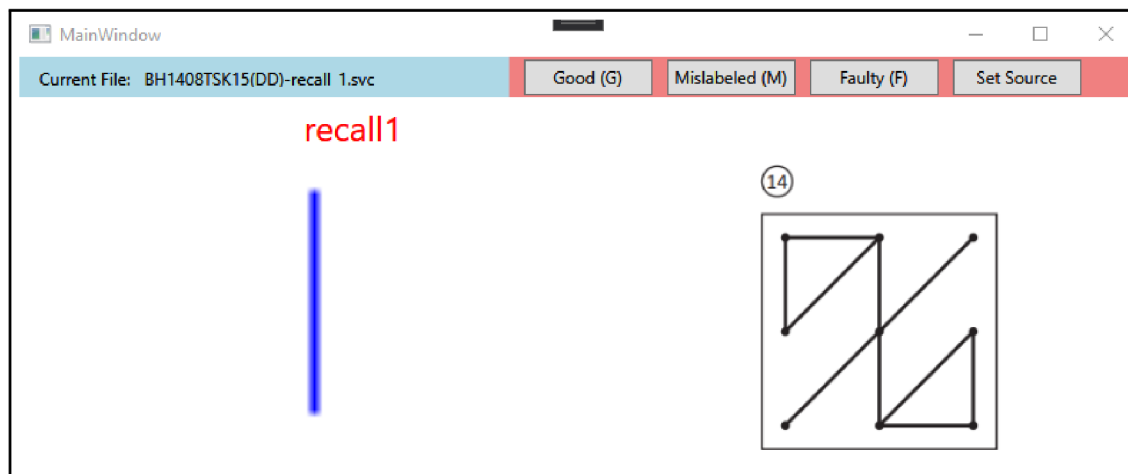


Obr. 1.6: Body zaznamenaného cvičení jsou od sebe výrazně vzdáleny, i když s jedná o jeden tah.

### Chyby kreslení

Některé soubory musely být vyřazeny z důvodu nejasnosti jejich obsahu. I když jejich data z technického pohledu nevykazují žádnou vadu, jejich obsah může být nepoužitelný. Především do této kategorie spadají soubory obsahující kresbu, ze které není

jasné zdali se vůbec jedná o správné cvičení a zároveň se však nepodobá žádnému jinému cvičení z protokolu. Tyto chyby nemusí být vždy naprosto nenávratné, neboť existuje šance, že došlo k chybě při segmentaci, či při labelování. Soubor však musí být vyřazen a chyba může být vyřešena jen případnou opakovanou segmentací zdrojového souboru.



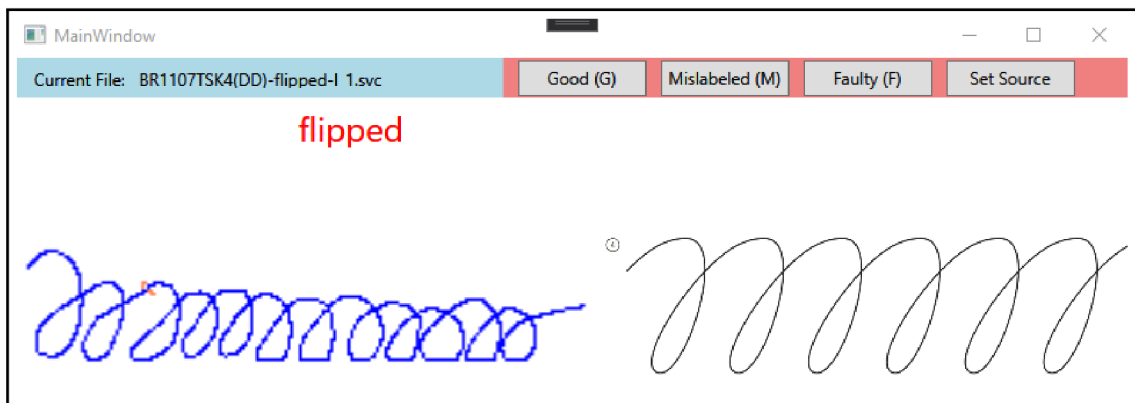
Obr. 1.7: U tohoto cvičení není jasné, zdali došlo k chybě při segmentaci, nebo jestli dítě vůbec kreslilo odpovídající cvičení

### Ostatní chyby

Chyby které nespádají ani do jedné z výše zmíněných kategorií byly poměrně vzácné, ale lze jako příklad uvést třeba chyby vzniklé nevhodným zacházením s vybavením, ne však chybou samotného zařízení, viz obr. 1.8.

## 1.4 Složení databáze

Po odstranění souborů obsahujících některou z chyb zůstal počet subjektů stejný, tedy 340, ale množství jednotlivých cvičení kleslo na 8 685 (průměrně 25,5 na dítě), bylo tedy odstraněno 371 souborů. Obsahem databáze jsou však bohužel i prvky, které nelze pro účely této práce použít z jiných důvodů, než je narušení dat samotných. V rámci metadat o testovaných subjektech jsou informace o jejich věku, třídě, pohlaví, diagnózách dysgrafie a dalších poruch, které by mohly ovlivnit jejich grafomotorickou zdatnost a skóre HPSQ-C. Především údaje o třídě, HPSQ-C a diagnostikované dysgrafii jsou pro účely zkoumání nových parametrů důležité. Bylo tedy nutné z databáze vyřadit všechny subjekty o kterých jsou tyto údaje neznámé.



Obr. 1.8: Cvičení, které bylo pravděpodobně kresleno na samém kraji tabletu a ve spodní části došlo k zachycení stylusu. Tímto může dojít k ovlivnění všech tabletem zachycených parametrů.

Nejvíce takto zasaženou skupinou byly předškolní děti, u kterých nebylo HPSQ-C vyhodnoceno v žádném případě.

Tab. 1.1: Tabulka popisuje složení databáze dětí, které vyplnily kreslicí protokol. V tabulce jsou uvedeny pouze děti, u kterých existuje záznam o třídě, diagnostikované dysgrafii, nebo skóre HPSQ-C. Jako třída "0." jsou označeni žáci prvního pololetí první třídy, kteří ještě neumí psát. Pro početní informaci jsou uvedeny i děti posledního roku před nástupem do školy (školka), i když jejich data nejsou použita jako předloha pro návrh parametrů.

Složení databáze						
Třída	n	chlapci	dívky	Dysgrafici	HPSQ-C avg	HPSQ-C std
školka	91	50	41	0	-	-
0.	17	10	7	0	12,47	5,47
1.	32	16	16	0	11,31	4,40
2.	41	20	21	1	13,60	4,55
3.	31	16	15	4	13,29	5,56
4.	82	55	27	5	13,76	6,04
celkem	203	117	86	5	13,17	5,43

Po finální úpravě databáze bylo tedy ponecháno 203 subjektů, ke kterým patří celkem 5821 validovaných souborů cvičení, tedy 28,67 na jedno dítě.





## 2 Sběr experimentálních dat

Z předešlé kapitoly a tabulky 1.1 je naprosto zřejmé, že se mezi dětmi nachází poměrně malé množství diagnostikovaných dysgrafiků, přičemž u některých z ostatních dětí nebyla diagnostika provedena. Není tedy možné je s jistotou označit za děti bez dysgrafie. Z tohoto důvodu může být použití dysgrafie jako validačního kritéria pro posouzení nových parametrů, nespolehlivé.

Tento nedostatek by mohl být alespoň z části napraven použitím nové subjektivní škály hodnotící grafomotorickou zdatnost dětí, která byla vyvinuta ve spolupráci se speciálními pedagogy. Jedná se o Likertovu škálu v rozsahu pěti celočíselných hodnot [0; 4], kde hodnotou 0 je označeno cvičení, které nevykazuje známky toho, že by jeho autor trpěl grafomotorickými obtížemi a hodnotou 4 naopak cvičení, které vykazuje nejvyšší míru grafomotorických obtíží při kreslení, resp. psaní. V rámci práce bude toto hodnocení označeno jako SHP - Subjektivní hodnocení speciálním pedagogem. Zároveň je potřeba ohodnotit ne jen jednotlivá cvičení, ale všechna cvičení dohromady. Pro sběr těchto dat bylo potřeba vyvinout systém, který by umožnil zvolené skupině psychologů procházet existující bankou kreslených cvičení a přiřazovat jim tyto hodnoty.

Za tímto účelem měla být vytvořena aplikace, s následujícími prvotními požadavky:

1. Aplikace má být naprosto nezávislá na ostatních systémech v počítači a internetovém připojení, tedy tzv. standalone a offline.
2. Nasazení a následné použití aplikace musí být rychlé a jednoduché s co nejmenšími požadavky na expertízu uživatele, tedy tzv. plug-and-play.
3. Data, které aplikace vyprodukuje musí být přístupná a zpracovatelná bez použití specializovaného software.
4. Vzhledem k velkému datasetu musí aplikace umožňovat rychlé a plynulé procházení bankou cvičení.

### 2.1 Vývoj aplikace

V této sekci bude popsán vývoj aplikace dle zadání a její výsledná podoba a funkcionality.

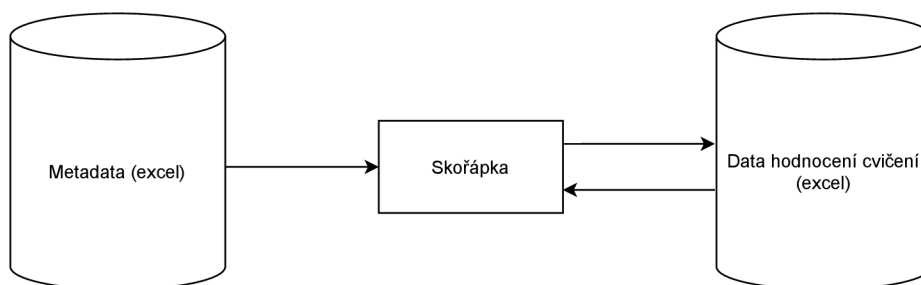
Aplikace byla vyvíjena téměř v provozu se zpětnou vazbou potenciálních uživatelů. V průběhu tohoto procesu tedy vznikaly nové požadavky na funkce a featury, kterými bude program disponovat. Výsledkem je tedy poměrně robustní aplikace *Skořápka - ScoreApp*, disponující pokročilými funkcemi navigace mezi soubory, náhledy a funkcí zoom.

## 2.1.1 Data management

Vzhledem k prvnímu požadavku, nebyla použita žádná standardní metoda pro ukládání dat. Ideální metodou by samozřejmě byla možnost použití některého typu SQL databáze, například PostgreSQL, nebo MS SQL. Toto by s sebou ale přineslo nutnost SQL serveru. V případě hostování online je nutné zajistit přístup uživatele k internetu a bezpečné připojení k serveru, na kterém SQL instance běží. Toto řešení navíc přináší i bezpečnostní rizika úniku dat, kterým je dobré se vzhledem k citlivosti údajů vyhnout. Možnost hostování SQL serveru lokálně se nabízí jako alternativa, avšak s sebou naopak nese nedostatky v podobě nároků na uživatelské znalosti v oblasti IT. Nastavení lokální instance SQL je poměrně složité a časově náročné, aplikace by tak nespĺňovala požadavek na jednoduchou instalaci a užívání. Ani export dat z relačních tabulek navíc není velmi přímočarý a je k němu zapotřebí buď specializovaného softwaru, nebo implementace skriptu.

Vzhledem k tomu, že finálním a cílovým formátem dat by dříve nebo později byl formát tabulkového programu MS Excel (tedy soubory .xls a jeho varianty jako.xlsx atp.), byla i samotná aplikace navržena tak, uměla číst a ukládat data prostřednictvím těchto souborů. Tohoto prvku aplikace využívá ve dvou případech:

1. Čtení metadat, jako je věk dítěte, třída atd., z databázové tabulky.
2. Čtení a ukládání hodnocení do uživatelem zvolené tabulky .xlsx.



Obr. 2.1: Proud dat mezi aplikací a soubory na disku počítače uchovávající metadata a nové informace.

Použití tabulek místo dedikovaného databázového systému má však svoje úskalí. Pod povrchem formátu.xlsx se ukrývá několik souborů ve formátu XML - Extensible Markup Language. Ve své podstatě se jedná o textové soubory dodržující striktně daný formát. Přímá práce s takovými soubory je obtížná, je tedy nutné je před použitím deserializovat. Tím dojde k přečtení veškerých dat a případně k uložení do

paměti programu. Z tohoto plynou určité nedostatky, především nemožnost vyhledat v záznamu konkrétní políčko bez přečtení celého souboru. Stejně tak je při ukládání obtížné a riskantní (z pohledu bezpečnosti ovlivnění ostatních dat) zapisovat jednu konkrétní hodnotu do konkrétního políčka. Je tedy jednodušší přečíst celý soubor, provést změny a původní soubor následně nahradit novým, obsahujícím změny. Je jasné, že tímto vzniká poměrně velký overhead při manipulaci se souborovým systémem. Data, která těmito cestami proudí jsou v poměru k dnešnímu běžnému výpočetnímu výkonu malá, je však třeba podotknout, že tento přístup není vhodný pro příliš velké množství informací. Již zmíněné SQL databáze jsou optimalizovány pro rychlé vyhledávání a čtení a zápis konkrétních polí v rámci obsažených tabulek. Pro přístup k excelovým souborům aplikace využívá knihoven `ExcelDataReader` [1] a `EPPlus` [9].

Druhým velkým nedostatkem tohoto řešení absence konvenčních datových typů „databáze“. I když MS Excel jako takový umožňuje jednotlivá pole označit za konkrétní datové typy, jako datum, desetinné číslo apod., tyto typy nemusí odpovídat formátům, se kterými běžně operuje programovací jazyk. Políčka jsou tedy většinou čtena jako textové pole `string` a následně parsována do požadovaného typu v rámci aplikace. Toto přináší další místo možného selhání aplikace a výkonový overhead.

## 2.1.2 Dataflow

Excelová data jsou v rámci aplikace držena ve dvou zásobnících, implementovaných pomocí kolekcí typu `List<MetaData>` a `List<SubjectScore>`, kde obě třídy obsahují nezbytná pole pro uchování dat a parametr potřebný k identifikaci subjektu, ke kterému náleží. V rámci celé aplikace je k tomuto účelu použit identifikační kód subjektu, většinou ve formátu `[XX][N]`, kde `X` je alfabetický znak a `N` několikacíferné číslo. Zde je možné poprvé poukázat na nedostatek aplikace plynoucí z výrazného iterativního vývoje aplikace. Původní zadání bylo velmi minimalistické a aplikace měla být velmi jednoduchá, bez složitějších funkcí. Právě proto kód disponuje určitými nedostatky, které se při chodu malé aplikace neprojeví, avšak jak aplikace roste, prodlevy v chodu aplikace mohou být patrné. V tomto případě jde o nevhodně zvolené kolekce `List<>`, které by při refaktoraci měly být nahrazeny vhodnějším typem `Dictionary`.

Zatímco zásobník pro `MetaData` je nutno pro chod aplikace naplnit, zásobník pro `SubjectScore` nikoliv. Tento případ nastává, pokud je program spuštěn nově a žádná cvičení ještě nebyla hodnocena, nebo pokud uživatel nezvolí žádný excelový soubor pro načtení dat. Aplikace tak slouží čistě jako prohlížeč cvičení, bez možnosti hodnocení.

Posledním důležitým zásobníkem je `List<ExerciseFile>`, ve kterém jsou drženy

údaje o souborech obsahující konkrétní cvičení. V určitém smyslu je tento zásobník centrálním prvkem aplikace. Zásobník neobsahuje cvičení jako taková, tedy údaje o samotných tazích a podobně. Pouze základní údaje o subjektu, jako věk, pohlaví a identifikační kód a umístění příslušného souboru cvičení v rámci souborového systému.

Data by samozřejmě mohla být ukládána v jednom zásobníku, který by obsahoval třídu zapouzdřující všechny informace. Z určitého hlediska je však práce se třemi zásobníky jednodušší. V případě této aplikace je důvodem převážně to, že každý zásobník představuje jinou množinu dat a ty nemusí být stejně velké, ale zároveň nelze ani zaručit, že některá množina bude obsahovat všechny odpovídající informace k prvkům z druhé množiny. Může se tedy stát, že pro některé cvičení nebudou existovat `MetaData`, nebo naopak. Stejně tak, mohou být některá cvičení, nebo `MetaData` vymazána až potom co byla ohodnocena, přičemž již existující hodnocení je potřeba zachovat. Pokud bychom tedy kterýkoliv dataset označili jako množinu **A** a **B**, následující relace 2.1 a 2.1 platit mohou a v praktickém provozu tomu tak většinou je, ale nelze je zaručit.

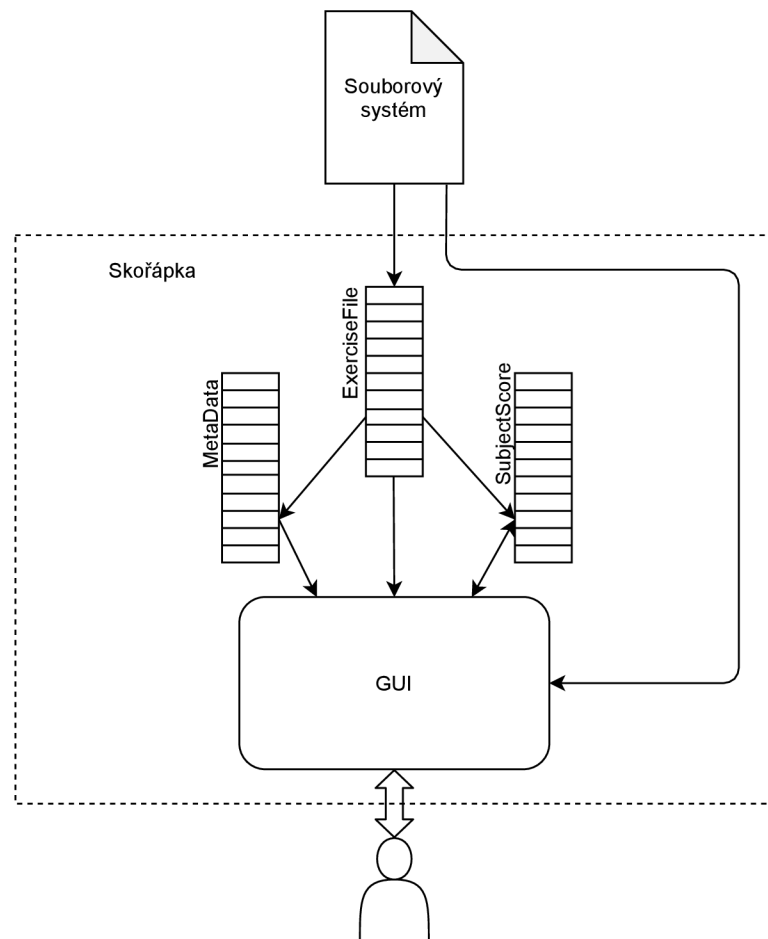
$$A \subseteq B \tag{2.1}$$

$$A \cap B \tag{2.2}$$

Zásobník `List<ExerciseFile>` je označován za centrální prvek, neboť slouží jako primární navigační prvek a právě na něj ukazují ukazatele, traverzující zásobník podle pokynů uživatele a zároveň určuje kam se mají zapisovat data hodnocení. Pro snadnou navigaci uživatele jsou cvičení seřazena podle třídy do které subjekt dochází od nejnižší po nejvyšší. Věk je při hodnocení relevantní informací a zároveň je pro hodnotitele jednodušší, když jsou děti stejné věkové kategorie v řadě za sebou.

Tento postup tedy zaručuje, že aplikace bude funkční, i při chybějícím záznamu `MetaDat` a uživatel bude moci provést hodnocení, pokud má aplikace přístup k souboru samotného cvičení. Uživateli však budou chybět informace o věku a pohlaví subjektu. Zároveň je možné všechny množiny zvětšovat i zmenšovat bez obav ze ztráty vytvořených dat. Soubory cvičení i excelové soubory s `MetaDaty` a hodnoceními by měly být bezpečně modifikovatelné i za běhu aplikace, je však nutno podotknout, že to není doporučený postup a aplikaci je vždy lepší vypnout před zásahem do dat, ke kterým přistupuje.

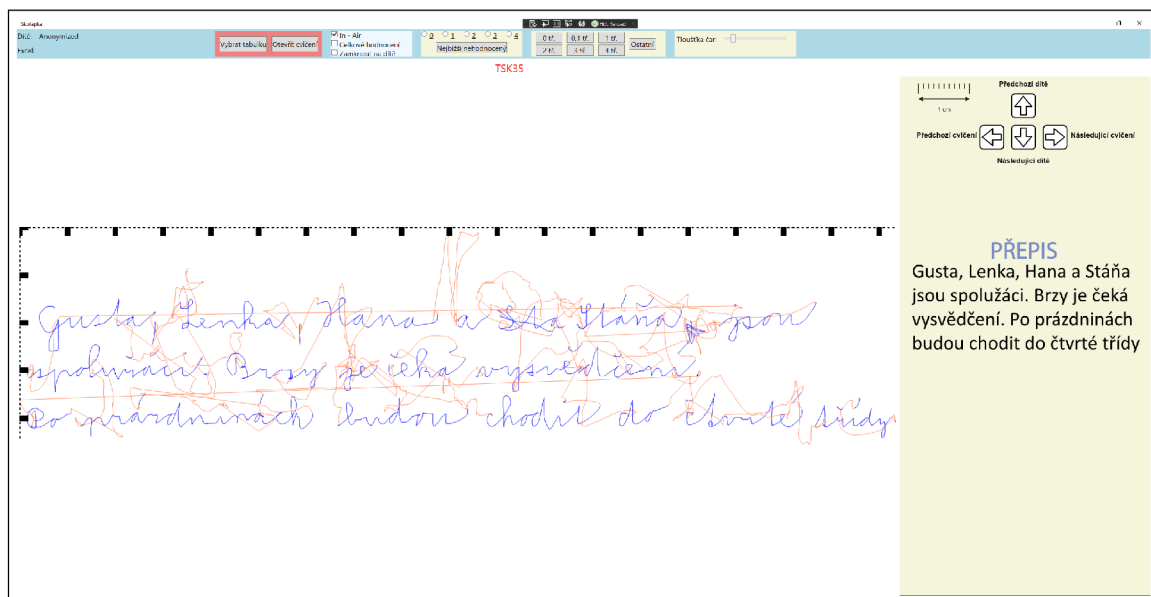
Pro navigaci ve cvičeních slouží ukazatel, který se pomocí inkrementace, nebo dekrementace pohybuje po zásobníku `List<ExerciseFile>`. Díky modulu funkci lze daty procházet v cyklech. Uživatel ukazatel ovládá pomocí šipek na klávesnici, kde šipky do stran jednoduše inkrementují a dekrementují, přičemž šipky nahoru a dolů přechází mezi subjekty. Uživatel má také možnost skočit na určitou věkovou kategorii pomocí tlačítka v GUI. Všechny tyto funkcionality jsou implementovány pomocí jednoduchých inkrementačních metod `moveRight`, resp. `moveLeft`. V případě větších skoků jsou tyto metody prostě volány dokud není splněna potřebná podmínka, například tedy, že ukazatel ukazuje na cvičení jiného subjektu. Stejným způsobem je implementována i možnost skočit na nejbližší nehodnocený soubor. Program zároveň disponuje jednoduchým textovým souborem, pomocí něhož lze nastavit, které cvičení mají být uživateli zobrazena. Pokud je tedy potřeba ohodnotit pouze podмноžinu cvičení, uživatel nemusí nežádaná cvičení přeskakovat.



Obr. 2.2: Obecné schéma celé aplikace Skořápka. Aplikace čte data ze souborového systému do tří nezávislých zásobníků a zobrazuje je v GUI. Uživatel s GUI interaguje pomocí myši a klávesnice.

Po dokončení operace navigace následuje otevření cvičení, jehož umístění je uloženo v právě vybrané instanci **ExerciseFile** a jeho obsah je parsován do celých tahů pro správné zobrazení. V případě, že je zvolena možnost přiřazení celkového hodnocení za všechny cvičení daného subjektu, jsou otevřeny všechny odpovídající soubory. Takto parsované tahy jsou uloženy do paměti programu pro zobrazení v GUI. Některé další funkce a proces zobrazení budou vysvětleny v následující kapitole, GUI.

### 2.1.3 GUI



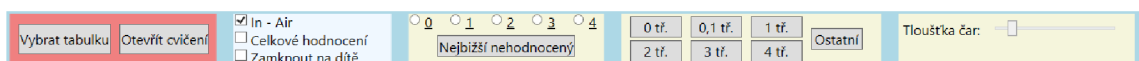
Obr. 2.3: GUI aplikace Skořápka. V horní části ovládací panel aplikace. V pravé části legenda ovládaní a předloha cvičení. V bílé ploše uprostřed je zobrazeno právě hodnocené cvičení.

Pro hodnocení kreslených cvičení je důležitý čistý a přehledný vzhled aplikace, přičemž největší prostor by měl bezesporu patřit právě zobrazenému cvičení. Aplikace byla tedy navržena tak, aby ovládací prvky zabíraly pouze minimum obrazovky a jsou umístěny v horním ovládacím panelu, spolu s informacemi o právě hodnoceném subjektu a právě otevřených souborech.

V pravé části se nachází doplňkový panel, který neobsahuje žádné ovládací prvky, ale pouze předlohu právě hodnoceného cvičení a jednoduchou legendu ovládaní. Zbytek plochy je věnován čistě zobrazenému cvičení. Pro zobrazování předloh je použita knihovna **Svg2Xaml** [16].

Jak je patrné z původních požadavků na aplikaci, její správný chod měl být bezproblémový na široké škále zařízení. Toto přináší určitý požadavek adaptability programu na zobrazovací zařízení. V dnešní době je dominantním rozlišením monitoru 1920x1080 px, avšak zařízení s tímto rozlišením mají různou velikost a není to ani taková vzácnost narazit na zařízení s nižším, či vyšším rozlišením, nebo úplně jiným poměrem stran. Zatímco nejpůvodnější rozlišení 1920x1080, FHD, disponuje poměrem stran 16:9, velmi rozšířená je i varianta 16:10 a v případě starších zařízení i 4:3. GUI aplikace tedy bylo navrženo tak, aby bylo co nejpřizpůsobivější prostředí, ve kterém běží, bez zásahu uživatele. Většina prvků GUI počítá s tím, že uživatelé budou pracovat na různých zařízeních, nebo nebudou aplikaci používat v režimu fullscreen. Velikosti jednotlivých oken se tedy velikostí přizpůsobí tak, aby bylo dosaženo co největší přehlednosti aplikace ve všech podmínkách. Přepočítání velikosti jednotlivých komponent je vyvoláno změnou velikosti okna aplikace. Při změně velikosti tažením myši tato událost však vzniká několikrát za vteřinu, což vytváří vysoké nároky na výpočet obsahu komponent uvnitř. Přepočítání je tedy voláno s prodlevou 100 ms po dokončení resize operace. V CodeBehind GUI je implementován 100 ms časovač, který je resetován a spuštěn vždy, když dojde kterýmkoliv způsobem ke změně velikosti okna, i když je velikost měněna nepřerušeně delší dobu. Po konečném uplynutí časovače proběhne event, který aplikaci signalizuje aby přepočítala velikosti vnitřních oken.

## Ovládací panel



Obr. 2.4: Ovládací panel aplikace obsahující možnosti otevírání souborů, přepínání režimů zobrazení, hodnocení cvičení a navigace.

V červeném poli v levé části panelu se nachází tlačítka pro zahájení práce, respektive, před zahájením práce je potřeba vybrat pracovní tabulku pro ukládání hodnocení, případně načtení existujících hodnot. **Otevřít cvičení** otevře dialogové okno, pro navigaci ke konkrétnímu souboru, toto je možné použít i pro skok ke konkrétnímu cíli v průběhu práce, bez použití navigace pomocí šipek.

V bleděmodrém poli nalezneme přepínací **CheckBox** tlačítka, které ovlivní globální chod aplikace.

- **In - Air** přepíná, zdali aplikace vykresluje oranžové linky pohybu dětí nad povrchem tabletu.

- **Celkové hodnocení** přepne aplikaci do módu pro udělení celkové známky subjektu. Na pracovní ploše se zobrazí všechny povolené cvičení, kreslené právě zvoleným dítětem.
- **Zamknout na dítě** Zamkne navigaci na aktuálním dítěti, dokud není přidělena známka všem cvičením včetně známky celkové. Uživatel může nadále používat šipek a zbytek ovládacího panelu pro pohyb mezi jednotlivými cvičeními daného dítěte.

První žluté pole obsahuje `RadioButton` tlačítka pro přidělení známky. Po jejich stlačení dojde k uložení, případně změně hodnoty v paměti aplikace. Hodnota je zároveň okamžitě propsána do zvolené excelové tabulky. Aplikace tedy neobsahuje tlačítko pro uložení; ukládání probíhá automaticky při každé změně. Pokud si uživatel zobrazí cvičení, které už v minulosti ohodnotil, tlačítko bude automaticky vybráno. Pod známkovacími tlačítky se nachází tlačítko umožňující skok na nejbližší další (za předpokladu, že uživatel začíná hodnotit od nejmladších dětí a pohybuje se šipkami doprava) neohodnocené cvičení, pokud není nastaven zámek na neohodnocené dítě.

V dalším žlutém poli nalezneme čistě navigační tlačítka. Ta prochází seznam cvičení, dokud není nalezeno první cvičení zvolené věkové skupiny. Pod hodnotami 0. a 0,1. třídy se nachází předškolní děti a děti docházející do první poloviny první třídy. Jde tedy o děti, které ještě neumí psát.

V posledním žlutém poli nalezneme `Slider` ovládající tloušťku písma.

## Pracovní plocha

Smyslem pracovní plochy je jednoduše co nejpřehledněji zobrazit aktuálně zvolené cvičení, nebo skupinu cvičení. Při zobrazení jednotlivého cvičení je kolem cvičení zobrazeno měřítko, určující velikost cvičení. Malé a velké dílky mají velikosti jednoho milimetru, respektive centimetru. Dané cvičení je vždy zobrazeno v co největší velikosti. Vzhledem k tomu, že cvičení je v datech zaneseno jako body v souřadnicovém systému  $[X; Y]$ , mezi body jsou nejprve nalezeny body s nejnižší hodnotou na každé ose. Tyto hodnoty jsou odečteny od každého bodu, čímž dojde k efektivnímu posunutí celého cvičení k nulovým osám. Následně jsou cvičení normalizována na maximální možnou šířku. Obě operace lze popsat rovnicemi 2.3, 2.4 a 2.5. V některých případech se může stát, že výsledný tvar bude vyšší, než okno pro zobrazení. S tímto případem si už ale umí poradit samo prostředí WPF a obrázek zmenšit. Je ale potřeba cvičení zvětšit na maximum, neboť i když je svým způsobem možné data cvičení označit za vektorovou grafiku, prostředí WPF pracuje s výsledkem vykreslo-



vací operace jako s obrázkem a obyčejně zvětšený obrázek může vypadat rozmazaně.

$$R = \frac{W_W}{\max(X) - \min(X)} \quad (2.3)$$

Pro každý bod cvičení je pak spočítáno (znak  $\Leftarrow$  označuje operaci přiřazení):

$$point_X \Leftarrow (point_X - \min(X)) \cdot R \quad (2.4)$$

$$point_Y \Leftarrow (point_Y - \min(Y)) \cdot R \quad (2.5)$$

Kde  $R$  je poměr aktuální šířky cvičení ku šířce zobrazovací komponenty  $W_W$ ,  $X$  je množina všech souřadnic cvičení na ose  $X$ .

Při vykreslování více cvičení najednou je situace složitější. Cvičení se musí vykreslit do pole předem neznámé velikosti (myšleno z pohledu tvorby programu). Zároveň i počet cvičení může být různý. V tuto chvíli je maximální počet cvičení na dítě 36. Jednoduchou možností by tedy bylo rozřezat pole na 36 dílků a ty postupně plnit, avšak v běžném případě je zkoumáno zhruba 10 cvičení. Velká část obrazovky by tak zůstala nevyužita. Program je tedy napsán tak, aby počet a rozložení těchto dlaždic přizpůsobil množství aktuálně zobrazovaných cvičení a velikosti aplikace. Aktuálním množstvím cvičení je myšleno skutečné množství cvičení, která mají být zobrazena, ne množství cvičení, která jsou nastavena pro hodnocení. Program tedy vlastnosti polí přepočítává pro každé dítě zvlášť a rozložení změní i při změně velikosti okna. Vzhledem k tomu, že v logice WPF se nulový bod nachází vertikálně v opačné pozici, je každé cvičení nutno odzrcadlit podle horizontální osy. Tato operace je jednoduchá pro jedno cvičení a může být provedena již existující metodou `flipYAxis` těsně před vykreslením, avšak v případě více cvičení musí být převrácena každá dlaždice zvlášť ručně.

Pro výpočet rozložení polí je nejprve určen počet řádků, podle 2.6

$$RowCount = \lfloor W_H \cdot \sqrt{\frac{W_W \cdot W_H}{E}} + 1 \rfloor \quad (2.6)$$

Kde  $W_W$  a  $W_H$  představují šířku a výšku zobrazovacího pole v pixelech,  $E$  pak počet cvičení k zobrazení.

Pomocí 2.7 a 2.8 pak vertikální a horizontální rozměry jedné dlaždice.

$$TileHeight = \lfloor \frac{W_H}{RowCount} \rfloor \quad (2.7)$$

$$TileWidth = \lfloor \frac{W_W \cdot RowCount}{E} \rfloor \quad (2.8)$$

Nakonec je podle 2.9 určen počet sloupců.

$$ColCount = \lceil \frac{E}{RowCount} \rceil \quad (2.9)$$

Pro výpočet poměru úpravy velikosti však nyní nestačí použít vzorec 2.3, neboť každé cvičení se musí svým rozměrem vejít do své dlaždice. Pro některá cvičení je tak limitující výška, pro některá šířka. Zároveň musí být zachován poměr velikosti mezi cvičeními. Program tedy vypočítá poměr pro každé cvičení podle 2.3 a upravenou variantu pro vodorovnou osu, dosazením  $Y$  hodnot místo  $X$  a výšky okna, místo jeho šířky. Jako poměr  $R$  pro všechna cvičení je pak použita nejmenší z hodnot pro všechna cvičení.

Obdobou operací 2.4 a 2.5, bez násobení poměrem  $R$  jsou body cvičení posunuty k počátkům os. Následně jsou pro body všech cvičení spočítány pozice v rámci přidělené dlaždice podle vztahů 2.10 a 2.11.

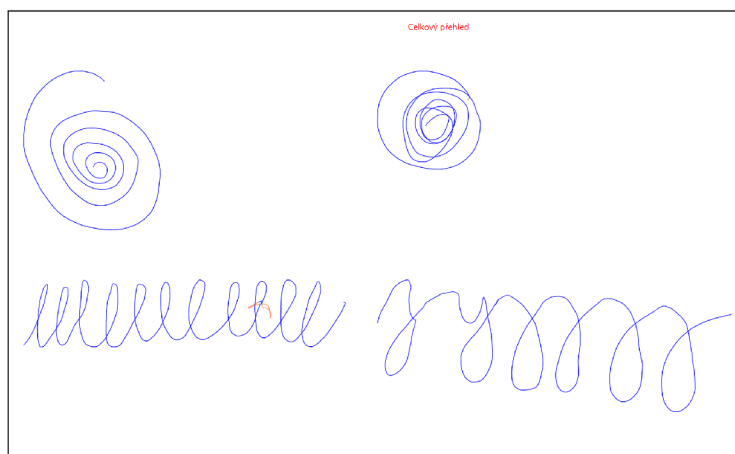
$$point_X \Leftarrow point_X \cdot R + c \cdot TileWidth \quad (2.10)$$

$$point_Y \Leftarrow (max(Y) - point_Y) \cdot R + r \cdot TileHeight \quad (2.11)$$

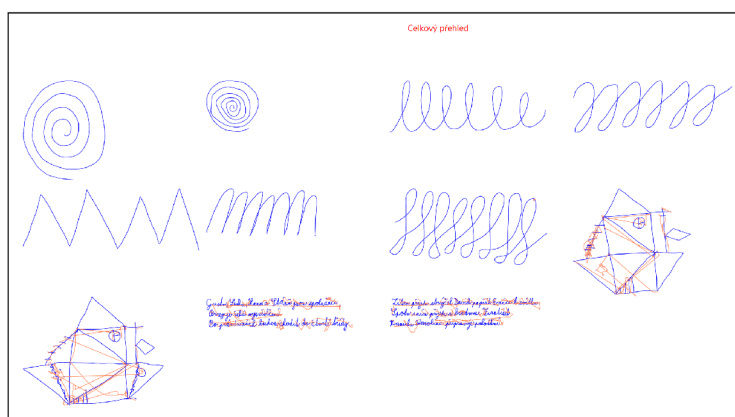
Kde  $c$  je pořadí sloupce do kterého cvičení patří v rozsahu  $[0; ColCount]$  a obdobně  $r$  je pořadí řádku cvičení v rozsahu  $[0; RowCount]$ .

Dalším krokem při vykreslení je spojení individuálních bodů cvičení spadajících do stejného tahu. Cvičení na obrázcích 2.5 a 2.8 jsou vykreslena už se spojnicemi mezi body. Právě kvůli tomuto kroku bylo nutné cvičení nejprve přizpůsobit na maximální možnou velikost, neboť interní .NET knihovna **Graphics** vykresluje tahy ve formátu bitmap a zvětšené bitmapy vypadají nevzhledně a kostičkovane. Knihovna **Graphics** také potřebuje jako vstupní parametr tloušťku čáry v pixelech, kterou bude cvičení vykresleno, což vnáší další prvek, který musí být přizpůsoben rozlišení celého okna, respektive displaye. Tohoto je docíleno pomocí následujících kroků. Nejprve je spočítán koeficient *LineWidthHelper*, podle vztahu 2.12 jehož záměrem je zmenšovat tloušťku čar na širokoúhlých obrazovkách. V případě vykreslování jednoho cvičení je *LineWidthHelper* ještě upraven podle vztahu 2.13.

$$LineWidthHelper \Leftarrow \left(\frac{W_H}{W_W}\right)^2 \quad (2.12)$$



Obr. 2.5: Zobrazení při nízkém počtu cvičení. Aplikace přizpůsobí rozložení a velikost cvičení aby došlo k co největšímu zobrazení jednotlivých cvičení.



Obr. 2.6: Zobrazení při vyšším počtu cvičení. Aplikace přizpůsobí rozložení a velikost cvičení aby došlo k co největšímu zobrazení jednotlivých cvičení.

$$LineWidthHelper \Leftarrow 1 - \left( \frac{LineWidthHelper}{2} \right) \quad (2.13)$$

Následně je pomocí vztahů 2.14 a 2.15 spočítána výsledná tloušťka čar.

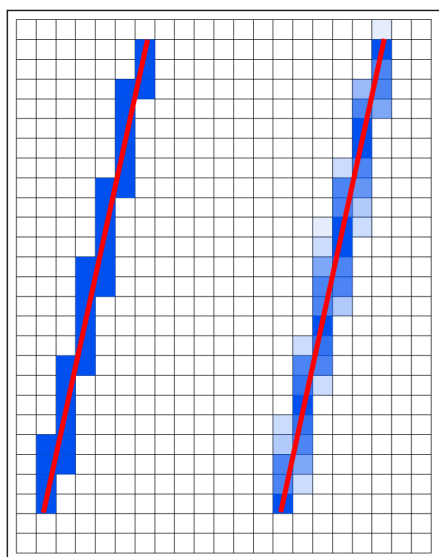
$$LineWidth \Leftarrow \left( \frac{LineWidthHelper \cdot W_W}{200} \right) < 5 \quad (2.14)$$

$$LineWidth \Leftarrow LineWidth \cdot \frac{\max(Y)}{\max(X)} \cdot WidthSlider \quad (2.15)$$

Kde  $X$  a  $Y$  jsou množiny souřadnic odpovídající osy cvičení a  $WidthSlider$  je hodnota nastavitelná táhlem v grafickém rozhraní, s hodnotami v rozsahu  $[0,1; 4]$ .

Protože vykreslování probíhá na formát bitmap, jednotlivým pixelům je zjednodušeně přiřazena hodnota buď 1, nebo 0. To se na výsledném obrázku může projevit kostičkovaným vzhledem, především u křivek a šikmých čar. Do obrázku je zanesen tzv. aliasing. Z hlediska zpracování signálu, k aliasingu dochází při záznamu, nebo reprodukci signálů, které obsahují spektrální složky nad tzv. Nyquistovým kmitočtem, který se rovná polovině vzorkovací frekvence. Tyto frekvence zanáší do přenášeného signálu zkreslení. Cílem antialiasingového filtru je tedy odstranění těchto spektrálních složek. Takovýto filtr tedy můžeme označit za filtr typu dolní propust. Stejný princip platí pro jednorozměrné signály, jako je třeba zvuk, tak pro vícerozměrné signály, jako je právě obraz.

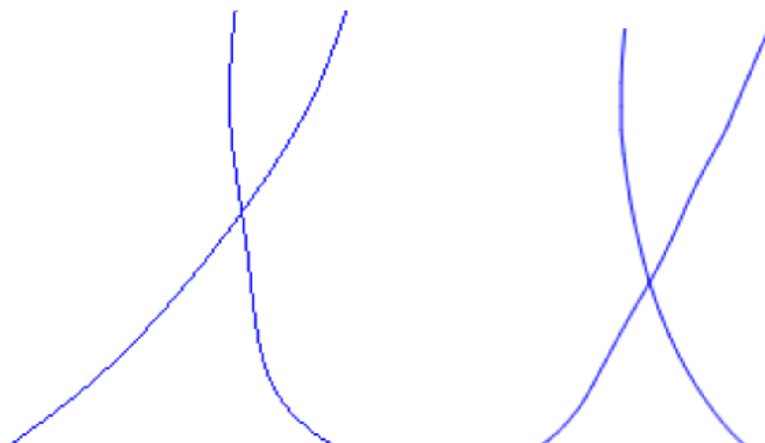
Pro potřeby této práce je ale možné jev aliasingu v obrazech možné zjednodušit. K aliasingu totiž v tomto případě dochází, pokud čára cvičení skrze pixely pod úhlem, který není kolmý na některou z os. Aktivovány jsou pak všechny pixely, kterými čára prochází. Jednoduchou metodou antialiasingu je tedy pouze částečná aktivace pixelu podle vzdálenosti jeho středu k nejbližšímu bodu úsečky cvičení.



Obr. 2.7: Vlevo úsečka s viditelnými hranami a aliasingem. Vpravo stejná úsečka s náznakem jednoduchého antialiasingu.

Nevýhodou některých antialiasingových metod může být, že výsledný obraz působí, vlivem aktivace většího množství pixelů, rozmazaně. V praxi je tedy důležité vybrat vhodný antialiasingový algoritmus. Tato volba je nejpodstatnější v oblasti náročných grafických aplikací, jako jsou například počítačové hry, kde se v jeden moment na obrazovce nachází velké množství, složitých objektů a tak je třeba hledat kompromis mezi výkonem a příjemným vzhledem. V případě této aplikace není

antialiasing jednoduchých tvarů značně náročný a volba antialiasingové metody tedy není tak podstatná.



Obr. 2.8: Vlevo vykreslení části cvičení bez vyhlazování hran (antialiasing), vpravo část cvičení s vyhlazováním hran

## Zoom

Při testovacím provozu aplikace vznikl také požadavek na možnost přiblížení cvičení. U cvičení jako je jednoduchá spirála není přiblížení pravděpodobně až tak podstatné, ale při hodnocení například diktátů, nebo Reyových figur může uživatel potřebovat zvětšit konkrétní část úlohy. Stejně tak při udělování celkové známky může uživatel chtít analyzovat jedno ze cvičení blíž, ale přepínat zpět do režimu jednotlivého zobrazení a listovat mezi cvičeními by bylo příliš zdlouhavé.

Technik pro implementaci přiblížení je několik. Nejsnazší je pravděpodobně přidat `slider` jako je tomu u velikosti písma a při nastavení hodnoty větší, než 1, obrázek poměrně přiblížit. U této metody je však problémem navigace. Obrázek se standardně přiblíží a okraje obrázku se tedy skryjí mimo pole. K pracovní ploše by tedy bylo nutné přidat `scrollbary`, kterými by uživatel ovládal zorné pole zoomu, toto řešení i když jednoduché, jde proti principům fungování aplikace, tedy co nejjednoduššího ovládání ze strany uživatele.

Obdobnou, implementačně stále poměrně jednoduchou variantou je zoomovat menší oblast nacházející se pod kurzorem myši a zobrazovat ji někde stranou, například v informačním panelu v pravé části aplikace, avšak to by znamenalo zvětšit informační panel, nebo zmenšit prvky, které se v něm už nacházejí. Celkově by tedy došlo ke ztrátě dostupného pracovního prostoru.

Nakonec zvolenou technikou přiblížení je zoom a zároveň zobrazení přiblížené oblasti místo kurzoru myši, pokud se myš zrovna nachází nad obrázkem cvičení.

Tento postup se nakonec ukázal být implementačně poměrně náročný. Problematiku lze rozdělit do dvou částí:

1. Detekce kurzoru myši nad obrázkem
2. Výpočet velikosti přibližovacího okna
3. Výpočet proporcí zvětšeného cvičení

Pro řešení prvního problému se nabízí zdánlivě jednoduché řešení. Stejně jako většina nástrojů pro tvorbu GUI, i WPF nabízí možnosti nastavit na svoje komponenty eventy, které volají vývojářem definované funkce. Takovým eventem může být například `mouseenterEvent`, který při zachycení události vjetí myši na polohu komponentu zavolá určenou funkci. Tohoto a opačného `mouseleaveEvent` lze využít pro otevření a zavření zoomovacího popup okénka. Následně stačí pomoci `mouseoverEventu` určit polohu myši a popup okénko přesunout pod ní. Toto ale vede k okamžitému zavolání `mouseleaveEventu` prohlížeče cvičení, neboť kurzor myši se nyní nachází na nově otevřeném zoom podokénku. Ve chvíli kdy okénko zmizí, proběhne `mouseenterEvent` a celý proces se zacyklí.

Pro řešení problému tedy bylo nutné obejít se bez jinak naprosto jednoduchého řešení použitím těchto eventů a implementovat vlastní řešení. Na celém okně je implementován `mousemoveEvent`, který při každé změně polohy myši vně okna aplikace reportuje polohu myši relevantně vůči svému nulovému bodu v levém horním rohu a na této pozici provede tzv. `HitTest`. `HitTest` je ve své podstatě paprsek, vystřelený směrem od uživatele skrze obrazovku, reportující komponenty aplikace, která jej zavolala. Zároveň nabízí možnost ignorovat komponenty podle typu.

---

```
HitTestFilterBehavior HitTestFilter(DependencyObject o)
{
    if (o.GetType() == typeof(Image))
    {
        return HitTestFilterBehavior.Continue;
    }
    else return HitTestFilterBehavior.ContinueSkipSelf;
}
```

---

Výpis 2.1: `HitTestFilterBehaviour`, který přeskočí vše, co není typu `Image`

Po filtru je seznam všech zasažených komponent filtrován ještě jednou, a pokud seznam neobsahuje konkrétní komponenty `imageView`, je volána funkce `closePopup()`. Pro předcházení problému s blikajícím popup okénkem je aplikace vybavena `bool` proměnnou `popupOpen`, jejíž hodnota uvádí, jestli je popup otevřený, nebo zavřený

a časovačem, který je spuštěn vždy, když dojde k zavření okénka. `popupOpen` je přepnuta do polohy `false` až po uplynutí 50 ms; do té doby se popup nemůže znovu otevřít. Při otvírání zoomovacího okénka je vypočítána i jeho velikost. Přibližovací popup je vždy čtvercový a jeho strana *PopupSize* se počítá podle jednoduchého vztahu 2.16.

$$PopupSize \Leftarrow \frac{\max(W_W, W_H)}{5} \quad (2.16)$$

Nyní je nutné určit obsah popup okénka. WPF nám umožňuje získat polohu kurzoru relativně k jakémukoliv komponentu. Pokud tedy `Mouse.GetPosition()` zavoláme s argumentem komponentu `ImageViewer`, získáme polohu relativní k tomuto komponentu, z čehož můžeme vypočítat polohu v obrázku. V případě, že je komponenta `ImageViewer` větší, než obrázek, který je do ní vložen jako zdroj (`source`), komponenta si jej mírně roztáhne. Toto ale znamená, že je nejprve nutné spočítat skutečnou polohu kurzoru vůči zdrojovému obrázku a ne jen vůči komponentu. WPF totiž neumožňuje použít jako obrazový zdroj pro popup část už vykresleného obrázku, ale musí být použit původní datový zdroj.

Je tedy nutné spočítat velikost a polohu čtverce, který "vyřízne" část původního cvičení. Toho dosáhneme vztahem 2.17.

$$XRatio \Leftarrow \frac{IV_W}{IV_{SW}} \quad (2.17)$$

---

```
1 var XRatio = sourceViewer.ActualWidth / sourceViewer.Source.Width;
```

---

Kde  $IV_W$  je šířka komponenty `ImageViewer` a  $IV_{SW}$  je šířka obrazového zdroje komponenty `ImageViewer`. Obdobným způsobem dosáhneme i hodnoty *YRatio*, pouze použitím výšky jednotlivých prvků.

Následně spočítáme souřadnice počátku, tedy levého horního rohu čtverce podle 2.18.

$$RectX \Leftarrow \max\left(0, \left\lfloor \left( MouseImPos_X - \frac{PopupSize}{ZoomScale \cdot 2} \right) \cdot \frac{1}{XRatio} \right\rfloor \right) \quad (2.18)$$

---

```
1 int RectX = Math.Max((int)((mouseImagePosition.X - (popupSize / (
    zoomScale * 2))) / XRatio), 0);
```

---

Kde  $MouseImPos_X$  je polohová souřadnice  $X$  relativně k počátku komponenty `ImageViewer` a  $ZoomScale$  je faktor přiblížení. Ten je pro jednotlivé cvičení nastaven na hodnotu 2, pro skupinu cvičení na hodnotu 3. Obdobně jako u rovnice 2.17, je potřeba vypočítat i hodnotu *RectY* dosazením odpovídajících hodnot  $Y$

---

```

1 //Vytvoření vykrajovacího čtverce
2 var rect = new Int32Rect(RectX, RectY, rectXSize, rectYSize);
3 //Ošetření přesahu čtverce
4 if (rect.X + rect.Width > popupImage.Source.Width)
5 {
6     rect.X = (int)popupImage.Source.Width - rect.Width;
7 }
8 //Vykrojení dlaždice pro zvětšení z původního obrazového zdroje
9 var crop = new CroppedBitmap(popupImage.Source, rect);

```

---

Výpis 2.2: Vytvoření vykrajovacího čtverce a následné vykrojení z původního obrázku

osy. Funkce `max()` je zde nutností, aby vykrajovací čtverec nezasáhl mimo rozsah souřadnic zdrojového obrázku.

Velikost strany vykrajovacího čtverce je poté určena podle 2.19.

$$RectXSize \Leftarrow \lfloor \frac{PopupSize}{ZoomScale \cdot XRatio} \rfloor \quad (2.19)$$

---

```

1 var rectXSize = (int)Math.Floor(ImageZoom_Popup.Width / (zoomScale
    * XRatio));

```

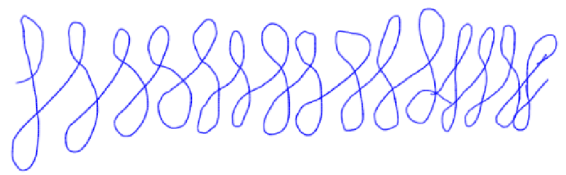
---

Stejně jako u 2.18 musíme ošetřit aby ani na druhé straně nedošlo k přesahu souřadnic čtverce. Případný přesah v každé z os, je odečten od počátkových souřadnic vykrajovacího čtverce.

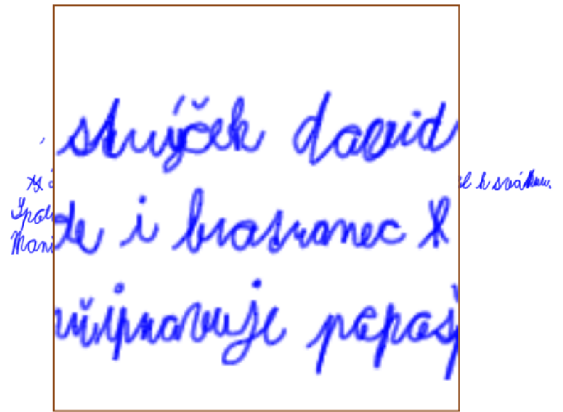
Nakonec je spočítána poloha popup okénka relativně k celému oknu. Vzhledem k tomu, že WPF umožňuje získat polohu myši, stačí od  $X$  a  $Y$  souřadnic myši odečíst polovinu hodnoty  $PopupSize$ , neboť popup okénko je vykreslováno od počátku, tedy horního levého rohu.

Celkově byla implementace přiblížení velmi náročná, protože bylo nutné implementovat vlastní řešení detekce myši pro tento specifický problém. Funkcionalita byla poměrně důkladně otestována avšak její chování stále obsahuje nějaké nedostatky. Především chování v případě, kdy aplikaci z části překrývá okno jiného programu. V běžném provozu je funkcionalita ale naprosto spolehlivá.





Justa, Lenka, Jana a Ivana jsou spoluzáci.  
Buzis je Čláčekův vyjádření.  
Do pánáčnicků, Ando<sup>o</sup> chodil do křesťanské školy.



Obr. 2.9: Ukázka vyskakovacího zoom okénka při zobrazení více cvičení najednou



## 3 Návrh parametrů a statistická analýza

Vzhledem k tomu, že některé dostupné cvičení jsou už do určité míry parametrizovány (např. z hlediska rychlosti, zrychlení, třesu atp.) a analyzovány [12], byly nové parametry zaměřeny na cvičení, které prozatím patří k těm neprozkoumanějším. Pro návrh byly vybrány cvičení 8-17 (Pro konzistenci s pojmenováním souborů protokolu budu používat zkratku anglického task - **TSK**, tedy **TSK8**, **TSK9**...). Jedná se o deset cvičení, která jsou tvořena pěti páry. V každém páru je vzorové cvičení, označené **Recognition**, které dítě kreslí do vymezeného prostoru se zachytnými tečkami, přičemž vidí předlohu cvičení. Poté je předloha zakryta a dítě kreslí stejné cvičení z paměti. Má však k dispozici stejný prostor se zachytnými body. Toto cvičení v páru je označeno jako **Recall**. Stejný princip platí i u **TSK27** a **TSK28**, kde se jedná o kreslení Reyovy figury, také jednou s předlohou a jednou z paměti.

Takto spárované cvičení umožňují porovnávat parametry písma u cvičení, která jsou v zásadě stejná, ale jsou vypracovány za odlišných podmínek. Tohoto principu využívá například Pagliarini et al. při zkoumání dětmi psaných písmen, které dostávaly tři různé pokyny ke psaní, spontánně, rychle a velkým písmem [14].

Zatímco u samostatných cvičení je možné extrahovat parametry, které jsou přítomny v daném cvičení, např. rychlost celku, či jednotlivých tahů. Páry vypracování téhož cvičení za různých podmínek umožňují spočítat rozdíl mezi parametry každého páru, případně přímo porovnávat samotná cvičení pro každé dítě a pokusit se tak najít souvislost parametru s jeho grafomotorickou zdatností [14].

### 3.0.1 Normalizovaný centroid

Centroidem se v geometrii nazývá bod, jehož souřadnice  $x$  a  $y$  jsou aritmetickým průměrem všech bodů dané figury. Dají se tedy vyjádřit rovnicemi 3.1 a 3.2.

$$C_x = \frac{x_1 + x_2 + x_3 + \dots + x_n}{N} \quad (3.1)$$

$$C_y = \frac{y_1 + y_2 + y_3 + \dots + y_n}{N} \quad (3.2)$$

Kde  $x_n$ , resp.  $y_n$  jsou polohové složky jednotlivých bodů a  $N$  je jejich celkové množství.

Pro zjednodušení práce s tímto parametrem je poloha centroidu normalizována do hodnot v rozsahu  $[0; 1]$ ; souřadnice  $[0,5; 0,5]$  tak označují prostředek cvičení. Tímto dojde ke kompenzaci vlivu velikosti daného cvičení na výslednou hodnotu parametru a bude tak možné jednodušeji odhalit například efekt, kdy dané dítě

kreslí všechna cvičení tak, že umisťuje více bodů do jejich pravé části, i když se jejich velikost bude lišit. Zároveň je tímto ošetřen případný problém s řády vstupních hodnot. Předchozí verze záznamového zařízení ukládala informace o souřadnicích  $x$  a  $y$  v jednotkách obrazových bodů, hodnoty tedy dosahovaly řádově desetitisíců. Aktuální verze zařízení ukládá polohové souřadnice v milimetrech, rozsahy hodnot pro jedno cvičení jsou tedy pouze desítky až stovky. Parametry získané na těchto dvou odlišených verzích by tedy nebylo možné přímo srovnávat. Normalizací podle rovnic 3.3 a 3.4 je tento problém odstraněn, navíc existuje předpoklad kompatibility i s budoucími verzemi zařízení za předpokladu, že informace o poloze budou mít lineární charakter a počátek souřadnicového systému bude stejně umístěn, resp. osy souřadného systému budou stejně orientovány.

$$C_{xr} = \frac{C_x}{\max(X)} \quad (3.3)$$

$$C_{yr} = \frac{C_y}{\max(Y)} \quad (3.4)$$

Parametr je možné využít dvěma způsoby. V první řadě je možné jej uvažovat samostatně a zkoumat, jestli jeho hodnota nějak koreluje s diagnózou dysgrafie, nebo s HPSQ-C skóre napříč celým datasetem. V druhé řadě je možné spočítat rozdíl hodnot pro pár cvičení Recognition-Recall a určit tak jak velký vliv má změna podmínek kreslení cvičení na dané dítě vyplňující dotazník. Pro účely této práce je jako parametr považována právě druhá varianta.

### 3.0.2 Normalizovaný korelační koeficient

V oblasti číslicového zpracování signálů se pro určení podobnosti dvou signálů používá operace křížové korelace definované vztahem 3.5. Takto definovaná operace však počítá pouze s jednorozměrným signálem. Pro práci s dvourozměrným signálem lze použít vzorec podobný, 3.6.

$$C(k) = \sum_{m=0}^{M-1} S1(m)S2(m+k) \quad (3.5)$$

Kde  $S1$  a  $S2$  jsou posloupnosti čísel.

$$C(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T1(m,n)T2(m+k,n+l) \quad (3.6)$$

$T1$  a  $T2$  jsou matice tvořené párem cvičení Recognition a Recall.  $C(k)$ , resp.  $C(k,l)$  jsou korelační koeficienty na pozici  $(k)$ , resp.  $(k,l)$ .

Abychom mohli pomocí vzorce určit podobnost obou cvičení, je nejprve nutné převést informace o tazích cvičení do matic. Záznamové rozlišení použitých digitizérů je ale natolik vysoké (5600 řádků na palec), že i když jsi jsou dvě cvičení velice podobné, hodnoty křížové korelace jsou velmi nízké právě proto, že dva tahy, které se lidskému oku mohou zdát na stejném místě jsou ve skutečnosti posunuté o jeden, či více pixelů. Je proto nutné snížit rozlišení obrázku. V úvahu připadají dvě operace.

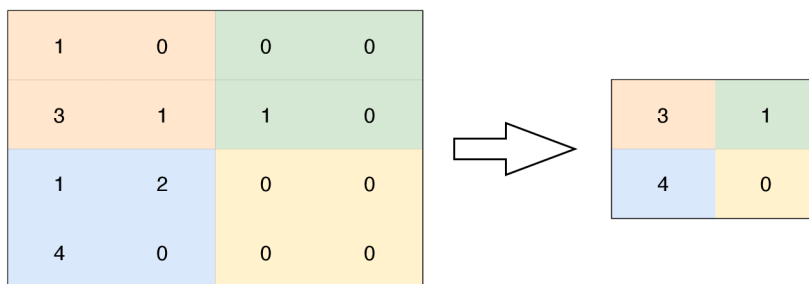
### **Podvzorkování**

Podvzorkování (z ang. downsampling) funguje na principu vypouštění konkrétních vzorků. Například pro zmenšení matice na čtvrtinu by bylo nutné vypustit každý druhý sloupec a každý druhý řádek. Existuje však riziko, že by tímto mohlo dojít k vypuštění velké části zachyceného cvičení. Právě díky vysokému rozlišení je v rámci matice každého cvičení jeden tah reprezentován jako tenký proužek jedniček v poměrně velkém poli nul. i když je poměrně nepravděpodobné, že by se ve cvičení nacházel tah, který je perfektně kolmý na jednu z os, alespoň z části tomu tak být může a právě takovýto tah by mohl být vynechán. Došlo by tak ke ztrátě velkého množství relevantních dat.

### **Pooling**

Pooling je v dnešní době často používanou operací v rámci konvolučních neuronových sítí, kde je používán právě pro redukci rozměrů matic mezi vrstvami sítě. Pooling funguje na principu rozdělení matice do malých, samostatných podmatic, např. o rozměru 2x2. Na každou podmatici je poté aplikována zvolená operace a její výsledek je uložen jako nová hodnota na odpovídající pozici. Při poolingů 2x2 s krokem 2 tak rovněž dojde ke snížení obrazových bodů na čtvrtinu. Velice často používaným druhem poolingů je tzv. max pooling, při kterém je jako výsledek každé podmatice zvolena její nejvyšší hodnota, viz obr. 3.1. Zásadním relevantním rozdílem oproti podvzorkování je, že nemůže dojít k vypuštění osamocené nenulové hodnoty v poli nul. Pro snížení rozlišení cvičení byl právě proto zvolen Max Pooling.

Implementace Max Poolingu v rámci tvoření matice je velice jednoduchá právě díky formátu vstupních dat. Polohová informace každého bodu je uložena v milimetrech. Při rozlišení digitizéru 5600 řádků na palec, tedy přibližně 220 řádků na milimetr, dosahují polohové hodnoty zkoumaných cvičení zhruba hodnot v rozsahu desítek milimetrů, avšak s desetinnými hodnotami. Jednoduché implementace Max Poolingu dosáhneme už pouhým zaokrouhlením každé polohové souřadnice na nejbližší celé číslo. Pro každé cvičení je tak vytvořena matice o rozměrech  $\text{round}(\max(x))+1 \times \text{round}(\max(y))+1$ . Pro cvičení jehož maximální prostorové souřadnice jsou [42,365; 47,845] mm, bude tedy vytvořena matice o rozměrech 43 × 49.



Obr. 3.1: Max Pooling

Je třeba podotknout, že takto implementovaný Max Pooling lze považovat za ne-standardní variantu, neboť první a poslední podmatice nebudou mít stejnou velikost jako zbytek, neboť k nule je zaokrouhlováno pouze zprava. Stejně tak i k nejvyšší hodnotě může být zaokrouhlován menší počet hodnot.

Použitím vzorce 3.6 vypočítáme koeficienty křížové korelace, z nichž zvolíme maximální hodnotu. Ta nám udává míru podobnosti obou cvičení. Vzhledem k tomu, že se jedná o binární matici, hodnota koeficientu nám vlastně udává, kolik bodů s hodnotou 1 se při každé iteraci protnulo. Tato hodnota je však stejně jako u předchozího parametru ovlivněna celkovým počtem nenulových bodů cvičení. Kresby, které jsou větší, nebo jednoduše plnější tak mohou dosahovat větších hodnot, což zamezuje jednoduchému porovnávání. Pro řešení tohoto problému se dá použít vzorec 3.7 pro normalizování křížové korelace. Ten však předpokládá, že obě matice mají stejný rozměr, což v tomto případě není možné zaručit.

$$C_N(k, l) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T1(m, n)T2(m + k, n + l)}{\sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T1(m, n)^2} \sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T2(m + k, n + l)^2}} \quad (3.7)$$

$T1$  a  $T2$  jsou matice tvořené párem cvičení Recognition a Recall.

Účel rovnice 3.7 je normalizace do hodnot v rozsahu  $[0; 1]$ . i když pro případ matic různých velikostí jej nelze použít, je možné jej pro tyto účely zjednodušit. Je totiž jasné, že aby byl výsledek pro dvě binární matice normalizován do stejného rozsahu  $[0; 1]$ , je nutné výsledek rovnice 3.6 podělit nejvyšší teoreticky dosažitelnou hodnotou, tedy počtem nenulových právě té matice, která jich obsahuje méně. Vzhledem k tomu, že cílem tohoto parametru je sledovat podobnost matice vzniklé ze cvičení Recall, k matici cvičení Recognition, bude hodnota koeficientu korelace vždy dělena počtem nenulových bodů matice Recognition. Vzniká tak vzorec 3.8.

Jako parametr je pak použita maximální hodnota vzniklé korelační matice.

$$C_{NB}(k, l) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T1(m, n)T2(m + k, n + l)}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T1(m, n)} \quad (3.8)$$

$T1$  a  $T2$  jsou matice tvořené párem cvičení Recognition a Recall, přičemž je nutné, aby za  $T1$  byla vždy dosazována matice **TSK** odpovídající cvičení Recognition.

Při předběžné analýze se tento parametr vykázal poměrně slibné výsledky, proto byly vyvinuty ještě další varianty parametru založeném na korelační matice. Zatímco původní varianta používá maximální hodnotu normalizované korelační matice, další varianty využívají příbuzné hodnoty. Pro praktické využití pravděpodobně nebudou vhodné všechny varianty parametru, neboť je vysoce pravděpodobné, že vzhledem k velmi podobným metodám jejich extrakce, budou mezi sebou parametry vzájemně silně korelovat. Je však možné, že se některá z variant ukáže být výrazně lepší než jiná. Případně se může ukázat, že každá z nich bude spolehlivější pro jiné cvičení, nebo jinou věkovou skupinu.

## Percentil

Tato varianta je ve své podstatě nejpodobnější té původní. Na rozdíl od maximální hodnoty, je použita hodnota 95. percentilu. Percentil se v tomto tvaru používá většinou jako prevence vlivu jakýchsi krajních, extrémních hodnot, které jsou silně odlišné od zbytku. Výsledná hodnota tedy může v některých případech lépe odrážet skutečnou míru shody obou cvičení. Vzhledem k tomu, že korelační matice je velmi řídká a obsahuje velké množství nulových prvků, je percentil počítán pouze z těch nenulových.

## Průměrná hodnota

Parametr je počítán jako aritmetický průměr nenulových hodnot výsledné korelační matice. Záměr tohoto parametru je podobný jako u percentilu, tedy získat další metriku shody cvičení. Vzhledem k vlastnostem porovnávaných cvičení, je totiž možné, že spolu budou korelovat po částech, za předpokladu, že například horní půlka kresleného cvičení bude dítětem zanesena s lehkým posunem do jedné strany. Maximální hodnota korelace tak nedosáhne tak vysokého maxima, ale matice bude obsahovat dvě, řádově stejně vysoké hodnoty. Průměr korelační matice, by toto mohl odhalit.

## Směrodatná odchylka

Směrodatná odchylka je hodnota, která je většinou uváděna a počítána spolu s průměrem. Ve své podstatě jde o hodnotu, která udává jak se vzájemně podobají jednotlivé hodnoty množiny. Pokud je směrodatná odchylka menší, hodnoty vybraného vzorku jsou si bližší a naopak. V praxi, kdy neznáme rozdělení množiny dat, je odchylka počítána pouze ze vzorku populace. Počítáme tak tzv. výběrovou směrodatnou odchylku  $s$  podle vzorce 3.9 pomocí aritmetického průměru.

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (3.9)$$

Kde  $N$  je počet prvků množiny,  $x_i$  představuje naměřené hodnoty a  $\bar{x}$  jejich aritmetický průměr.

### 3.0.3 Isochornie

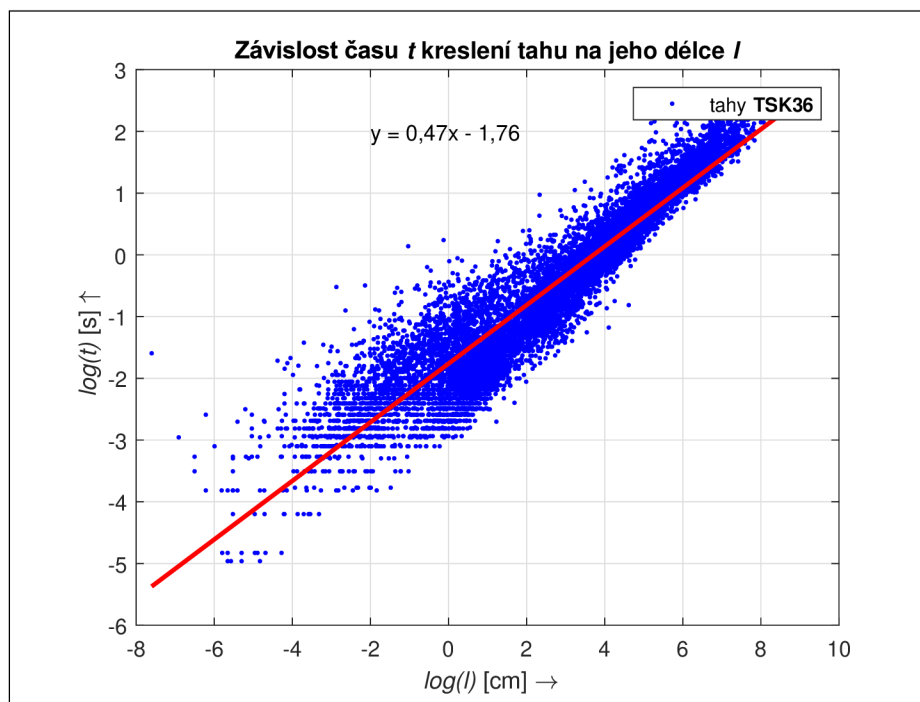
Při psaní a kreslení dochází k vytváření pohybů, které nabývají různých tvarů a především délek. Zdá se být samozřejmé, že delší pohyby budou trvat delší dobu. I když čas potřebný k vykreslení tahu narůstá lineárně s jeho délkou, nárůst doby je pomalejší, než nárůst délky. Princip isochornie [14] toto vysvětluje jako snahu udržet celkovou dobu potřebnou pro nakreslení tahu libovolné délky relativně konstantní [21]. Tento princip navíc můžeme pozorovat už u dětí od pěti let jejich věku, i když jeho vlastnosti se s věkem mění [22]. Při kreslení křivek lze pozorovat, že úhlová rychlost tahu je konstantní a tangenciální rychlost tedy narůstá v závislosti na poloměru [23]. Například při kreslení elips o desetinásobné velikosti, zjistil Viviani et al. u jednoho ze zkoumaných subjektů pouze 50% nárůst v době kreslení [20].

Principy isochornie lze navíc pozorovat i u jiných dobrovolných pohybů, jako je například zvedání předmětů, či ukazování [14], ale i u jiných živočišných druhů, například Makaků [19].

Pagliari et al. navíc ukazuje, že děti, které trpí samotnou dyslexií, nebo dyslexií spolu s dysgrafií, principy isochornie porušují [13].

Zajímavým faktem ohledně isochornie pro zkoumání vývojových poruch psaní u dětí může být také to, že podle Pagliarini et al. je isochornie nezávislá na věku dítěte u dětí docházejících do první až páté třídy ( $n=298$ ) [14]. Věkové složení dětí, ze kterých se skládá datová sada pro tuto práci je v podobném rozsahu. Zatímco ostatní parametry většinou vykazují závislost na věku dítěte, respektive na třídě do které dítě dochází, parametr založený na principu isochornie by mohl nabízet možnost srovnání dětí z širšího věkového rozsahu, což by mohlo pomoci vypořádat se s relativně malým množstvím dat v některých věkových kategoriích poskytnutého datasetu.





Obr. 3.2: Závislost doby tahu na jeho délce pro psané cvičení 36 a všechny subjekty datasetu. Z grafu je patrný lineární vztah mezi oběma proměnnými, v souladu s předpokladem Isochronie.

Na rozdíl od předchozích navržených parametrů, zkoumání isochronie nevyžaduje cvičení kreslená v páru za různých podmínek ale je možné ji zkoumat na každém jednotlivém cvičení zvlášť. Je však důležité podotknout, že vzhledem k tomu, že jde ve své podstatě o zkoumání závislosti doby vykreslení tahu  $t$  [s] na jeho délce  $l$  [cm], může se ukázat jako podstatné volit cvičení, které disponují větším množstvím tahů různých délek. Především se tedy může jednat o cvičení psaného textu, jako jsou **TSK30** až **TSK36**, které obsahují směsici textů, které dítě píše tiskacím i psacím písmem.

Z hlediska konkrétní analýzy je tento parametr zamýšlen jako sada dvojic hodnot, tedy páru hodnot (doba a délka) pro každý tah na papíře ve zkoumaném cvičení. Každý datový řádek v souboru cvičení obsahuje informaci o čase (timestamp), polohu  $x$  a  $y$  a informaci, zda se pero dotýká podložky. Je tedy poměrně jednoduché rozdělit body do seskupení představující tahy a z nich poté odečíst dobu tahu odečtením první a poslední časové hodnoty a celkovou uraženou vzdálenost použitím upraveného vzorce pro Pythagorovu větu, viz vzorec 3.10.

$$l = \sum_{n=1}^N \sqrt{(x[n] - x[n-1])^2 + (y[n] - y[n-1])^2} \quad (3.10)$$

Index výpočtu sumy začíná na indexu 1 v souladu s indexováním většiny programovacích jazyků, tedy od 0. Pro  $n = 1$  tedy dochází k porovnání  $x[1]$  a  $x[0]$ , resp.  $y[1]$  a  $y[0]$ , což jsou první a druhé prvky pole hodnot. Takto získané dvojice lze použít pro další analýzu, například určit sklon lineární závislosti pomocí lineární regrese a ty porovnat napříč datasetem.

Vzhledem k tomu, že Pagliarini et al. [13] hovoří o porušování principu isochronie, byl i v této práci navržen parametr snažící se tuto skutečnost zachytit. Při statistické analýze se pro vyjádření míry lineární korelace dvou proměnných, mimo jiné, používá také Pearsonův korelační koeficient  $\rho$ , který dosahuje hodnot v rozsahu  $[-1; 1]$ , kde hodnoty  $-1$  a  $1$  vyjadřují dokonalou negativní, respektive pozitivní korelaci a  $0$  žádnou korelaci mezi dvěma proměnnými. Podobně jako u směrodatné odchylky, i při výpočtu korelačního koeficientu musíme zohlednit fakt, že počítáme pouze se vzorkem populace, vztah je tedy upraven do podoby 3.11. Vzhledem k tomu, že isochronie předpokládá lineární vztah mezi délkou tahu a jejich dobou, je možné vytvořit hypotézu, že děti netrpící grafomotorickými potížemi budou dosahovat v absolutní hodnotě vyšších hodnot  $\rho$ , než děti vykazující znaky grafomotorických potíží.

$$\rho = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}} \quad (3.11)$$

Kde  $N$  je počet párů hodnot proměnných  $(x_1, y_1), \dots, (x_i, y_i)$  a  $\bar{x}$  a  $\bar{y}$  jsou jejich aritmetické průměry.

### 3.1 Programové řešení extrakce parametrů

Pro extrakci a výpočet parametrů byl opět jako u validace databáze zvolen jazyk C#, tentokrát v podobě konzolové aplikace na platformě .NET Core 3.1. Funkce .NET Core poskytují velkou škálu funkcí pro práci s daty. Velikou předností je v tomto případě i jednoduchá integrace SQL databáze pomocí Entity Framework a veliká dostupnost knihoven třetích stran, jako například ExcelDataReader [1]. I když C# nebyl navržen pro výpočetní úkony, stále v této disciplíně nabízí poměrně slušný, ale především snadno dostupný výkon. Je pravda, že existují i výkonnější jazyky, jako například C++. Implementační náročnost je však daleko vyšší. Jazyk jako Python je sám o sobě zase pomalejší a pro dosažení velké výkonnosti vyžaduje zásadní optimalizaci s použitím knihoven třetích stran. Poslední zvažovanou variantou byl Matlab, který také obecně nepatří mezi nejvýkonnější, nejde-li o silně optimalizovaný kód. Jeho výhodou je však široká nabídka dostupných knihoven obsahující implementované matematické funkce. Proto byl Matlab v rámci práce využíván při

návrhu parametrů, pro jakési prototypování. Následně však byly všechny parametry reimplementovány nativně přímo v C#.

C# tak nabízí zlatou střední cestu mezi rychlostí a jednoduchostí implementace. V případě nutnosti integrovanějšího, nebo rychlejšího přístupu je zde navíc možnost implementovat exekučně pomalé úseky, či výpočetně náročné úseky (obecně zvané bottleneck) ve výpočetně efektivnějším jazyku, jako je C++ a v kódu C# referencovat exportovanou .dll knihovnu. Stejná možnost existuje i pro funkce nabízené Matlabem, avšak v tomto případě ne s cílem zvýšení výpočetní rychlosti, ale pro zjednodušení, či zvýšení míry automatizace získávání parametrů.

Vzhledem k tomu, že charakter jednotlivých parametrů byl poměrně výstižně popsán už v první části kapitoly 3 a v odpovídajících podkapitolách, bude se tato sekce věnovat čistě aspektu softwarového řešení problematiky.

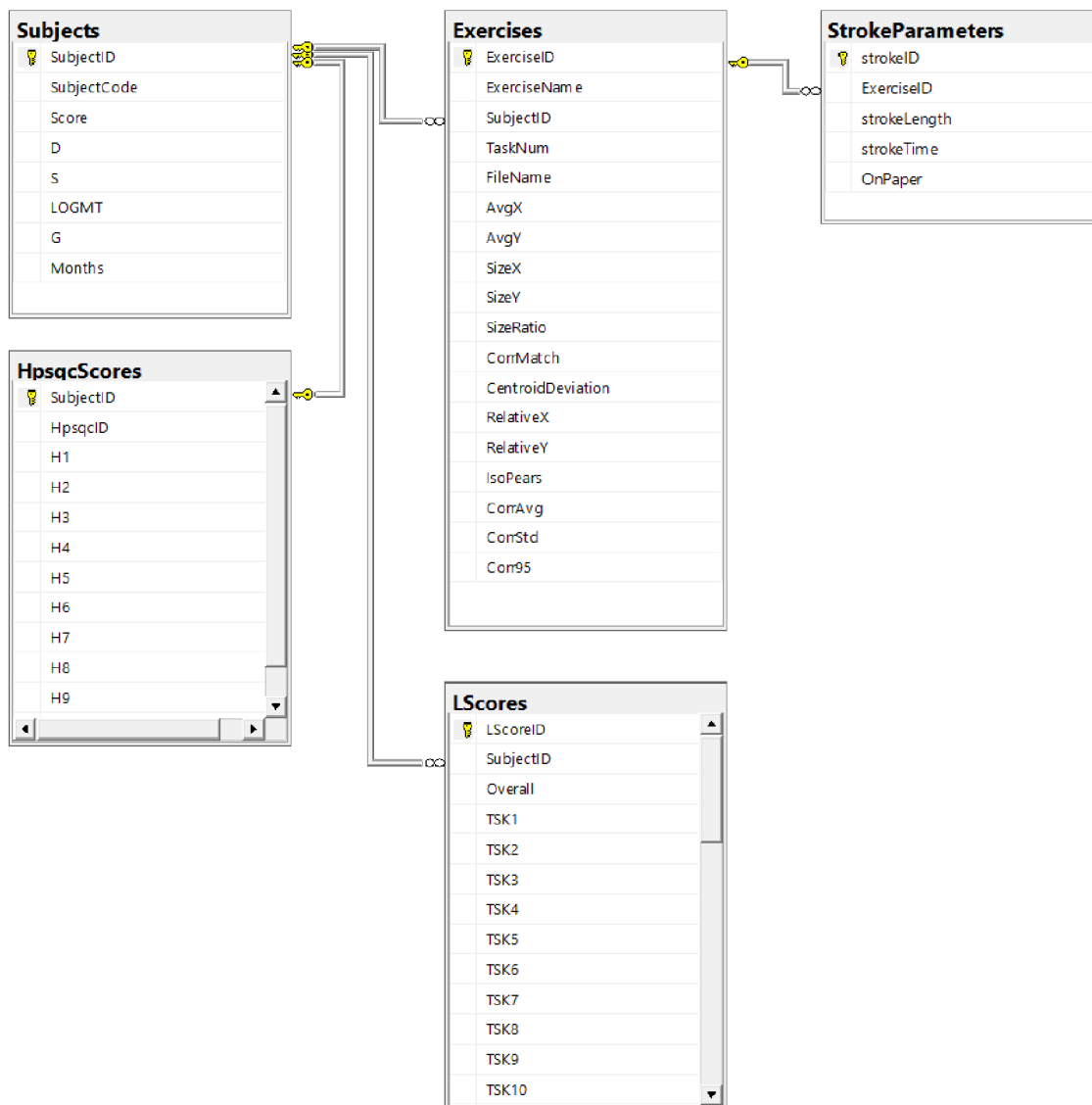
### 3.1.1 Objektová databáze

Pro snadnou práci s extrahovanými daty byla vytvořena instance databáze SQL. V ní byly vytvořeny tři tabulky odpovídající abstraktním objektům navržených tak, aby představovaly třídění a hierarchii zpracovaných dat. Jsou jimi:

- Subjects - obsahující obecné informace o subjektu
- Exercises - obsahující obecné informace o cvičení a vypočítané parametry a hodnoty některých mezivýpočtů
- StrokeParameters - obsahující obecné informace o jednotlivých tazích, či extrahovaných parametrech
- HpsqcScores - obsahující dílčí HPSQ-C hodnocení každého dítěte
- LScores - obsahující hodnocení jednotlivých cvičení 1-36 a jedno celkové hodnocení pro každé dítě podle škály v kapitole 2

V rámci této struktury nejsou ukládána surová data pro výpočet parametrů. Není tedy možné cvičení z databáze znovu získat, nebo znovu provést výpočet parametrů. K tomu je nutné mít lokální přístup k segmentovaným zdrojovým souborům.

Mezi tabulkami jsou vazby typu One-To-Many, s výjimkou vazby na tabulku HpsqcScores, kde je vazba typu P-ne-To-One. Prostřednictvím knihovny Entity Framework tato implementace nabízí velice jednoduché filtrování dat, které spolu v daném smyslu souvisí. Například při načítání dat pro jeden subjekt z databáze mohou být automaticky načteny všechny řádky tabulky Exercises, které splňují podmínku dané vazby, tedy odpovídající klíč `exerciseID`.



Obr. 3.3: Diagram databáze sloužící pro ukládání metadat cvičení a následně vypočtených parametrů a mezivýsledků výpočtu.

## Subjects

Tabulka zapouzdřuje abstrakci subjektu - dítěte - které vyplňovalo dotazník. V proměnných jsou uchovány následující hodnoty: SubjectCode - unikátní kódové označení subjektu, Score - HPSQ-C skóre, D - diagnosa dysgrafie, G - třída, S - pohlaví. Ke každému subjektu náleží právě jeden prvek v tabulce subjects.

## Exercises

Jedná se o abstrakci jednotlivých cvičení. Mezi tabulkou Subjects a Exercises je vazba typu One-To-Many, neboť k jednomu subjektu náleží zpravidla více cvičení. Tabulka obsahuje sloupce, představující jednotlivé získané parametry a základní informace o souboru, například:

- FileName - umístění v rámci souborového systému, na kterém byl spuštěn extrakční program
- TaskNum - odpovídající označení tasku, např. **TSK16**
- avgX - průměrná souřadnice x
- sizeX - celková velikost na ose X
- convMatch - výstupní hodnota parametru normalizované korelace viz kapitola 3.0.2
- centroidDeviation - rozdíl relativní polohy normalizovaných centroidů párových cvičení viz. kapitola 3.0.1
- relativeX - normalizovaná poloha x v rozsahu 0 až 1

## StrokeParameters

Obsahuje informace o jednotlivých tazích. Vzhledem k tomu, že tabulka obsahuje informace pouze pro jeden parametr, isochronii, viz kapitolu 3.0.3, je velmi jednoduchá a názvy jejích sloupců jsem poměrně výstižné, není tedy třeba je komentovat. Mezi tabulkami Exercises a StrokeParameters je opět vazba typu One-To-Many, protože až na výjimky náleží ke každému cvičení několik tahů.

## HpsqcScores a Lscores

Tabulky obsahují číselné hodnoty hodnocení příslušejících k danému dítěti. Jediný rozdíl mezi těmito tabulkami je typ vazby, kde HpsqcScores má na tabulku Subjects vazbu One-To-One a tabulka Lscores vazbu Many-To-One, neboť ke každému dítěti může náležet hodnocení více expertů.

### 3.1.2 Implementace extrakce parametrů v kódu

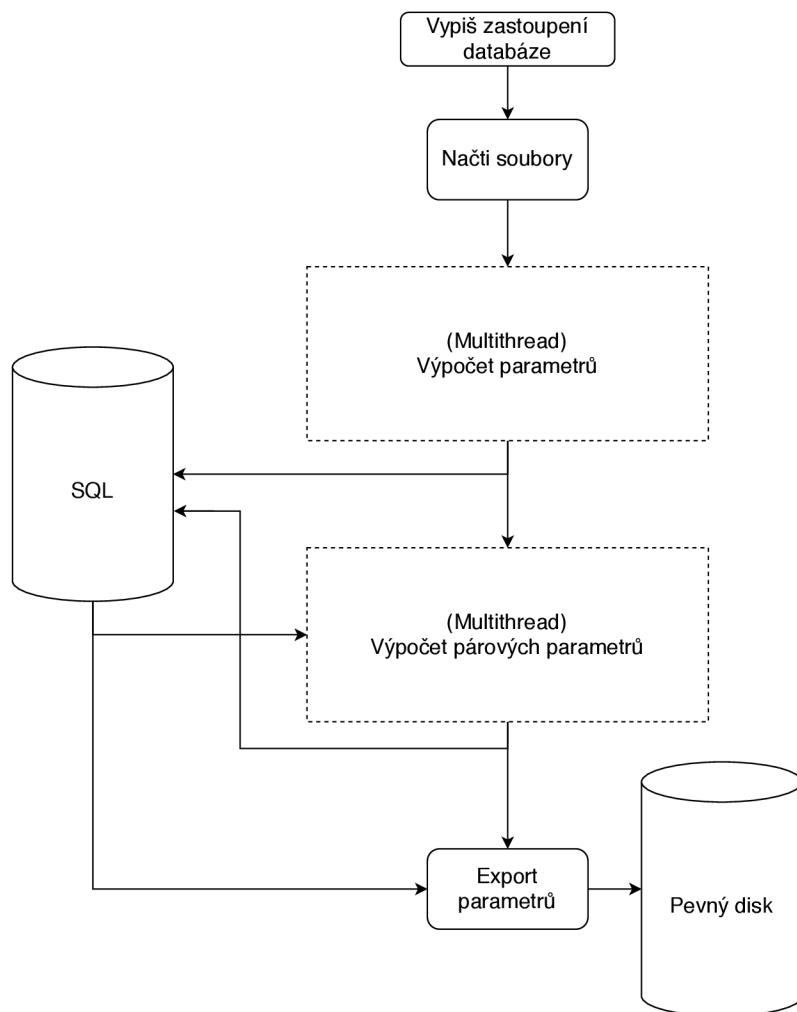
Program nedisponuje žádným grafickým rozhraním a ani nepřijímá žádné argumenty v rámci konzole, takže veškeré nastavení musí být manuálně upravováno přímo v kódu. Pro úspěšné spuštění je však pouze potřeba správně nastavit umístění složky se zdrojovými daty pro extrakci parametrů, excelového souboru obsahujícím metadata subjektů a cestu kam má být uložena matice extrahovaných parametrů. Jedná se o proměnné ExercisesPath, CorpusPath a ParameterMatrixPath v rámci třídy

`getParams`. Program očekává umístění složky, která obsahuje podsložky představující jednotlivé subjekty. Jde tedy o stejný souborový systém, jako na obr. 1.2. Vzhledem k absenci nastavení, je kód možné chápat v širším pojetí jako skript, tj. při každém spuštění je potřeba upravit volané metody odkomentováním, zakomentováním, nebo modifikací příkazů. Pro většinu účelů je však možné ponechat program v současné podobě. Pro výpočet, či export parametrů je možné určit subjekty pro které budou parametry spočítány. V určitém smyslu lze program brát jako knihovnu v tom smyslu, že jde o soubor funkčních metod pro výpočet a export parametrů, ale je potřeba určité programátorské zručnosti pro jejich správné seřazení a volání s platnými argumenty. Následující popis průběhu programu popisuje jakýsi jeden „ideální“ průběh programu, avšak metody jsou psány tak, aby nevyžadovaly žádný vstup z předchozích metod, ale načítaly si existující data rovnou z databáze. Je tedy možné některé části programu zavolat jednou a pro další spuštění metody vůbec nevolat pro šetření výpočetního času.

Po spuštění programu je do konzole vypsána jednoduchá statistika počtu dětí ve třídách, obdobná tabulce 1.1. Následně je vytvořena instance třídy `Extractor`, která zapouzdřuje metody a proměnné potřebné k extrakci parametrů, a je volána její metoda `GetParams`. V rámci této metody probíhá základní zpracování a analýza cvičení. Jsou získány informace o cvičeních, jako jsou délky a doba tahů, absolutní velikost cvičení, průměrná poloha bodů v rámci cvičení apod. Program načte cesty ke všem `.svc` souborům ve všech podsložkách nacházejících se v cestě zadané v proměnné `ExercisesPath` a provede pro ně výpočty.

Zpracovány jsou všechny soubory, které prošly procesem validace databáze. Zpracování je tak časově lehce náročné, v řádech minut. Pro větší efektivitu zpracování je použita paralelizace, která je v `C#` v tomto případě velice jednoduše implementovatelná, protože každému vláknu může být přidělen právě jeden soubor a zároveň spolu soubory nesouvisí v tom smyslu, že by vlákna potřebovala vzájemně přistupovat k paměti. Přidělování souborů do vláken obstarává sama metoda `Parallel.ForEach`. V rámci smyčky je zařazeno jednoduché vypisování počtu již zpracovaných souborů pro přibližný odhad potřebné doby pro zpracování a zároveň pro kontrolu, že nedošlo k zamrznutí programu. I když důsledkem paralelizace dochází k daleko větší náročnosti na paměť systému, doba zpracování je řádově kratší. Na systému s procesorem disponujícím osmi procesními vlákny je doba zpracování, přibližně 8680 souborů, necelé dvě minuty.

V rámci každé iterace smyčkou tedy dojde k načtení jednoho přiděleného souboru, z něhož je vytvořena instance třídy `Parameters`, která představuje právě jedno cvičení. Do této instance je načten obsah souboru a jeho název a umístění na disku. Každý datový řádek zdrojového souboru je následně interpretován třídou `point`. Ze všech získaných `pointů` jsou následně získány jednotlivé tahy.



Obr. 3.4: Zjednodušený flowchart výpočetního algoritmu pro výpočet nových parametrů on-line písma.

Soubor transformovaný do objektové podoby je následně předán do metody `GetParametersAsync`, ve které už dochází k samotné extrakci některých parametrů, nebo jejich mezivýpočtů. Především jde o průměrnou pozici bodů, tedy centroid, celkovou velikost a páry parametrů jednotlivých tahů pro parametr isochronie, tedy délky a doby každého tahu.

Extrahované parametry jsou vloženy do odpovídající instance třídy `Parameters` a ta je vložena do seznamu parametrů (`List<Parameters>`) připravených na uložení do SQL databáze.

Vzhledem k tomu, že mezi ukládanými daty musí vznikat vazby a je potřeba zaručit, že pro každý subjekt bude existovat pouze jeden záznam, probíhá vkládání do databáze pouze na jednom vlákně. Paralelní implementace ukládání je sice možná, ale časový zisk běhu programu je poměrně malý oproti potřebné době pro vytvoření

a odladění algoritmu.

Následuje volání `CompareRecallRecognition`, metody, ve které dochází k výpočtu parametrů, při kterých dochází k porovnání páru cvičení `Recognition` a `Recall`. Z databáze jsou načteny všechny dostupné páry pro každý subjekt. Pokud u některého subjektu chybí jeden z páru, není pro něj parametr spočítán. Pro každý pár je poté opět paralelně spočítána normalizovaná poloha centroidu a rozdíl těchto hodnot mezi oběma cvičeními. Všechny hodnoty jsou posléze vloženy do databáze, přičemž hodnota pro rozdíl cvičení je vždy vkládána do řádku cvičení `Recognition`.

Další volanou metodou je `CorrelateRecallRecognition` v níž dochází k výpočtu korelační matice pro páry cvičení a všech parametrů z tohoto vycházejících. Z pohledu výpočetní náročnosti jde o nejproblematictější metodu, neboť v rámci ní dochází k výpočtu korelací poměrně velkých matic; v případě cvičení **TSK27** a **TSK28**, tedy Reyových figur, jde o matice rozměrů řádově 100x100.

V průběhu metody, jsou nejprve vytvořeny páry cvičení `Recall` a `Recognition`. Pro každé cvičení z páru je následně volána funkce `CreateMatrix`, která provede pooling, a konvertuje data cvičení do typu `Matrix<double>`. Pro maticovou reprezentaci a některé výpočty je použita knihovna `Math.Net` [18], která je pro tento výpočet užitečná především implementací rychlého násobení matic. Matematické knihovny pro platformu `.NET` bohužel nejsou tak obsáhlé jako knihovny jiných jazyků, jako je třeba `Python`, nebo `Matlab`. Bylo tedy nutné vytvořit vlastní implementaci metody pro dvourozměrnou křížovou korelaci. Principem korelace je ve své podstatě vzájemné posouvání obou matic přes sebe. Při každém posunu jsou překrývající se hodnoty vynásobeny a sečteny. Výsledná hodnota je uložena na odpovídající pole výsledné korelační matice. Tato operace je pak provedena pro každou možnou polohu.

Odpovídající algoritmus je možné implementovat pomocí dvou vnořených `for` cyklů, kde jeden iteruje přes řádky výsledné korelační matice a druhý přes její sloupce. Při každé iteraci jsou vypočteny souřadnice překrývajících se hodnot matic a ty jsou extrahovány jako samostatné submatice, mezi kterými je následně proveden Hadamardův součin, tedy násobení po složkách, a jednotlivé výsledky jsou sečteny a uloženy do korelační matice.

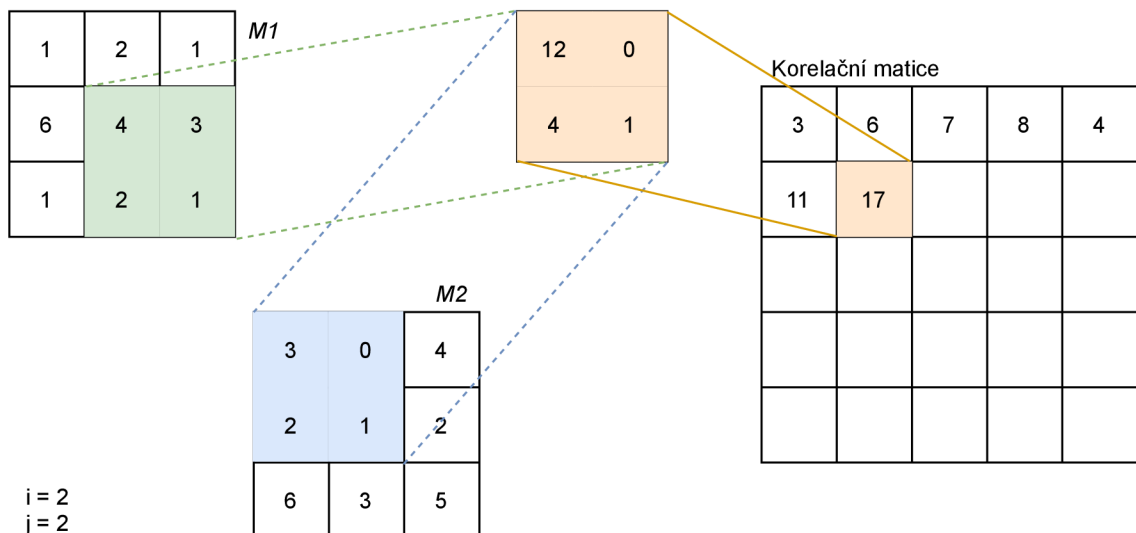
Velikost korelační matice se řídí podle vztahů 3.12 a 3.13.

$$C_X = M1_X + M2_X - 1 \quad (3.12)$$

$$C_Y = M1_Y + M2_Y - 1 \quad (3.13)$$

Kde  $M1$  a  $M2$  jsou vstupní matice do metody `CrossCorrelate`.





Obr. 3.5: Diagram výpočtu korelační matice.  $i$  a  $j$  představují iterační proměnné vnořených for cyklů. Vypočtená hodnota je vložena do korelační matice na pozici  $[j; i]$ .

Následující funkce `GetMetaDataFromExcel` a `GetLScoresFromExcel` jsou implementačně triviální. Dochází pouze k extrakci metadat a hodnocení cvičení z přiložených excelových tabulek k odpovídajícím subjektům a cvičením. Stejně jako v aplikaci *Skořápka*, i zde jsou používány knihovny EPPlus [9] a ExcelDataReader [1].

Poslední extrakční metodou je `CalculateIsochrony`. Program načte data o délkách a trvání tahů všech uložených cvičení. Pro každé cvičení je pak spočítán Pearsonův korelační koeficient těchto dvou proměnných, který je pak uložen do databáze.

Voláním metody `GenerateExcel` je pak generován excelový .xlsx soubor, do kterého je vložena matice spočítaných parametrů včetně hodnocení a některých metadat, jako např. třída a věk subjektu.

Tímto je běžný chod programu ukončen. V kódu se však nalézá několik dalších exportních metod, které se dají všechny zavolat zapouzdřující funkcí `ExportParameters`. Funkce slouží k exportování spočtených prvotních, jednoduchých parametrů, jako jsou délky a doby tahů, polohy centroidů a podobně. Tyto metody jsou principiálně velice jednoduché a nemá význam se jim podrobně věnovat, jelikož se jedná pouze o export do generovaných textových (.txt) souborů s využitím nativních funkcí C#. Co v této části ovšem stojí za zdůraznění je právě už jednou zmíněná funkce platformy .NET, LINQ, respektive její konkrétnější část spojená s databázovou knihovnou Entity Framework, LINQ to Entities, která umožňuje velice rychle, jednoduché a především přehledné vytvoření Query pro načtení požadovaných dat.

---

```

var centroidHighHPSQC = context.Exercises //Tabulka
    Exercises
    .Where(x => x.Subject.G == 3)
    .Where(x => x.TaskNum == 9)
    .OrderByDescending(x => x.Subject.Score)
    .Take(5);

```

---

Výpis 3.1: Příklad načítání dat z SQL

Na ukázce kódu 3.1 je patrných hned několik věcí:

- Vazby mezi tabulkami - i když query primárně vstupuje do tabulky Exercises, na řádku 2 provádíme filtraci a na řádku 4 seřazení podle proměnných z tabulky Subjects.
- Selekcce subjektů - Díky seřazení a následujícímu příkazu `Take()` můžeme jednoduše provádět podmíněnou selekci. V tomto případě „Načti 5 cvičení **TSK9** dětí ze třetí třídy, jejichž HPSQC skóre je nejvyšší“.

## 3.2 Analýza dat

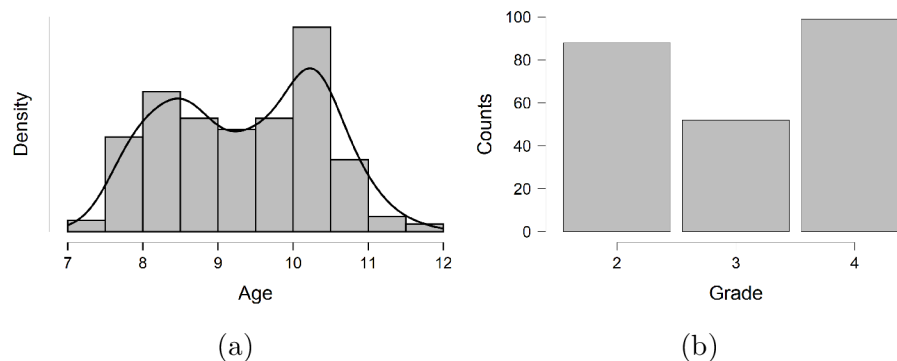
Pro statistickou analýzu a vyhodnocení parametrů byly zvoleny pouze děti druhé, třetí a čtvrté třídy. Navíc, po provedení třídění databáze popsaném v kapitole 1 došlo ještě k dalšímu sběru dat, které už však nebylo potřeba podrobit stejnému třídění, tabulka 1.1 je tedy neobsahuje. Tabulky 3.1 a 3.2 představují konečné složení databáze, použité pro analýzu. Pro analýzu byl použit program JASP [7].

Tab. 3.1: Věková statistika databáze

	Třída	Věk
Ar. průměr	3.046	9.360
Modus	4.000	10.333
Std	0.885	1.030

Tab. 3.2: Rozložení databáze do tříd

Třída	počet	%	% kumulativně	dysgrafie
2	88	36.820	36.820	1
3	52	21.757	58.577	6
4	99	41.423	100.000	8
Celkem	239	100.000		15



Obr. 3.6: Rozložení věku (a) a tříd (b)

Tab. 3.3: Cvičení označená hvězdičkou jsou tzv. párová (viz kap. 3). Parametr je počítán ze cvičení uvedeného a cvičení následujícího, tedy **TSK8\*** byl počítán z **TSK8** a **TSK9**. V závorkách jsou uvedeny zkratky značení parametrů použité při analýze.

Korelační koeficient (Corr)	Isochronie (IsoPears)	Centroid (Cent)
Tsk8*	Tsk33	Tsk8*
Tsk10*	Tsk34	Tsk10*
Tsk12*	Tsk35	Tsk12*
Tsk14*	Tsk36	Tsk14*
Tsk16*		Tsk16*
Tsk27*		Tsk27*

Jako validační kritéria bylo použito SHP, popsané v kapitole 2, tedy vždy hodnocení odpovídající příslušnému cvičení, nebo celkové hodnocení žáka. Dále subskóry HPQS-C 1, 2, 3, 9 a 10, kde 1, 2 a 10 hodnotí čitelnost výsledného produktu (legibility) a 3, 9 časový výkon dítěte (performance time). Dále byla jako validační kritérium použita diagnóza dysgrafie a hodnocení SHP pod zkratkou **LScore**, kde číslo za zkratkou určuje cvičení, kterému hodnocení patří, přičemž pomocí SHP pro cvičení **TSK27** a **TSK28** bylo vytvořeno také subhodnocení **LScore2827**, které je definováno jako prostý rozdíl odpovídajících hodnocení.

Provedeny byly tři metody analýzy:

1. Pearsonův korelační koeficient mezi jednotlivými vypočítanými parametry a validačními kritérii založených na dotazníku HPSQ-C
2. Spearmanův korelační koeficient mezi jednotlivými vypočítanými parametry a validačními kritérii založených SHP

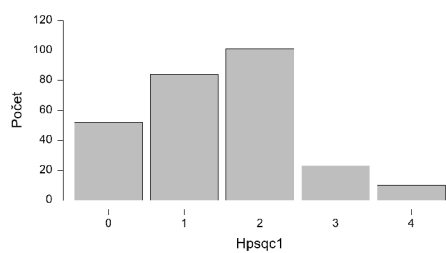
### 3. Mann-Whitney U-Test mezi diagnózou dysgrafie a jednotlivými vypočítanými parametry

Analýzy byly provedeny vždy pro každou třídu odděleně. Tabulka 3.3 ukazuje pro která cvičení byl spočítán každý parametr z kapitoly 3. Parametry *Corr* jsou vždy ještě doplněny o klíčové slovo (viz kap. 3.0.2):

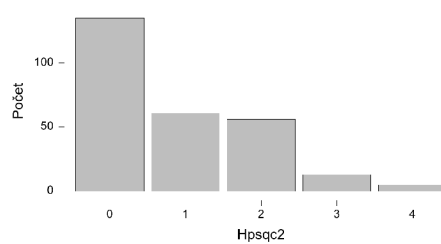
- *Max* - Maximální hodnota korelační matice
- *Avg* - Průměrná hodnota nenulových prvků korelační matice
- *Std* - Směrodatná odchylka nenulových hodnot korelační matice
- *95* - 95. percentil nenulových hodnot korelační matice

#### **Rozdělení hodnot validačních kritérií**

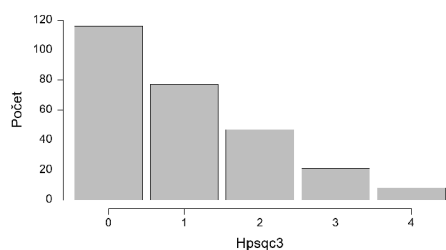
Jak už bylo zmíněno v předchozí sekci, jedním z použitých validačních kritérií je subjektivní hodnocení cvičení SHP. Vzhledem k teprve probíhajícímu sběru dat je k dispozici pouze jeden set hodnocení provedený pedagogem základní školy.



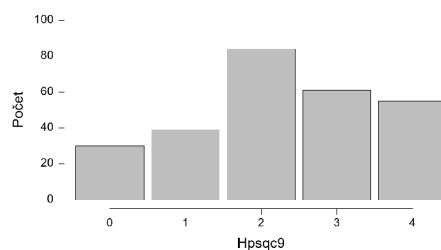
(a)



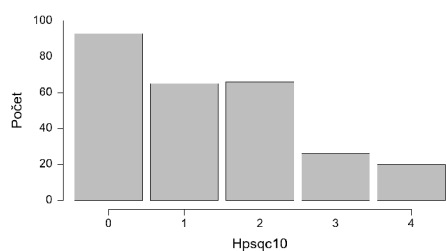
(b)



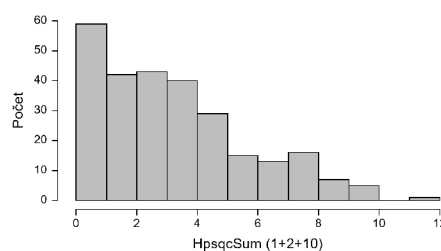
(c)



(d)

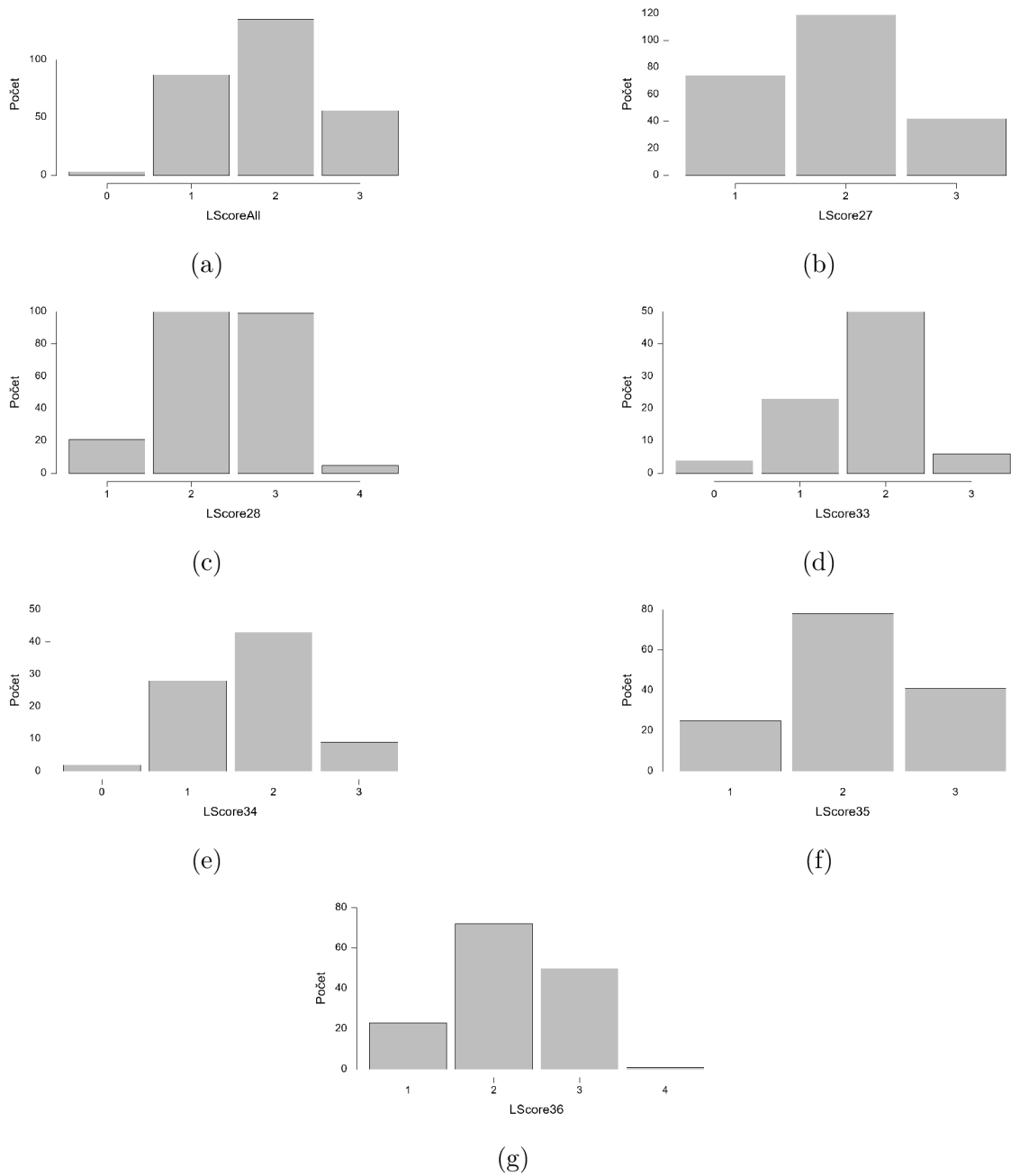


(e)



(f)

Obr. 3.7: Histogramy použitých subskórů hodnocení HPSQ-C napříč celým použitým datasetem (a-e), včetně sumy subskórů 1, 2 a 10 (f).



Obr. 3.8: Histogramy SHP napříč celým použitým datasetem. Celkového hodnocení (a) a jednotlivá hodnocení cvičení použitých pro analýzu (b-g).

## 4 Výsledky

### 4.1 Statistická analýza

V této sekci budou popsány výsledky statistické analýzy rozdělené do tříd a následně podle validačního kritéria. V tabulkách nejsou pro zkrácení uvedeny nesignifikantní výsledky (s pravděpodobností náhody  $p > 0.05$ ). U políček označených „-“ nebyl koeficient vůbec vypočítán.

#### Mann-Whitney U-Test

U dětí druhé třídy nebylo možné U-Test vůbec provést, neboť dataset dané kategorie neobsahuje dostatek subjektů s diagnózou dysgrafie. U dětí třetí a čtvrté třídy byl test proveden, ale neodhalil žádné signifikantní korelace.

#### 4.1.1 2. třída

##### HPSQ-C

Z korelační analýzy parametrů a HPSQ-C vyplývá významná korelace mezi HPSQ-C subskórem 1 a korelačními parametry pro cvičení **TSK10**. Dále mezi subskórem HPSQ-C 3 a korelačními parametry pro cvičení **TSK14**, **TSK16** a **TSK27**.

##### SHP

Korelační analýza (tab. 4.2) v tomto případě odhalila významné korelace napříč mnoha parametry a celkovým hodnocením SHP, především však u korelačních parametrů pro cvičení **TSK12** a **TS27**.

Tab. 4.1: Korelační analýza (Pearson) nových parametrů a hodnocení HPSQ-C u dětí 2. třídy

Variable	Hpsqc1	Hpsqc2	Hpsqc3	Hpsqc10	HpsqcSum
	Pearson (r)	Pearson (r)	Pearson (r)	Pearson (r)	Pearson (r)
Tsk8CorrMax	-0,273 *				
Tsk10CorrMax	-0,267 *				
Tsk10CorrAvg	-0,276 *			-0,291 *	-0,298 **
Tsk10CorrStd	-0,333 **			-0,276 *	-0,343 **
Tsk10Corr95	-0,332 **			-0,267 *	-0,328 **
Tsk12CorrMax	-0,254 *		-0,272 *		-0,263 *
Tsk12CorrAvg		-0,247 *			
Tsk12CorrStd		-0,273 *			
Tsk12Corr95		-0,268 *			-0,235 *
Tsk14CorrAvg			-0,285 *		
Tsk14CorrStd			-0,306 **		
Tsk14Corr95			-0,309 **		
Tsk16CorrMax	-0,259 *				-0,243 *
Tsk16CorrAvg			-0,262 *		
Tsk16CorrStd			-0,285 *		
Tsk16Corr95			-0,256 *		
Tsk27CorrMax	-0,241 *		-0,251 *		-0,264 *
Tsk27CentDev	0,313 **				0,260 *

\*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$



Tab. 4.2: Korelační analýza (Spearman) nových parametrů a SHP u dětí 2. třídy

Parametr	LScoreAll	LScore27	LScore28	LScore33
	$\rho$	$\rho$	$\rho$	$\rho$
Tsk27CorrMax	-0,372 ***	-0,308 **	-0,531 ***	-
Tsk27CorrAvg	-0,320 **	-0,217 *	-0,420 ***	-
Tsk27CorrStd	-0,290 **		-0,419 ***	-
Tsk27Corr95	-0,293 **		-0,387 ***	-
Tsk16Cent	0,234 *	-	-	-
Tsk27Cent		0,230 *	0,360 ***	-
Tsk33IsoPears	-0,297 **	-	-	-0,367 ***
Tsk8CorrMax	-0,448 ***	-	-	-
Tsk8CorrAvg	-0,374 ***	-	-	-
Tsk8CorrStd	-0,382 ***	-	-	-
Tsk8Corr95	-0,361 ***	-	-	-
Tsk10CorrMax	-0,372 ***	-	-	-
Tsk10CorrAvg	-0,343 **	-	-	-
Tsk10CorrStd	-0,397 ***	-	-	-
Tsk10Corr95	-0,401 ***	-	-	-
Tsk12CorrMax	-0,581 ***	-	-	-
Tsk12CorrAvg	-0,536 ***	-	-	-
Tsk12CorrStd	-0,539 ***	-	-	-
Tsk12Corr95	-0,578 ***	-	-	-
Tsk14CorrMax	-0,451 ***	-	-	-
Tsk14CorrAvg	-0,340 **	-	-	-
Tsk14CorrStd	-0,398 ***	-	-	-
Tsk14Corr95	-0,361 ***	-	-	-
Tsk16CorrMax	-0,267 *	-	-	-
Tsk16CorrAvg	-0,341 **	-	-	-
Tsk16CorrStd	-0,347 ***	-	-	-
Tsk16Corr95	-0,307 **	-	-	-

\*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$

## 4.1.2 3. třída

### HPSQ-C

Tab. 4.3: Korelační analýza (Pearson) nových parametrů a hodnocení HPSQ-C u dětí 3. třídy

Parametr	Hpsqc1 Pearson (r)	Hpsqc2 Pearson (r)	Hpsqc3 Pearson (r)
Tsk8CorrAvg	-0,340 *		
Tsk8CorrStd	-0,333 *		
Tsk8Corr95	-0,327 *		
Tsk12CorrStd	-0,276 *		
Tsk12Corr95	-0,287 *		
Tsk14CorrStd			0,291 *
Tsk14Corr95			0,279 *
Tsk16CorrAvg		0,29 *	

\*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$

Korelační analýza pro HPSQ-C u dětí třídy odhalila lehce signifikantní korelace u korelačních parametrů několika cvičení, avšak počet korelujících parametrů byl nižší, než 5% celkového počtu pro tento set.

## SHP

Tab. 4.4: Korelační analýza (Spearman) nových parametrů a SHP u dětí 3. třídy

Parametr	LScoreAll	LScore27	LScore28	LScore36
	$\rho$	$\rho$	$\rho$	$\rho$
Tsk27CorrMax	-0,520 ***	-0,447 **	-0,498 ***	
Tsk27CorrAvg	-0,349 *	-0,336 *	-0,514 ***	-
Tsk27CorrStd	-0,321 *	-0,335 *	-0,484 ***	-
Tsk27Corr95	-0,347 *	-0,326 *	-0,467 **	-
Tsk16Cent	0,409 **	-	-	-
Tsk27Cent	0,344 *	0,322 *	0,477 ***	-
Tsk36IsoPears		-	-	-0,405 **
Tsk8CorrMax	-0,376 **	-	-	-
Tsk10CorrMax	-0,458 ***	-	-	-
Tsk10CorrStd	-0,277 *	-	-	-
Tsk12CorrMax	-0,437 **	-	-	-
Tsk12CorrAvg	-0,336 *	-	-	-
Tsk12CorrStd	-0,384 **	-	-	-
Tsk12Corr95	-0,358 **	-	-	-
Tsk16CorrMax	-0,297 *	-	-	-

\*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$

Spearmanova korelační analýza v tomto případě odhalila signifikantní korelace mezi korelačním parametrem *CorrMax* a celkovým hodnocením pro cvičení **Tsk27** a **Tsk10**. Cvičení **Tsk27** také významně koreluje všemi svými parametry s vlastním hodnocením **LScore28**.

### 4.1.3 4. třída

#### HPSQ-C

Tab. 4.5: Korelační analýza (Pearson) nových parametrů a hodnocení HPSQ-C u dětí 4. třídy

Parametr	Hpsqc1 r	Hpsqc2 r	Hpsqc3 r	Hpsqc9 r	Hpsqc10 r	HpsqcSum r
Tsk8CorrAvg	-0,219 *					-0,211 *
Tsk8CorrStd	-0,198 *					-0,208 *
Tsk8Corr95						-0,199 *
Tsk10CorrMax	-0,299 **	-0,279 **			-0,199 *	-0,306 **
Tsk10CorrStd					-0,228 *	-0,226 *
Tsk10Corr95					-0,215 *	-0,206 *
Tsk12CorrMax	-0,230 *					
Tsk14CorrMax					-0,207 *	
Tsk14CorrAvg		-0,203 *		0,199 *		
Tsk14CorrStd		-0,223 *				
Tsk14Corr95		-0,219 *		0,204 *		
Tsk27CorrAvg	-0,208 *					
Tsk8Cent		0,209 *				
Tsk12Cent				-0,255 *		
Tsk14Cent			-0,213 *			
Tsk27Cent		0,298 **		0,201 *		0,230 *
Tsk36IsoPears			-0,224 *			

\*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$

Nejsilněji korelujícím parametrem z datasetu čtvrté třídy s kritériem HPSQ-C (1, 2, 10 a jejich součet, HpsqcSum) byl parametr *CorrMax* cvičení **TSK10**.

#### SHP

Nejsilněji korelujícími parametry jsou *Corr* parametry cvičení **Tsk27** s hodnocením **LScore28** a cvičení **Tsk10**, **Tsk14** a **Tsk16** s celkovým hodnocením **LScoreAll**.

Tab. 4.6: Korelační analýza (Spearman) nových parametrů SHP u dětí 4. třídy

Variable	LScoreAll		LScore27		LScore28		LScore36	
	$\rho$		$\rho$		$\rho$		$\rho$	
Tsk27CorrMax	-0,264	**			-0,489	***	-	
Tsk27CorrAvg					-0,395	***	-	
Tsk27CorrStd	-0,225	*			-0,470	***	-	
Tsk27Corr95					-0,428	***	-	
Tsk27Cent			0,228	*	0,202	*	-	
Tsk36IsoPears	-0,214	*					-0,202	*
Tsk8CorrMax	-0,257	*	-		-		-	
Tsk8CorrAvg	-0,245	*	-		-		-	
Tsk8CorrStd	-0,248	*	-		-		-	
Tsk10CorrMax	-0,250	*	-		-		-	
Tsk10CorrAvg	-0,330	***	-		-		-	
Tsk10CorrStd	-0,407	***	-		-		-	
Tsk10Corr95	-0,346	***	-		-		-	
Tsk12CorrMax	-0,224	*	-		-		-	
Tsk12CorrStd	-0,203	*	-		-		-	
Tsk14CorrMax	-0,247	*	-		-		-	
Tsk14CorrAvg	-0,282	**	-		-		-	
Tsk14CorrStd	-0,353	***	-		-		-	
Tsk14Corr95	-0,336	***	-		-		-	
Tsk16CorrMax	-0,333	***	-		-		-	
Tsk16CorrAvg	-0,249	*	-		-		-	
Tsk16CorrStd	-0,278	**	-		-		-	
Tsk16Corr95	-0,283	**	-		-		-	
Tsk12Cent	0,236	*	-		-		-	
Tsk16Cent	0,200	*	-		-		-	

\*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$



## 5 Diskuse

### 5.1 Validace databáze

Z původních 9056 souborů jich bylo vyřazeno 371, viz kap. 1. U žádného subjektu však nedošlo k vyřazení všech, či většiny cvičení. Počet subjektů zůstal stejný, tedy 340. Z těch bylo nemalé množství vyřazeno kvůli chybějícím informacím, ve většině případů vyhodnocení skóre HPSQ-C. Nakonec tedy zůstalo 203 validních subjektů s 5821 jednotlivými cvičeními. I když je databáze poměrně rozsáhlá, disponuje malým množstvím subjektů s potvrzenou diagnózou dysgrafie. Tato skutečnost může výrazně ztížit vyhodnocení navržených parametrů. Složení databáze popisuje tabulka 1.1.

Chyby, kvůli kterým byly soubory vyřazeny jsou z většinové části odstranitelné bez nutnosti opakování záznamu protokolu. Mezi tyto chyby většinou patří chyba při segmentaci a labelování souborů. I když jsou chyby opravitelné, nejefektivnější by bylo jim předcházet. Není však jasná přímá příčina chyb. Segmentace i labelování probíhá do určité míry manuálně, může tedy jít o chyby lidského faktoru. K chybám může také docházet vlivem nevhodného software, například vlivem nízké přehlednosti jeho chodu.

Mezi neodstranitelné chyby ve velké většině patřily chyby záznamového zařízení. Ty se projevovaly špatným záznamem cvičení. Takto znehodnocené soubory už dál není možné použít a bohužel není možné je nijak rekonstruovat. V tomto případě lze pouze doporučit vyšší ostražitost personálu při záznamu cvičení a při projevech neobvyklého chování zařízení chybu odstranit. Nelze vyloučit chybovost softwaru, ani hardwaru zařízení.

Implementovaný program se ukázal být jako spolehlivý a stabilní. V průběhu používání se nevyskytla žádná neočekávaná chyba programu. Program v současné době slouží pouze k validaci kreslicích cvičení v konkrétním formátu, avšak jeho případné rozšíření na jiné datové struktury je možné a nenáročné.

### 5.2 Aplikace pro sběr hodnocení

Aplikace *Skořápka* byla navržena pro rychlý, pohodlný a stabilní sběr hodnocení uživatelem. Disponuje funkcemi, které tuto činnost velmi usnadňují a zpřehledňují. Aplikace v současnosti běží na serveru v několika instancích přístupných z internetu. V tomto režimu je používána speciálními pedagogy a učiteli. V tuto chvíli bylo dokončeno hodnocení celé databáze jedním učitelem. Je tedy možné aplikaci označit za funkční a otestovanou. *Skořápka* má podobné limitace jako aplikace pro třídění databáze, tedy, že v současné době umí otevírat pouze specifický formát souborů,

avšak s tím rozdílem, že její kód je napsán rozšiřitelnějším způsobem, než je tomu u validačního nástroje. V případě potřeby je tedy možné ze *Skořápky* vytvořit univerzální nástroj pro hodnocení obrazových souborů, čímž by mohla výrazně usnadnit sběr hodnocení potřebných k vývoji dalších parametrů.

### 5.3 Statistická analýza nových parametrů

V rámci práce byly navrženy a naprogramovány nové parametry, které by mohly být používány k vyhodnocování vývojových grafomotorických potíží. U datasetu druhé třídy bohužel nebylo možné použít Mann-Whitney U-Test, kvůli nedostatečnému počtu diagnostikovaných dysgrafií v dané skupině, nebylo tedy možné ověřit použitelnost kombinace tohoto testu vůči parametrům. U datasetu třetí a čtvrté třídy bylo možné Mann-Whitney analýzu provést, ale žádná z hodnot nebyla signifikantní, což vzhledem k počtu diagnostikovaných dysgrafií v těchto skupinách může být důsledkem právě jejich nízkého výskytu, viz tab. 3.2.

U dětí ze druhé třídy byly nalezeny četné korelace mezi *Corr* parametry a subskórem HPSQ-C 3 ( $r = [-0,251; -0,309]$ ), který vypovídá o tom, zda má dítě dost času na opsání zdrojového materiálu. Korelace s parametry, které porovnávají kreslení stejného cvičení jednou s předlohou a jednou z paměti, by mohlo znamenat, že tyto děti mají problém s pracovní pamětí (viz tabulka 4.1). S pracovní pamětí souvisí také HPSQ-C 9, se kterým korelovaly *Corr* a *Cent* parametry u dětí čtvrté třídy, ovšem ne do takové míry jako právě u druháků (tabulka 4.5).

U dětí z každé třídy byla odhalena korelace mezi parametrem *IsoPears* a SHP. Ze všech tříd byla tato korelace nejslabší u dětí 4. třídy, kde  $\rho = -0,202$  (tab. 4.6), zároveň ale u těchto jediných koreloval parametr i s hodnocením HPSQ-C (tab. 4.5). Nejsilnější korelace ( $\rho = -0,405$ ) parametru je u dětí 3. třídy v tabulce 4.4. Pagliarini et. al říká, že děti trpící dyslexií a dysgrafií principy isochronie porušují [13]. Tato odhalená korelace nasvědčuje možnosti, že principy isochronie porušují i děti trpící grafomotorickými potížemi nezávisle na věku, resp. třídě v rozsahu [2; 4]. Zvončák et al. dosáhli Spearmanova korelačního koeficientu  $\rho$  s HPSQ-C v rozsahu  $[-0,29; -0,34]$ , resp.  $[0,27; 0,31]$  [25]. Zvončák et al. také dosáhli  $\rho = 0,37$  a  $\rho = -0,38$  při korelační analýze pohybu nad tabletem při psaní textu [24].

Velmi zajímavými se ukázaly být parametry pro **TSK27**. Parametry *Corr* i *Cent* silně korelují s hodnotou *LScore28*, tedy tím cvičením z páru, kresleným z paměti. Předpokladem hodnot *LScore27* a *LScore28* je, že děti nakreslí cvičení **TSK28** hůř a hodnota *LScore28* bude tedy vyšší. Vzhledem k vysoké složitosti cvičení Reyovy figury, bude tento rozdíl hodnot vyšší u dětí trpícími grafomotorickými potížemi. Nejvyšší hodnoty korelace je možné najít v tabulce 4.2. Nejvyšší hodnotu korelace pro parametr *Cent* pak v tabulce 4.4 pro třetí třídu.



Napříč celou analýzou je patrné, že parametry *Corr* pro cvičení **TSK8** až **TSK16** signifikantně korelují s hodnocením HPSQ-C i hodnotami *LScoreAll*. Toto je nejpatrnější v tabulkách pro druhou třídu, 4.1 a 4.2. Z variant parametru *Corr* nejčastěji dosahuje nejvyšší hodnoty korelace i signifikance varianta *Max*. Zároveň se nejčastěji vyskytuje u cvičení, kde se vyskytuje jako jediná signifikantní varianta parametru. Další v pořadí nejvyšších hodnot je varianta *Std*. Mezi variantami parametru však nejsou velké rozdíly, což ukazuje i na možnost dalšího výzkumu, buď za cílem dalšího zdokonalení parametru, nebo alespoň otestování, která varianta je skutečně ta nejspolehlivější.

### 5.3.1 Limitace

Mezi hlavní limitace analýzy patří set hodnocení SHP, který je vzhledem k pilotního sběru hodnocení zatím jediný. Hodnocení prováděl pedagog základní školy bez odpovídajícího výcviku, nebo specializace. Jak je patrné z tabulek v sestavě 3.8, dominantní hodnotou hodnocení je hodnota 2, která už představuje dítě poměrně postižené grafomotorickými potížemi. Toto rozložení nesouhlasí s rozložením HPSQ-C (legibility) (viz Obr. 3.7 (f)), jehož nejvíce zastoupenou hodnotou je právě 0, což souhlasí i množstvím diagnostikovaných dysgrafií podle tabulky 3.2. Je tedy možné, že pedagog provádějící hodnocení byl příliš přísný a hodnocení jsou posunuta doprava.

Dalšími limity práce může být stále nedostatečná databáze, především její obsah dysgrafií (tab. 3.2), ale také nerovnoměrné rozložení dětí do tříd, viz tab. 3.6.



# Závěr

Práce se zabývá vývojem softwarových nástrojů pro zjednodušení zkoumání objektivních parametrů on-line písma, především na úrovni sběru a třídění dat. Dále se práce zabývá návrhem, naprogramováním a korelační analýzou nových objektivních parametrů on-line písma a validačních kritérií v podobě hodnocení HPSQ-C a subjektivním hodnocením psacího produktu odborníkem na Likertově škále.

Diplomová práce si kladla následující tři cíle, pro které bylo navrženo odpovídající řešení:

1. **Validace rozsáhlého datasetu poskytnutého pro výzkum. Vzhledem k množství dat bude nutné vyvinout softwarový nástroj pro usnadnění třídění dat.**

V rámci práce byl implementován jednoduchý softwarový nástroj, který umožňuje třídění dat do tří kategorií - validní, návratně poškozený, nenávratně poškozený. Databáze byla následně pomocí nástroje přetříděna. (viz kapitola 1).

2. **Návrh a implementace desktopové aplikace pro subjektivní hodnocení kreslicích a psacích cvičení odborníkem.**

Podobně jako pro řešení předchozího cíle, i zde byl vyvinut program, který uživateli umožňuje zobrazovat cvičení a přiřazovat jim známky na základě subjektivního hodnocení, případně jiného kritéria. Aplikace nabízí funkce usnadňující hodnocení velkého objemu souborů, díky implementaci pokročilých navigačních funkcí a zobrazovacích funkcí, jako je zoom.

3. **Návrh několika nových parametrů online písma, které by mohly být využity pro diagnostiku grafomotorických potíží u dětí.**

Práce popisuje návrh tří nových parametrů - Odchylka normalizovaných centroidů, Pearsonův korelační koeficient ověřující principy isochronie, Normalizovaný korelační koeficient a jeho varianty. Parametry jsou následně zkoumány pomocí korelační analýzy (Pearsonův korelační koeficient, Spearmanův korelační koeficient, Mann-Whitney U-Test).

Z analýzy je patrné, že navržené parametry mají určitý potenciál, avšak je potřeba provést korelaci znovu proti robustnější sadě hodnocení SHP. Implementace parametrů také naskýtá spoustu možností pro návrh variant, které by mohlo být užitečné prozkoumat, neboť by mohly fungovat lépe, než varianty prozatím používané. Lákavou oblastí dalšího výzkumu je hlubší prozkoumání parametrů založených na principu isochronie, které dosud nejsou v soudobé literatuře pro tento účel důkladně prozkoumány.

Další možnou cestou navázání na tuto práci by mohlo být využití strojového učení ať už pro optimalizaci parametrů, nebo pro predikci grafomotorických potíží.



# Literatura

- [1] ExcelDataReader. 2020.  
URL <https://github.com/ExcelDataReader/ExcelDataReader>
- [2] Asselborn, T.; Chapatte, M.; Dillenbourg, P.: Extending the Spectrum of Dysgraphia: A Data Driven Strategy to Estimate Handwriting Quality. *Scientific Reports*, roènik 10, è. 1, Feb 2020: str. 3140, ISSN 2045-2322, doi: 10.1038/s41598-020-60011-8.  
URL <https://doi.org/10.1038/s41598-020-60011-8>
- [3] Asselborn, T.; Gargot, T.; Kidziński, Ł.; aj.: Automated human-level diagnosis of dysgraphia using a consumer tablet. *npj Digital Medicine*, roènik 1, è. 1, Aug 2018: str. 42, ISSN 2398-6352, doi:10.1038/s41746-018-0049-x.  
URL <https://doi.org/10.1038/s41746-018-0049-x>
- [4] Engel-Yeger, B.; Nagauker-Yanuv, L.; Rosenblum, S.: Handwriting Performance, Self-Reports, and Perceived Self-Efficacy Among Children With Dysgraphia. *American Journal of Occupational Therapy*, roènik 63, è. 2, Bøezen 2009: s. 182–192, doi:10.5014/ajot.63.2.182.  
URL <https://doi.org/10.5014/ajot.63.2.182>
- [5] Feder, K. P.; Majnemer, A.: Handwriting development, competency, and intervention. *Developmental Medicine & Child Neurology*, roènik 49, è. 4, 2007: s. 312–317, doi:<https://doi.org/10.1111/j.1469-8749.2007.00312.x>, <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8749.2007.00312.x>.  
URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8749.2007.00312.x>
- [6] Gavenčiak, M.: Systém poloautomatizované segmentace online písma. 2019.
- [7] JASP Team: JASP (Version 0.14.1)[Computer software]. 2020.  
URL <https://jasp-stats.org/>
- [8] Kushki, A.; Schwellnus, H.; Ilyas, F.; aj.: Changes in kinetics and kinematics of handwriting during a prolonged writing task in children with and without dysgraphia. *Research in Developmental Disabilities*, roènik 32, è. 3, Kvìten 2011: s. 1058–1064, doi:10.1016/j.ridd.2011.01.026.  
URL <https://doi.org/10.1016/j.ridd.2011.01.026>
- [9] Källman, J.; Alm, M.: EPPlus. 2021.  
URL <https://epplussoftware.com/>

- [10] McHale, K.; Cermak, S. A.: Fine Motor Activities in Elementary School: Preliminary Findings and Provisional Implications for Children With Fine Motor Problems. *American Journal of Occupational Therapy*, ročník 46, è. 10, Øíjen 1992: s. 898–903, doi:10.5014/ajot.46.10.898.  
URL <https://doi.org/10.5014/ajot.46.10.898>
- [11] Mekyska, J.; Faundez-Zanuy, M.; Mzourek, Z.; aj.: Identification and Rating of Developmental Dysgraphia by Handwriting Analysis. *IEEE Transactions on Human-Machine Systems*, ročník 47, è. 2, 2017: s. 235–248, ISSN 2168-2291, doi:10.1109/THMS.2016.2586605.  
URL <http://ieeexplore.ieee.org/document/7530869/>
- [12] Mekyska, J.; Galaz, Z.; Safarova, K.; aj.: Computerised Assessment of Graphomotor Difficulties in a Cohort of School-aged Children. In *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2019, s. 1–6, doi:10.1109/ICUMT48472.2019.8970767.
- [13] Pagliarini, E.; Guasti, M. T.; Toneatto, C.; aj.: Dyslexic children fail to comply with the rhythmic constraints of handwriting. *Human Movement Science*, ročník 42, 2015: s. 161 – 182, ISSN 0167-9457, doi: <https://doi.org/10.1016/j.humov.2015.04.012>.  
URL <http://www.sciencedirect.com/science/article/pii/S0167945715000834>
- [14] Pagliarini, E.; Scocchia, L.; Vernice, M.; aj.: Children’s first handwriting productions show a rhythmic structure. *Scientific Reports*, ročník 7, è. 1, Èervenec 2017, doi:10.1038/s41598-017-05105-6.  
URL <https://doi.org/10.1038/s41598-017-05105-6>
- [15] Palmis, S.; Danna, J.; Velay, J.-L.; aj.: Motor control of handwriting in the developing brain: A review. *Cognitive Neuropsychology*, ročník 34, è. 3-4, Kvìten 2017: s. 187–204, doi:10.1080/02643294.2017.1367654.  
URL <https://doi.org/10.1080/02643294.2017.1367654>
- [16] Richter, B.: Svg2Xaml. 2020.  
URL <https://github.com/smischke/svg2xaml>
- [17] Rosenblum, S.; Dror, G.: Identifying Developmental Dysgraphia Characteristics Utilizing Handwriting Classification Methods. *IEEE Transactions on Human-Machine Systems*, ročník 47, è. 2, 2017: s. 293–298, doi:10.1109/THMS.2016.2628799.

- [18] Ruegg, C.; Cuda, M.; Gael, J. V.: MathNet.Numerics. 2021.  
URL <https://numerics.mathdotnet.com/>
- [19] Sartori, L.; Camperio-Ciani, A.; Bulgheroni, M.; aj.: Reach-to-grasp movements in *Macaca fascicularis* monkeys: the Isochrony Principle at work. *Frontiers in Psychology*, roènik 4, 2013, doi:10.3389/fpsyg.2013.00114.  
URL <https://doi.org/10.3389/fpsyg.2013.00114>
- [20] Viviani, P.; Cenzato, M.: Segmentation and coupling in complex movements. *Journal of Experimental Psychology: Human Perception and Performance*, roènik 11, è. 6, 1985: s. 828–845, doi:10.1037/0096-1523.11.6.828.  
URL <https://doi.org/10.1037/0096-1523.11.6.828>
- [21] Viviani, P.; McCollum, G.: The relation between linear extent and velocity in drawing movements. *Neuroscience*, roènik 10, è. 1, Záóí 1983: s. 211–218, doi: 10.1016/0306-4522(83)90094-5.  
URL [https://doi.org/10.1016/0306-4522\(83\)90094-5](https://doi.org/10.1016/0306-4522(83)90094-5)
- [22] Viviani, P.; Schneider, R.: A developmental study of the relationship between geometry and kinematics in drawing movements. *Journal of Experimental Psychology: Human Perception and Performance*, roènik 17, è. 1, 1991: s. 198–218, doi:10.1037/0096-1523.17.1.198.  
URL <https://doi.org/10.1037/0096-1523.17.1.198>
- [23] Viviani, P.; Terzuolo, C.: Trajectory determines movement dynamics. *Neuroscience*, roènik 7, è. 2, 1982: s. 431 – 437, ISSN 0306-4522, doi: [https://doi.org/10.1016/0306-4522\(82\)90277-9](https://doi.org/10.1016/0306-4522(82)90277-9).  
URL <http://www.sciencedirect.com/science/article/pii/0306452282902779>
- [24] Zvoncak, V.; Mekyska, J.; Safarova, K.; aj.: New Approach of Dysgraphic Handwriting Analysis Based on the Tunable Q-Factor Wavelet Transform. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, s. 289–294, doi: 10.23919/MIPRO.2019.8756872.
- [25] Zvoncak, V.; Mucha, J.; Galaz, Z.; aj.: Fractional Order Derivatives Evaluation in Computerized Assessment of Handwriting Difficulties in School-aged Children. In *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2019, s. 1–6, doi: 10.1109/ICUMT48472.2019.8970811.





## Seznam symbolů a zkratek

<b>WPF</b>	Windows Presentation Foundation
<b>GUI</b>	Graphical User Interface
<b>LINQ</b>	Language-Integrated Query
<b>XML</b>	Extensible Markup Language
<b>SHP</b>	Subjektivní hodnocení speciálním pedagogem



# Seznam příloh

A Obsah přiloženého adresáře

91



## A Obsah přiloženého adresáře

V přiloženém .zip souboru se nachází zdrojový kód všech vyvinutých programů. Aplikace Skořápka se nachází samostatně v adresáři ScoreApp. Programy pro validaci dat a extrakce parametrů jsou strukturovány jako dva projekty v jednom solution. Otevření celého solution je možné pomocí souboru "Dys-LabelCheck.sln", k čemuž je potřeba mít k dispozici vývojové prostředí s platformou .NET Core 3.1.

Ve výpisu nejsou uvedeny naprosto všechny soubory, neboť vzhledem k členění zdrojového kódu aplikací, nemají některé soubory samostatnou funkci.

```
/ ..... Kořenový adresář příloh
├── Dys-LabelCheck ..... Obsah složky Dys-LabelCheck
│   ├── Dys-LabelCheck ..... Program pro validaci dat
│   │   ├── ModelsLocal ..... Soubory modelů SQL databáze
│   │   ├── Protocols ..... Obrázky předloh jednotlivých cvičení
│   │   ├── App.xaml
│   │   ├── App.xaml.cs ..... Hlavní soubor se zdrojovým kódem
│   │   ├── Dys-LabelCheck.csproj
│   │   ├── MainWindow.xaml ..... Nastavení a rozložení GUI
│   │   └── MainWindow.xaml.cs ..... Zdrojový kód ovládající GUI
│   ├── Dys-ParameterExtractor ..... Program pro extrakci paramaterů
│   │   ├── Models ..... Soubory modelů SQL databáze
│   │   ├── Requests
│   │   │   └── ParametersRequests ..... Kód pro vkládání do databáze
│   │   ├── Dys-ParameterExtractor.csproj
│   │   ├── Extensions.cs
│   │   ├── PairComp.cs
│   │   ├── Parameters.cs
│   │   ├── Program.cs
│   │   └── StrokeProcessor.cs
│   └── Dys-LabelCheck.sln
├── ScoreApp ..... Obsah složky ScoreApp
│   ├── ScoreApp ..... Program pro validaci dat
│   │   ├── Protocols ..... Předlohy cvičení ve formátu jpg a svg
│   │   ├── App.xaml
│   │   ├── App.xaml.cs ..... Hlavní soubor se zdrojovým kódem
│   │   ├── ScoreApp.csproj
│   │   ├── MainWindow.xaml ..... Nastavení a rozložení GUI
│   │   └── MainWindow.xaml.cs ..... Zdrojový kód ovládající GUI
│   └── ScoreApp.sln
```