

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Návrhový modul pro výpočet pevnosti šroubovaných spojů**

Místo této strany bude  
zadání práce.

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 2. května 2017

Patrik Patera

V bakalářské práci jsou použity názvy programových produktů, firem, apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

# Poděkování

Rád bych poděkoval Ing. Kamilu Ekšteinovi, Ph.D. za ochotu a pomoc při vedení práce. Dále Tomáši Potůčkovi za design ikon.

## **Abstract**

The main objective of the thesis is create an application such as a module to design and visualise of the geometry bolted joints for design engineers on Department of Machine Design at the Faculty of Mechanical Engineering at University of West Bohemia in Pilsen. The application will be based on a simple CAD system. The first part describes common features and components of choosen CAD systems. The second part is focused on description of bolted joints, requirements of the application and design of the application. The third part describes implementation where the reader is familiar with the algoirhms and architecture of the application. The final part contains testing of the application, description of test methods and a summary after testing.

## **Abstrakt**

Cílem této bakalářské práce je vytvořit aplikaci, která bude fungovat jako modul pro vizuální návrh geometrie šroubovaných spojů pro konstruktéry Katedry konstruování strojů Fakulty strojní Západočeské univerzity v Plzni. Výsledná aplikace se bude chovat jako jednoduchý CAD systém. V první části jsou popsány společné vlastnosti a komponenty vybraných CAD systémů. Druhá část je zaměřena na popis šroubovaných spojů, požadavky na aplikaci a celkový návrh aplikace. V třetí části je popsána implementace, aby se čtenář seznámil s algoritmy a celkovou architekturou aplikace. Poslední část je zaměřena na testování aplikace, kde je popsán způsob testování a následné shrnutí výsledků z testování.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>11</b>
<b>2</b>	<b>CAD systémy</b>	<b>12</b>
2.1	Komerční systémy . . . . .	12
2.1.1	AutoCAD . . . . .	13
2.1.2	CATIA . . . . .	14
2.2	Open-source systémy . . . . .	14
2.2.1	QCAD . . . . .	15
2.2.2	FreeCAD . . . . .	15
2.3	Společné funkce . . . . .	17
2.4	Obecné ovládání . . . . .	18
2.4.1	Manipulace s plátnem . . . . .	18
2.4.2	Manipulace s objekty . . . . .	19
<b>3</b>	<b>Návrh</b>	<b>20</b>
3.1	Popis šroubovaného spoje . . . . .	20
3.2	Požadavky na aplikaci . . . . .	21
3.3	Framework . . . . .	22
3.3.1	JavaFX . . . . .	22
3.3.2	Qt . . . . .	23
3.4	Geometrická primitiva . . . . .	23
3.4.1	Bod . . . . .	24
3.4.2	Přímka . . . . .	24
3.4.3	Úsečka . . . . .	25
3.4.4	Kružnice . . . . .	26
3.4.5	Kruh . . . . .	27
3.4.6	Kruhová výseč . . . . .	28
3.4.7	Polygon . . . . .	29
3.5	Spojovací součástky . . . . .	31
3.5.1	Šroub . . . . .	31
3.5.2	Kolík . . . . .	32
3.5.3	Péro . . . . .	32
3.5.4	Zátěžová plocha . . . . .	33
3.6	Grafické uživatelské rozhraní . . . . .	34
3.6.1	Rozložení komponent . . . . .	34
3.6.2	Strom objektů . . . . .	34

3.6.3	Panel vlastností . . . . .	35
3.6.4	Dialogová okna . . . . .	36
3.7	Výstupní formát dat . . . . .	36
3.7.1	Struktura objektu . . . . .	36
3.7.2	Struktura skupiny objektů . . . . .	37
3.7.3	Struktura plátna . . . . .	37
3.7.4	Dodatečné vlastnosti objektu . . . . .	37
3.7.5	XSL Transformace . . . . .	37
<b>4</b>	<b>Implementace</b>	<b>39</b>
4.1	Architektura MVC . . . . .	39
4.2	Aplikační logika . . . . .	40
4.2.1	PCanvasController . . . . .	41
4.2.2	PTabPageController . . . . .	41
4.2.3	PToolController . . . . .	42
4.2.4	PInfoPaneController . . . . .	43
4.2.5	PWindowController . . . . .	43
4.3	Modely . . . . .	43
4.3.1	PArc . . . . .	44
4.3.2	PCircle . . . . .	46
4.3.3	PPolygon . . . . .	46
4.3.4	PComposite . . . . .	48
4.4	Nástroje . . . . .	48
4.4.1	Rotace . . . . .	48
4.4.2	Vracení změn . . . . .	49
4.4.3	Kopírování objektu . . . . .	49
4.4.4	Označení objektu . . . . .	50
<b>5</b>	<b>Testování</b>	<b>51</b>
5.1	Použitelnost v praxi . . . . .	51
5.2	Testování podle scénáře . . . . .	52
5.2.1	Funkčnost . . . . .	53
5.2.2	Ovladatelnost a manipulace s objekty . . . . .	55
5.2.3	Kontrola uložení/načtení objektů a plátna . . . . .	56
5.2.4	Modelování návrhu spoje . . . . .	57
5.3	Monkey testování . . . . .	58
5.4	Shrnutí . . . . .	58
<b>6</b>	<b>Závěr</b>	<b>60</b>
	<b>Literatura</b>	<b>61</b>

<b>A</b>	<b>Uživatelská příručka</b>	<b>63</b>
A.1	Popis GUI . . . . .	63
A.1.1	Panel s geometrickými primitivy . . . . .	64
A.1.2	Panel se součástkami . . . . .	64
A.1.3	Panel s nástroji . . . . .	64
A.1.4	Panel s menu . . . . .	64
A.1.5	Panel s plátnem . . . . .	64
A.1.6	Panel s umístěnými objekty . . . . .	64
A.1.7	Panel s vlastnostmi objektu . . . . .	65
A.1.8	Informační panel . . . . .	65
A.1.9	Dialogové okno k určení objektu . . . . .	65
A.1.10	Klávesové zkratky . . . . .	66
<b>B</b>	<b>Záznamy z testování</b>	<b>67</b>
B.1	Uživatel 1 . . . . .	67
B.1.1	Funkčnost . . . . .	67
B.1.2	Ovladatelnost a manipulace s objekty . . . . .	69
B.1.3	Kontrola uložení/načtení objektů a plátna . . . . .	71
B.1.4	Modelování návrhu spoje . . . . .	72
B.2	Monkey testování . . . . .	72
B.3	Uživatel 2 . . . . .	73
B.3.1	Funkčnost . . . . .	73
B.3.2	Ovladatelnost a manipulace s objekty . . . . .	75
B.3.3	Kontrola uložení/načtení objektů a plátna . . . . .	77
B.3.4	Modelování návrhu spoje . . . . .	78
B.4	Monkey testování . . . . .	78



# Seznam obrázků

2.1	GUI AutoCAD 2016 . . . . .	13
2.2	GUI CATIA (Zdroj: <a href="http://engineering.com/">http://engineering.com/</a> ) . . . . .	14
2.3	GUI QCAD . . . . .	15
2.4	GUI FreeCAD . . . . .	16
3.1	Šroubovaný spoj . . . . .	20
3.2	Umístění součástky . . . . .	22
3.3	Znázornění bodu . . . . .	24
3.4	Přímka . . . . .	24
3.5	Úsečka . . . . .	25
3.6	Kružnice . . . . .	26
3.7	Vlastnosti kružnice . . . . .	27
3.8	Kruh . . . . .	28
3.9	Kruhová výseč . . . . .	28
3.10	Porovnání polygonu a lomené čáry . . . . .	29
3.11	Porovnání polygonů . . . . .	30
3.12	Nákres šroubu (zdroj: <a href="http://www.ebolt.co.uk/">http://www.ebolt.co.uk/</a> ) . . . . .	31
3.13	Nákres kolíku (zdroj: <a href="http://www.konarik.cz/">http://www.konarik.cz/</a> ) . . . . .	32
3.14	Nákres těsného péra . . . . .	32
3.15	Nákres obdélníkové zátěžové plochy . . . . .	33
3.16	Návrh GUI . . . . .	35
3.17	Stromová struktura . . . . .	35
3.18	XSL Transformace . . . . .	38
4.1	Diagram MVC . . . . .	40
4.2	Diagram tříd ovladačů . . . . .	40
4.3	Vazby mezi ovladači . . . . .	41
4.4	Závislost mezi ovladačem a plátnem . . . . .	41
4.5	Vazby mezi tlačítky a ovladačem . . . . .	42
4.6	Panel s vlastnostmi objektu . . . . .	43
4.7	Informační panel . . . . .	43
4.8	Diagram tříd těles . . . . .	44
4.9	Test bodu v kruhové výseči . . . . .	46
4.10	Test bodu v polygonu . . . . .	47
5.1	GUI aplikace . . . . .	52
5.2	Vzor šroubovaného spoje . . . . .	57

A.1 GUI aplikace . . . . .	63
A.2 Výběr objektu . . . . .	65

# 1 Úvod

Konstruktéři z Katedry konstruování strojů (KKS) Fakulty strojní (FST) na Západočeské univerzitě v Plzni (ZČU) používají k návrhu šroubovaných spojů zastaralý program, který již nevyhovuje současným požadavkům. K modelování složitějších geometrických modelů používají různé CAD systémy, které jsou v oblasti modelování (2D i 3D) prakticky nenahraditelnými nástroji, avšak pro návrh šroubovaných spojů jsou příliš komplexní a nemají možnost výsledný návrh uložit ve formátu, který konstruktéři z KKS potřebují pro výpočetní program, který provádí výpočet pevnosti šroubovaných spojů.

Cílem této bakalářské práce je prozkoumat a seznámit se s technikami práce běžně používaných CAD systémů. Na základě těchto poznatků posléze navrhnout a implementovat aplikaci, která bude fungovat jako návrhový modul šroubovaných spojů s podobnými vlastnostmi, jež nabízí běžně používané CAD systémy. Výsledná aplikace bude obsahovat vybraná geometrická primitiva, která budou sloužit jako základní prvky pro návrh jednotlivých součástí šroubovaných spojů, a součástky, které konstruktéři běžně používají ke spojování strojních součástí.

Dále bude zapotřebí navrhnout vhodný výstupní formát navrženého šroubovaného spoje tak, aby byl dále použitelný pro výpočetní program.

## 2 CAD systémy

CAD (Computer-Aided Design) neboli počítačem podporované kreslení se obecně jedná o komplexní systémy, které jsou určeny k modelování geometrických modelů v mnoha oblastech. Největší zastoupení je především v těchto oblastí:

- Strojírenství
- Elektrotechnika
- Stavební inženýrství
- Textilní průmysl
- Zdravotnictví

V současné době existuje nesmírné množství CAD systémů, které jsou schopné vytvářet 2D (dvourozměrný) i 3D (trojrozměrný) geometrické modely v závislosti na konkrétní oblasti, pro kterou má být model zhotoven. Většinou je však možné mezi náhledy přepínat, kde 2D náhled se nejvíc uplatní pro úpravu 3D modelu v libovolném průmětu (jedná se např. o nárýs, půdorys či bokorys) pro jednodušší manipulaci s modelem.

CAD systémy neslouží pouze k modelování, ale většina z nich umožňuje i následnou simulaci s vymodelovanými objekty na základě podmínek, které mohou nastat v reálném světě. Pod simulací si můžeme představit např. náraz automobilu do různého typu materiálu pro demonstraci jeho deformace, sledování proudění vzduchu kolem letadla a mnoho dalšího. Virtuální simulace je hlavně jakýsi prototyp, který šetří čas a peníze, před provedením reálné simulace, avšak některé simulace (FEM/CFD) mohou být ve větší shodě s realitou.

Dále se v této kapitole budu zabývat několika CAD systémy zaměřenými především na modelování ve strojírenství. Nalezneme zde jejich obecný popis, společné funkce a postupy ovládání.

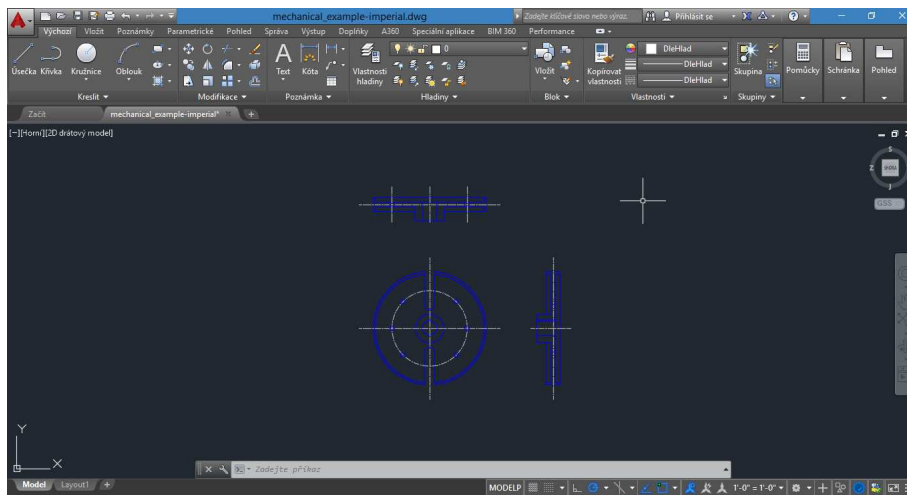
### 2.1 Komerční systémy

Komerční CAD systémy jsou převážně využívané společnostmi, které vyžadují vysokou kvalitu výsledného modelu. Licence těchto systémů jsou zpoplatněny a jejich cena se pohybuje v řádech tisíců až statisíců korun, podle

jejich využití. Jedny z předních společností jsou SolidWorks, CATIA, Siemens NX, Autodesk, Inc a Pro/ENGINEER. Všechny uvedené CAD systémy jsou si velice podobné jak rozmístěním jednotlivých komponent, tak funkcemi, které nabízejí, a proto zde bude popis pouze dvou z nich.

### 2.1.1 AutoCAD

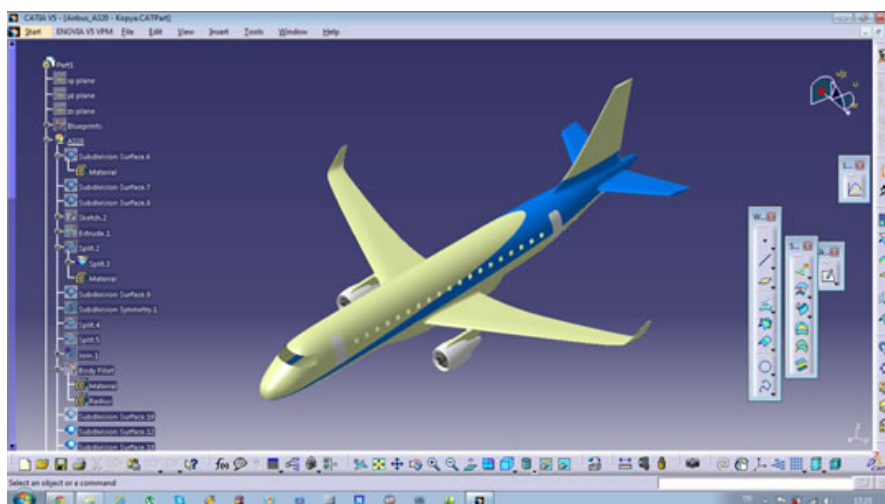
AutoCAD [2] je jedním z produktů firmy Autodesk, Inc. Je to profesionální software určený pro modelování (ve 2D i 3D) v mnoha oblastech. Vývoj je datován od roku 1977, kdy se v jmenoval Interact CAD. Tento CAD systém podporuje mnoho světových jazyků včetně češtiny a je poskytován ve více variantách, např. jako studentská verze určená pro studenty nebo AutoCAD LT, což je zjednodušená verze, která je finančně dostupnější a pro menší projekty zcela postačující. Jeho základní výstupní formát je DWG (což je nativní formát výkresů programu AutoCAD), popřípadě DXF (formát výkresů, který byl vyvinut firmou Autodesk a umožňuje výměnu mezi AutoCADem a jinými programy). Dále je možnost exportu do jiných běžně používaných formátů. AutoCAD je především vyvíjen pro operační systém Microsoft® Windows™. Poslední verze produktu dostupná pro operační systém MacOSX® je z roku 2016. Grafické uživatelské rozhraní (GUI – *Graphic User Interface*) AutoCADu 2016 s jednoduchým nákresem si je možné prohlédnout na obrázku 2.1.



Obrázek 2.1: GUI AutoCAD 2016

### 2.1.2 CATIA

CATIA [5] (*Computer-Aided Three-Dimensional Interactive Application*) je další software z řady CAD systémů pro modelování ve 2D i 3D. Je vyvíjen francouzskou firmou Dassault Systèmes působící na trhu od roku 1977. Využití nalezneme zejména v oblasti letectví a automobilovém průmyslu. Je používán předními společnostmi jakými je např. The Boeing Company, Airbus, BMW, Porsche, ale i nám velice dobře známá Škoda Auto. Základním výstupním formátem je CATpart, popřípadě CATproduct, ale i mnoho dalších běžně používaných formátů. Dostupný je pro operační systém Microsoft® Windows™, ale i pro většinu UNIXových systémů. Grafické uživatelské rozhraní CATIA s návrhem Airbusu (dopravního letadla) můžeme vidět na obrázku 2.2.



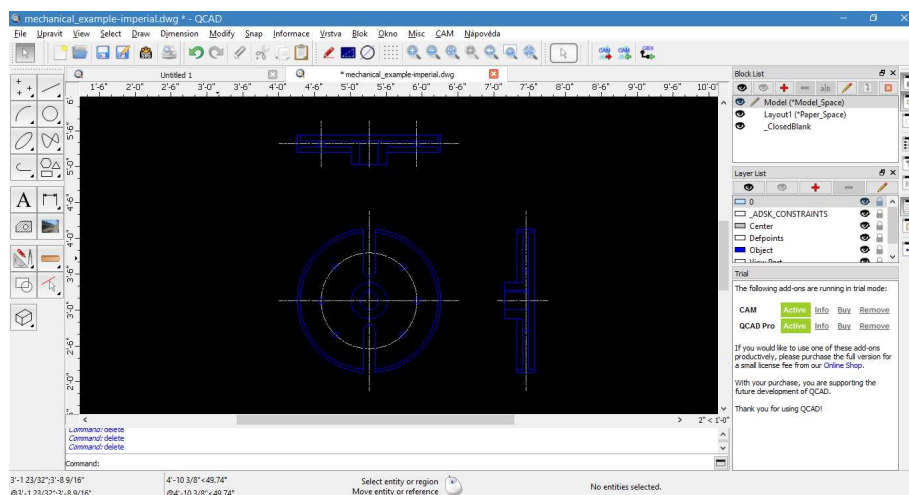
Obrázek 2.2: GUI CATIA (Zdroj: <http://engineering.com/>)

## 2.2 Open-source systémy

Existuje i řada volně dostupných CAD systémů, a to i s otevřeným zdrojovým kódem, které se svými funkcemi a ovládáním velice podobají profesionálním systémům z části 2.1. Jejich výhodou je, že jsou zdarma k použití, ale i právě díky otevřenému zdrojovému kódu se na vývoji může podílet širší komunita lidí, a tak vznikají nové nástroje a vylepšení pod podmínkou, že jejich zdrojový kód bude opět volně dostupný. Jsou využívány nejen pro soukromé účely, ale také i pro komerční (jen zde nemáme zajištěnou profesionální funkčnost a kvalitu).

## 2.2.1 QCAD

QCAD je volně dostupný 2D CAD systém s otevřeným zdrojovým kódem, který je především určen pro technické výkresy budov, interiérů, mechanických součástí a různých schémat. Je dostupný pro operační systémy Microsoft® Windows™, MacOS®X a Linux. Zdrojový kód je vydán pod licencí GPL verze 3 (GPLv3). Jeho předností je minimalistický vzhled a intuitivní ovládání, kde i neznalý uživatel je schopen vytvořit jednoduchý výkres. Podporuje formát DWG i DXF, stejně jako komerční produkty, a mnoho dalších používaných formátů. Existuje i jeho placená verze nazvaná QCAD Professional, která stojí řádově desítky eur a obsahuje vylepšené funkce oproti standardní volně dostupné verzi. Příklad grafického uživatelského rozhraní s výkresem je k nahlédnutí na obrázku 2.3.

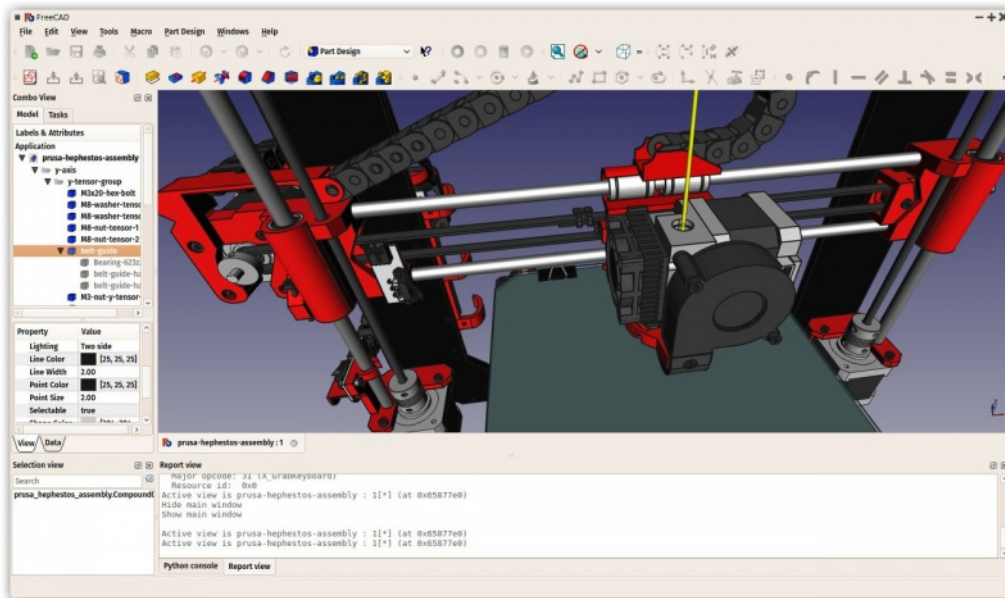


Obrázek 2.3: GUI QCAD

## 2.2.2 FreeCAD

FreeCAD je volně dostupný 3D CAD systém s otevřeným zdrojovým kódem. Je zaměřen přímo pro strojírenskou oblast, ale jeho uplatnění nalezneme i např. v architektuře a navrhování interiérů budov. Vyvíjen je pro operační systémy Microsoft® Windows™, MacOS®X a Linux. Zdrojový kód je chráněn licencí LGPLv2. Vzhled působí minimalisticky, ale skrývá mnoho užitečných funkcí, které jsou chytře skryty v různých módech modelování, a tak není menu s nástroji zbytečně přeplněno zrovna nepotřebnými funkcemi. Ovládání je vcelku jednoduché, až na poněkud nezvyklý posun po plátně (např. rotace je možná stiskem prostředního a pravého tlačítka myši), ale

jinak se ovládá podobně jako většina jiných CAD systémů. FreeCAD podporuje formát DWG i DWF a export do dalších běžně používaných formátů. Na obrázku 2.4 je možné vidět grafické uživatelské rozhraní spolu s modelem 3D tiskárny.



Obrázek 2.4: GUI FreeCAD



## 2.3 Společné funkce

V této části jsou rozebrány zejména důležité funkce již zmíněných CAD systémů v této kapitole. Přednostně jsem se zaměřil na funkce, které by mohly být užitečné a nepostradatelné v rámci požadované aplikace a jsou převážně určené pro modelování ve 2D. Výrazem „plátno“, který bude dále používán, je myšlen virtuální prostor CAD systémů pro vizuální modelování objektů. Operace typu výběr objektu, úprava rozměrů, pohyb po plátně, apod., beru jako základní funkce, které jsou více méně samozřejmostí u tohoto typu programů.

- **Mřížka** – Je užitečná funkce při umisťování objektu na plátno pomocí myši. Při aktivní mřížce jsou na plátně vykresleny body s danou konstantní vzdáleností (zadanou manuálně nebo automaticky vypočtenou v závislosti na měřítku). Pokud se k nějakému z těchto bodů přiblížíme, pak se kurzor myši k bodu přichytí a je tak jednodušší umístit objekt na konkrétní pozici.
- **Parametrické zadávání hodnot** – Slouží především k umístění objektu na přesně dané souřadnice. V závislosti na daném objektu je možné měnit jeho vlastnosti, kterými mohou být např. středový bod, výška, šířka, apod.
- **Rotace a posun** – Jak již plyne z názvu, jde o možnost rotovat či posouvat zvolený objekt, jak pomocí kurzoru myši, tak zadáním přesných souřadnic. Pro rotaci je možné zvolit i referenční bod jako střed rotace (kolem tohoto bodu bude zvolený objekt rotovat), protože není vždy žádoucí, aby objekt rotoval pouze kolem svého středového bodu; může rotovat např. kolem středu jiného objektu.
- **Uchycování k hlavním osám** – Je podobná funkce jako mřížka, ale s tím rozdílem, že při umisťování objektu je kurzor myši uchycen k přímkě, která je rovnoběžná s osou  $x$  nebo  $y$ . To se hodí zejména, pokud chceme zvolený objektu posouvat pouze po zvolené ose.
- **Měření vzdálenosti** – Slouží jako jakési dynamické pravítko, kterým jsme schopni změřit vzdálenost mezi dvěma zvolenými body (např. jak daleko od sebe jsou středy některých objektů).
- **Vracení změn** – Pomocí určených tlačítek v menu (či využitím klávesové zkratky **Ctrl + Z**) je možné vracet vykonané změny do předchozího stavu.

- **Booleovské operace:** Tři základní operace, které slouží k snadnějšímu modelování složitějších objektů.
  - **Sjednocení** – Pomocí této operace jsme schopni sjednotit vybrané objekty, které mají společné body, do jednoho celku. Výsledný objekt se posléze chová jako jeden objekt a veškeré další transformace se aplikují na všechny jeho části.
  - **Průnik** – Průnikem dvou objektů je možné dostat takovou část, kterou mají objekty společnou. Podmínkou tedy musí být, aby zvolené objekty měly společné body – alespoň dva.
  - **Oříznutí** – Oříznutí (rozdíl) dvou objektů je operace, která ořízne část z jednoho objektu. Jejich společná oblast bude z objektu vyříznuta (podmínkou je, že objekty mají společné body – alespoň dva). U této operace závisí na pořadí výběru objektů (např. první vybraný objekt bude ten, z kterého se část vyřízne).
- **Seznam objektů** – Některé CAD systémy mají speciální komponentu, která uchovává v seznamu (popř. stromu) dosud vložené objekty. Díky této funkci jsme schopni rychleji zvolit požadovaný objekt nebo ho rovnou modifikovat. Je zde i možnost si objekty pojmenovat dle libosti a to zvyšuje přehlednost vložených objektů.

## 2.4 Obecné ovládání

Jednotlivé funkce je možné ovládat jak kurzorem myši, tak i různými klávesovými zkratkami, které ulehčují a hlavně urychlují proces modelování. Bude zde uvedeno několik základních ovládacích prvků, které jsou běžné v zmíněných CAD systémech a zejména využitelné pro 2D. U některých CAD systémů je možné akce vykonávat pomocí vkládání příkazů do integrované příkazové řádky. Příkazy se liší podle toho, který CAD systém zrovna používáme a většinou bývají k nalezení na webových stránkách nebo v manuálu programu.

### 2.4.1 Manipulace s plátnem

Nové plátno můžeme vytvořit dvěma způsoby. První je kliknutím na příslušné tlačítko umístěné v menu aplikace (většinou skryté pod tlačítkem *Soubor*) nebo stisknutím kombinace tlačítek **Ctrl + N**.

Pohybovat se můžeme i po ploše plátna, a to při stisku prostředního tlačítka myši nebo kombinací klávesy **Ctrl + levého tlačítka myši**. Takováto

akce je většinou doprovázena změnou kurzoru na takzvanou *uzavřenou ruku* nebo *osový kříž*. To nám dává najevo, že následným tažením myši se budeme pohybovat po ploše plátna.

Rolováním kolečkem myši je intuitivně docíleno škálování plátna. Díky tomu můžeme měnit aktuální měřítko a tím jsme schopni manipulovat s objekty, které mají velké, respektive malé rozměry oproti ostatním objektům.

## 2.4.2 Manipulace s objekty

Při průniku kurzoru s objektem na plátně je pomocí levého tlačítka myši možné daný objekt označit a následně s ním provádět různé operace. Posun objektu lze většinou vykonat tažením jeho středového bodu na jiné místo. Kolem objektu bývají také body, kterými můžeme pomocí myši manipulovat tak, že měníme jejich pozici a objekt tím transformujeme (např. měníme jeho šířku, výšku či upravujeme jeho tvar).

Kromě posunu a škálování označeného objektu je samozřejmě možné ho z plátna smazat pomocí klávesy *Delete*. Dalším možným způsobem smazání je vyvolání kontextového menu pravým tlačítkem myši a výběr příslušné položky pro smazání (označeno popiskem *Smazat*, *Vymazat* či *Odstranit*).

Pokud to daný objekt dovoluje, jsme schopni mu přidávat a odebírat body. Tím měníme tvar (popř. šířku, výšku, apod.) zvoleného objektu. Tato akce je možná při přidržení tlačítka *Ctrl* (pro přidání) nebo *Shift* (pro odebrání) a následném stisku levého tlačítka myši. Odebírat či přidávat body můžeme v závislosti na pravidlech určených objektem (např. polygon musí obsahovat nejméně 3 body, atd.).

Objekty je kromě mazání také možné uložit do souboru pro jejich budoucí použití. Pomocí klávesové zkratky *Ctrl + Shift + S* máme možnost uložit celé plátno se všemi objekty. Operace se nazývá *Uložit jako* a pomocí následného dialogu provedeme uložení plátna pod vlastním názvem (cestou) a případně i formátem souboru. Pokud chceme průběžně stav plátna ukládat, stačí nám stisk kombinace kláves *Ctrl + S*, která zaručí uložení pod již zvoleným názvem (cestou) – pokud existuje; jinak se objeví stejné dialogové okno jako při akci *Uložit jako*. Obě operace je možné provádět i kurzorem myši z nabídky menu nazvané *Soubor*.

Import objektů ze souboru na plátno se provádí pomocí kombinace kláves *Ctrl + I* nebo *Ctrl + Shift + I*. V následném dialogovém okně vybereme potřebný soubor obsahující objekt a ten se vloží na právě aktivní plátno. To samé platí i pro import celého plátna jen s rozdílem, že se vytvoří nová záložka pro importované plátno.

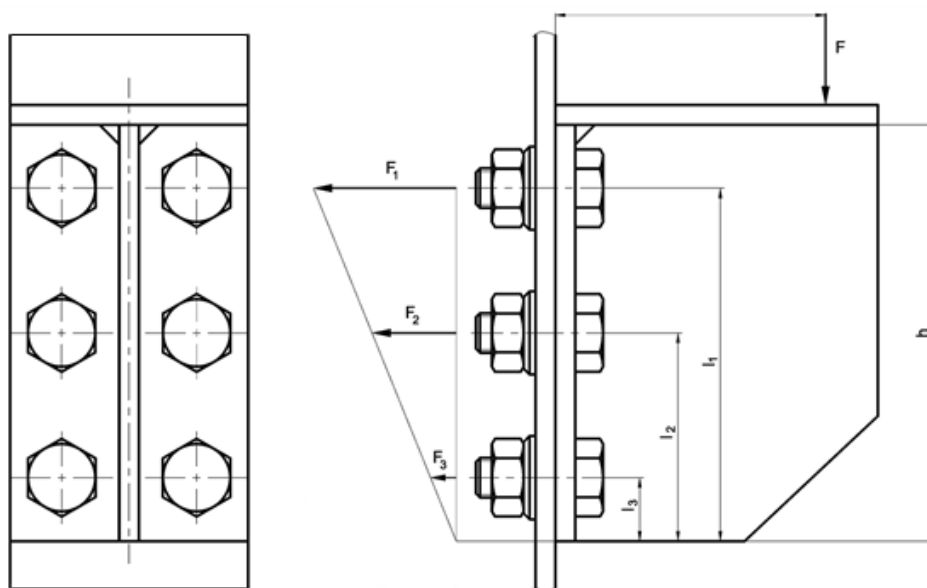
# 3 Návrh

## 3.1 Popis šroubovaného spoje

Šroubované spoje [4] jsou jedním z nejčastějších způsobů spojování konstrukčních součástí. Hlavní spojovací součástí pro šroubované spoje je šroub. Existují však i další součástky, kterými můžeme podpořit rozložení působících sil na šroubovaný spoj, a těmi mohou být: kolík, péro, zátěžová plocha (ta sice není jako spojovací prvek, ale působí na plochu určitou silou, aby nedocházelo např. ke smyku desek). Více o součástkách se dozvíme v 3.5.

U těchto typů spojů – šroubovaných – hovoříme o tzv. *rozzebíratelných* spojích, které je možné snadno a bez poškození spojovacích součástí rozebrat a opět je spojit dohromady. Pro *nerozzebíratelné* spoje platí, že pokud bychom spojovací části chtěli od sebe rozdělit, pak trvale deformujeme jejich plochu (spojení těchto částí dosáhneme např. svařování, pájením nebo lepením).

Příklad spoje je možné vidět na obrázku 3.1 (vlevo se nachází půdorys a vpravo bokorys).



Obrázek 3.1: Šroubovaný spoj

## 3.2 Požadavky na aplikaci

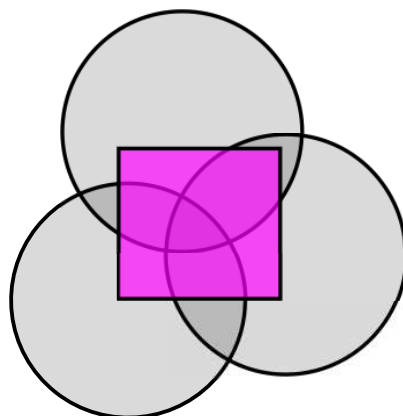
Výsledná aplikace by se měla chovat jako jednoduchý 2D CAD systém, který bude umět umisťovat na plátno základní objekty (geometrická primitiva) pomocí kurzoru myši, ale i parametrickým zadáním přesných souřadnic středu objektu. S těmito objekty bude možné, i po umístění na plátno, dále manipulovat (přesouvat, upravovat rozměry a měnit jejich tvar) v závislosti na typu konkrétního objektu. Ze základních objektů je možné sestavit kompozitum (skupinu objektů), tedy komplexní objekt složený z více primitiv najednou. Šroubovaný spoj bude navrhován v půdorysu (tedy jako bychom se na spoj dívali shora).

Kromě geometrických primitiv musí být možné vytvářet i spojovací součástky (dále jen součástky) – šroub, kolík, péro a obecnou zátěžovou plochu, více bude popsáno v části 3.5. Součástky nebude možné samovolně přidat na plátno, ale musí splňovat určité náležitosti (požadavky), aby byly považovány za validní a umístěny na plátno. Požadavky pro správné umístění jsou:

- Součástku lze umístit pouze celým svým obsahem na plochu jiného objektu. Je-li na plátně více objektů, které mají různé společné body, pak je možné, aby součástka byla umístěna i do oblasti jejich průniků, nesmí však svou částí vyčnívat mimo oblast, kterou tyto objekty vytvářejí a všechny objekty musí být součástí jedné skupiny, jak je patrné z obrázku 3.2 (součástka zvýrazněna).
- Součástku nelze umístit na jinou součástku (např. šroub na šroub nelze).
- Součástka nelze umístit na díru v jiném objektu.
- Jakýkoliv jiný objekt přidaný na součástku bude umístěn po danou součástkou.
- Díru nelze vyříznout na místě, kde se již nachází jakákoliv součástka.

Nad objekty by mělo být možné provádět operaci sloučení (bez ní by nemohla vzniknout skupina). Sloučení objektů lze provést pouze tehdy, pokud mají objekty společný alespoň jeden bod nebo je jeden uvnitř druhého.

Aplikace bude schopna uložit plátno s objekty do souboru pro budoucí zpracování výpočetním softwarem k tomu určeným.



Obrázek 3.2: Umístění součástky

## 3.3 Framework

Před samotnou implementací aplikace bylo nutné provést výběr frameworku, který poskytne potřebnou funkcionalitu pro vývoj aplikace s ohledem na požadavky aplikace (vizte 3.2). Stručně zde popíšu vybrané kandidáty – *Qt* a *JavaFX*. Oba tyto frameworky nabízejí veškerou potřebnou funkčnost k vývoji aplikace typu CAD systém, ale díky nabitým zkušenostem s frameworkem *JavaFX*, u kterého mohu nejvíce uplatnit veškeré své dovednosti, jsem se rozhodl zvolit ho k implementaci této aplikace.

### 3.3.1 JavaFX

*JavaFX* [15] je open-source framework napsaný v jazyce *Java* a slouží pro vývoj grafických uživatelských aplikací. Díky jazyku *Java* je zajištěna přenositelnost mezi operačními systémy pomocí *JVM* [10] (*Java Virtual Machine*). Je považován za nástupce zastaralé knihovny *Swing* [9]. Knihovna je k dispozici jako veřejné rozhraní (API) pro programování aplikací. Je možné využít všechny výhody, který jazyk *Java* přináší, jako např. použití *multithreadingu* (využití více vláken), *generičnost* a *lambda výrazy*. Uživatelské rozhraní (UI) může být napsáno přímo v jazyce *Java* nebo za použití *FXML* (založeno na značkovacím jazyce *XML*, kde je definováno UI deklarativně). Často se však využívá kombinace obou variant (GUI v *FXML* a aplikační logika v *Javě*). *JavaFX* není žádnou novinkou, vývoj je datován od roku 2008, ale prošel výraznými změnami a v současné době je vydán ve verzi 8.0 s mnoha vylepšeními a širší paletou komponent.

*JavaFX* nabízí řadu základních komponent, jako např. tlačítka (různého typu), textová pole, popisky, apod., ale také sofistikovanější komponenty, které jsou např.: seznamy, stromy, tabulky, *HTML* editory, atd. Obsahuje

i komponentu plátno, která je podle [15] nejlepším možným řešením pro kreslení různých tvarů (i po jednotlivých pixelech). K úpravě vzhledu jednotlivých komponent lze využít i kaskádové styly (CSS). Nabízí klasickou obsluhu událostí pomocí myši, ale také podporuje gesta, což je vhodné při obsluze aplikace na dotykovém zařízení. K jednoduchému umístění jednotlivých komponent je dostupný nástroj *Scene Builder*, který funguje na principu *Drag & Drop* (táhni a pusť). Komponentu jednoduše umístíme v návrháři na požadované místo a kód se automaticky vygeneruje.

### 3.3.2 Qt

Qt [11] je multiplatformní aplikační framework určený k vývoji pro stolní počítače, vestavěné systémy a mobilní zařízení. Je napsán v jazyce C++ a je dostupný pod licencí GPL, LGPL a komerční licencí. Využívá preprocessor MOC (*Meta-Object Compiler*), který rozšiřuje jazyk C++ o vlastnost *Signals & Slots* (sloužící pro komunikaci mezi objekty). MOC nejdříve zpracuje zdrojový kód a následně vygeneruje přeložitelný C++ kód.

Uživatelské rozhraní může být napsáno přímo v jazyce C++, a to za pomoci *Widgets*. Nabízí i možnost navrhnout UI pomocí interaktivního grafického nástroje *Qt Designer*, který generuje XML s umístěnými komponenty.

Dalším způsobem, kterým lze napsat UI, je využití *QtQuick* modulů. Uživatelské rozhraní je psáno v QML (*Qt Meta Language* nebo *Qt Modeling Language*), což je deklarativní jazyk s integrovaným JavaScriptem a CSS. Celé UI je možné napsat přímo v QML nebo kombinaci s C++ (GUI napsáno v QML a aplikační logika v C++ – častější varianta).

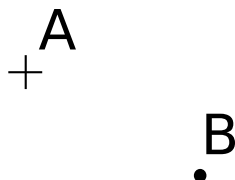
Qt se nevyužívá pouze jako GUI nástroj, ale také nabízí spoustu dalších modulů pro síťování, databáze, OpenGL (3D grafická knihovna), webové technologie, komunikační protokoly a mnohem víc.

## 3.4 Geometrická primitiva

V této části jsou popsána jednotlivá geometrická primitiva, která jsou nezbytná k zkonstruování složitějších tvarů, jež jsou potřeba pro vytvoření šroubovaného spoje. Všechny níže popsané geometrické útvary jsou probírané pouze v rovině a za použití kartézské souřadné soustavy, což je plně postačující pro popis šroubovaného spoje. Veškeré zde uvedené vzorce a definice jsou čerpány z knihy [3].

### 3.4.1 Bod

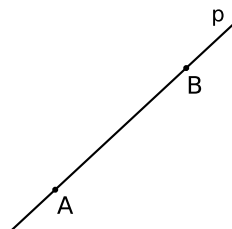
Bod patří mezi základní geometrické útvary. Lze ho definovat jako bezrozměrný útvar, který může být specifikovaný v  $n$ -rozměrných prostorech (potom obsahuje právě  $n$  souřadnic  $x_1, x_2, \dots, x_n$ ). Bod označujeme velkým písmenem a graficky ho značíme např. křížkem či kroužkem, jak je uvedeno na obrázku 3.3.



Obrázek 3.3: Znázornění bodu

### 3.4.2 Přímka

Přímku označujeme malými latinskými písmeny. Máme-li dva různé body  $A$  a  $B$ , pak pomocí nich můžeme vytvořit přímku  $\overleftrightarrow{AB}$ , která je nekonečně dlouhá a prochází právě těmito dvěma body (vizte obrázek 3.4).



Obrázek 3.4: Přímka

Rovinnou přímku lze popsat několika možnými způsoby, avšak vždy se jedná o lineární rovnici. Rovnice 3.1 popisuje přímku v takzvaném směrnicovém tvaru:

$$y = kx + q, \quad (3.1)$$

kde  $k = \tan \varphi$  je směrnice přímky ( $\varphi$  je orientovaný úhel mající vrchol v počátku souřadnicové soustavy) a  $q$  je tzv. úsek na ose  $y$ , tedy druhá souřadnice průsečíku přímky s osou  $y$ . Proměnná  $k$  udává, zdali je přímka grafem: rostoucí funkce (pro  $k > 0$ ), klesající funkce ( $k < 0$ ) nebo je přímka rovnoběžná s osou  $x$  ( $k = 0$ ). Pro  $q = 0$  platí, že přímka prochází počátkem  $O$  kartézské souřadné soustavy.



Přímku můžeme i popsat pomocí rovnice obecného tvaru přímky:

$$ax + by + c = 0, \quad (3.2)$$

kde  $a$ ,  $b$ ,  $c$  jsou konstanty, přičemž konstanty  $a$  a  $b$  nejsou zároveň rovny nule.

Dále lze přímku vyjádřit i v parametrickém tvaru, který je určen těmito rovnicemi:

$$\begin{aligned} x &= x_0 + at, \\ y &= y_0 + bt, \end{aligned} \quad (3.3)$$

kde  $x_0$  a  $y_0$  jsou souřadnice kteréhokoliv bodu přímky,  $(a, b)$  je směrnice přímky, které nesmějí být zároveň rovny nule a  $t \in (-\infty, +\infty)$  je nezávisle proměnný parametr.

Užitečná je rovnice přímky dané dvěma body  $P_1[x_1, x_2]$  a  $P_2[x_2, y_2]$  pro ( $x_1 \neq x_2$ ):

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} \quad (3.4)$$

po úpravě:

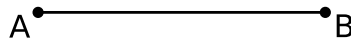
$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1). \quad (3.5)$$

Rovnici 3.4 je také možno zapsat ve tvaru matice:

$$\begin{vmatrix} x & y & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = 0.$$

### 3.4.3 Úsečka

Úsečka je speciální případ přímky, která má konečnou délku a je ohraničena dvěma body  $A$  a  $B$ . Značí se buď „úsečka  $AB$ “, nebo stručněji  $AB$ . Příklad takovéto úsečky je uveden na obrázku 3.5.



Obrázek 3.5: Úsečka

Pro úsečku  $AB$  jsme schopni vypočítat její délku, která se značí  $|AB|$  nebo vzdálenost dvou bodů  $d(A, B)$ . Výpočet provádíme podle vzorce 3.6:

$$|AB| = \sqrt{(B_x - A_x)^2 + (B_y - A_y)^2} \quad (3.6)$$

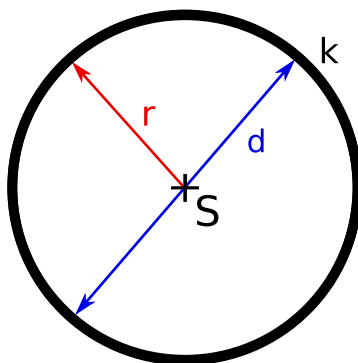
Dále můžeme u úsečky vypočítat její středový bod, který lze spočítat podle vzorce 3.7:

$$\begin{aligned} S_x &= \frac{A_x + B_x}{2}, \\ S_y &= \frac{A_y + B_y}{2}, \end{aligned} \quad (3.7)$$

kde  $S_x$  je  $x$ -ová souřadnice bodu a  $S_y$  je  $y$ -ová souřadnice středového bodu.

### 3.4.4 Kružnice

Kružnici  $k$  lze definovat jako rovinný útvar se středem  $S$  a poloměrem  $r$  (vizte obrázek 3.6), kde pro všechny body množiny kružnice  $k$  platí, že jejich vzdálenost od pevného bodu  $S$  je právě rovna poloměru  $r$ . Poloměrem kružnice nazýváme kteroukoliv délku úsečky, která má jako krajní body střed  $S$  a libovolný bod kružnice  $k$ . Dále určujeme průměr kružnice  $d$ , který je roven dvojnásobku poloměru, tedy délce tětiny, jejíž krajní body leží na kružnici a prochází středem  $S$ .



Obrázek 3.6: Kružnice

Pro výpočet délky kružnice (značíme znakem  $o$ ) je možné použít rovnici 3.8, která počítá jak s poloměrem  $r$ , tak s průměrem  $d$ .

$$o = 2\pi r = \pi d. \quad (3.8)$$

Pro popis kružnice, která má střed  $S$  v počátku kartézské soustavy  $O[0, 0]$ , slouží rovnice:

$$x^2 + y^2 = r^2. \quad (3.9)$$

Pomocí rovnice 3.9 však není možné kružnici umístit na libovolné souřadnice v kartézské soustavě, a proto je definována takzvaná obecná rovnice kružnice 3.10.

$$(x - m)^2 + (y - n)^2 = r^2, \quad (3.10)$$

kde  $m$  a  $n$  jsou souřadnice středu kružnice  $S[m, n]$  s poloměrem  $r$ .

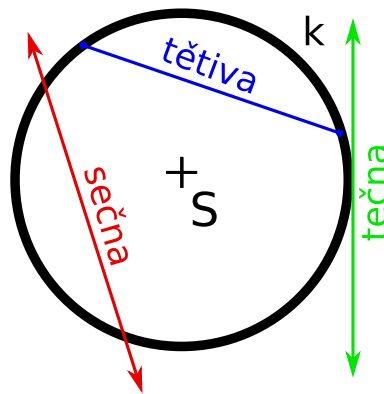
Rovnice 3.11 slouží k parametrickému vyjádření kružnice se středem  $S[m, n]$  a poloměrem  $r$ . Díky této rovnici je možné snadno získat libovolný bod kružnice na souřadnicích  $[x, y]$ .

$$\begin{aligned} x &= m + r \cos(t), \\ y &= n + r \sin(t), \end{aligned} \quad (3.11)$$

kde  $t \in \langle 0, 2\pi \rangle$  je nezávislý proměnný parametr pro vyjádření úhlu v obloukové míře.

Další vlastnosti můžeme vidět na obrázku 3.7, kde je znázorněna tečna, tětiva a sečna. Definice těchto pojmů se především hodí pro hledání průsečíku různých přímek, které protínají kružnici  $k$ .

- **Tečna** – Je přímka, která protíná kružnici v právě jednom společném bodě.
- **Tětiva** – Je úsečka, jejíž dva krajní body jsou vzájemně různé a oba leží na kružnici.
- **Sečna** – Je přímka, která protíná kružnici právě ve dvou různých bodech.



Obrázek 3.7: Vlastnosti kružnice

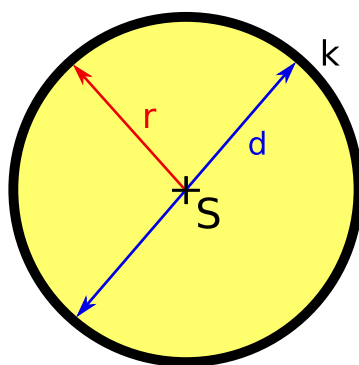
### 3.4.5 Kruh

Kruh  $k$  má stejné vlastnosti jako kružnice ze sekce 3.4.4 jen s tím rozdílem, že vzdálenost všech bodů množiny (kruhu  $k$ ) je od středu kruhu  $S$  nejvýše rovna poloměru kruhu  $r$ . Na obrázku 3.8 je pomocí barevné výplně znázorněna množina bodů, které tvoří kruh.

Pro kruh je možné spočítat nejen jeho obvod, který má stejný vzorec (vizte předchozí rovnice 3.8), ale navíc i jeho obsah  $s$ , který je dán rovnicí 3.12.

$$s = \pi r^2 = \frac{\pi d^2}{4}, \quad (3.12)$$

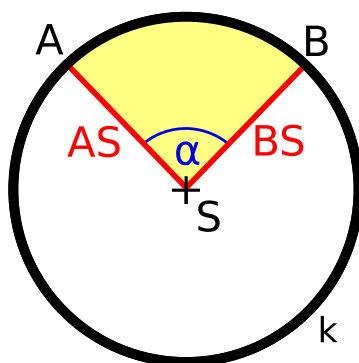
kde  $r$  je poloměr kruhu a  $d$  průměr kruhu.



Obrázek 3.8: Kruh

### 3.4.6 Kruhá výseč

Kruhá výseč je určitá část kruhu, která je tvořena třemi body, kde dva z nich ( $A$  a  $B$ ) leží přímo na obvodu kruhu (tedy na kružnici  $k$ ) a třetím je střed  $S$ . Z těchto bodů vytvoříme úsečky  $AS$  a  $BS$ , které svírají určitý úhel  $\alpha$  a ohraničují tak danou výseč (délka těchto úseček je poloměrem kruhu), jak je možné vidět na obrázku 3.9.



Obrázek 3.9: Kruhá výseč

Pro obvod  $o$ , respektive délku oblouku  $l$ , nyní platí rovnice:

$$l = \frac{\pi r \alpha}{180} = \frac{\pi d \alpha}{360} = r \operatorname{arc}(\alpha), \quad (3.13)$$

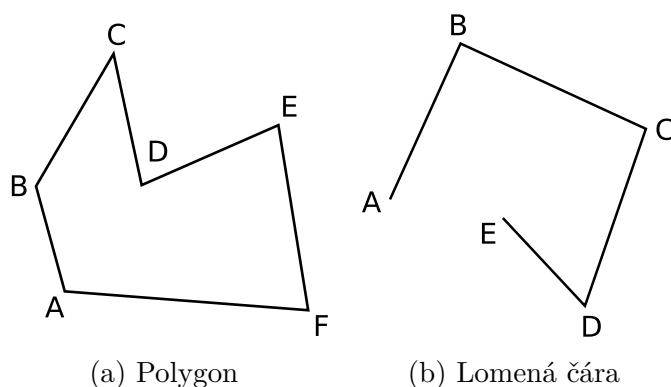
kde  $r$  je poloměr kruhu,  $d$  průměr kruhu,  $\operatorname{arc}$  je závislost obloukové míry na stupňové míře a  $\alpha$  velikost středového úhlu ve stupňové míře.

Pro výpočet obsahu  $s$  kruhové výseče slouží rovnice:

$$s = \frac{\pi r^2 \alpha}{360} = \frac{1}{2} r^2 \operatorname{arc}(\alpha) = \frac{lr}{2}. \quad (3.14)$$

### 3.4.7 Polygon

Polygon (mnohoúhelník) je rovinný útvar, který pomocí úseček spojuje  $n$  různých bodů, kde  $n \in \mathbb{N}$  a  $n \geq 3$ . Tyto úsečky nazýváme strany polygonu a body vrcholy polygonu. Množinu úseček spojujících dané body lze považovat za uzavřenou lomenou čáru pod podmínkou, že i poslední bod je spojen úsečkou s prvním bodem (tedy  $A_1A_2, A_2A_3, \dots, A_nA_1$ ), pak tato lomená čára tvoří uzavřenou oblast procházející množinou bodů, kterou nazýváme polygonem. Na obrázku 3.10a můžeme vidět příklad polygonu, kdežto na obrázku 3.10b je pouze příklad lomené čáry.

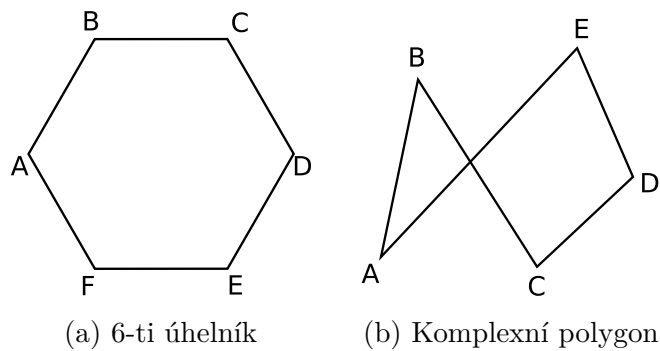


Obrázek 3.10: Porovnání polygonu a lomené čáry

Na základě různých vlastností můžeme rozlišovat několik skupin polygonů, kterými jsou:

- **Konvexní** – Konvexní znamená, že každý vnitřní úhel polygonu je menší nebo roven  $180^\circ$ .
- **Konkávni** – Konkávni naopak znamená, že má alespoň jeden vnitřní úhel větší než  $180^\circ$ .
- **Pravidelný** – Pro pravidelný polygon platí, že všechny jeho úhly a strany jsou stejně velké.
- **Nepřavidelný** – Nepřavidelný polygon je opak pravidelného, tedy takový, který má různě velké úhly a strany.
- **Jednoduchý** – Jednoduchý polygon je takový, že uzavřená lomená čára, která ho tvoří, neprotíná sama sebe v žádném bodě.
- **Komplexní** – Komplexní polygon je opačný případ jednoduchého, jeho uzavřená lomená čára protíná sama sebe alespoň v jednom bodě.

Příklad nekonvexního, nepravidelného a jednoduchého polygonu můžeme vidět na předchozím obrázku 3.10a. Konvexní a pravidelný polygon je pak znázorněn na obrázku 3.11a jako 6-ti úhelník. Komplexní polygon představuje obrázek 3.11b.



Obrázek 3.11: Porovnání polygonů

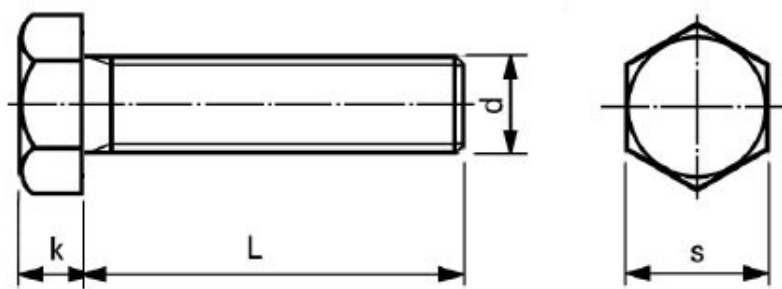
## 3.5 Spojovací součástky

Pro spojení jednotlivých součástí šroubovaného spoje se primárně používá součástka – šroub. Dále je možné použít i jiné součástky, které pomohou k rozložení sil působící na spoj. Jednotlivé typy součástek mají obecně zabránit různým pohybům spojovaných částí šroubovaného spoje, protože v určitou chvíli mohou na každou část spojení působit jiné síly je nutné, aby daná spojovací součástka sílu rozložila a nedovolila tak posunu jednotlivých (či rotaci) částí šroubovaného spoje, proto existuje několik základních součástek, které jsou popsány níže. Součástky byly zvoleny na základě požadavků konstruktérů (vizte Požadavky na aplikaci 3.2).

Grafický tvar těchto prvků v aplikaci je možné i ručně složit ze základních primitiv, ale pro jednodušší a rychlejší návrh spoje jsou již předem definovány. Navíc je u těchto objektů kladen důraz na jejich umístění podle předem jasně daných pravidel, které jsou uvedeny v části 3.2.

### 3.5.1 Šroub

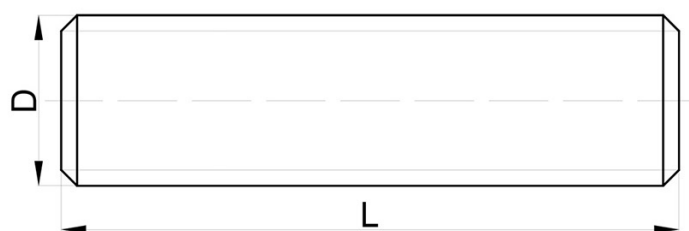
Šroub je tvořen hlavou (různého tvaru) a dříkem. Existuje velké množství různých typů šroubu a každý z nich se hodí k jinému účelu. V aplikaci se setkáme s jedním typem, a to je šroub s šestiúhelníkovou hlavou, jehož velikost se měří podle průměru (či poloměru) dříku a jeho závit je po celé délce dříku, který je uveden v nákresu na obrázku 3.12. Šroub v praxi zabraňuje, aby se jednotlivé části spoje pohnuly v osách kolmých a rovnoběžné na šroub (aby nedocházelo ke smyku a oddělení materiálů od sebe), ale také k rotačnímu pohybu částí spoje (pokud je šroubů více než jeden).



Obrázek 3.12: Nákres šroubu (zdroj: <http://www.ebolt.co.uk/>)

### 3.5.2 Kolík

Kolík je obvykle kuželový či válcový spojovací prvek. Podle jeho tvaru se dále dělí na různé další druhy. Podle tvaru a druhu kolíku se určují jeho rozměry. V aplikaci bude použit kolík s válcovým tvarem, u kterého je možné v půdorysu definovat jeho průměr, respektive poloměr. Nákres válcového kolíku je možné vidět na obrázku 3.13. Kolík v praxi umožňuje rychlé rozebrání jednotlivých částí spoje a zajišťuje vzájemnou polohu dvou součástí (*lícovací kolík*), ale zejména má zamezit posuvu jedné části po druhé (u tzv. *pojišťovacích kolíků* i zabránit rotaci).



Obrázek 3.13: Nákres kolíku (zdroj: <http://www.konarik.cz/>)

### 3.5.3 Péro

Péro je podlouhlá součástka většinou s pravoúhlým průřezem (hrany mohou být i oblé). V materiálu je pro něj vyfrézován otvor, do které je posléze pero vloženo. Velikost se udává jeho šířkou, výškou (průřezu) a délkou. V nákresu je však důležitá pouze hodnota šířka a délka. Tvar péra může být různý a závisí pouze na účelu využití. Nákres *těsného péra* je uveden na obrázku 3.14, tento typ péra je použit v aplikaci a má zabránit jak otáčení spojovaných součástí materiálu, tak i jejich vzájemnému pohybu.

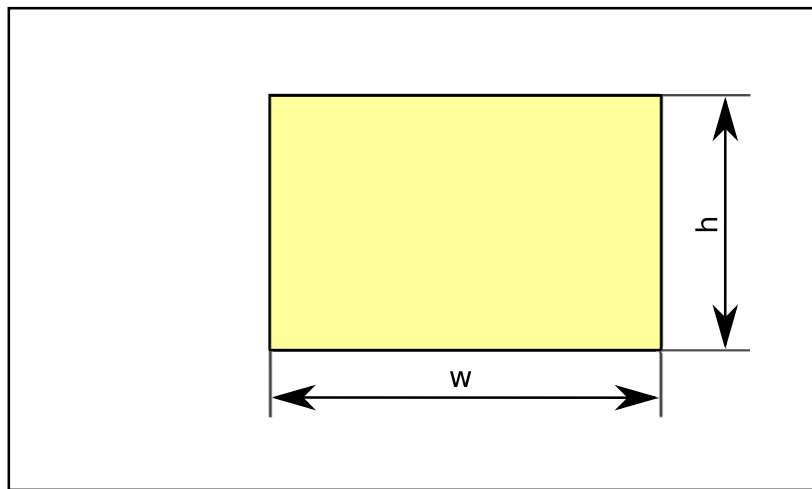


Obrázek 3.14: Nákres těsného péra



### 3.5.4 Zátěžová plocha

Zátěžová plocha je místo, který působí silou na spojovací materiál. Zvyšuje třecí sílu mezi spojenými materiály, ale také zabraňuje celkovému pohybu (díky zvýšení hmotnosti). Obvyklý tvar je obdélníkový (ale i jakýkoliv jiný, záleží k jakému účelu bude zátěžová plocha použita) a udává se šířkou a délkou. Pro nákres jsou důležité hodnoty šířka a délka. Příklad obdélníkové zátěžové plochy, která je umístěna na spoji, je možné vidět na obrázku 3.15 (zvýrazněná oblast je zátěžová plocha).



Obrázek 3.15: Nákres obdélníkové zátěžové plochy

## 3.6 Grafické uživatelské rozhraní

GUI (*Graphical User Interface*), tedy grafické uživatelské rozhraní, jež umožňuje ovládat aplikaci pomocí grafických komponent. V GUI bude kladen důraz na typické rozmístění základních komponent tak, aby se co nejvíce podobalo již existujícím CAD systémům, ale zároveň aplikace působila minimalistickým dojmem (žádné nadbytečné komponenty, které způsobují nepřehlednost). U GUI je především důležité, aby se znalý (ale i částečně nezalý) uživatel neztrácel v aplikaci, ale intuitivně zvládal použít jednotlivé nástroje, které aplikace poskytuje. Proto je vhodné, aby komponenty reprezentující nástroje měly takovou ikonu, která nejvíce vystihuje její funkci. Základní komponenty těchto typů aplikací jsou:

- Menu.
- Plátno pro objekty.
- Nástroje pro práci s objekty.
- Prvky pro umístění jednotlivých objektů.
- Panel sloužící k parametrické úpravě vlastností objektu.
- Seznam umístěných objektů.

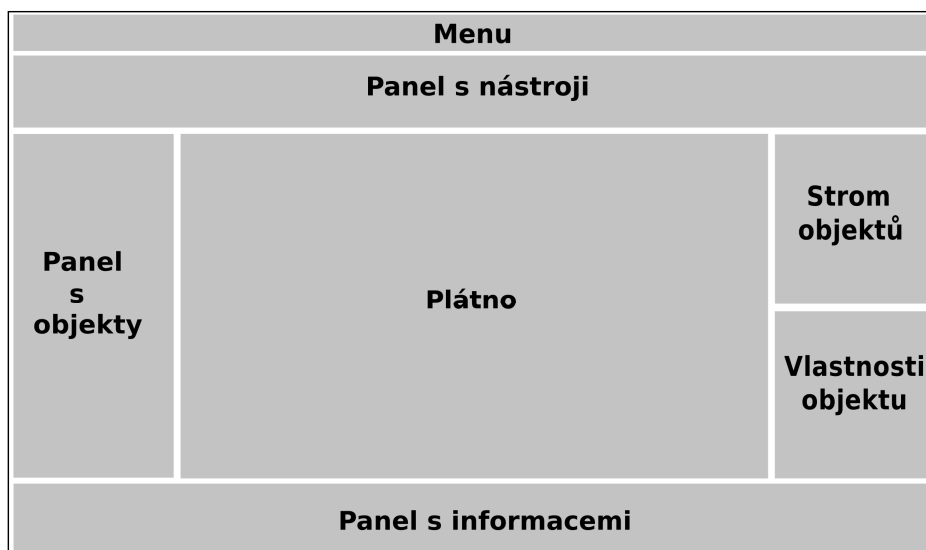
Komplexnější CAD systémy nabízejí daleko větší množství komponent, ale pro tento typ CADu tyto základní funkce zcela postačují, aby aplikace byla použitelná při praktickém nasazení.

### 3.6.1 Rozložení komponent

Cílová skupina uživatelů, která bude aplikaci používat, jsou strojaři z Katedry konstruování strojů, a ti jsou zvyklí používat CAD systémy uvedené v kapitole 2. Proto by bylo žádoucí, aby i výsledná aplikace byla do určité míry podobná zmiňovaným CAD systémům (avšak s ohledem na její požadavky 3.2). Na obrázku 3.16 je možné vidět umístění komponent v aplikaci.

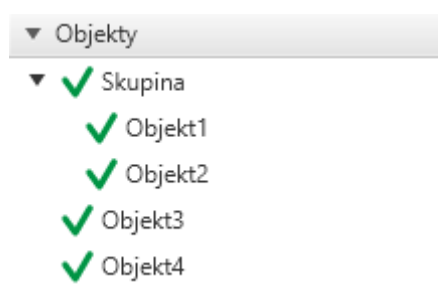
### 3.6.2 Strom objektů

Jednou z komponent, která ulehčuje manipulaci s objekty, je strom objektů. Tato komponenta ukládá veškeré umístěné objekty do stromu. Strom je vybrán z toho důvodu, že komplexní objekt bývá tvořen z více jiných objektů



Obrázek 3.16: Návrh GUI

(tzv. skupina objektů) a tyto objekty je nutné zobrazit tak, aby bylo jasné, že patří k jedné skupině, což strom umožňuje. Všechny objekty jedné skupiny jsou zarovnány pod objektem skupiny a to působí dojmem, že jsou součástí jednoho objektu. Další užitečnou funkcí je možnost tyto objekty skrýt (pokud je kompozit složen z hodně objektů, pak zaplňují velký prostor a zneřehledňují celý strom). K získání představy slouží obrázek 3.17, který ukazuje příklad jednoduchého stromu s kompozitem (složen ze dvou objektů) a dvěma dalšími základními objekty.



Obrázek 3.17: Stromová struktura

### 3.6.3 Panel vlastností

Panel vlastností slouží k zobrazení a úpravě aktuálně vybraného objektu. Díky tomu, že je panel umístěn přímo v základním rozložení aplikace, je úprava jednotlivých objektů pohodlnější a rychlejší. Pro tento účel je snazší

použít panel, než např. dialogové okno (samostatné okno v aplikaci), které se nemusí vždy zobrazit v popředí aplikace, ale může být schované v pozadí (přepínání oken zbytečně prodlužuje práci).

### 3.6.4 Dialogová okna

V některých případech je použití dialogových oken vhodné. Jedná se především o situace, kdy chceme uživatele po nějaké události informovat, že daná operace nebyla provedena správně nebo je nutné ji dokončit manuálně. Takovým případem může být přichycení jednoho objektu k druhému, kdy objekt, který se snažíme přichytit, se nachází na ploše více objektů, které nejsou v jedné skupině, potom je nutné ručně určit, ke kterému z těchto objektů se přichytí. Dalším případem je načtení objektu ze souboru, kdy pomocí dialogového okna můžeme vybrat soubor s objektem (to samé i s uložením).

## 3.7 Výstupní formát dat

Pro uložení tohoto typu dat byl vybrán formát XML [16] (*eXtensible Markup Language*), tedy rozšiřitelný značkovací jazyk, protože bývá označen jako standardní formát pro výměnu informací a lze ho snadno převést do jiných formátů (bude blíže popsáno dále). XML je určen pro přenos dat mezi různými aplikacemi. Jednotlivá data jsou vkládána mezi předem určené tagy (značky), které jsou zvolené podle potřeby. Musí se však splnit pravidla, aby daný XML soubor byl validní (pravidla můžeme nalézt v [16]).

### 3.7.1 Struktura objektu

Objekt je uvozen počáteční značkou `<shape>` a ukončovací značkou `</shape>` značkou. Mezi těmito dvěma značkami jsou uloženy veškeré informace o objektu. Informace jsou různé dle typu objektu. Formát jednotlivých vlastností objektu má však stejnou šablonu, každá vlastnost je uvozena počáteční značkou `<variables type="typ_proměnné" name="název_proměnné">` a ukončovací značkou `</variables>`, kde právě `typ_proměnné` a `název_proměnné` se liší podle typu objektu k jeho uložení do souboru. Každý objekt může ještě obsahovat seznam dodatečných informací, které si mohou uživatelé sami definovat a ty jsou pak uloženy ve počáteční značce `<properties>` a ukončovací značce `</properties>` se stejnou hierarchií, jako vlastnosti objektu uvedené v tomto odstavci.

### 3.7.2 Struktura skupiny objektů

Skupina objektů je uvozena počáteční značkou `<composite>` a ukončovací značkou `</composite>`. V těchto značkách jsou obsaženy jednotlivé objekty s jejich vlastnostmi ze skupina složena (vizte 3.7.1). Za výčtem všech jeho objektů jsou ještě uloženy konkrétní informace o celé skupině, které jsou stejné, jako informace objektu (ve výsledku je skupina také pouze objekt). To znamená, že používají stejnou šablonu pro uložení vlastností.

### 3.7.3 Struktura plátna

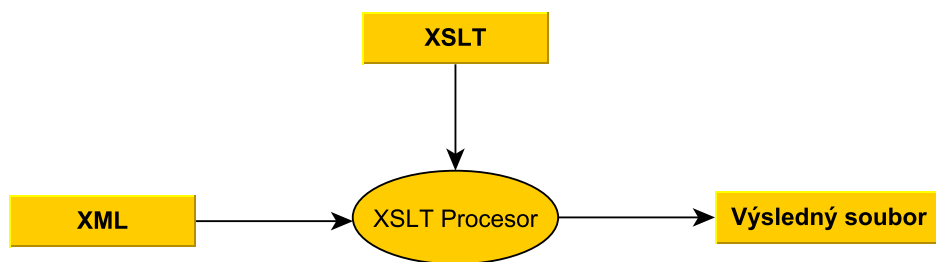
Plátno je uloženo jako výčet všech objektů, které jsou na něm umístěny. Všechny jeho objekty jsou uloženy mezi značkami `<canvas>` a `</canvas>`. Plátno si neukládá žádné další speciální informace, pouze shromažďuje veškeré objekty v jednom souboru.

### 3.7.4 Dodatečné vlastnosti objektu

V XML souboru v kořenovém adresáři aplikace *properties.xml* jsou pro každé primitivum a součástku připraveny kontejnery, kam lze doplnit dodatečné vlastnosti daného objektu. Uživatel si podle příkladu (uveden v komentáři souboru) může ke každému objektu definovat další vlastnosti navíc, které potřebuje uložit s daným objektem. V těchto souborech jsou uloženy typy a názvy proměnných, které následně bude možné editovat přímo v aplikaci. Při definování je možné i vyplnit výchozí hodnoty, které budou automaticky zapsány do seznamu vlastností při vytvoření objektu.

### 3.7.5 XSL Transformace

XSL [17] (*eXtensible Stylesheet Language*) je rozšiřitelný stylový jazyk, který slouží k popisu, jak se má XML dokument transformovat, ale také slouží k definici formátování vzhledu XML dokumentů (vzhled jednotlivých elementů). Díky této transformaci je možné převést XML dokument na jakýkoliv jiný typ. K tomu nám postačí pouze napsat soubor XSLT [18] (*XSL Transformace*), který obsahuje značky pro XSLT procesor a značky výsledného souboru. K transformaci XML dokumentu použijeme vytvořený XSLT soubor (který musíme sami napsat v závislosti na vstupním – XML – a výstupním souboru), tyto dva soubory necháme zpracovat tzv. XSLT procesorem, který je obvykle součástí programovacího jazyka nebo jako knihovny, a pomocí něho je vytvořen výsledný soubor. Příklad, jak probíhá taková transformace, je uveden na obrázku 3.18.



Obrázek 3.18: XSL Transformace

## 4 Implementace

Aplikace je implementována v programovacím jazyce Java s využitím technologie JavaFX (důvod výběru vizte 3.3) za dodržování zásad OOP [14] (*Objektově orientované programování*). Díky zásadám OOP bude možné v budoucnu aplikaci dále rozvíjet (např. přidání nového nástroje či objektu). K verzování zdrojového kódu byl použit verzovací systém Git [7] na domácím uložišti autora.

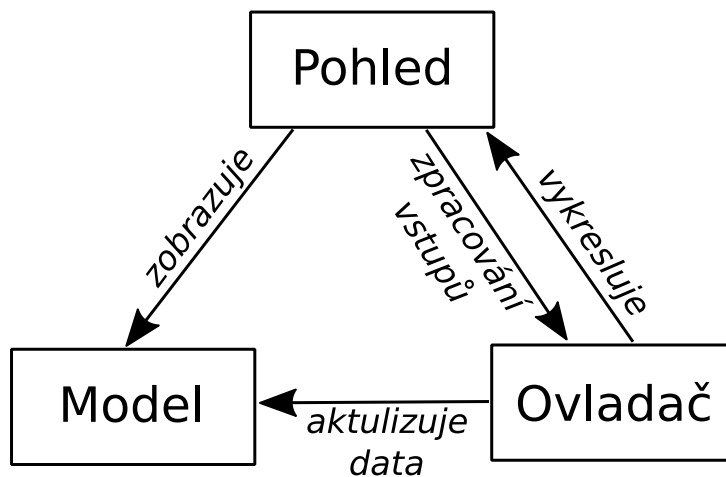
Jednotlivé třídy (základní prvek OOP) aplikace jsou podle účelu uloženy Java v balíku (*package* – sdružuje třídy nebo další balíčky v jednom adresáři), který má následující strukturu:

- **controllers** – obsahuje všechny ovladače (obstarávají logiku aplikace), který mají na starosti aplikační logiku.
- **utils** – obsahuje užitečné třídy, které slouží pro výpočty nebo k uchování různých informací.
- **models** – zde jsou uloženy jednotlivé třídy, které reprezentují modely (geometrické tvary, součástky, tlačítka, apod.) aplikace.
- **tools** – obsahuje třídy, které reprezentují nástroje (pro manipulaci s objekty – geometrickými tělesy) v aplikaci.
- **views** – obsahuje jednotlivé grafické prvky (FXML, CSS a ikony) a hlavní třídu aplikace.

### 4.1 Architektura MVC

Celá aplikace je implementována podle principu modelu MVC (*Model-View-Controller*), podrobný popis lze nalézt v [15]. Je to softwarová architektura, která se nejvíce používá pro vytváření GUI aplikací. Obsahuje tři základní komponenty: *model*, *pohled* a *ovladač*. *Model* reprezentuje objekt, u kterého potřebujeme uchovat jeho vlastnosti a definovat chování (např. jak se bude vykreslovat). *Pohled* je grafický prvek, který zobrazuje vlastnosti objektu nebo přebírá vstupy zadané uživatelem. *Ovladač* má na starosti celou aplikační logiku, tedy zpracovává vstupy od uživatelů a na základě nich aktualizuje *model* a posléze opět předá *pohledu*. Existuje mnoho přístupů, jak

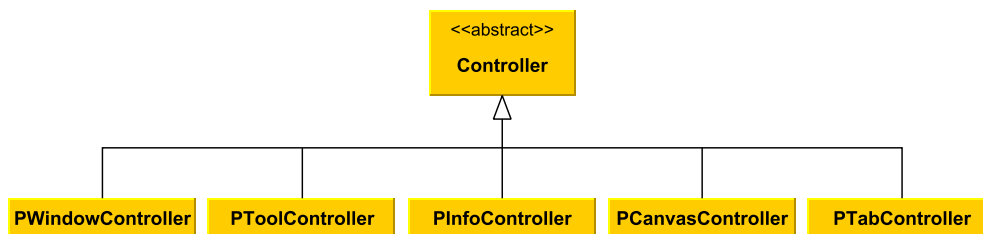
postavit aplikaci na architektuře MVC, ale především jde o to, aby byl dodržen princip jednotlivých vrstev – mít oddělenou logiku od *modelu* a *pohledu*. Z obrázku 4.1 je patrné, jak MVC funguje.



Obrázek 4.1: Diagram MVC

## 4.2 Aplikační logika

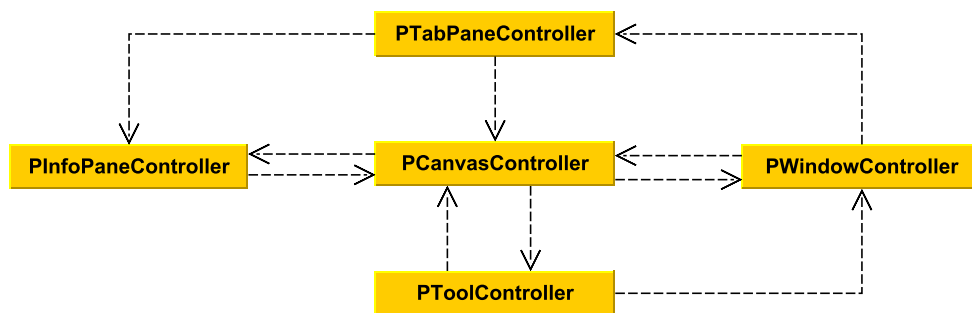
O veškerou aplikační logiku se starají tzv. *ovladače*, které mezi sebou komunikují a předávají si informace, podle nichž provádějí akce. Aplikace obsahuje pět základních ovladačů: PCanvasController, PInfoPaneController, PTabPaneController, PToolController a PWindowController. Tyto zmíněné ovladače dědí (vlastnost OOP) od abstraktního ovladače PController, jak je možné vidět na obrázku 4.2. Na *ovladač* lze nahlížet jako na návrhový vzor *Prostředník* (*Mediator* – více o návrhových vzorech je možné nalézt v [13]).



Obrázek 4.2: Diagram tříd ovladačů

Na obrázku 4.3 je patrné, jak ovladače mezi sebou komunikují.

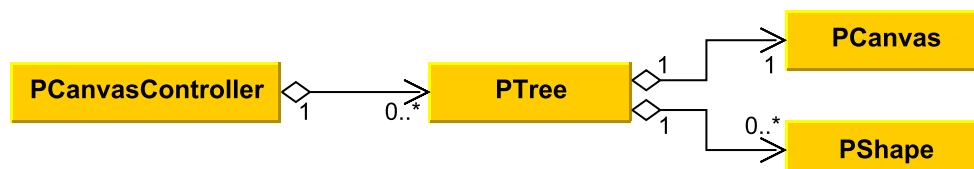




Obrázek 4.3: Vazby mezi ovladači

### 4.2.1 PCanvasController

Tento ovladač má na starosti uchovávání veškerých vytvořených pláten v seznamu a provádět nad nimi akce (např. překreslení plátna). Samotná třída `PCanvas`, která reprezentuje plátno, však není uložena přímo v ovladači, ale nejdříve je vytvořena třída `PTree`, která ve stromové struktuře uchovává všechny objekty, jež jsou umístěné na plátně a jako kořenový prvek je plátno. Pokud uživatel hodlá přidat např. nový objekt (na plátno), vše probíhá právě přes tento ovladač pomocí metody `void addShape(PShape shape)`, která dále zařídí i překreslení plátna zavoláním metody `void redrawCanvas()`. Příklad vazby mezi ovladačem a plátnem je možné vidět na obrázku 4.4.



Obrázek 4.4: Závislost mezi ovladačem a plátnem

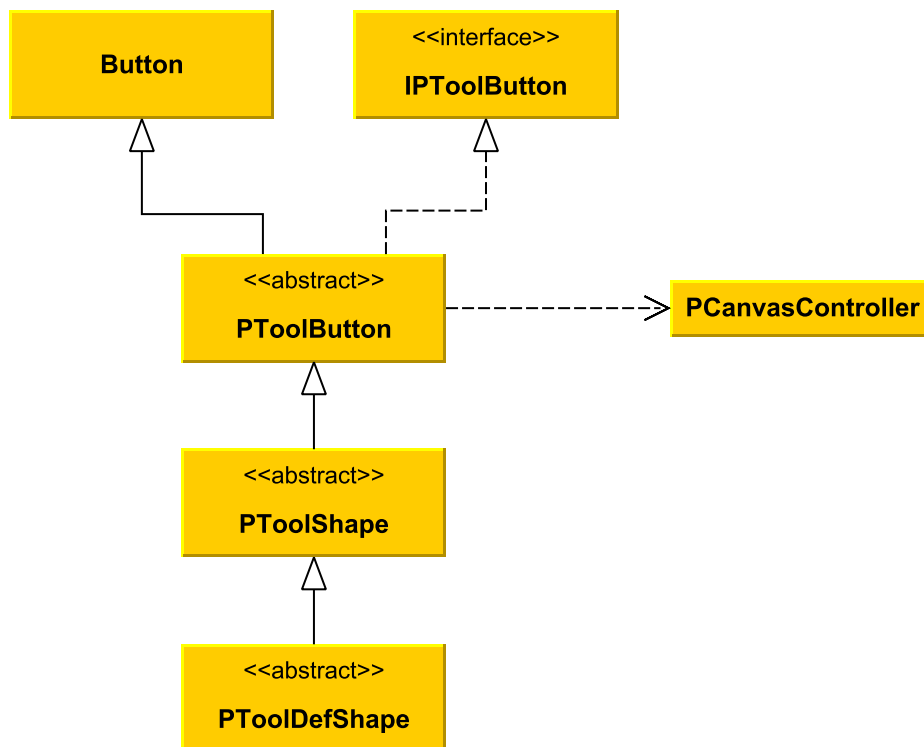
### 4.2.2 PTabPaneController

Ovladač se stará o to, aby se nové plátno přidalo do panelu (grafické komponenty API JavaFX), který jednotlivá plátna zobrazuje v záložkách. V Javě se tento panel vytváří pomocí třídy `TabPane`. Díky tomu lze pracovat s více plátny a je možné i přenášet objekty z jedno plátna do jiného (zkopírováním objektu). `PTabPaneController` komunikuje s `PCanvasController`, a to tak, že po přidání nové záložky se vytvoří plátno, respektive třída `PTree`, a pomocí metody `void addCanvas(PTree canvas)` je předáno ovladači, které

se stará o plátna. To samé platí i pro uzavření záložky s plátnem, jen se volá metoda `void removeCanvas(PTree canvas)`.

### 4.2.3 PToolController

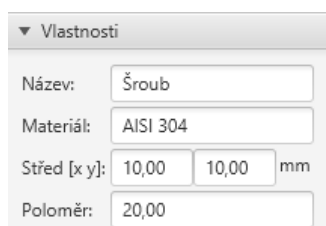
Ovladač se stará veškerá tlačítka a pomocí nich zajišťuje interakci s uživatelem. Tlačítka se dále dělí na tři skupiny: tlačítka pro umístování geometrických primitiv, tlačítka pro umístování součástek a tlačítka s nástroji. Po stisku libovolného tlačítka je ovladači předán pokyn, aby nastavil události (dle typu tlačítka) všem aktivním plátnům a předchozí události smazal. Události jsou řešeny pomocí rozhraní `EventHandler` a tyto události jsou uchovány v přepravce `PEvent` i s typem události (typ události – klávesa, tlačítko myši, apod.). Každé tlačítko dědí od abstraktní třídy `PToolButton`, které implementuje rozhraní `IPToolButton` a dědí třídu `Button` (komponenta pro tlačítko). Jednotlivé závislosti jsou znázorněny na obrázku 4.5. Od abstraktní třídy `PToolShape` dědí tlačítka, která na plátno přidávají geometrická primitiva a od `PToolDefShape` dědí tlačítka, která na plátno přidávají součástky.



Obrázek 4.5: Vazby mezi tlačítky a ovladačem

#### 4.2.4 PInfoPaneController

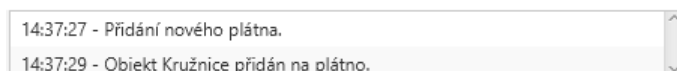
Tento ovladač se stará o to, aby panel (`TreeView`), který zobrazuje informace o umístěných objektech na plátně, byl stále aktuální (např. při odstranění objektu z plátna je odstraněn i z tohoto panelu). Dále zobrazuje panel s vlastnostmi jednotlivých objektů, zpracovává zadané hodnoty (manuální úprava vlastností objektu) a odesílá je ovladači, který se stará o plátno (ten pak aktualizuje hodnoty upravovaného objektu). Panel s vlastnostmi šroubu je možné vidět na obrázku 4.6.



Obrázek 4.6: Panel s vlastnostmi objektu

#### 4.2.5 PWindowController

Ovladač má na starosti obsluhu menu (komponenta `Menu`) a informační panel – `ListView` (informuje uživatele o provedené akci, vizte obrázek 4.7) ve spodní části aplikace. Zpracovává aktuální pozici myši a pomocí komponenty `Label` (slouží pro zobrazení popisku) ji zobrazuje uživateli. Dále se vytváří dialogová okna k uložení/načtení objektu či plátna.

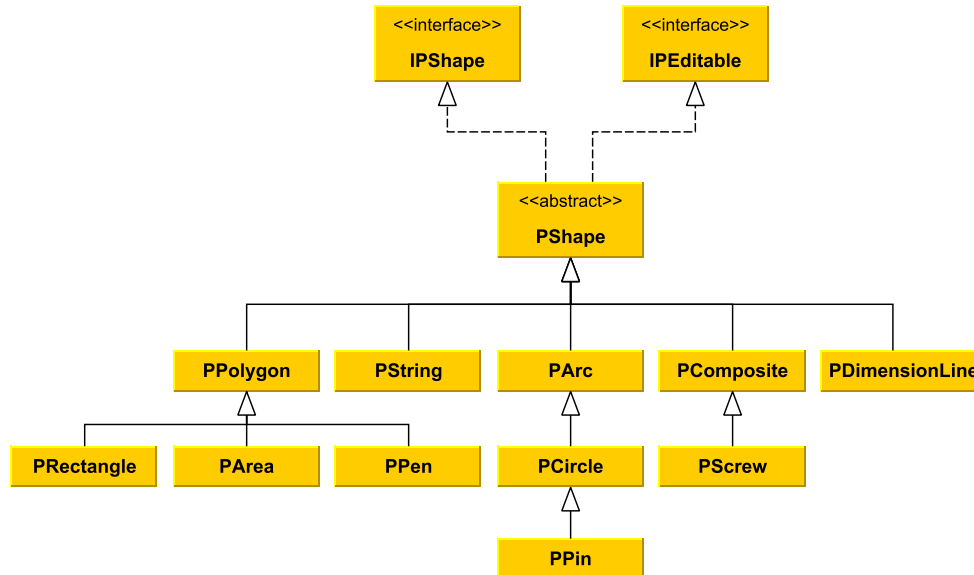


Obrázek 4.7: Informační panel

### 4.3 Modely

V této části bude popsána implementace jednotlivých geometrických primitiv z části 3.4 a důležité výpočty, jako je např. výpočet průsečíků a test, zda je bod uvnitř tělesa. Součástky z části 3.5 jsou složeny z geometrických primitiv, a to znamená, že od nich přebírají veškeré vlastnosti, pouze jsou u nich definována speciální pravidla pro umístění na plátno (vizte 3.2). Vazby mezi těmito třídami reprezentující tělesa (primitiva a součástky) lze

vidět na obrázku 4.8. Dále se v této části může objevit pojem *instance třídy* a *překrytí metody* (prvky OOP, jejichž popis lze nalézt v [14]).



Obrázek 4.8: Diagram tříd těles

K vykreslování těles na plátno, musí třídy reprezentující jednotlivá geometrická tělesa překrýt metodu `void drawing(PGraphicContext pgc)` v níž je uvedeno, jak se má těleso vykreslovat. K vykreslování jednotlivých objektů slouží třída `PCanvas`, která dědí od třídy `Canvas` (JavaFX komponenta pro plátno). Existují dva typy výplně objektů: s výplní (pevné těleso) nebo bez výplně (díra).

### 4.3.1 PArc

`PArc` je třída reprezentující kruhovou výseč. K vytvoření lze použít tři různé konstruktory, které se liší vstupními parametry:

- `PArc(PPoint pcenter, PPoint pstartPoint, PPoint pendPoint, PShapeType ptype)` – Vytvoří kruhovou výseč na základě zadaného středu kruhové výseče, počátečního bodu, koncového bodu a typu výplně.
- `PArc(PPoint pcenter, PPoint pstartPoint, double pangle, PShapeType ptype)` – Vytvoří kruhovou výseč na základě středu kruhové výseče, počátečního bodu, velikosti úhlu a typu výplně.

- `PArc(PArc arc)` – Kopírovací konstruktor, vytvoří novou instanci třídy z již existující.

### Výpočet průsečíků kruhové výseče s úsečkou

K výpočtu průsečíku (bodu) je použita parametrická rovnice přímky z 3.4.2. Průsečík s kruhovou výsečí musí splňovat podmínku, aby jeho vzdálenost od středu kruhové výseče byla rovna poloměru kruhové výseče. Dále je nutné ověřit, jestli leží v oblasti kruhové výseče (plocha kruhové výseče) a nachází se na testované úsečce (protože přímka je nekonečně dlouhá). Vychází se z toho, že průsečík musí ležet ve vzdálenosti, která je rovna poloměru  $r$  kruhové výseče:

$$|ku + q - s|^2 = r^2, \quad (4.1)$$

kde  $s$  je střed kruhové výseče. Rovnice je umocněna, aby se testovaly obě strany kruhové výseče. Po úpravě dostaneme:

$$k^2(u^2) + 2k(u(q - s)) + (q^2 + ss - 2qs - r^2) = 0, \quad (4.2)$$

podle výpočtu diskriminantu  $D$  (vizte [3]) této kvadratické rovnice mohou nastat tři případy:  $D < 0$ , tudíž neexistuje průsečík;  $D = 0$ , existuje právě jeden průsečík; a  $D > 0$ , existují dva průsečíky. Pomocí vzorce pro výpočet kořenů kvadratické rovnice dostaneme parametr  $k$ :

$$k_{1,2} = \frac{-b \pm \sqrt{D}}{2a}, \quad (4.3)$$

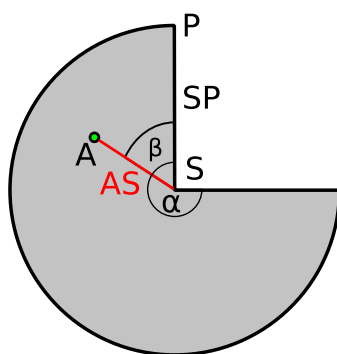
kde  $a$ ,  $b$ ,  $c$  jsou koeficienty z rovnice 4.2. Dosazením získaného parametru  $k_1$  a  $k_2$  do:

$$\begin{aligned} x &= k_{1,2}u + x_0, \\ y &= k_{1,2}u + y_0, \end{aligned} \quad (4.4)$$

kde  $u$  je směrový vektor přímky a  $x_0$ ,  $y_0$  jsou souřadnice libovolného bodu ležícího na této přímce, dostaneme souřadnice průsečíku (bodu).

### Test bodu uvnitř kruhové výseče

Pro test, jestli bod  $A$  leží uvnitř kruhové výseče, je sestrojena orientovaná úsečka  $SA$  (střed kruhové výseče s testovaným bodem  $A$ ) a  $SP$  (střed a počáteční bod kruhové výseče). Následně je vypočtena odchylka  $\beta$  (úhel) a pokud platí, že  $0 \leq \beta \leq \alpha$ , kde  $\alpha$  je úhel kruhové výseče, a délka úsečky  $AS$  je menší nebo rovna poloměru kruhové výseče, pak tento bod leží uvnitř kruhové výseče, jak je možné vidět na obrázku 4.9.



Obrázek 4.9: Test bodu v kruhové výseči

### 4.3.2 PCircle

PCircle je třída reprezentující kružnici. K vytvoření lze použít dva různé konstruktory, které se liší vstupními parametry:

- `PCircle(PPoint pcenter, double pradius, PShapeType ptype)` – Vytvoří kružnici na základě středu kružnice, poloměru kružnice a typu výplně.
- `PCircle(PCircle circle)` – Kopírovací konstruktor, vytvoří novou instanci třídy z již existující instance.

#### Výpočet průsečíků kružnice s úsečkou

Je použit stejný postup jako u kruhové výseče (vizte 4.3.1) s jedním rozdílem, že není nutné testovat, jestli nalezené průsečíky leží v oblasti, kterou kruhová výseč určuje, protože kružnice má úhel  $360^\circ$ .

#### Test bodu uvnitř kružnice

Pro test bodu u kružnice stačí zjistit jeho vzdálenost od středu kružnice a pokud je tato vzdálenost menší nebo rovna poloměru kružnice, pak testovaný bod leží v kružnici.

### 4.3.3 PPolygon

PPolygon je třída reprezentující polygon. K vytvoření lze použít jeden konstruktor:

- `PPolygon(List<PPoint> pointList, PShapeType ptype)` – Vytvoří polygon, kde vstupem je seznam jeho bodů.

K vytvoření kopie polygonu je připravena metoda `PPolygon copy` (`PPolygon polygon`).

### Výpočet průsečíků polygonu s úsečkou

V cyklu se postupně projde každý bod polygonu (v seznamu jsou body uloženy proti směru hodinových ručiček) a každé dva sousední body (v seznamu) se spojí úsečkou. Každá tato úsečka je posléze testována se vstupní úsečkou (tj. úsečka, která má mít společný bod s polygonem). K nalezení průsečíku dvou různých úseček budeme vycházet z těchto dvou rovnic přímek (vizte část 3.4.2):

$$\begin{aligned} a_1x + b_1y + c_1 &= 0, \\ a_2x + b_2y + c_2 &= 0 \end{aligned} \tag{4.5}$$

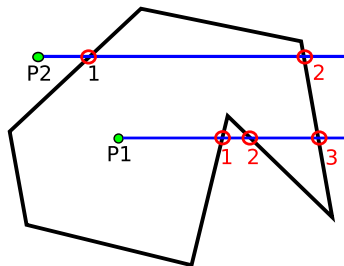
a dostaneme:

$$\begin{aligned} x_p &= \frac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1}, \\ y_p &= \frac{c_1a_2 - c_2a_1}{a_1b_2 - a_2b_1}, \end{aligned} \tag{4.6}$$

kde  $x_p$  a  $y_p$  jsou souřadnice bodu průsečíku za podmínky, že  $(a_1b_2 - a_2b_1) \neq 0$ . Pokud by se jmenovatel rovnal 0, pak průsečík neexistuje. Nalezený průsečík se posléze otestuje, zda leží na obou úsečkách (z důvodu, že přímka je nekonečně dlouhá).

### Test bodu uvnitř polygonu

K testování, jestli leží bod uvnitř polygonu, je použit algoritmus *Ray Crossing* [12]. Činnost algoritmu nejlépe vystihuje obrázek 4.10, kde bod  $P_1$  leží uvnitř polygonu, z něhož je vedena přímka rovnoběžná s osou  $x$  směrem doprava. Následně jsou nalezeny všechny průsečíky s touto přímkou a pokud je lichý počet průsečíků, pak tento bod leží v polygonu. Příklad bodu, který v polygonu neleží, je  $P_2$  (přímka má s polygonem sudý počet průsečíků).



Obrázek 4.10: Test bodu v polygonu

### 4.3.4 PComposite

Třída dovoluje uchovat libovolný počet ( $> 2$ ) objektů tvořících jednu skupinu, která se bude chovat jako jeden objekt (objekty jsou uloženy v seznamu). Veškeré aplikované transformace na tuto skupinu (objekt), se projeví na všech obsažených objektech. Aby bylo možné z objektů vytvořit skupinu, musí se buďto překrývat (mají společně body – jedná se o sjednocení) nebo se nacházejí vně jiného objektu (jedná se o ukotvení, popř. vyříznutí). Každému objektu přidanému do skupiny je nastaven i jeho rodič (tj. reference na skupinu – skupina je jeden objekt). Díky tomu je možné přímo z objektu získat referenci na rodiče, jež se hodí např. pro oddělení objektu ze skupiny (nemusí se procházet veškeré skupiny a hledat objekt, který má být oddělen). Třída nevyžaduje žádné speciální algoritmy (veškeré algoritmy již obsahují objekty skupiny). K vytvoření lze použít dva různé konstruktory, které se liší vstupními parametry:

- `PComposite(PShape shape1, PShape shape2)` – Vytvoří skupinu ze dvou objektů.
- `PComposite(List<PShape> shapeList)` – Vytvoří skupinu ze seznamu objektů.

K vytvoření kopie skupiny slouží metoda `PComposite copy(PComposite composite)`.

## 4.4 Nástroje

V této části budou popsány pouze nástroje, které se netýkají vyříznutí, sjednocení nebo ukotvení (tyto nástroje jsou založeny na hledání průsečíků nebo testování bodu uvnitř objektu, jež je popsáno v 4.3), ale převážně takové nástroje, jejichž funkčnost nebyla popsána, jako např.: rotace, vrácení změn, kopírování objektů, apod.

### 4.4.1 Rotace

K rotaci libovolného objektu je použita matice rotace (čerpáno z [6]). Pomocí matice rotace můžeme nechat objekt rotovat kolem jakéhokoliv zvoleného



bodu o libovolný úhel. Matice rotace vypadá následovně:

$$\begin{aligned}
 Rot(x_0, y_0) &= T(-x_0, -y_0)Rot(\varphi)T(x_0, y_0) \\
 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_0 & -y_0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_0 & y_0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ (-x_0\cos(\varphi) & (-x_0\sin(\varphi) & 1 \\ +y_0\sin(\varphi) + x_0) & -y_0\cos(\varphi) + y_0) \end{pmatrix},
 \end{aligned} \tag{4.7}$$

kde  $\varphi$  je velikost úhlu v obloukové (úhel, o který chceme rotovat objekt) a  $x_0$ ,  $y_0$  jsou souřadnice bodu, podle kterého bude objekt rotovat. Body objektu, kterým chceme rotovat, se umístí do matice:

$$\begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix}, \tag{4.8}$$

následně se vynásobí maticí rotace (4.7) a výsledná matice bude obsahovat nové body objektu (body změny polohu aplikováním rotace). Body budou v matici uloženy stejně jako v (4.8). Matici rotace lze získat pomocí metody `double[][] getMatRot(double angle, PPoint center)` (`angle` je úhel ve stupňové míře – metoda si převede na obloukovou míru – a `point` je bod, podle kterého bude provedena rotace), která se nachází ve třídě `PShape`.

#### 4.4.2 Vracení změn

Vracení změn je v aplikaci implementováno pomocí *zásobníku*, do kterého je před změnou kteréhokoliv objektu uložena celá kopie plátna (kopie všech jeho objektů). V *zásobníku* se uchovává maximálně pět pláten. Při stisku klávesové zkratky `Ctrl + Z`, je ze zásobníku vyjmuto plátno (poslední uložené) a to je nastaveno jako aktuální.

#### 4.4.3 Kopírování objektu

Kopírování je provedeno tak, že se po stisku klávesové zkratky `Ctrl + C` do pomocné proměnné uloží kopie vybraného objektu (zavoláním metody `PShape copy()`). Vložení objektu (klávesová zkratka `Ctrl + V`) je provedeno tak, že se kopie objektu z pomocné proměnné vloží do aktuálního plátna.

#### 4.4.4 Označení objektu

Při pohybu kurzoru myši po plátně se testuje, jestli se kurzor nachází v blízkosti nějakého objektu, avšak někdy bývá výpočet náročný (podle typu objektu) a pokud by na plátně bylo umístěno mnoho objektů, pak by aplikace nemusela fungovat plynule. Proto se nejdříve zjišťuje, jestli se kurzor nachází v obdélníkovém „obalu“, tzv. *bounding box* (jednodušší výpočet), který ohraničuje každý objekt.

# 5 Testování

V této kapitole budou popsány jednotlivé testovací případy, které prověří kvalitu výsledné aplikace. Testování je důležitá část vývoje aplikace, která by se neměla opomíjet. Díky testování jsme schopni odhalit případné chyby ještě před samotným používáním aplikace v praxi. Bez testování bychom nemohli zaručit jistou kvalitu aplikace. Samozřejmě i po testování se v aplikaci mohou nacházet nějaké chyby, ale díky testování je výrazně redukovány. Existuje celá řada způsobů, jak můžeme aplikaci testovat: Já si vybral funkční testování, testování podle scénářů a *monkey* testování (nahodilé klikání na komponenty v aplikaci). Důležité bylo především nechat aplikaci otestovat cílovými uživateli (strojaři z KKS), kteří by měli být schopni aplikaci ovládat bez větších obtíží (funkční testování).

## 5.1 Použitelnost v praxi

Testování použitelnosti proběhlo v přítomnosti strojařů z KKS (dále jen uživatel), kterým byla aplikace spuštěna a ponechána k otestování. Uživatel nebyl nijak poučen, jak aplikace funguje, pouze bylo připomenuto k čemu přesně slouží. Uživateli byl zadán úkol, aby vytvořil jednoduchý náčrt libovolného šroubovaného spoje. Bez sebemenších obtíží byl schopný použít základní primitiva pro modelování. Dále mu chybělo jako základní primitivum obdélník, ze kterého ve většině případu vychází při návrhu šroubovaného spoje a z něho dále tvaruje komplexnější tvary.

Umísťování jednotlivých objektů proběhlo úspěšně, jak parametrickým zadáním hodnot v panelu s vlastnostmi, tak za použití kurzoru myši. Parametrické zadávání sice fungovalo dobře, ale uživateli se nelíbilo, jakým způsobem musí hodnoty zadávat. Při zadání souřadnic středu objektu jsou jednotlivé složky od sebe odděleny čárkou (desetinná místa pak oddělena tečkou) v jednom vstupním poli, uživateli by více vyhovovalo, kdyby byla vstupní pole separována (zadání by probíhalo ve dvou polích) a po stisku klávesy **Enter** se automaticky kurzor přesunul do dalšího pole (než klikáním myši). U objektu typu *šroub* chyběl parametr pro možnost zadání jeho velikosti (poloměru závitového dřívku).

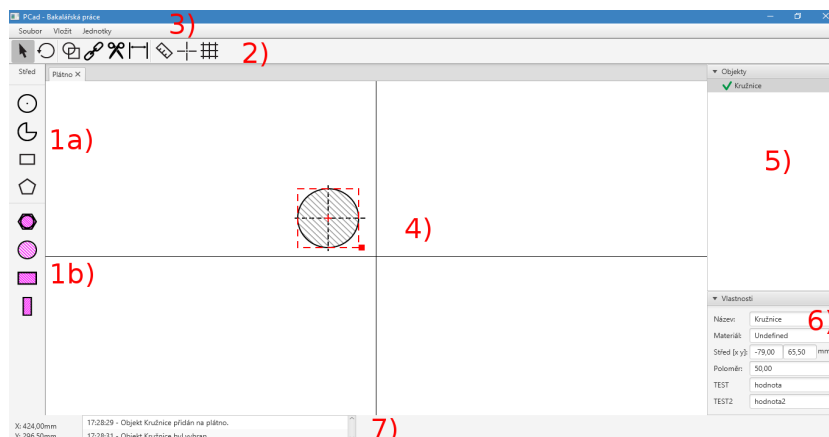
Dále byly vyzkoušeny jednotlivé nástroje pro práci s objekty. Uživatel intuitivně používal i nápovědy jednotlivých nástrojů (při najetí kurzorem myši na daný nástroj), pokud si nebyl zcela jistý, jakou dané tlačítko vykonává funkci. Nejvíce uživatele zajímala funkce, která dokáže nezávislé ob-

jekty ukotvit k sobě a vytvořit tím skupinu. Další velice potřebnou funkcí, kterou uživatel vyzkoušel, byla *mřížka*, která ovšem v té době byla ve vývoji a nemohla tak být otestována její plná funkčnost. Ostatní nevyzkoušené funkce (z panelu nástrojů) byly uživateli představeny a především byl potěšen funkcí *kóta*, která nejen že měří vzdálenost, ale při změně její délky lze posouvat druhým z bodů, který byl zadán (pokud je uchycen k bodu jiného objektu, pak pohne i s tímto objektem). Při manipulaci s objekty na plátně uživateli chyběla akce *zpět*, která by měla umožnit vrátit změny do původního stavu před manipulací s daným objektem, a také možnost zkopírovat objekt a následně ho vložit.

Na konci testování byl uživatel seznámen s ostatními dostupnými funkcemi a vlastnostmi aplikace. Zejména byla vysvětlena pravidla, jak funguje odstraňování jednotlivých prvků ze skupiny a kam je možné umístit definovaný objekt (součástku). Uživatel neměl již žádné další výhrady ani připomínky a tím testování použitelnosti skončilo.

## 5.2 Testování podle scénáře

V této části je napsána řada scénářů, které jsou rozděleny do tří kategorií. Každá kategorie má za úkol otestovat různé vlastnosti aplikace. Uživatel bude postupovat podle pokynů uvedených v jednotlivých scénářích a na základě vykonané akce oznamovat, jestli proběhla úspěšně či neúspěšně (popř. slovně popíše problém, který nastal). Na obrázku 5.1 je vidět GUI aplikace s přidáním očíslováním důležitých komponent, na které se ve scénářích bude odkazovat, aby se uživatel lépe orientoval v aplikaci. Pojmeme objekt se rozumí geometrická primitiva a součástky.



Obrázek 5.1: GUI aplikace

### 5.2.1 Funkčnost

Tento scénář má za úkol ověřit veškeré funkce, které aplikace poskytuje pro modelování šroubovaných spojů. Scénář je rozdělen na dvě části. V první bude testováno přidávání jednotlivých objektů na plátno. Druhá část je zaměřena na testování jednotlivých nástrojů, které umožňují manipulaci s přidávanými objekty.

- **Přidávání objektů** – Veškeré změny provedené parametrickým zadáním v panelu 6) se potvrzují klávesou **Enter**.
  1. K přidávání objektů je nutné vytvořit plátno. To je možné pomocí klávesové zkratky **Ctrl+N** nebo výběru nového plátna v menu 3) – *Soubor*. Povedlo se vytvořit plátno?
  2. Postupně umístěte všechna geometrická primitiva, která se nachází v panelu 1a) na plátno pomocí kurzoru myši. Povedlo se umístit geometrická primitiva?
  3. Postupně umístěte všechny součástky, které se nachází v panelu 1b) na plátno pomocí kurzoru myši. Tyto objekty lze umístit pouze na geometrická primitiva. Povedlo se umístit součástky?
  4. Znovu se pokuste umístit geometrická primitiva, ale teď pomocí parametrického zadání souřadnic středu v panelu 6). Povedlo se umístit geometrická primitiva?
  5. Znovu se pokuste umístit součástky, ale teď pomocí parametrického zadání souřadnic středu v panelu 6). Povedlo se umístit součástky?
  6. Postupně vyberte jednotlivé součástky a pokuste se je umístit pomocí kurzoru myši na součástky, které jsou již umístěny na plátně. Povedlo se umístit součástky?
- **Funkcionalita nástrojů** – Zde uživatel otestuje veškeré nástroje, které se nacházejí v panelu 2), počínaje prvním nástrojem zleva. Uživatel může přidávat nové objekty dle libosti tak, aby byl schopný otestovat daný nástroj.
  1. Zvolte nástroj *Výběr* a pomocí něho postupně označte objekty (v jednom okamžiku lze pouze jeden). Povedlo se?
  2. Zvolte nástroj *Rotace* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se otočit s objektem?

3. Zvolte nástroj *Výběr* a přesuňte libovolný objekt k jinému tak, aby se protínaly, následně zvolte nástroj *Seskupení* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se seskupit objekty?
4. Zvolte nástroj *Výběr* a přesuňte libovolný objekt do jiného tak, aby se celou plochou nacházel uvnitř, následně zvolte nástroj *Ukotvení* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se ukotvit objekt?
5. Zvolte nástroj *Výběr* a přesuňte geometrické primitivum do jiného tak, aby se celou plochou nacházelo uvnitř, následně zvolte nástroj *Vyříznutí* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se vytvořit díru do objektu?
6. Klikněte pravým tlačítkem na objekt *Díra* v panelu 5) a zvolte možnost *Rozdělit*. Zrušila se skupina (tj. díra stala opět pevným tělesem)?
7. Zvolte nástroj *Měření* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se změřit vzdálenost dvou bodů?
8. Klikněte pravým tlačítkem myši na nástroj *Měření* a zvolte možnost *S úpravou vzdálenosti*. První bod umístěte do středu libovolného objektu a druhý též do středu, ale jiného objektu. Následně vyberte kótu a v panelu 6) změňte její délku. Posunul se druhý objekt a ukazuje popisek správnou délku?
9. Aktivujte nástroj *Měřítka* a zkontrolujte, jestli se v levém dolním rohu plátna 4) nachází měřítko. Vykresluje se měřítko?
10. Deaktivujte nástroj *Měřítka* a zkontrolujte, jestli zmizelo měřítko z plátna 4). Přestalo se vykreslovat měřítko?
11. Aktivujte nástroj *Vodící čáry* a zkontrolujte, jestli se na plátně 4) vykreslují vodící čáry (při pohybu kurzoru po plátně). Vykreslují se vodící čáry?
12. Deaktivujte nástroj *Vodící čáry* a zkontrolujte, jestli se na plátně 4) přestaly vykreslovat vodící čáry. Přestaly se vykreslovat vodící čáry?
13. Aktivujte nástroj *Mřížka* a zkontrolujte, jestli se na plátně 4) vykreslují pomocné body. Vykreslují se pomocné body?
14. Po aktivaci nástroje *Mřížka* se pod panel 6) přidal nový panel, kde upravte vzdálenost bodů na ose  $x$  a  $y$  (kladná hodnota a větší než 0). Změnilo se rozmístění bodů *Mřížky*?

15. Zvolte nástroj *Výběr* a pokuste se pohnout libovolným objektem. Uchycuje se objekt k pomocným bodům?
16. Deaktivujte nástroj *Mřížka* a zkontrolujte, jestli se přestaly vykreslovat body. Zmizely body z plátna?
17. V panelu 3) zvolte tlačítko *Jednotky* a zvolte možnost *cm*. Nyní pohněte kurzorem myši po plátně. Provedl se přepočítání jednotek v panelu 7)?

### 5.2.2 Ovladatelnost a manipulace s objekty

V tomto scénáři uživatel interaktivně otestuje ovladatelnost a manipulaci s jednotlivými objekty, které jsou umístěny na plátně.

1. Vytvořte nové plátno, na které umístíte veškeré dostupné základní objekty z panelu 1a). Povedlo se to?
2. Označte objekt, pomocí kurzoru myši ho přesuňte na jinou pozici a porovnejte jeho souřadnice středu v panelu 6) s aktuální pozicí kurzoru myši v panelu 7) (*X* a *Y* souřadnice). Shodují se souřadnice v panelu 6)? Toto opakujte pro všechny ostatní objekty.
3. Označte objekt a v panelu 6) upravte jeho název a materiál. Byl objekt přejmenován i v panelu 5)? Zůstaly hodnoty uloženy i v panelu 6)? Toto opakujte pro všechny ostatní objekty.
4. Označte objekt *Kružnice* a pomocí kurzoru myši změňte jeho velikost (táhnutím červeného bodu). Zvolte nástroj *Měření* a změřte jeho poloměr. Shoduje se naměřený poloměr s poloměrem v panelu 6) (nemusí být přesné, ale odchylka by neměla být více jak 0,5 mm)?
5. Označte objekt *Kruhová výseč* a pomocí kurzoru myši změňte jeho velikost (táhnutím červeného bodu). Zvolte nástroj *Měření* a změřte jeho poloměr. Shoduje se naměřený poloměr s poloměrem v panelu 6) (nemusí být přesné, ale odchylka by neměla být více jak 0,5 mm)?
6. Označte objekt *Kruhová výseč* a pomocí kurzoru myši změňte polohu jednoho z modrých bodů tak, abyste docílili změny úhlu na 180°. Povedlo se změnit úhel?
7. Označte objekt *Obdélník*, přesuňte libovolný modrý bod a poznamenejte si jeho polohu (souřadnice v panelu 7)). Nachází bod s těmito souřadnicemi i v seznamu bodů v panelu 6)?

8. Označte objekt *Polygon* a proveďte to samé, co s obdélníkem. Dosáhli jste stejného výsledku?
9. Postupně označujte objekty a odstraňte je z plátna pomocí klávesy Delete nebo klikněte pravým tlačítkem myši na objekt v panelu 5) a vyberte možnost smazat (kombinujte obě možnosti). Povedlo se smazat objekty z plátna?
10. Vytvořte jeden velký obdélník, na který umístíte všechny součástky z panelu 1b). Povedlo se je umístit?
11. Pro veškeré umístěné součástky proveďte akci 2., 3. a vyzkoušejte změnu jejich velikosti pomocí červeného bodu tak, aby se vzájemně překrývaly. Zobrazil se v panelu 5) u jednotlivých součástí červený vykřičník? Posléze všechny objekty smažte.
12. Umístíte všechny základní objekty na plátno vedle sebe a na každém proveďte jednu transformaci. Nyní pomocí klávesové zkratky **Ctrl + Z** vraťte objekty do původního stavu (maximálně lze vrátit 5 změn). Podařilo se vrátit objekty do původního stavu?
13. Každý objekt se pokuste klávesovou zkratkou **Ctrl + C** zkopírovat a následně stiskem **Ctrl + V** objekt vložit na plátno. V panelu 5) sledujte, jestli se zkopírované objekty skutečně přidávají. Povedlo se zkopírovat objekty?

### 5.2.3 Kontrola uložení/načtení objektů a plátna

V tomto scénáři uživatel otestuje správnou funkčnost ukládání objektů/plátna do souboru a jejich pozdější načtení zpět do aplikace.

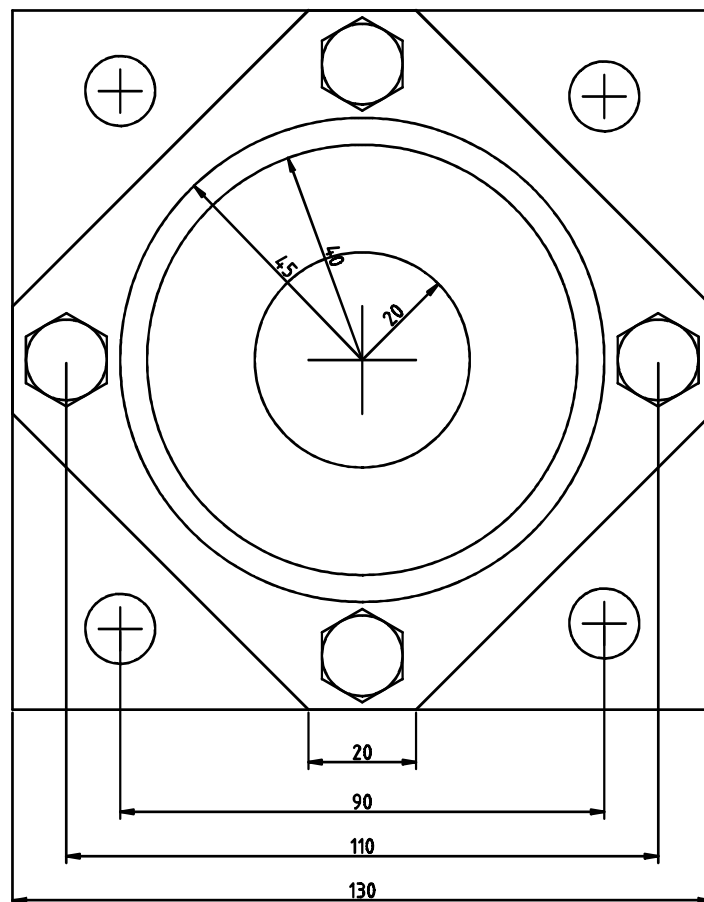
1. Vytvořte nové plátno a umístíte na něj několik různých objektů (alespoň 4) a pomocí klávesové zkratky **Ctrl + S** uložte pod libovolným jménem. Vytvořil se výsledný soubor?
2. Vyberte si jeden z objektů, který přejmenujte a změňte mu materiál (libovolný text). V panelu 5) na něj klikněte pravým tlačítkem myši a zvolte *Uložit*. Vytvořil se výsledný soubor?
3. Nyní vytvořte nové plátno z panelu 3) pod tlačítkem *Vložit* nebo stiskem klávesové zkratky **Ctrl + I** načtete uložený objekt. Přidal se objekt na plátno?



4. Pomocí tlačítka *Soubor* z panelu 3) zvolte *Načíst plátno* nebo stiskněte klávesovou zkratku **Ctrl + M** a vyberte uložené plátno. Přidalo se uložené plátno do aplikace?
5. Otevřete soubor s uloženým objektem v textovém formátu, smažte část jeho obsahu a pokuste se objekt načíst v aplikaci. Podařilo se objekt načíst?
6. Otevřete soubor s uloženým plátnem v textovém formátu, smažte část jeho obsahu a pokuste se plátno načíst v aplikaci. Podařilo se plátno načíst?

### 5.2.4 Modelování návrhu spoje

V tomto scénáři se uživatel pokusí podle zadané předlohy navrhnout šroubovaný spoj v aplikaci. Na obrázku 5.2 je zobrazen výkres jednoduchého šroubovaného spoje, který slouží jako předloha.



Obrázek 5.2: Vzor šroubovaného spoje

- Pokuste se v aplikaci navrhnout podobný spoj, jako je zobrazen na obrázku 5.2. Podařilo se navrhnout šroubovaný spoj? Pokud ne, popište v čem byly problémy.

## 5.3 Monkey testování

*Monkey* testování spočívá v náhodném klikání na komponenty v aplikaci. Díky tomu je možné zachytit chyby, které by mohly zapříčinit celkovou nefunkčnost aplikace, ale ostatní testy tyto chyby nemusely odhalit, protože se při tomto typu testování aplikace ovládá nezvyklým způsobem (různé kombinace nástrojů, zběsilé klikání všude možné, apod.). Každý z uživatelů, který se na testování podílel, měl za úkol splnit i následující body:

1. Klikajte na všechna možná tlačítka, která aplikace nabízí, bez předem vytvořeného plátna. Stalo se něco neobvyklého?
2. Proveďte to samé, co v bodě 1, ale s vytvořeným plátnem. Stalo se něco neobvyklého?
3. Přidejte na plátno několik objektů a náhodně se snažte pomocí panelu vlastností upravit dostupné hodnoty vkládáním různých znaků. Stalo se něco neobvyklého?
4. Odstraňte všechny objekty z plátna a pro každou funkci z panelu nástrojů náhodně klikajte po plátně. Stalo se něco neobvyklého?
5. V posledním kroku zavřete záložku s plátnem a proveďte to samé, co v bodě 1. Stalo se něco neobvyklého?

U každého z uvedených kroků odpovězte ano či ne. Pokud je odpověď ano, stručně popište, co se stalo a jak k tomu došlo.

## 5.4 Shrnutí

Při opravách jsem se zejména zaměřil na chyby nalezené po testu použitelnosti 5.1. Testování použitelnosti odhalilo několik chyb, jako např.: nemožnost zadat velikost šroubu a špatná funkce nástroje *vyříznutí*, které byly posléze opraveny. Dále byly přidány nové vlastnosti, které strojaři uvedli v 5.1 jako např.: rozdělení pole pro zadávání souřadnic a automatický posun klávesou *Enter*, přidání primitiva Obdélník, přepracování mřížky a vytvoření funkce pro kopírování objektů a vracení změn.

Některé drobné chyby byly ještě nalezeny během testování podle scénáře, šlo o špatný přepoččet velikosti šroubu, nepřesné přidání objektů na plátno po parametrickém zadání souřadnic a chybné uložení/načtení plátna. *Monkey* testování neodhalilo žádné podivné chování aplikace. Všechny chyby byly opraveny a autorem byla ověřena správná funkčnost.

## 6 Závěr

V této práci jsem se účelově seznámil s funkcemi uživatelských rozhraní běžně používaných, ale i volně dostupných CAD systémů. Na základě získaných poznatků a požadavků konstruktérů z KKS jsem nejprve navrhl a následně implementoval intuitivní (jak prokázalo testování v praxi) aplikaci, která má konstruktérům usnadnit práci při návrhu geometrie šroubovaných spojů. Aplikace je snadná na ovládání a i nezkušený uživatel je schopný navrhnout jednoduchý šroubovaný spoj. Grafické uživatelské prostředí je navrženo minimalisticky, ale s ohledem na klasické rozmístění komponent a nástrojů v běžně používaných CAD systémech.

Žádná jiná aplikace, která by přesně splňovala požadavky konstruktérů KKS neexistuje, a proto je tato aplikace zcela unikátní.

Velkou výhodou aplikace je, že díky zvolenému výstupnímu formátu XML je možné pomocí XSLT uložené objekty transformovat do jakéhokoliv jiného formátu a přenést tak objekt do jiné aplikace.

Uživatelé, kteří aplikaci testovali, ji hodnotili jak po funkční, tak po grafické stránce velice pozitivně. Líbilo se jim jednoduché a srozumitelné ovládání.

Budoucím rozšířením aplikace by mohlo být přidání nových funkcí, jako je např. kruhové pole. Kruhové pole slouží k umístění určitého počtu objektů se zvoleným odstupem po obvodu kružnice. Dalším možným rozšířením může být přidání předem vytvořených XSLT souborů, aby samotná aplikace nabízela uložení návrhu šroubovaného spoje do běžně používaných formátů.

# Literatura

- [1] *Ant* [online]. The Apache Ant Project, 2017. [cit. ]. Dostupné z: <http://ant.apache.org/>.
- [2] *AutoCad* [online]. Autodesk, Inc, 2017. [cit. ]. Dostupné z: <http://www.autodesk.cz/products/autocad/overview>.
- [3] BARTSCH, H.-J. *Matematické vzorce*. Mladá fronta, 1996. ISBN 80-204-0607-7.
- [4] BOLEK, A. a. J. K. *Části strojů. 1. svazek*. SNTL, 1989. ISBN 80-0-00046-7.
- [5] *CATIA* [online]. Dassault Systemes, 2017. [cit. ]. Dostupné z: <https://www.3ds.com/products-services/catia/>.
- [6] DUNCAN, M. *Applied Geometry for Computer Graphics and CAD*. Springer, 2005. ISBN 1-85233-801-6.
- [7] *Git* [online]. Software Freedom Conservancy, Inc., 2017. [cit. ]. Dostupné z: <https://git-scm.com/sfc>.
- [8] *Java* [online]. Oracle, 2017. [cit. ]. Dostupné z: <http://www.oracle.com/technetwork/java/javase>.
- [9] *Swing* [online]. Oracle, 2016. [cit. ]. Java SE Documentation. Dostupné z: <https://docs.oracle.com/javase/8/docs/technotes/guides/swing/>.
- [10] *Java Virtual Machine Technology* [online]. Oracle, 2016. [cit. ]. Java SE Documentation. Dostupné z: <https://docs.oracle.com/javase/8/docs/technotes/guides/vm/>.
- [11] *Framework Qt* [online]. Qt Company, 2017. [cit. ]. Dostupné z: <https://www.qt.io/>.
- [12] *Ray-crossing (Crossing Number Method)* [online]. softSurfer, 2012. [cit. ]. Dostupné z: [http://geomalgorithms.com/a03-\\_inclusion.html](http://geomalgorithms.com/a03-_inclusion.html).
- [13] RUDOLF, P. *Návrhové vzory*. Computer Press, 2015. ISBN 978-80-251-2126-9.
- [14] RUDOLF, P. *OOP: Naučte se myslet a programovat objektově*. Computer Press, 2010. ISBN 978-80-251-2126-9.
- [15] SHARAN, K. *Learn JavaFX 8: Building User Experience and Interfaces with Java 8*. Apress, 2015. ISBN 978-1-4842-1142-7.

- [16] *XML format* [online]. W3C, 2008. [cit. ]. Dostupné z:  
<https://www.w3.org/TR/xml/>.
- [17] *XSL* [online]. W3C, 2006. [cit. ]. Dostupné z:  
<https://www.w3.org/TR/xsl/>.
- [18] *XSLT* [online]. W3C, 2008. [cit. ]. Dostupné z:  
<https://www.w3.org/TR/xslt>.

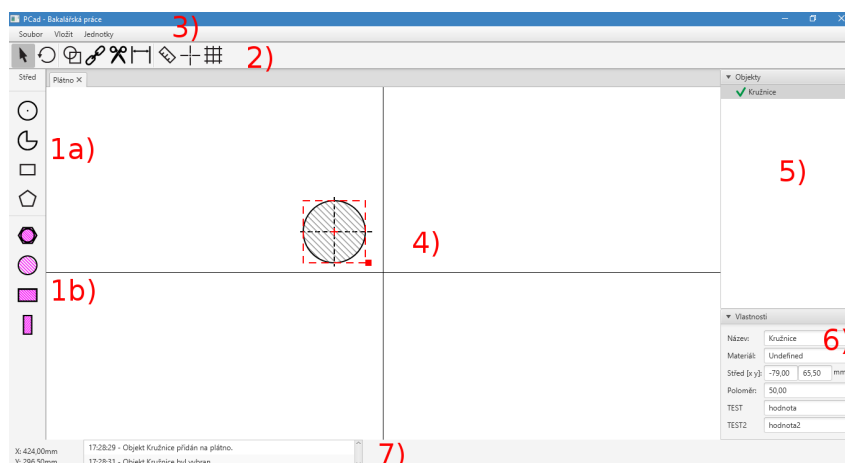
# A Uživatelská příručka

Ke spuštění aplikace je nutné mít nainstalovanou Javu [8]. Aplikace byla vyvíjena pod verzí 1.8.0\_92, ale byla vyzkoušena i na verzi 1.8.0\_131 pod Microsoft Windows 10. Aplikaci je možné spustit dvěma způsoby:

1. Spustit přímo z vytvořeného *jar* souboru, který se nachází v adresáři *aplikace* a jmenuje se *PCad.jar*. Aplikaci můžeme spustit dvojným kliknutím levého tlačítka myši (nebo napsáním příkazu `java -jar PCad.jar` v příkazové řádce).
2. Vygenerovat si spustitelný *jar* soubor přímo ze zdrojového kódu pomocí nástroje *Ant* [1] (vyzkoušenou na verzi 1.10.0). Stačí spustit příkazovou řádku a přesunout se do kořenové složky a napsat příkaz `ant` – ten za pomoci přiloženého souboru *build.xml* vytvoří spustitelný soubor *PCad.jar* (taktéž do kořenového adresáře), kterým můžeme aplikaci spustit dvojným kliknutím levého tlačítka myši (nebo napsáním příkazu `java -jar PCad.jar` v příkazové řádce).

## A.1 Popis GUI

Aplikace se ovládá velice podobně jako běžně používané CAD systémy, takže znalý člověk by neměl mít problém s ovládáním této aplikace. Na obrázku A.1 je možné vidět grafické uživatelské rozhraní s očíslovanými panely, které budou dále blíže popsány.



Obrázek A.1: GUI aplikace

### **A.1.1 Panel s geometrickými primitivy**

Panel s geometrickými primitivy se na obrázku A.1 nachází pod označením **1a**). Jednotlivé ikony není nutné popisovat, jelikož jasně znázorňují, které primitivum reprezentuje. Stačí kliknout na příslušnou ikonu a poté je možné pomocí kurzoru myši nebo zadáním přesných hodnot v panelu **6**) umístit dané primitivum na plátno. Navíc se zde nachází tlačítko *Střed*, který přesune pohled do středu plátna.

### **A.1.2 Panel se součástkami**

Panel se součástkami se na obrázku A.1 nachází pod označením **1b**). Při najetí kurzorem myši na příslušnou ikonu je možné zjistit o jakou součástku se jedná. Stačí kliknout na příslušnou ikonu a poté je možné pomocí kurzoru myši nebo zadáním přesných hodnot v panelu **6**) umístit danou součástku na plátno.

### **A.1.3 Panel s nástroji**

Panel s nástroji se na obrázku A.1 nachází pod označením **2**). Zde jsou umístěny veškeré nástroje určené k manipulaci s objekty, které aplikace nabízí. Při najetí kurzorem myši na příslušnou ikonu je možné zjistit o jaký nástroj se jedná a dále postupovat podle instrukcí, které nástroj nabízí.

### **A.1.4 Panel s menu**

Panel s nástroji se na obrázku A.1 nachází pod označením **3**). Je to klasické menu, na které je uživatel zvyklý. Nachází se zde možnosti jako: vytvořit nové plátno, uložit plátno, načíst plátno, ukončit aplikaci, načíst objekt a přepočítat jednotky.

### **A.1.5 Panel s plátnem**

Panel s plátnem se na obrázku A.1 nachází pod označením **4**). Je to plátno, kam se umísťují jednotlivé objekty. Pomocí klávesové zkratky *Delete* je možné odstranit označený objekt.

### **A.1.6 Panel s umístěnými objekty**

Panel s umístěnými objekty se na obrázku A.1 nachází pod označením **5**). Panel slouží k zobrazení veškerých umístěných objektů na plátně **4**). Pokud



na objekt klikneme pravým tlačítkem myši je možné uložit vybraný objekt nebo ho odstranit (popř. ho oddělit od skupiny). Oddělit objekt od skupiny lze pouze tehdy, pokud se uvnitř něj nenachází jiný objekt.

### A.1.7 Panel s vlastnostmi objektu

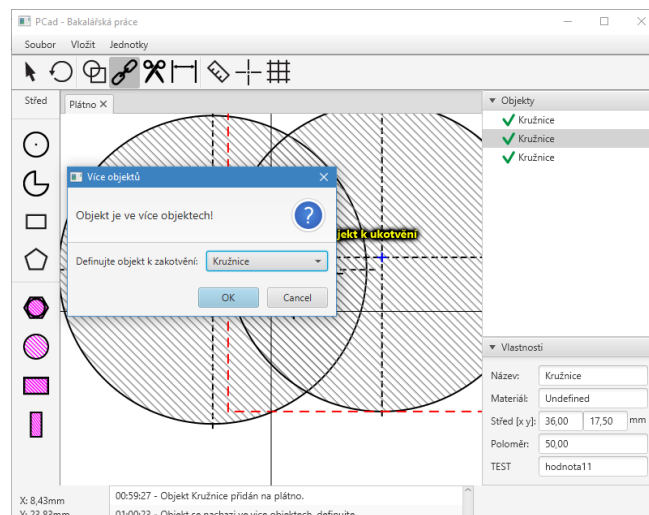
Panel s vlastnostmi objektu se na obrázku A.1 nachází pod označením 6). Zde je možné upravovat veškeré vlastnosti označeného objektu. Úprava každé informace je nutné potvrdit klávesou Enter. Veškeré číselné hodnoty se zadávají v *mm* pokud není popiskem uvedeno jinak.

### A.1.8 Informační panel

Informační panel se na obrázku A.1 nachází pod označením 7). Nachází se zde aktuální pozice kurzoru myši na plátně a popisky po provedení různých akcí.

### A.1.9 Dialogové okno k určení objektu

Pokud provedeme akci *Ukotvení* nebo *Vyříznutí* a požadovaný objekt se nachází současně celou svou plochou na ploše více objektů, které nejsou součástí jedné skupiny, je uživatel vyzván k výběru objektu, pro který se akce provede. Tento dialog je možné vidět na obrázku A.2.



Obrázek A.2: Výběr objektu

### A.1.10 Klávesové zkratky

Zde jse výčet jednotlivých klávesových zkratek, které aplikace nabízí:

- Ctrl + N – vytvoření nového plátna.
- Ctrl + M – načíst existující plátno ze souboru.
- Ctrl + S – uložit aktivní plátno.
- Ctrl + I – načíst existující objekt ze souboru.
- Ctrl + C – kopírování označeného objektu.
- Ctrl + V – vložení zkopírovaného objektu.
- Ctrl + Z – vrácení změn.
- Esc – zvolení nástroje *Výběr*.
- Delete – odstranění označeného objektu z plátna.
- Rolováním kolečka myši – škálování plátna.
- Přidržení prostředního tlačítka myši – pohyb po plátně.
- Ctrl + levé tlačítko myši – následným posunem myši se lze pohybovat po plátně.
- Shift + levé tlačítko myši – odebrání bodů polygonu na kterém se nachází kurzor myši.
- Dvojklik levého tlačítka myši – přidání bodů polygonu na hranu polygonu, kde se nachází kurzor myši.

# B Záznamy z testování

## B.1 Uživatel 1

- **Počítačová gramotnost** – Pokročilý.
- **Znalost v oblasti návrhu spojů** – Částečně znalý, student FST (Fakulta strojní).
- **Operační systém** – Windows 10 64-Bit.
- **Verze Javy** – 1.8.0\_92.

### B.1.1 Funkčnost

- **Přidávání objektů** – Veškeré změny provedené parametrickým zadáním v panelu 6) se potvrzují klávesou **Enter**.
  1. K přidávání objektů je nutné vytvořit plátno. To je možné pomocí klávesové zkratky **Ctrl+N** nebo výběru nového plátna v menu 3) – *Soubor*. Povedlo se vytvořit plátno?  
**Ano**
  2. Postupně umístěte všechna geometrická primitiva, která se nachází v panelu 1a) na plátno pomocí kurzoru myši. Povedlo se umístit geometrická primitiva?  
**Ano**
  3. Postupně umístěte všechny součástky, které se nachází v panelu 1b) na plátno pomocí kurzoru myši. Tyto objekty lze umístit pouze na geometrická primitiva. Povedlo se umístit součástky?  
**Ano**
  4. Znovu se pokuste umístit geometrická primitiva, ale teď pomocí parametrického zadání souřadnic středu v panelu 6). Povedlo se umístit geometrická primitiva?  
**Ne**
  5. Znovu se pokuste umístit součástky, ale teď pomocí parametrického zadání souřadnic středu v panelu 6). Povedlo se umístit součástky?  
**Ne**

6. Postupně vyberte jednotlivé součástky a pokuste se je umístit pomocí kurzoru myši na součástky, které jsou již umístěny na plátně. Povedlo se umístit součástky?

**Ne**

- **Funkcionalita nástrojů**

1. Zvolte nástroj *Výběr* a pomocí něho postupně označte objekty (v jednom okamžiku lze pouze jeden). Povedlo se?

**Ano**

2. Zvolte nástroj *Rotace* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se otočit s objektem?

**Ano**

3. Zvolte nástroj *Výběr* a přesuňte libovolný objekt k jinému tak, aby se protínaly, následně zvolte nástroj *Seskupení* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se seskupit objekty?

**Ano**

4. Zvolte nástroj *Výběr* a přesuňte libovolný objekt do jiného tak, aby se celou plochou nacházel uvnitř, následně zvolte nástroj *Ukotvení* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se ukotvit objekt?

**Ano**

5. Zvolte nástroj *Výběr* a přesuňte geometrické primitivum do jiného tak, aby se celou plochou nacházelo uvnitř, následně zvolte nástroj *Vyříznutí* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se vytvořit díru do objektu?

**Ano**

6. Klikněte pravým tlačítkem na objekt *Díra* v panelu 5) a zvolte možnost *Rozdělit*. Zrušila se skupina (tj. díra stala opět pevným tělesem)?

**Ano**

7. Zvolte nástroj *Měření* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se změřit vzdálenost dvou bodů?

**Ano**

8. Klikněte pravým tlačítkem myši na nástroj *Měření* a zvolte možnost *S úpravou vzdálenosti*. První bod umístěte do středu libovolného objektu a druhý též do středu, ale jiného objektu. Následně vyberte kótu a v panelu 6) změňte její délku. Posunul se druhý

objekt a ukazuje popisek správnou délkou?

**Ano**

9. Aktivujte nástroj *Měřítko* a zkontrolujte, jestli se v levém dolním rohu plátna 4) nachází měřítko. Vykresluje se měřítko?

**Ano**

10. Deaktivujte nástroj *Měřítko* a zkontrolujte, jestli zmizelo měřítko z plátna 4). Přestalo se vykreslovat měřítko?

**Ano**

11. Aktivujte nástroj *Vodící čáry* a zkontrolujte, jestli se na plátně 4) vykreslují vodící čáry (při pohybu kurzoru po plátně). Vykreslují se vodící čáry?

**Ano**

12. Deaktivujte nástroj *Vodící čáry* a zkontrolujte, jestli se na plátně 4) přestaly vykreslovat vodící čáry. Přestaly se vykreslovat vodící čáry?

**Ano**

13. Aktivujte nástroj *Mřížka* a zkontrolujte, jestli se na plátně 4) vykreslují pomocné body. Vykreslují se pomocné body?

**Ano**

14. Po aktivaci nástroje *Mřížka* se pod panel 6) přidal nový panel, kde upravte vzdálenost bodů na ose  $x$  a  $y$  (kladná hodnota a větší než 0). Změnilo se rozmístění bodů *Mřížky*?

**Ano**

15. Zvolte nástroj *Výběr* a pokuste se pohnout libovolným objektem. Uchycuje se objekt k pomocným bodům?

**Ano**

16. Deaktivujte nástroj *Mřížka* a zkontrolujte, jestli se přestaly vykreslovat body. Zmizely body z plátna?

**Ano**

17. V panelu 3) zvolte tlačítko *Jednotky* a zvolte možnost *cm*. Nyní pohněte kurzorem myši po plátně. Provedl se přepočít jednotek v panelu 7)?

**Ano**

## B.1.2 Ovladatelnost a manipulace s objekty

1. Vytvořte nové plátno, na které umístíte veškeré dostupné základní objekty z panelu 1a). Povedlo se to?

**Ano**

2. Označte objekt, pomocí kurzoru myši ho přesuňte na jinou pozici a porovnejte jeho souřadnice středu v panelu 6) s aktuální pozicí kurzoru myši v panelu 7) ( $X$  a  $Y$  souřadnice). Shodují se souřadnice v panelu 6)? Toto opakujte pro všechny ostatní objekty.

**Ano**

3. Označte objekt a v panelu 6) upravte jeho název a materiál. Byl objekt přejmenován i v panelu 5)? Zůstaly hodnoty uloženy i v panelu 6)? Toto opakujte pro všechny ostatní objekty.

**Ano**

4. Označte objekt *Kružnice* a pomocí kurzoru myši změňte jeho velikost (táhnutím červeného bodu). Zvolte nástroj *Měření* a změřte jeho poloměr. Shoduje se naměřený poloměr s poloměrem v panelu 6) (nemusí být přesné, ale odchylka by neměla být více jak 0,5 mm)?

**Ano**

5. Označte objekt *Kruhová výseč* a pomocí kurzoru myši změňte jeho velikost (táhnutím červeného bodu). Zvolte nástroj *Měření* a změřte jeho poloměr. Shoduje se naměřený poloměr s poloměrem v panelu 6) (nemusí být přesné, ale odchylka by neměla být více jak 0,5 mm)?

**Ano**

6. Označte objekt *Kruhová výseč* a pomocí kurzoru myši změňte polohu jednoho z modrých bodů tak, abyste docílili změny úhlu na  $180^\circ$ . Povedlo se změnit úhel?

**Ano**

7. Označte objekt *Obdélník*, přesuňte libovolný modrý bod a poznamenejte si jeho polohu (souřadnice v panelu 7)). Nachází bod s těmito souřadnicemi i v seznamu bodů v panelu 6)?

**Ano**

8. Označte objekt *Polygon* a proveďte to samé, co s obdélníkem. Dosáhli jste stejného výsledku?

**Ano**

9. Postupně označujte objekty a odstraňte je z plátna pomocí klávesy Delete nebo klikněte pravým tlačítkem myši na objekt v panelu 5) a vyberte možnost smazat (kombinujte obě možnosti). Povedlo se smazat objekty z plátna?

**Ano**

10. Vytvořte jeden velký obdélník, na který umístíte všechny součástky z panelu 1b). Povedlo se je umístit?

**Ano**

11. Pro veškeré umístěné součástky proveďte akci 2., 3. a vyzkoušejte změnu jejich velikosti pomocí červeného bodu tak, aby se vzájemně překrývaly. Zobrazil se v panelu 5) u jednotlivých součástí červený vykřičník? Posléze všechny objekty smažte.

**Ano**

12. Umístíte všechny základní objekty na plátno vedle sebe a na každém proveďte jednu transformaci. Nyní pomocí klávesové zkratky **Ctrl + Z** vraťte objekty do původního stavu (maximálně lze vrátit 5 změn). Podařilo se vrátit objekty do původního stavu?

**Ano**

13. Každý objekt se pokuste klávesovou zkratkou **Ctrl + C** zkopírovat a následně stiskem **Ctrl + V** objekt vložit na plátno. V panelu 5) sledujte, jestli se zkopírované objekty skutečně přidávají. Povedlo se zkopírovat objekty?

**Ano**

### **B.1.3 Kontrola uložení/načtení objektů a plátna**

1. Vytvořte nové plátno a umístíte na něj několik různých objektů (alespoň 4) a pomocí klávesové zkratky **Ctrl + S** uložte pod libovolným jménem. Vytvořil se výsledný soubor?

**Ne**

2. Vyberte si jeden z objektů, který přejmenujte a změňte mu materiál (libovolný text). V panelu 5) na něj klikněte pravým tlačítkem myši a zvolte *Uložit*. Vytvořil se výsledný soubor?

**Ano**

3. Nyní vytvořte nové plátno z panelu 3) pod tlačítkem *Vložit* nebo stiskem klávesové zkratky **Ctrl + I** načtete uložený objekt. Přidal se objekt na plátno?

**Ano**

4. Pomocí tlačítka *Soubor* z panelu 3) zvolte *Načíst plátno* nebo stiskněte klávesovou zkratku **Ctrl + M** a vyberte uložené plátno. Přidalo se uložené plátno do aplikace?

**Ne**

5. Otevřete soubor s uloženým objektem v textovém formátu, smažte část jeho obsahu a pokuste se objekt načíst v aplikaci. Podařilo se objekt načíst?

**Ne**

6. Otevřete soubor s uloženým plátnem v textovém formátu, smažte část jeho obsahu a pokuste se plátno načíst v aplikaci. Podařilo se plátno načíst?

**Ne, plátno nešlo uložit, proto nemohlo být načteno**

### B.1.4 Modelování návrhu spoje

- Pokuste se v aplikaci navrhnout podobný spoj, jako je zobrazen na obrázku 5.2. Podařilo se navrhnout šroubovaný spoj? Pokud ne, popište v čem byly problémy.

**Ano, povedlo se**

## B.2 Monkey testování

1. Klikajte na všechna možná tlačítka, která aplikace nabízí, bez předem vytvořeného plátna. Stalo se něco neobvyklého?

**Nestalo**

2. Proveďte to samé, co v bodě 1, ale s vytvořeným plátnem. Stalo se něco neobvyklého?

**Nestalo**

3. Přidejte na plátno několik objektů a náhodně se snažte pomocí panelu vlastností upravit dostupné hodnoty vkládáním různých znaků. Stalo se něco neobvyklého?

**Nestalo**

4. Odstraňte všechny objekty z plátna a pro každou funkci z panelu nástrojů náhodně klikajte po plátně. Stalo se něco neobvyklého?

**Nestalo**

5. V posledním kroku zavřete záložku s plátnem a proveďte to samé, co v bodě 1. Stalo se něco neobvyklého?

**Nestalo**



## B.3 Uživatel 2

- **Počítačová gramotnost** – Pokročilý.
- **Znalost v oblasti návrhu spojů** – Poučen.
- **Operační systém** – Windows 10 64-Bit.
- **Verze Javy** – 1.8.0\_131.

### B.3.1 Funkčnost

- **Přidávání objektů** – Veškeré změny provedené parametrickým zadáním v panelu 6) se potvrzují klávesou **Enter**.
  1. K přidávání objektů je nutné vytvořit plátno. To je možné pomocí klávesové zkratky **Ctrl+N** nebo výběru nového plátna v menu 3) – *Soubor*. Povedlo se vytvořit plátno?  
**Ano**
  2. Postupně umístěte všechna geometrická primitiva, která se nachází v panelu 1a) na plátno pomocí kurzoru myši. Povedlo se umístit geometrická primitiva?  
**Ano**
  3. Postupně umístěte všechny součástky, které se nachází v panelu 1b) na plátno pomocí kurzoru myši. Tyto objekty lze umístit pouze na geometrická primitiva. Povedlo se umístit součástky?  
**Ano**
  4. Znovu se pokuste umístit geometrická primitiva, ale teď pomocí parametrického zadání souřadnic středu v panelu 6). Povedlo se umístit geometrická primitiva?  
**Ne**
  5. Znovu se pokuste umístit součástky, ale teď pomocí parametrického zadání souřadnic středu v panelu 6). Povedlo se umístit součástky?  
**Ne**
  6. Postupně vyberte jednotlivé součástky a pokuste se je umístit pomocí kurzoru myši na součástky, které jsou již umístěny na plátně. Povedlo se umístit součástky?  
**Ne**
- **Funkcionalita nástrojů**

1. Zvolte nástroj *Výběr* a pomocí něho postupně označte objekty (v jednom okamžiku lze pouze jeden). Povedlo se?  
**Ano**
2. Zvolte nástroj *Rotace* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se otočit s objektem?  
**Ano**
3. Zvolte nástroj *Výběr* a přesuňte libovolný objekt k jinému tak, aby se protínaly, následně zvolte nástroj *Seskupení* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se seskupit objekty?  
**Ano**
4. Zvolte nástroj *Výběr* a přesuňte libovolný objekt do jiného tak, aby se celou plochou nacházel uvnitř, následně zvolte nástroj *Ukotvení* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se ukotvit objekt?  
**Ano**
5. Zvolte nástroj *Výběr* a přesuňte geometrické primitivum do jiného tak, aby se celou plochou nacházelo uvnitř, následně zvolte nástroj *Vyříznutí* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se vytvořit díru do objektu?  
**Ano**
6. Klikněte pravým tlačítkem na objekt *Díra* v panelu 5) a zvolte možnost *Rozdělit*. Zrušila se skupina (tj. díra stala opět pevným tělesem)?  
**Ano**
7. Zvolte nástroj *Měření* a dále postupujte podle instrukcí, které nástroj nabízí. Povedlo se změřit vzdálenost dvou bodů?  
**Ano**
8. Klikněte pravým tlačítkem myši na nástroj *Měření* a zvolte možnost *S úpravou vzdálenosti*. První bod umístěte do středu libovolného objektu a druhý též do středu, ale jiného objektu. Následně vyberte kótu a v panelu 6) změňte její délku. Posunul se druhý objekt a ukazuje popisek správnou délku?  
**Ano**
9. Aktivujte nástroj *Měřítka* a zkontrolujte, jestli se v levém dolním rohu plátna 4) nachází měřítka. Vykresluje se měřítka?  
**Ano**
10. Deaktivujte nástroj *Měřítka* a zkontrolujte, jestli zmizela měřítka

z plátna 4). Přestalo se vykreslovat měřítko?

**Ano**

11. Aktivujte nástroj *Vodící čáry* a zkontrolujte, jestli se na plátně 4) vykreslují vodící čáry (při pohybu kurzoru po plátně). Vykreslují se vodící čáry?

**Ano**

12. Deaktivujte nástroj *Vodící čáry* a zkontrolujte, jestli se na plátně 4) přestaly vykreslovat vodící čáry. Přestaly se vykreslovat vodící čáry?

**Ano**

13. Aktivujte nástroj *Mřížka* a zkontrolujte, jestli se na plátně 4) vykreslují pomocné body. Vykreslují se pomocné body?

**Ano**

14. Po aktivaci nástroje *Mřížka* se pod panel 6) přidal nový panel, kde upravte vzdálenost bodů na ose  $x$  a  $y$  (kladná hodnota a větší než 0). Změnilo se rozmístění bodů *Mřížky*?

**Ano**

15. Zvolte nástroj *Výběr* a pokuste se pohnout libovolným objektem. Uchycuje se objekt k pomocným bodům?

**Ano**

16. Deaktivujte nástroj *Mřížka* a zkontrolujte, jestli se přestaly vykreslovat body. Zmizely body z plátna?

**Ano**

17. V panelu 3) zvolte tlačítko *Jednotky* a zvolte možnost *cm*. Nyní pohněte kurzorem myši po plátně. Provedl se přepočítání jednotek v panelu 7)?

**Ano**

### B.3.2 Ovladatelnost a manipulace s objekty

1. Vytvořte nové plátno, na které umístíte veškeré dostupné základní objekty z panelu 1a). Povedlo se to?

**Ano**

2. Označte objekt, pomocí kurzoru myši ho přesuňte na jinou pozici a porovnejte jeho souřadnice středu v panelu 6) s aktuální pozicí kurzoru myši v panelu 7) ( $X$  a  $Y$  souřadnice). Shodují se souřadnice v panelu 6)? Toto opakujte pro všechny ostatní objekty.

**Ano**

3. Označte objekt a v panelu 6) upravte jeho název a materiál. Byl objekt přejmenován i v panelu 5)? Zůstaly hodnoty uloženy i v panelu 6)? Toto opakujte pro všechny ostatní objekty.  
**Ano**
4. Označte objekt *Kružnice* a pomocí kurzoru myši změňte jeho velikost (táhnutím červeného bodu). Zvolte nástroj *Měření* a změřte jeho poloměr. Shoduje se naměřený poloměr s poloměrem v panelu 6) (nemusí být přesné, ale odchylka by neměla být více jak 0,5 mm)?  
**Ano**
5. Označte objekt *Kruhová výseč* a pomocí kurzoru myši změňte jeho velikost (táhnutím červeného bodu). Zvolte nástroj *Měření* a změřte jeho poloměr. Shoduje se naměřený poloměr s poloměrem v panelu 6) (nemusí být přesné, ale odchylka by neměla být více jak 0,5 mm)?  
**Ano**
6. Označte objekt *Kruhová výseč* a pomocí kurzoru myši změňte polohu jednoho z modrých bodů tak, abyste docílili změny úhlu na  $180^\circ$ . Povedlo se změnit úhel?  
**Ano**
7. Označte objekt *Obdélník*, přesuňte libovolný modrý bod a poznamenejte si jeho polohu (souřadnice v panelu 7)). Nachází bod s těmito souřadnicemi i v seznamu bodů v panelu 6)?  
**Ano**
8. Označte objekt *Polygon* a proveďte to samé, co s obdélníkem. Dosáhli jste stejného výsledku?  
**Ano**
9. Postupně označujte objekty a odstraňte je z plátna pomocí klávesy Delete nebo klikněte pravým tlačítkem myši na objekt v panelu 5) a vyberte možnost smazat (kombinujte obě možnosti). Povedlo se smazat objekty z plátna?  
**Ano**
10. Vytvořte jeden velký obdélník, na který umístíte všechny součástky z panelu 1b). Povedlo se je umístit?  
**Ano**
11. Pro veškeré umístěné součástky proveďte akci 2., 3. a vyzkoušejte změnu jejich velikosti pomocí červeného bodu tak, aby se vzájemně

překrývaly. Zobrazil se v panelu 5) u jednotlivých součástí červený vykřičník? Posléze všechny objekty smažte.

**Ano**

12. Umístěte všechny základní objekty na plátno vedle sebe a na každém proveďte jednu transformaci. Nyní pomocí klávesové zkratky **Ctrl + Z** vraťte objekty do původního stavu (maximálně lze vrátit 5 změn). Podařilo se vrátit objekty do původního stavu?

**Ano**

13. Každý objekt se pokuste klávesovou zkratkou **Ctrl + C** zkopírovat a následně stiskem **Ctrl + V** objekt vložit na plátno. V panelu 5) sledujte, jestli se zkopírované objekty skutečně přidávají. Povedlo se zkopírovat objekty?

**Ano**

### **B.3.3 Kontrola uložení/načtení objektů a plátna**

1. Vytvořte nové plátno a umístěte na něj několik různých objektů (alespoň 4) a pomocí klávesové zkratky **Ctrl + S** uložte pod libovolným jménem. Vytvořil se výsledný soubor?

**Ano**

2. Vyberte si jeden z objektů, který přejmenujte a změňte mu materiál (libovolný text). V panelu 5) na něj klikněte pravým tlačítkem myši a zvolte *Uložit*. Vytvořil se výsledný soubor?

**Ano**

3. Nyní vytvořte nové plátno z panelu 3) pod tlačítkem *Vložit* nebo stiskem klávesové zkratky **Ctrl + I** načtěte uložený objekt. Přidal se objekt na plátno?

**Ano**

4. Pomocí tlačítka *Soubor* z panelu 3) zvolte *Načíst plátno* nebo stiskněte klávesovou zkratku **Ctrl + M** a vyberte uložené plátno. Přidalo se uložené plátno do aplikace?

**Ano**

5. Otevřete soubor s uloženým objektem v textovém formátu, smažte část jeho obsahu a pokuste se objekt načíst v aplikaci. Podařilo se objekt načíst?

**Ne**

6. Otevřete soubor s uloženým plátnem v textovém formátu, smažte část jeho obsahu a pokuste se plátno načíst v aplikaci. Podařilo se plátno načíst?

**Ne**

### B.3.4 Modelování návrhu spoje

- Pokuste se v aplikaci navrhnout podobný spoj, jako je zobrazen na obrázku 5.2. Podařilo se navrhnout šroubovaný spoj? Pokud ne, popište v čem byly problémy.

**Ano, povedlo se**

## B.4 Monkey testování

1. Klikajte na všechna možná tlačítka, která aplikace nabízí, bez předem vytvořeného plátna. Stalo se něco neobvyklého?

**Nestalo**

2. Proveďte to samé, co v bodě 1, ale s vytvořeným plátnem. Stalo se něco neobvyklého?

**Nestalo**

3. Přidejte na plátno několik objektů a náhodně se snažte pomocí panelu vlastností upravit dostupné hodnoty vkládáním různých znaků. Stalo se něco neobvyklého?

**Nestalo**

4. Odstraňte všechny objekty z plátna a pro každou funkci z panelu nástrojů náhodně klikajte po plátně. Stalo se něco neobvyklého?

**Nestalo**

5. V posledním kroku zavřete záložku s plátnem a proveďte to samé, co v bodě 1. Stalo se něco neobvyklého?

**Nestalo**