

# **Softwarová aplikace pro vkládání vstupních dat do modelu pro testování tepelné stability místnosti**

Tomáš Holík

---

Bakalářská práce  
2018

 Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2017/2018

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš Holík**

Osobní číslo: **A16598**

Studijní program: **B3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Forma studia: **kombinovaná**

Téma práce: **Softwarová aplikace pro vkládání vstupních dat do modelu pro testování tepelné stability místnosti**

Téma anglicky: **A Software Application for Inputting Data into a Room Thermal Stability Testing Model**

Zásady pro vypracování:

1. Nastudujte základní mechanismy sdílení tepla ve vytápěných místnostech obytných domů.
2. Seznamte se se základními parametry a funkcemi modelu pro testování tepelné stability místnosti, připraveného v programu COMSOL Multiphysics.
3. Proveďte analýzu geometrických a termofyzikálních vstupních parametrů studovaného modelu.
4. Vypracujte návrh uživatelského prostředí pomocné softwarové aplikace pro vkládání vstupních dat potřebných pro simulaci rozložení teploty připraveným modelem.
5. Na základě vypracovaného návrhu naprogramujte v prostředí JAVA softwarovou aplikaci pro vkládání vstupních dat a jejich export v podobě TXT souborů.
6. Ověřte funkčnost naprogramované aplikace a možnost importu z ní exportovaných TXT souborů do modelu místnosti připraveného v programu COMSOL Multiphysics.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **PŘÍHODA, Miroslav a Miroslav RÉDR.** Sdílení tepla a proudění. Ostrava: VŠB – Technická univerzita Ostrava, 2008. ISBN 978-80-248-1748-4.
2. **KULHÁNEK, František.** Stavební fyzika II. Stavební tepelná technika. 1. vyd. Praha: ČVUT Praha, 2013. ISBN 978-80-01-04957-0.
3. Introduction to COMSOL Multiphysics [online]. 2017 [cit. 2017-11-29]. Dostupná z: <http://www.humusoft.cz>.
4. **HORSTMANN, Cay S.** Core Java. Tenth edition. Boston: Prentice Hall, 2016. ISBN 0-13-417730-4.
5. **SCHILDT, Herbert.** Java: the complete reference. Ninth edition. New York: McGraw-Hill Education, 2014. ISBN 0071808558.
6. **VOS, Johan.** Pro JavaFX 8: a definitive guide to building desktop, mobile, and embedded Java clients / Johan Vos, Weiqi Gao, Stephen Chin, Dean Iverson, James Weaver. Berkeley, CA: Apress, 2014. Expert's voice in Java. ISBN 1430265744.

Vedoucí bakalářské práce:

**Ing. Hana Charvátová, Ph.D.**

Ústav automatizace a řídicí techniky

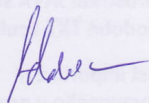
Datum zadání bakalářské práce:

**15. prosince 2017**

Termín odevzdání bakalářské práce:

**25. května 2018**

Ve Zlíně dne 15. prosince 2017



doc. Mgr. Milan Adámek, Ph.D.  
*děkan*



prof. Ing. Vladimír Vašek, CSc.  
*ředitel ústavu*

#### Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářské práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky, Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považuji se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

#### Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 22. 5. 2024

.....  
podpis autora

## **ABSTRAKT**

Bakalářská práce se zabývá tvorbou pomocné Java aplikace pro vkládání vstupních dat do parametrizovaného modelu pro testování tepelné stability místnosti, připraveného v programu COMSOL Multiphysics. Cílem práce bylo navrhnout a vytvořit uživatelské rozhraní umožňující zadávání všech požadovaných vstupních hodnot a následný export do souboru ve formátu podporovaného programem COMSOL Multiphysics. Návrh aplikace byl vytvořen na základě analýzy modelu a jeho geometrických a termofyzikálních vstupních parametrů, které vychází ze základních zákonitosti sdílení tepla ve studovaném modelu místnosti. Funkčnost aplikace byla ověřena porovnáním simulací se vstupními daty vytvořenými pomocí Java aplikace a importovanými do programu COMSOL Multiphysics a simulací provedenou za stejných podmínek s daty vloženými do modelu přímo v uživatelském prostředí programu COMSOL Multiphysics.

Klíčová slova: JavaFX, FXML, JavaFX 3D grafika, desktopová aplikace, tepelná stabilita místnosti, sdílení tepla, simulace, COMSOL Multiphysics

## **ABSTRACT**

The bachelor thesis deals with creation of ancillary Java software application used for input of initial parametrized data of model for thermal stability in COMSOL Multiphysics. Purpose of the work was to design and implement graphical user interface enabling input of required data and export into file format supported by COMSOL Multiphysics. Design of the application was based on analysis of model and its geometrical and thermophysical input parameters, which relate to basic methods of heat transfer. Functionality was validated by comparison of simulation which was run with input parameters created by Java application and imported into COMSOL Multiphysics and second simulation executed with the same conditions and data inserted directly in program COMSOL Multiphysics.

Keywords: JavaFX, FXML, JavaFX 3D graphics, desktop application, thermal room stability, thermal transmittance, simulation, COMSOL Multiphysics

Rád bych touto cestou vyjádřil poděkování Ing. Haně Charvátové, Ph.D. za její cenné rady a trpělivost při vedení mé bakalářské práce.

## OBSAH

ÚVOD .....	9
<b>I TEORETICKÁ ČÁST .....</b>	<b>9</b>
<b>1 SDÍLENÍ TEPLA .....</b>	<b>11</b>
1.1 SDÍLENÍ TEPLA VEDENÍM - KONDUKCE.....	11
1.1.1 Časově ustálené 1D vedení tepla v rovinné stěně .....	13
1.2 SDÍLENÍ TEPLA PROUDĚNÍM - KONVEKCE.....	14
1.2.1 Kritéria podobnosti .....	15
1.3 SDÍLENÍ TEPLA SÁLÁNÍM - RADIACE .....	16
1.3.1 Sdílení tepla sáláním mezi dvěma nekonečně rozlehlými rovinnými šedými tělesy .....	18
1.4 SDÍLENÍ TEPLA PROSTUPEM .....	18
1.4.1 Prostup tepla rovinnou stěnou .....	19
1.5 TEPELNĚ TECHNICKÉ VLASTNOSTI STAVEBNÍCH MATERIÁLŮ.....	20
1.6 TEPELNÁ STABILITA .....	20
<b>2 POPIS PROGRAMOVACÍHO JAZYKA JAVA FX.....</b>	<b>22</b>
2.1 STRUKTURA UŽIVATELSKÉHO ROZHRANÍ JAVA FX .....	22
2.1.1 Stage .....	22
2.1.2 Scene .....	23
2.1.3 Grafické styly uzlů.....	23
2.1.4 Panely pro rozmístění prvků .....	24
2.2 FXML.....	25
2.2.1 Nastavení vlastností prvků v FXML .....	25
2.3 TVORBA 3D OBJEKTŮ .....	26
<b>II ANALYTICKÁ ČÁST .....</b>	<b>27</b>
<b>3 ANALÝZA MODELU PRO TESTOVÁNÍ TEPELNÉ STABILITY MÍST- NOSTI V PROGRAMU COMSOL MULTIPHYSICS.....</b>	<b>30</b>
3.1 GEOMETRICKÉ A TERMOFYZIKÁLNÍ VSTUPNÍ PARAMETRY STUDOVA- NÉHO MODELU .....	30
3.2 FORMÁT SOUBORU PRO IMPORT DO STUDOVANÉHO MODELU .....	32
3.3 IMPORT TEXTOVÉHO SOUBORU SE VSTUPNÍMI PARAMETRY DO STU- DOVANÉHO MODELU.....	33
<b>III PROJEKTOVÁ ČÁST.....</b>	<b>33</b>
<b>4 NÁVRH APLIKACE.....</b>	<b>36</b>

<b>5</b>	<b>POUŽITÉ TECHNOLOGIE .....</b>	<b>37</b>
5.1	NETBEANS .....	37
5.2	JAVA DEVELOPMENT KIT .....	37
5.3	MAVEN .....	37
<b>6</b>	<b>IMPLEMENTACE .....</b>	<b>38</b>
6.1	VYTVÁŘENÍ GRAFICKÉHO UŽIVATELSKÉHO ROZHRAŇÍ PRO ZADÁVÁNÍ VSTUPNÍCH PARAMETRŮ .....	38
6.2	TVORBA A ZOBRAZENÍ 3D MODELU STUDOVANÉ MÍSTNOSTI.....	39
6.3	EXPORT ZADANÝCH VSTUPNÍCH PARAMETRŮ DO SOUBORU .....	41
<b>7</b>	<b>TESTOVÁNÍ FUNKČNOSTI PROGRAMOVANÉ APLIKACE .....</b>	<b>43</b>
	<b>ZÁVĚR.....</b>	<b>46</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>47</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>49</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>50</b>



## ÚVOD

Úbytek neobnovitelných zdrojů energie a zvýšení povědomí o klimatických problémech navýšil v posledním desetiletí zájem o nízkoenergetické domy. Nicméně u takovýchto domů hraje klíčovou roli správný architektonický návrh, který je ale značně náročný. Při návrhu se využívají softwarové nástroje a simulace, které umožňují vzít v potaz všechny důležité konstantní i proměnné faktory pro zjištění tepelné stability.

V této práci se vychází z modelu pro testování tepelné stability místnosti obytného domu v zimním období vytvořeného v programu COMSOL Multiphysics (dále jen COMSOL), který je však velmi komplexní a pro běžného uživatele či studenta jeho použití bez znalosti práce s programem COMSOL je poměrně komplikované.

Pro usnadnění práce a také kvůli nutnosti manipulace se zdrojovým modelem je cílem této práce vytvořit externí grafické uživatelské rozhraní, které umožní uživateli zadat všechny informace nutné k simulaci tepelné stability místnosti. Dále poskytne funkci exportu zadaných parametrů do textového souboru, který bude možno nainportovat do uživatelského rozhraní programu COMSOL a tím se mu nastaví požadované vstupní parametry v modelu. Oproti v této práci používané verzi programu COMSOL je zadávání vstupních dat jednodušší a přehlednější.

V první kapitole teoretické části práce jsou popsány způsoby sdílení tepla a v jednotlivých podkapitolách se bere zřetel na děje, které probíhají v obytné místnosti podobné studovanému modelu.

Programovací jazyk JavaFX, pomocí kterého byla vytvořena pomocná aplikace, je popsán v druhé kapitole 2.1 teoretické části bakalářské práce. Je zde zmíněna struktura hlavních komponent JavaFX a podkapitola 2.3 obsahuje popis tvorby 3D objektů.

Analýza modelu pro testování tepelné stability místnosti je rozebírána v kapitole 3.1 a na jejím základě je vytvořen návrh aplikace v podkapitole 4. Dále jsou zde popsány veškeré vstupní parametry v podkapitole 3 a formát souboru pro import do COMSOLu je specifikován v podkapitole 3.2.

V praktické části je popsána samotná implementace aplikace v JavaFX od návrhu rozložení prvků v grafickém uživatelském rozhraní až po validaci a verifikaci vytvořené aplikace. Problematika vytváření a zobrazování 3D modelu studované místnosti v aplikaci je zmíněna v podkapitole 6.2 spolu s experimenty, které předcházely finálnímu řešení.

# I. TEORETICKÁ ČÁST

## 1 Sdílení tepla

Sdílení tepla je jev, který se vyskytuje ve všech prostředích. Podmínkou pro šíření energie obecně je rozdíl energických potenciálů. Jedná se o snahu dosáhnout ve všech bodech hmoty stejné energické bilance. V případě sdílení tepla se tímto gradientem rozumí rozdíl teplot. Tepelný tok se tedy uskutečňuje z míst o vyšší koncentraci tepelné energie do míst s koncentrací nižší. V praxi jsou rozlišovány tři druhy sdílení tepla, které se odlišují podmínkami a způsoby přenosu. Jedná se o sdílení kondukcí, konvekcí a radiací [1].

Sdílení tepla se obvykle nerealizuje pouze jedním způsobem, ale reálná výměna tepla je často jejich kombinací, jak bude uvedeno v podkapitole 1.4.

### 1.1 Sdílení tepla vedením - konduktce

Se sdílením tepla tohoto druhu se lze nejčastěji setkat u pevných látek, avšak lze se s ním za jistých podmínek setkat v látkách všech skupenství. Nastává předáváním kinetické energie mezi sousedícími částicemi látky vlivem vzájemných srážek. Teplejší, rychleji kmitající molekuly, předávají svou kinetickou energii sousedním, pomaleji kmitajícím molekulám.

Dochází k přenášení tepelné energie ve směru klesající teploty. Nutná podmínka je existence spojitého látkového prostředí a tedy ve vakuu vedení tepla nenastane. Způsob sdílení tepla vedením se vyskytuje téměř u všech stavebních konstrukcích.

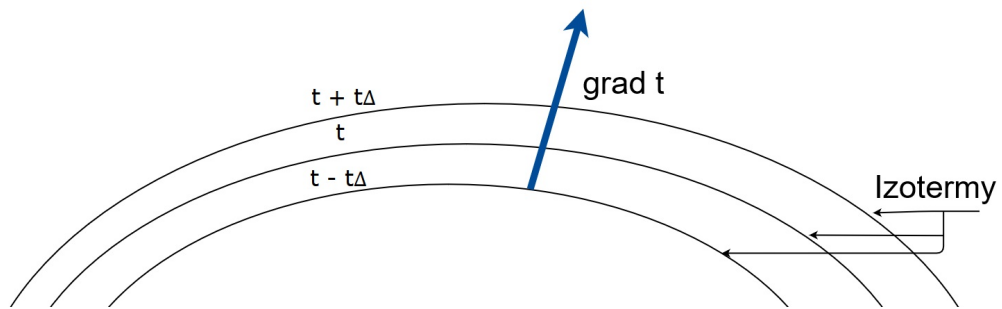
Při sdílení tepla vedením dochází ve zkoumaném objektu ke změně teploty v prostoru i čase. Tento proces je popsán teplotním polem, které přiřazuje každému bodu tělesa jeho teplotu. Teplotní pole se dělí na stacionární (ustálené), které je neměnné v čase a nestacionární (neustálené), kde je teplota v daném bodě funkcí místa i času.

Při neustáleném vedení tepla postupně dochází k postupnému vyrovnávání teplotních rozdílů mezi jednotlivými částmi tělesa. Na základě toho v jaké prostorové dimenzi se teplotní pole nachází, jej lze rozdělit na dvourozměrné (rovinné) a trojrozměrné (prostorové). Geometrické body o stejné teplotě se nazývají izotermickými plochami. Libovolná rovina protíná izotermické plochy v křivkách zvaných izotermy [2].

Gradient teplotního pole  $\nabla t$  je vektor, který je kolmý k izotermě a udává směr teplotního spádu. Lze ho definovat jako derivaci teploty  $t$  podle normály  $n$  [3]:

$$\nabla t = \text{grad } t = \frac{\partial t}{\partial n} \quad (1.1)$$

Základní fyzikální zákony popisující vedení tepla jsou první a druhý Fourierův zákon [3]. První Fourierův zákon popisuje závislost tepelného toku na gradientu teploty tedy na teplotním spádu. Platí pro stacionární teplotní pole, což je stav, kdy se rozložení



Obr. 1.1 Gradient teplotního pole [4]

teplot v tělese s časem nemění. Další předpoklad je, že těleso musí být homogenní, izotropní a má konstantní fyzikální vlastnosti. Pro vedení tepla v jednorozměrném tělese platí [4]:

$$q = \frac{\dot{Q}}{S} = -\lambda \frac{dT}{dx} \quad (1.2)$$

Kde:  $q$  tepelný tok vztažený na jednotku plochy  $S$ , tj. hustota tepelného toku  
 $\dot{Q}$  tepelný tok  
 $\lambda$  součinitel tepelné vodivosti  
 $S$  teplosměnná plocha tělesa

První Fourierův zákon pro třírozměrné těleso má tvar [3]:

$$\vec{q} = -\lambda \text{grad } t = -\lambda \nabla t = -\lambda \left( \frac{\partial t}{\partial x} + \frac{\partial t}{\partial y} + \frac{\partial t}{\partial z} \right) \quad (1.3)$$

Kde:  $q$  hustota tepelného toku  
 $\lambda$  součinitel tepelné vodivosti

Vzhledem k tomu, že tepelný tok je vektor, tak má tři složky do jednotlivých směrů:

$$q_x = -\lambda \frac{\partial t}{\partial x} \quad (1.4)$$

$$q_y = -\lambda \frac{\partial t}{\partial y} \quad (1.5)$$

$$q_z = -\lambda \frac{\partial t}{\partial z} \quad (1.6)$$

Druhý Fourierův zákon platí pro časově proměnné teplotní pole a pro zobrazení

v trojrozměrném prostoru má tvar [3]

$$\frac{\partial t}{\partial \tau} = a \left( \frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} + \frac{\partial^2 t}{\partial z^2} \right) \quad (1.7)$$

$$a = \frac{\lambda}{c_p \rho} \quad (1.8)$$

Kde:  $c_p$  měrná tepelná kapacita  
 $a$  teplotní vodivost  
 $\tau$  čas  
 $\rho$  hustota

### 1.1.1 Časově ustálené 1D vedení tepla v rovinné stěně

V praxi je možné některé případy vedení tepla považovat za jednodimenzionální. Pokud se teplota v daném tělese mění jen v závislosti na jedné souřadnici, tak hovoříme o jednorozměrném vedení tepla. Například u vedení tepla napříč rozlehlými stěnami domů lze předpokládat, že tepelný tok je hlavně přenášen ve směru přes stěnu a tedy teplota se mění je v závislosti na poloze napříč stěnou [4].

Pro určení tepelného toku u jednoduché rovinné stěny lze vycházet z prvního Fourierova zákona. Pro dané teploty na obou hranicích stěny lze zapsat tepelný tok, který je přenášený vedením z jedné strany stěny na druhou ve směru osy  $x$  [4].

$$\dot{Q} = -\lambda S \frac{dT}{dx} \quad (1.9)$$

Pomocí integrace dostaneme

$$\dot{Q} \int_0^\delta dx = -\lambda S \int_{T_1}^{T_2} dT \quad (1.10)$$

$$\dot{Q} \delta = -\lambda S (T_2 - T_1) \quad (1.11)$$

$$\dot{Q} = \frac{\lambda S (T_1 - T_2)}{\delta} \quad (1.12)$$

Kde:  $\dot{Q}$  tepelný tok  
 $\lambda$  součinitel tepelné vodivosti  
 $\delta$  tloušťka stěny ve směru tepelného toku  
 $S$  teplosměnná plocha stěny  
 $T$  teplota stěny

Jestliže například uvažujeme stěnu domu s omítkou na obou stranách, tak se z jednoduché rovinné stěny stává složena rovinná stěna se třemi vrstvami. Každá vrstva je definovaná svou tepelnou vodivostí a tloušťkou viz obrázek 1.2. Tepelný tok prochází jednotlivými vrstvami beze změny. Platí tedy [4]:

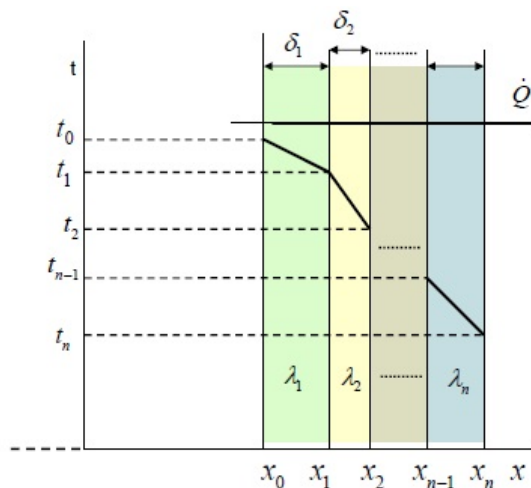
$$\dot{Q} = -\lambda_1 S \frac{T_2 - T_1}{\delta_1} = -\lambda_2 S \frac{T_3 - T_2}{\delta_2} = -\lambda_3 S \frac{T_4 - T_3}{\delta_3} \quad (1.13)$$

Po úpravě platí:

$$\dot{Q} = \frac{T_1 - T_4}{\frac{\delta_1}{\lambda_1 S} + \frac{\delta_2}{\lambda_2 S} + \frac{\delta_3}{\lambda_3 S}} \quad (1.14)$$

Pro složenou stěnu tedy obecně platí:

$$\dot{Q} = \frac{T_1 - T_{n+1}}{\sum_{j=1}^n \frac{\delta_j}{\lambda_j}} S \quad (1.15)$$



Obr. 1.2 Vedení tepla složenou stěnou [5]

Kde:  $\lambda_j$  tepelná vodivost dané vrstvy  
 $\delta_j$  tloušťka dané vrstvy

## 1.2 Sdílení tepla prouděním - konvekce

Proudění tepla je jeden ze způsobů sdílení tepla v tekutinách, při kterém se přemísťují přímo částice s větší energií a přenášejí přitom tepelnou energii. Pohybující se tekutina přitom přidává nebo odebírá tepelnou energii z povrchu okolních těles. Samovolné tepelné proudění stoupá vzhůru, protože teplejší části kapalin a plynů mají menší hustotu.

Proudění tepla lze dělit na přirozené (volné) proudění a nucenou konvekci. Přirozené proudění nejčastěji vzniká v silovém poli zemské tíže, ale může se objevit i v odstředivých či elektromagnetických polích [2]. Vynucené proudění je vyvolané vnějšími silami jako například ventilátorem, čerpadlem, tahem komína a pohybem tuhého tělesa.

V praxi se téměř vždy jedná o kombinaci volné a nucené konvekce. Podíl přirozené konvekce je tím větší, čím menší je rychlost nuceného proudění. Při vysokých rychlostech nuceného proudění je vliv přirozeného proudění zanedbatelný.

Pro výpočet hustoty tepelného toku při proudění se využívá Newtonův ochlazovací zákon [3].

$$\dot{Q} = \alpha \Delta t S \quad (1.16)$$

$$q = \alpha (t_p - t) \quad (1.17)$$

Kde:  $t_p$  teplota povrchu stěny  
 $t$  teplota proudící tekutiny  
 $\alpha$  součinitel přestupu tepla při proudění  
 $S$  teplosměnná plocha tělesa

### 1.2.1 Kritéria podobnosti

Pro většinu praktických problémů je jejich analytické řešení je natolik složité, že se využívá výsledků experimentálních výzkumů spolu s aplikací teorie podobnosti. U teorie podobnosti se vychází z toho, že jestliže je několik různých systémů navzájem podobných, tak je i jejich chování podobné. Pro porovnání podobnosti dvou systémů se využívají kritéria podobnosti, které berou v potaz závislosti mezi relevantními veličinami.

**Grashofovo číslo** - vyjadřuje podobnost při volném gravitačním neizotermickém proudění [2].

$$G_r = \frac{gl^3\beta\Delta t}{\nu^2} \quad (1.18)$$

**Prandtlovo číslo** - určuje podobnost mezi sdílením hybnosti a tepla [2].

$$P_r = \frac{\nu}{a} = \frac{c_p\eta}{\lambda} \quad (1.19)$$

**Pécletovo číslo** - porovnává přenos tepla konvekcí a přenos tepla vedením. Je využíván při nucené konvekci [2].

$$P_e = R_e P_r = \frac{vl}{a} \quad (1.20)$$

**Reynoldsovo číslo** [2]

$$R_e = \frac{vl}{\nu} = \frac{vl\rho}{\eta} \quad (1.21)$$

**Stantonovo číslo** [2]

$$S_t = \frac{N_u}{P_e} = \frac{\alpha}{\nu\rho c_p} \quad (1.22)$$

**Nusseltovo číslo** - patří mezi základní podobnostní čísla. Používá se při řešení přenosu tepla konvekcí a pro výpočet součinitele přestupu tepla  $\alpha$  [2].

$$N_u = C (G_r P_r)^n \quad (1.23)$$

Součinitel přestupu tepla  $\alpha$  při volné konvekci lze získat pomocí následujícího vztahu [5]:

$$\alpha = \frac{N_u \lambda}{l} \quad (1.24)$$

Kde:  $l$  je charakteristický rozměr

$\beta$  koeficient objemové roztažitosti

$\eta$  dynamická viskozita

$\nu$  kinematická viskozita

$\rho$  hustota

$v$  rychlost

### 1.3 Sdílení tepla sáláním - radiace

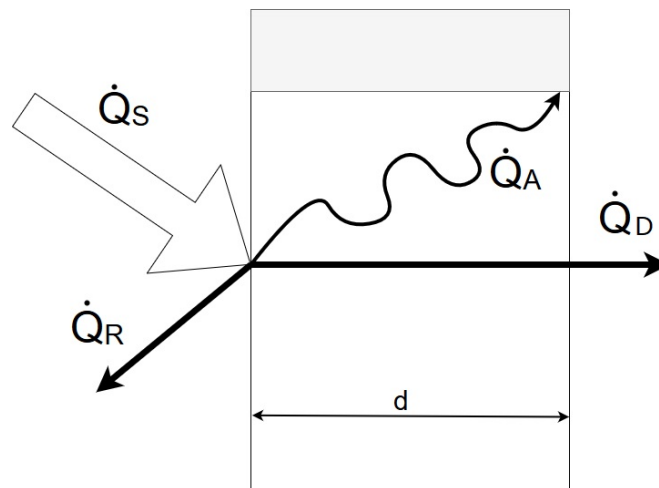
Při sdílení tepla sáláním dochází k přenosu elektromagnetického záření, které se pohybuje se ve vlnových délkách 0,1  $\mu\text{m}$  - 100  $\mu\text{m}$ . Každé těleso z libovolného materiálu o teplotě vyšší než 0 K v určité míře vyzařuje energii, ale zároveň ji částečně pohlcuje a propouští. Dochází tedy ke změně energie vyzářené jinými tělesy na teplo.

Na rozdíl od kondukce se tepelné záření nejlépe šíří ve vakuu. Kromě názvu sálání se pro tento způsob sdílení tepla využívají i synonyma záření a radiace. Rozdělení sálavého toku na jeho jednotlivé složky lze pozorovat na obrázku 1.3.

V důsledku energetické rovnováhy platí [6]:

$$\dot{Q}_S = \dot{Q}_A + \dot{Q}_R + \dot{Q}_D \quad (1.25)$$





Obr. 1.3 Rozdělení sálavého toku na jeho jednotlivé složky [6]

Kde:  $\dot{Q}_S$  celkový sálavý tok, který dopadá na stavební konstrukci  
 $\dot{Q}_A$  energie pohlcená konstrukcí  
 $\dot{Q}_R$  energie odražená konstrukcí  
 $\dot{Q}_D$  energie procházející konstrukcí

Po úpravě předchozí rovnice do poměrného tvaru platí:

$$\frac{\dot{Q}_A}{\dot{Q}_S} + \frac{\dot{Q}_R}{\dot{Q}_S} + \frac{\dot{Q}_D}{\dot{Q}_S} = 1 \quad (1.26)$$

Dále platí:

$$A + R + D = 1 \quad (1.27)$$

Kde:  $A$  pohltivost, tj. podíl pohlceného sálání a dopadajícího sálání  
 $R$  odrazivost, tj. podíl odraženého sálání a dopadajícího sálání  
 $D$  propustnost, tj. podíl propuštěného sálání a dopadajícího sálání

V případě, že je  $A = 1$  a zároveň  $R = D = 0$ , tak těleso pohltí veškerou sálavou energii na něj dopadající. Takové těleso se nazývá dokonale černé těleso, dokonalý zářič a nebo Planckův zářič.

Pokud je  $R = 1$  a také  $A = D = 0$ , pak to znamená, že těleso odrazí veškerou dopadající sálavou energii a nazývá se bílé těleso či dokonalé zrcadlo.

O dokonale propustné těleso se jedná, pokud  $D = 1$  a  $R = A = 0$

Stefan-Boltzmannův zákon, který popisuje hustotu sálavého tepelného toku, patří mezi základní zákony v oblasti sálání. Pro dokonale černé těleso platí [6]:

$$q_c = C_c \left( \frac{T}{100} \right)^4 \quad (1.28)$$

Kde:  $q_c$  hustota sálavého tepelného toku ( $\text{Wm}^{-2}$ )  
 $C_c$  součinitel sálání absolutně černého tělesa  $C_c = 5,67[\text{Wm}^{-2}\text{K}^{-4}]$   
 $T$  termodynamická (absolutní) teplota tělesa (K)

V praxi se černá tělesa nevyskytují, ale pouze takzvaná šedá tělesa, u kterých není všechna dopadající energie absorbována, ale část energie je odražena zpět. Může nastat i situace, kdy dochází k mnohonásobnému odrazu energie mezi dvěma tělesy. Pro součinitel sálání šedých těles platí:

$$C = A C_c \quad (1.29)$$

Kde:  $A$  pohltivost sálání "šedého" tělesa

### 1.3.1 Sdílení tepla sáláním mezi dvěma nekonečně rozlehlými rovinnými šedými tělesy

U šedých těles dochází k odrazu energie a tedy ne všechna odražená energie musí dopadnout na těleso, které původně energii vyslalo. Nicméně tento problém je u nekonečně rozlehlých paralelních povrchů eliminován, protože všechna energie, která opouští první povrch dopadá na povrch druhý [4]. Výsledný tepelný tok přenesený sáláním mezi oběma povrchy je dán následujícím vztahem:

$$\dot{Q} = \frac{\sigma (T_1^4 - T_2^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \quad (1.30)$$

Kde:  $\dot{Q}$  tepelný tok vyměněný mezi dvěma povrchy  
 $\epsilon$  emisivita, poměrná zářivost  
 $\sigma$  Stefan-Boltzmannova konstanta,  $\sigma = 5,669 \cdot 10^{-8} [\text{W} \cdot \text{m}^{-2} \text{K}^{-4}]$   
 $T_1$  termodynamická teplota prvního povrchu  
 $T_2$  termodynamická teplota druhého povrchu

## 1.4 Sdílení tepla prostupem

Sdílení tepla prostupem představuje kombinovaný přenos tepla mezi tekutinami (plyn či kapalina), které jsou oddělené pevnou přepážkou. Vzniká v důsledku toho, že na rozhraní stěny a kapaliny vytvoří tenká mezní vrstva, ve které dochází k prudké změně teploty.

### 1.4.1 Prostup tepla rovinnou stěnou

Mezi prvním prostředím a stěnou dochází ke sdílení tepla prouděním charakterizovaným součinitelem přestupu tepla ( $\alpha_1$ ). Ve stěně dochází k přenosu tepla vedením se součinitelem tepelné vodivosti ( $\lambda$ ) a ze stěny do druhého prostředí opět prouděním charakterizovaným součinitelem přestupu tepla ( $\alpha_2$ ). Dle zákona zachování energie platí [2]:

$$\begin{aligned}\dot{Q} &= \dot{Q}_1 = \dot{Q}_2 = \dot{Q}_3 \\ \dot{Q}_1 &= \alpha_1 S (t_1 - t_{s1}) \\ \dot{Q}_2 &= \frac{\lambda}{\delta} S (t_{s1} - t_{s2}) \\ \dot{Q}_3 &= \alpha_2 S (t_{s2} - t_2)\end{aligned}\tag{1.31}$$

Po úpravě:

$$\begin{aligned}t_1 - t_{s1} &= \frac{\dot{Q}}{\alpha_1 S} \\ t_{s1} - t_{s2} &= \frac{\dot{Q} \delta}{\lambda S} \\ t_{s2} - t_2 &= \frac{\dot{Q}}{\alpha_2 S}\end{aligned}\tag{1.32}$$

Po sečtení platí:

$$t_1 - t_2 = \frac{\dot{Q}}{S} \left( \frac{1}{\alpha_1} + \frac{d}{\lambda} + \frac{1}{\alpha_2} \right)\tag{1.33}$$

Kde:  $t_{s1}, t_{s2}$  teploty rozhraní jednotlivých vrstev

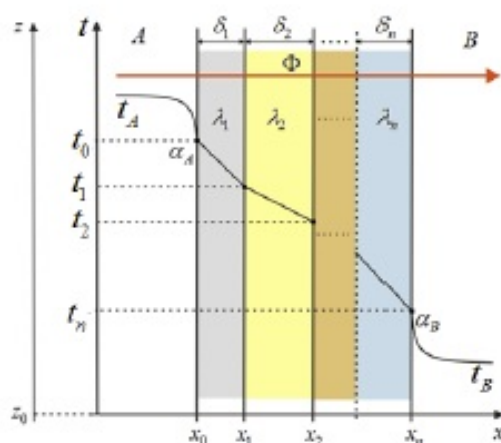
Z této rovnice lze vyjádřit  $\dot{Q}$ , což je tepelný tok při celkovém prostupu tepla rovinnou stěnou, která se skládá pouze z jedné vrstvy [5].

$$\dot{Q} = \frac{t_1 - t_2}{\frac{1}{\alpha_1} + \frac{\delta}{\lambda} + \frac{1}{\alpha_2}} S\tag{1.34}$$

Pro prostup tepla složenou rovinnou stěnou je situace velmi podobná, ale je potřeba brát v potaz několik vrstev, které se mohou skládat z odlišných materiálů [5].

$$\dot{Q} = \frac{t_A - t_B}{\frac{1}{\alpha_A} + \sum_{j=1}^n \frac{\delta_j}{\lambda_j} + \frac{1}{\alpha_B}} S\tag{1.35}$$

Kde:  $t_A, t_B$  teploty tekutin obklopujících stěnu



Obr. 1.4 Prostup tepla složenou rovinnou stěnou [5]

## 1.5 Tepelně technické vlastnosti stavebních materiálů

Pro základní tepelně technické výpočty je nezbytné mít k dispozici pro každý materiál jeho tepelnou vodivost, hustotu a měrnou tepelnou kapacitu. Součinitel tepelné vodivosti vyjadřuje schopnost materiálu ve větší nebo menší míře přenášet teplo vedením. Patří k nejdůležitějším vlastnostem stavebních materiálů ve stavební tepelné technice. Měrná tepelná kapacita udává, jaké množství tepla je třeba dodat jednomu kilogramu látky, aby se její teplota zvýšila o jeden stupeň Celsia. Hustota je fyzikální veličina, která vyjadřuje hmotnost objemové jednotky látky.

Součinitel tepelné vodivosti závisí na teplotě, tlaku a složení látky. Určuje se měřením hustoty tepelného toku, gradientu a poté výpočtem v vztahu [3]:

$$\lambda = -\frac{dQ}{\text{grad } t \cdot dS \cdot d\eta} \quad (1.36)$$

Sdílení tepla v tuhých tělesech závisí na tom, zda se jedná o kovy, polovodiče či izolanty. Obecně platí, že kovy jsou lepšími tepelnými vodiči než nekovy. Z hlediska tepelné vodivosti jsou zvláštní tuhá tělesa s pórovitou strukturou, jako jsou například cihly, beton a dřevo. Složitý děj sdílení tepla v tuhé hmotě a v pórech vyplněných plynem či kapalinou je výrazně závislý na vlhkosti, která při zaplnění pórů zvyšuje součinitel tepelné vodivosti. Při zvyšování teploty má u pórovitých těles stále větší vliv sálání [7].

## 1.6 Tepelná stabilita

Tepelná stabilita popisuje teplotní vlastnosti prostoru tvořeného stavebními konstrukcemi ve vztahu k vnějším podmínkám. Místnost se dá označit jako tepelně stabilní,

pokud její tepelný stav zůstává v daném časovém intervalu v dovolených mezích. Návrh dobré tepelné stability obytného objektu by měl zajistit, aby se daný objekt nebo jeho jednotlivá místnost v letním období nepřehřívala. V zimě má ve spojitosti s přerušovaným způsobem vytápění zajistit, aby teplota nepoklesla po přerušení vytápění pod určitou hodnotu [7].

## 2 Popis programovacího jazyka JavaFX

Hlavním cílem bakalářské práce je vytvoření aplikace v programovacím jazyce JAVA umožňující uživateli zadávat všechny nezbytné parametry pro spuštění a vyhodnocení simulace na modelu pro testování tepelné stability místnosti v zimním období, vytvořeném v programu COMSOL. Existuje několik variant pro tvorbu uživatelského rozhraní v JAVA, ale v této práci se budu výhradně věnovat programovacímu jazyku JavaFX.

Java podporovala tvorbu uživatelských rozhraní již od prvních verzí, kdy se využívalo AWT (Abstract Windows Toolkit). Poté se přešlo na Swing, který znamenal značné zlepšení, ale i tak mu chyběla funkcionalita pro snadnou tvorbu 2D a 3D objektů a jejich efektivní rendrování. JavaFX je následovník Swingu a do budoucna by ho měla úplně nahradit. Jedná se o moderní framework, který se využívá pro tvorbu multiplatformních aplikací. Aplikace tedy může běžet například na stolním počítači, mobilním telefonu a nebo na webovém prohlížeči. Obsahuje knihovny pro podporu obrázků, videa, hudby, grafů, CSS stylů, tvorbu 3D objektů a dalších technologií. JavaFX umožňuje tvorbu uživatelského rozhraní pomocí kódu v Java a nebo pomocí FXML, což bude více popsáno v podkapitole 2.2.

Pro zobrazení chybových hlášek uživateli se využívá vyskakovací modální okno, které neumožňuje uživateli další práci s aplikací, dokud toto okno nezavře [9].

### 2.1 Struktura uživatelského rozhraní JavaFX

V JavaFX se pro popis hlavních prvků uživatelského rozhraní využívá metafory s divadlem. Je zde jeviště (*Stage*) na kterém se vše odehrává a mohou se na něm měnit různé scény (*Scene*) [8]. Na scéně je možné umístit jednotlivé prvky.

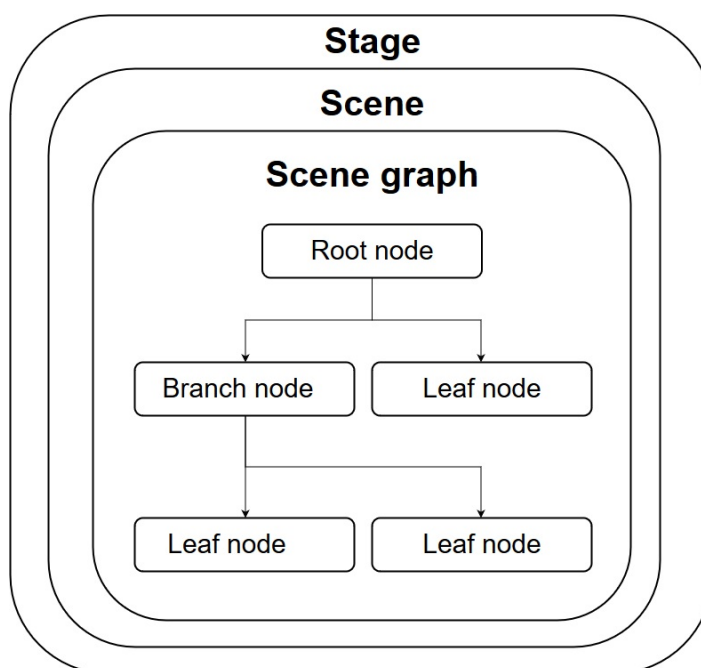
Dále v textu budu využívat anglický název *stage* místo českého překladu, jelikož při odkazování na existující třídy mně to přijde vhodnější.

#### 2.1.1 Stage

Třída *javafx.stage.Stage* je prvek nejvyšší úrovně v JavaFX. Při vývoji aplikací pro osobní počítače je *stage* hlavním oknem aplikace a kromě názvu a geometrických rozměrů umožňuje například u okna skrýt i všechny jeho ovládací prvky a zobrazit ho průhledně. U *stage* se nastavuje i modalita okna, která definuje, jestli okno blokuje další okna aplikace a vyžaduje přednostní obsluhu uživatele, což se využívá u vyskakovacích oken s chybovými hláškami nebo důležitými upozorněními [9].

### 2.1.2 Scene

Scéna reprezentuje vizuální obsah *stage* a v JavaFX se jedná o třídu *javafx.scene*. Scéna je v jeden moment připojená maximálně na jeden *stage*. Scéna slouží jako kontejner pro jeho jednotlivé prvky, které jsou hierarchicky členěné v grafu scény (scene graph), což je stromová datová struktura, jejíž elementy se nazývají uzly (node) a která má právě jeden kořenový uzel. Mezi uzly lze definovat jejich vztahy pomocí dědičnosti [8]. Toto hlavní hierarchické členění je zobrazeno na obrázku 2.1.



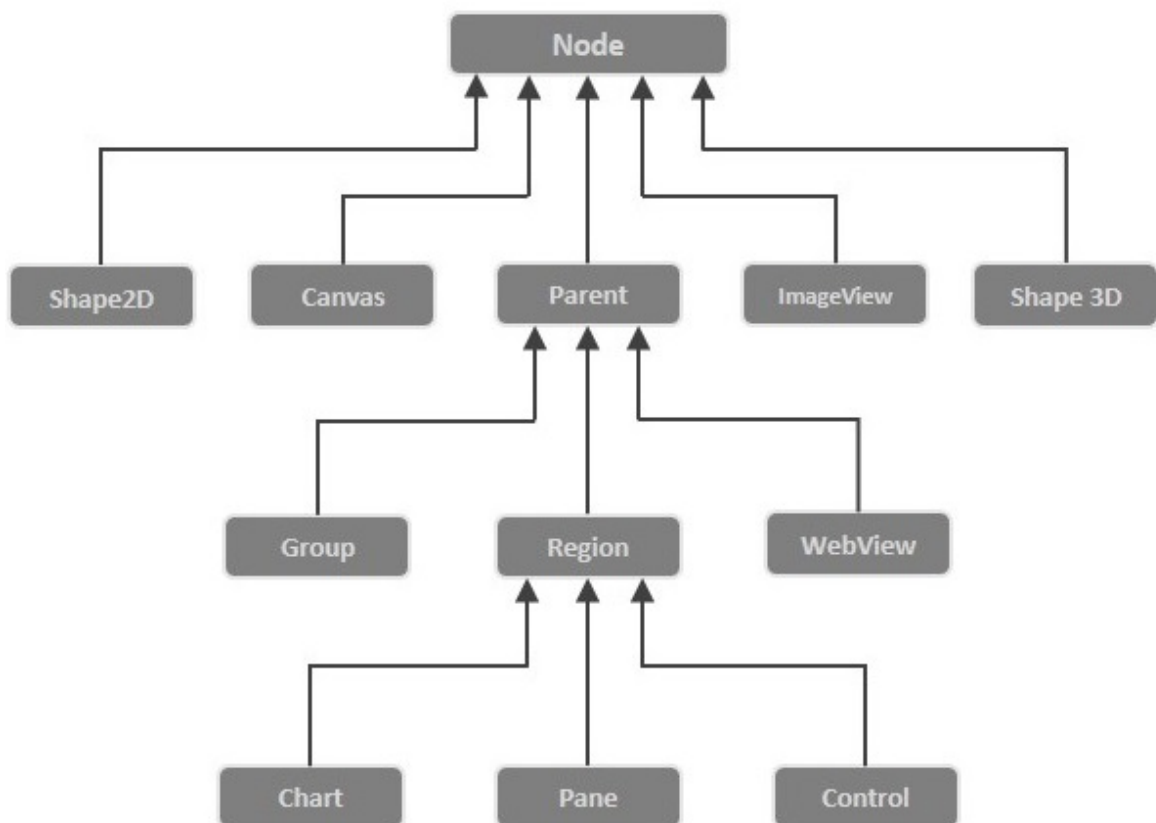
Obr. 2.1 Struktura hlavních komponent JavaFX [8]

V JavaFX je uzel definován abstraktní třídou *javafx.scene.Node* a všechny ostatní grafické uživatelské komponenty jsou potomky této třídy. Na obrázku 2.2 lze vidět hierarchii dědičnosti třídy uzel spolu s některými potomky.

### 2.1.3 Grafické styly uzlů

V JavaFX se pro nastavování stylů uzlů využívají kaskádové styly (Cascading Style Sheets-CSS) které umožňují popis grafického zobrazení elementů. Lze pomocí nich nastavit například barvu a velikost písma či pozadí prvků a jejich orámování. Kaskádové styly byly primárně vytvořené pro webové stránky a stylování HTML prvků. Soubor s kaskádovými styly bývá běžně oddělen od ostatního obsahu a dochází tím k oddělení logiky.

Práce s kaskádovými styly je v JavaFX velice podobná jako pro stylování HTML. Hlavní rozdíl je ve jmenné konvenci, kterou JavaFX využívá pro výběr prvků. Pro referenci na prvky se používá název třídy psaný malým písmem a pokud se jedná



Obr. 2.2 Hierarchie dědičnosti třídy uzel [8]

o víceslovný název, tak se mezi slova vloží pomlčka. Tedy například pro referenci na prvek *TextField* se využívá *text-field*. Na prvky je možné se odkazovat i přes jejich unikátní identifikátor o čemž se více rozepisují v sekci 2.2.

Jak již bylo zmíněno, tak kaskádové styly jsou uloženy ve vlastním souboru, který lze přiřadit k existující scéně například následujícím způsobem.

```
scene.getStyleSheets().add(getClass().getResource("views/style.css"));
```

#### 2.1.4 Panely pro rozmístění prvků

Panely pro rozvržení prvků vychází ze třídy *javafx.scene.layout.Pane* a jsou to kontejnery obsahující uzly, které jsou jeho potomky. Každá implementace definuje, jakým způsobem se zobrazují uzly, které obsahují. Například může zobrazit uzly horizontálně vedle sebe, vertikálně a nebo třeba v tabulce. Existuje několik tříd pro rozmístění prvků, a to například *HBox*, *VBox*, *FlowPane*, *BorderPane*, *GridPane*, *AnchorPane* a další.

- *HBox* umístí potomky do jediné horizontální řady a umožní nastavení mezer mezi uzly.
- *VBox* má velmi podobné chování jako *HBox*, ale zobrazí všechny své uzly naopak,



a to do vertikální řady.

- Panel *BorderPane* umožňuje komplexnější rozvržení umístění prvků do několika oblastí. Oblasti dělí se na horní, spodní, středovou, levou a pravou oblast. U každého uzlu je možné nastavit jednu z těchto variant.
- *GridPane* slouží pro rozmístí svých potomků do tabulkové struktury. Každý prvek má přesně definované ve kterém sloupci a řádku tabulky se má zobrazit.

## 2.2 FXML

FXML je formát založený na značkovacím jazyce eXtensible Markup Language (XML), který umožňuje vytváření uživatelského rozhraní. Usnadňuje rozdělení softwarové architektury aplikace do Model-view-controller (MVC), která odděluje datový model aplikace, uživatelské rozhraní a řídicí logiku tak, aby modifikace některé z nich měla minimální dopad na ostatní vrstvy. Soubor FXML lze editovat v běžných textových editorech, a po jeho úpravě není nutné program znovu zkompileovat, aby se do něj promítly tyto úpravy [8]. Vzhledem k tomu, že XML je hierarchicky strukturované, tak umožňuje intuitivní členění jednotlivých uzlů.

### 2.2.1 Nastavení vlastnosti prvků v FXML

V JavaFX obsahují prvky různé vlastnosti, jako například výchozí hodnotu v textovém poli a nebo nastavení zarovnání prvku, ale existují pouze dvě možnosti, jak je přes FXML nastavit. Vložení vlastností do atributů daného prvku:

```
<Label text="hodnota atributu"></Label>
```

A nebo přes vytvoření nového vnořeného prvku:

```
<Label>  
<text>hodnota atributu</text>  
</Label>
```

Další důležitá vlastnost je unikátní identifikátor prvku *fx:id*. Umožňuje odkazování na tento prvek z JavaFX kódu, což je využívá v kontrolérech, a nebo také například nastavení CSS přes referenci na tento objekt. Zde je ukázka nastavení identifikátoru u textového prvku:

```
<TextField fx:id="identifikátor"/>
```

Dále je možné v FXML definovat události vázané na prvky. Typicky se jedná o stisk tlačítka na klávesnici a nebo na myši, ale množina podporovaných událostí je v JavaFX mnohem obsáhlejší. Reakce na události a jejich zpracování je možné přímo v FXML pomocí bloku skriptu, který definuje funkci pro zpracování, a nebo v kontroléru [9].

V následujícím příkladu lze vidět nastavení události pro textové pole, která je spuštěna po uvolnění stisknutého tlačítka.

```
<TextField onKeyReleased="#handleKeyReleased" .. />
```

### 2.3 Tvorba 3D objektů

Od verze JavaFX 8 umožňuje vytvářet plnohodnotnou 3D scénu s nastavitelnými kamerami, světlem a modely. Lze ji tak využívat pro tvorbu animací nebo k prohlížení 3D modelů, ale ke tvorbě náročných her s propracovanou fyzikou není momentálně příliš vhodná. Trojrozměrné objekty je možné vytvářet pomocí následujících základních tvarů:

- koule
- krychle
- válec

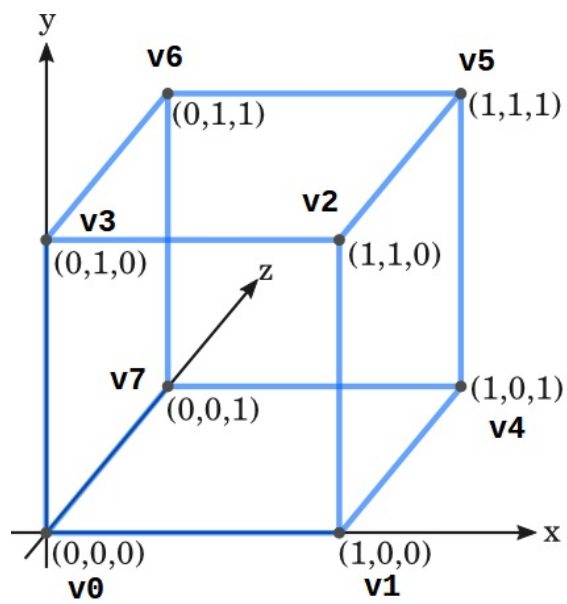
Reálné objekty je poměrně těžké vytvořit pomocí základních 3D tvarů, které JavaFX nabízí, a proto se využívá *MashView* umožňující definovat objekt jako množinu mnohoúhelníků. Pro popis topologie objektů se využívají trojúhelníky s definovanými vazbami mezi sebou a využívá se k tomu třída *TriangleMesh*. Pro úplný popis 3D objektu je potřeba nastavit souřadnice bodů (Points), souřadnice textur a plochy trojúhelníků (Faces).

V následujícím textu se zaměřím na tvorbu 3D krychle [8]. V trojrozměrném souřadnicovém systému je každý bod definován souřadnicemi  $x, y$  a  $z$ . U krychle jsou to její vrcholy a celý objekt je definovaný množinou souřadnic těchto osmi vrcholů.

```
float[] body = {  
    0, 0, 0,    //v0  
    1, 0, 0,    //v1  
    0, 0, 1,    //v7  
};
```

Při nastavení textury na daný objekt je potřeba přesně specifikovat, které části textury budou aplikované na jednotlivé trojúhelníky. Krychli rozložíme do 2D, kde  $v_{0-7}$  označují vrcholy krychle a  $t_{0-13}$  vrcholy trojúhelníků. Dále si texturu představíme jako box 1x1 označený čárkovane a vložíme ji pod model krychle. Zde je vidět, že na výsledném objektu bude pouze část této textury. Souřadnice  $x$  a  $y$  pro bod textury budou nabývat hodnot  $(0 - 1)$ . Například vrchol  $t_8$  bude nabývat hodnot  $(0.5, 0)$ .

```
float[] textury = {  
    0, 0.33f,    // t0
```



Obr. 2.3 3D krychle se souřadnicemi [8]

```

    0.25f, 0.33f, // t1
    0, 0.5f // t13
};

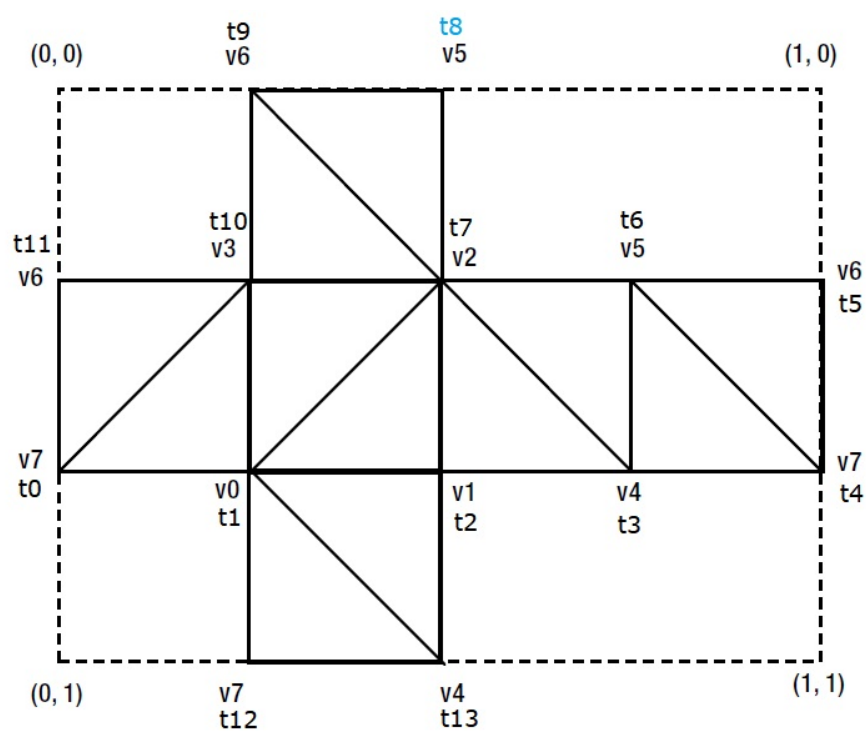
```

Posledním krokem je vytvoření jednotlivých trojúhelníků ze kterých se krychle bude skládat. Při tvorbě se bude odkazovat na indexy prvků tabulek *body* a *texture*.

```

int[] faces = new int[] {
    5,8,6,9,2,7 //v5,t8, v6,t9, v2,t7
};

```



Obr. 2.4 Povrch rozložené krychle na textuře [8]

## II. ANALYTICKÁ ČÁST

### 3 Analýza modelu pro testování tepelné stability místnosti v programu COMSOL Multiphysics

V této práci navazuji na již vytvořený model [13] pro testování tepelné stability místnosti a simulaci distribuce teploty v místnosti obytného domu v zimním období. Jedná se o parametrizovaný model, který umožňuje uživateli měnit geometrické rozměry jednotlivých prvků v místnosti, fyzikální vlastnosti použitých materiálů a podmínky simulace. Tyto parametrizované hodnoty je možné vkládat do modelu přímo v uživatelském prostředí COMSOL nebo je načíst do modelu přes textový soubor. Cílem této sekce bude analyzovat všechny parametrizovatelné vstupní parametry, což je rozepsáno v podkapitole 3.1, a zjistit formát textového souboru, který je možné importovat v uživatelském prostředí programu COMSOL. Na základě této analýzy bude proveden návrh aplikace.

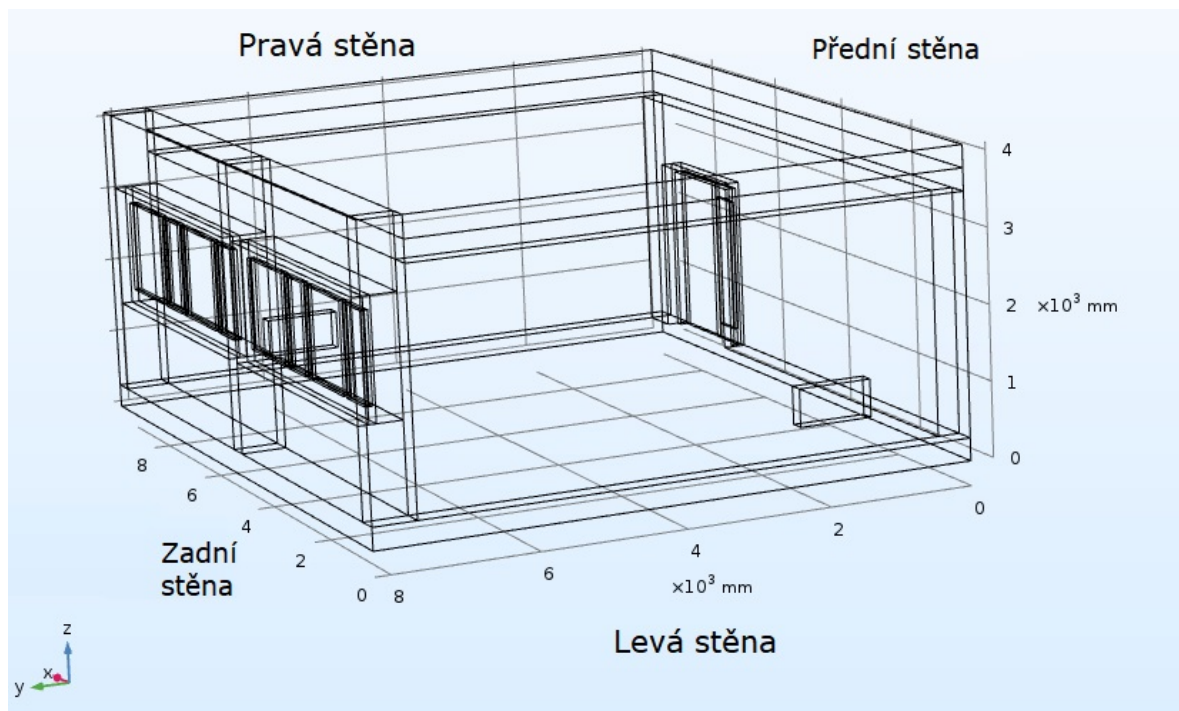
#### 3.1 Geometrické a termofyzikální vstupní parametry studovaného modelu

Z pohledu sdílení tepla v modelu dochází k několika důležitým dějům. Mezi místností a jejím okolím dochází k prostupu tepla jednoduchou nebo složenou stěnou, která je zde zastoupena jak již stěnami místností a jejich izolací a nebo také okny a dveřmi. Pro tento jev je nezbytné stanovit tloušťku pevné přepážky a materiál, ze kterého se skládá, který je podstatný pro součinitel tepelné vodivosti a byl popsán v podkapitole 1.5. Přímotopy jsou zdrojem sdílení tepla pomocí sálání a přestupu.

Geometrický náčrt modelu je znázorněn na obrázku 3.1. Představuje jej uzavřená neklimatizovaná místnost v obytném domě s vnější stěnou, na které je umístěno osm oken obklopených rámy. Dále v této místnosti, ze které vedou jediné dveře, jsou na protilehlých stěnách dva přímotopy. U zmíněných prvků je možné konfigurovat jednotlivé geometrické rozměry a umístění včetně tloušťky stropu, jeho izolace a podlahy. Pomocí parametrizovaných vstupů je možné měnit pouze hodnoty existujících prvků a není možné přidávat další, jako například přidat nové okno.

Celkový počet parametrizovaných vstupů je 195 a lze je rozdělit do několika kategorií.

- **Rozměry geometrických prvků místnosti** - model je trojrozměrný a tedy u všech prvků se nastavuje šířka, výška a tloušťka. Pro určení tepelné stability jsou důležité nejenom geometrické rozměry prvků, ale také jejich přesné umístění.
  - **Základní prvky konstrukce místnosti** - základní prvky konstrukce představují všechny obvodové stěny, podlaha a strop včetně jeho izolace, u nichž lze volit jednotlivé rozměry, jako je například tloušťka stropu a jeho tepelné izolace, výška místnosti, šířka sloupů ve stěně s okny a vzdálenost levého sloupu od vnějšího povrchu levé stěny.

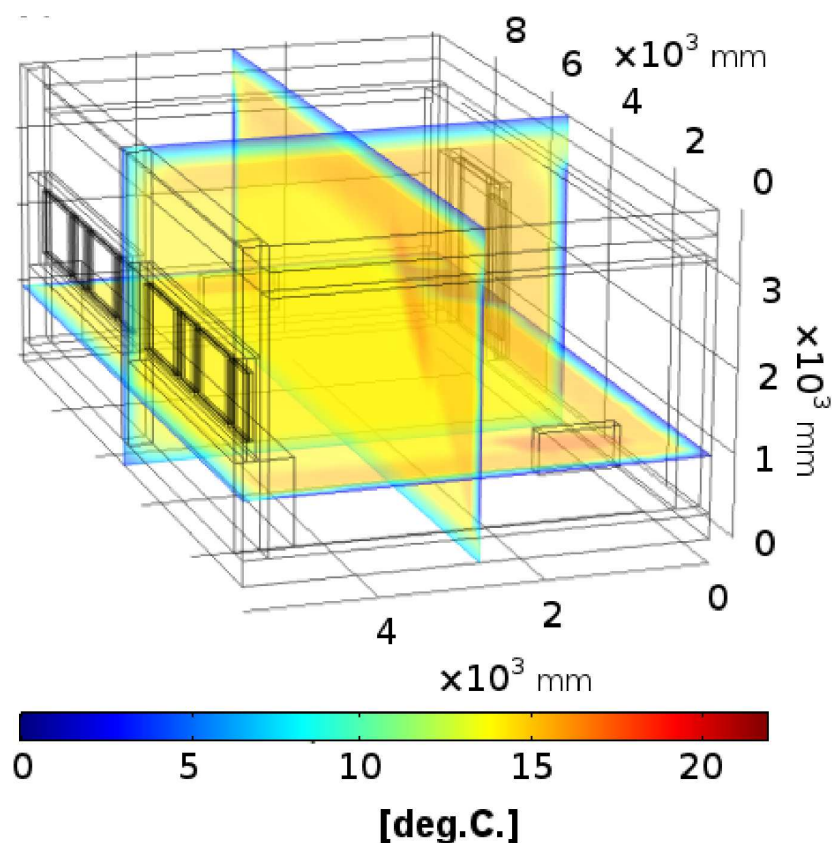


Obr. 3.1 Ukázka studovaného modelu místnosti

- **Okna včetně rámu** - každé okno obsahuje také rám a tyto prvky jsou specifikovány následujícími parametry: tloušťka, šířka a výška okna nebo rámu. Dále vzdálenost od podlah a také vzdálenost od vnějšího povrchu stěny.
- **Dveře včetně rámu a přímotopy** - dveře jsou složeny z několika částí a to křídla, plastové tabule a rámu. Pro přímotopy se nastavuje například vzdálenost od podlahy.
- **Fyzikální vlastnosti** - z hlediska výpočtu tepelné stability je nutné definovat u všech prvků představující tuhé materiály hodnoty těchto vlastností:
  - **Měrná tepelná kapacita**
  - **Součinitel tepelné vodivosti**
  - **Hustota**
- **Podmínky simulace a vyhodnocení**
  - **Počáteční a okrajové podmínky** - zde se nastavují následující parametry: součinitel přestupu tepla na venkovní stěně budovy a uvnitř budovy a také počáteční teplota vzduchu uvnitř místnosti.
  - **Požadavky na simulaci** - vzhledem k tomu, že chování modelu je popsáno diferenciální rovnicí, která je zde řešena numerickou metodou, tak je

nutné zadat počáteční čas a krok numerického řešení. Zadává se i celková doba ohřevu/chladnutí místnosti jako ukončující časová podmínka pro běh simulace.

- **Podmínky vyhodnocení výsledků** - výstupem simulace je také zobrazení teploty a tepelného toku a to pomocí řezů místností v jednotlivých rovinách. Pro obě tyto varianty se v simulaci vytvoří tři řezy a je u nich nutné nastavit přesnou geometrickou pozici viz obrázek 3.2.



Obr. 3.2 Zobrazení teploty v řezích modelem po ukončení simulace

Protože se jedná o parametrizovaný model, tak má být uživateli umožněno zadávat libovolné hodnoty a jediné omezení tedy bude, aby se jednalo o reálné čísla. U všech parametrů existuje nějaká výchozí hodnota, kterou je nutné použít i v aplikaci, což může být vhodné, pokud uživatel změní jen některé hodnoty.

### 3.2 Formát souboru pro import do studovaného modelu

Jednotlivé prvky modelu mají v COMSOLu unikátní identifikaci a tedy díky tomu je možné k nim přistupovat a měnit jejich hodnotu přes import souboru obsahující hodnoty prvků. Tento soubor musí být v určitém formátu a lze jej jednoduše do v COMSOLu importovat.



Import podporuje formáty .csv, .xsl a textový formát. U textového formátu každý řádek v souboru reprezentuje nastavení jednoho atributu a obsahuje unikátní identifikátor, hodnotu s její jednotkou a slovní popis. Jednotka se nezadává u všech parametrů, ale pouze u geometrických hodnot, časových jednotek a teploty. Řádky jsou ve formátu:

```
identifikátor hodnota[jednotka] "slovní popis hodnoty"
```

Pro .csv je formát souboru velmi podobný, ale jednotlivé hodnoty nejsou oddělené mezerami nýbrž čárkou.

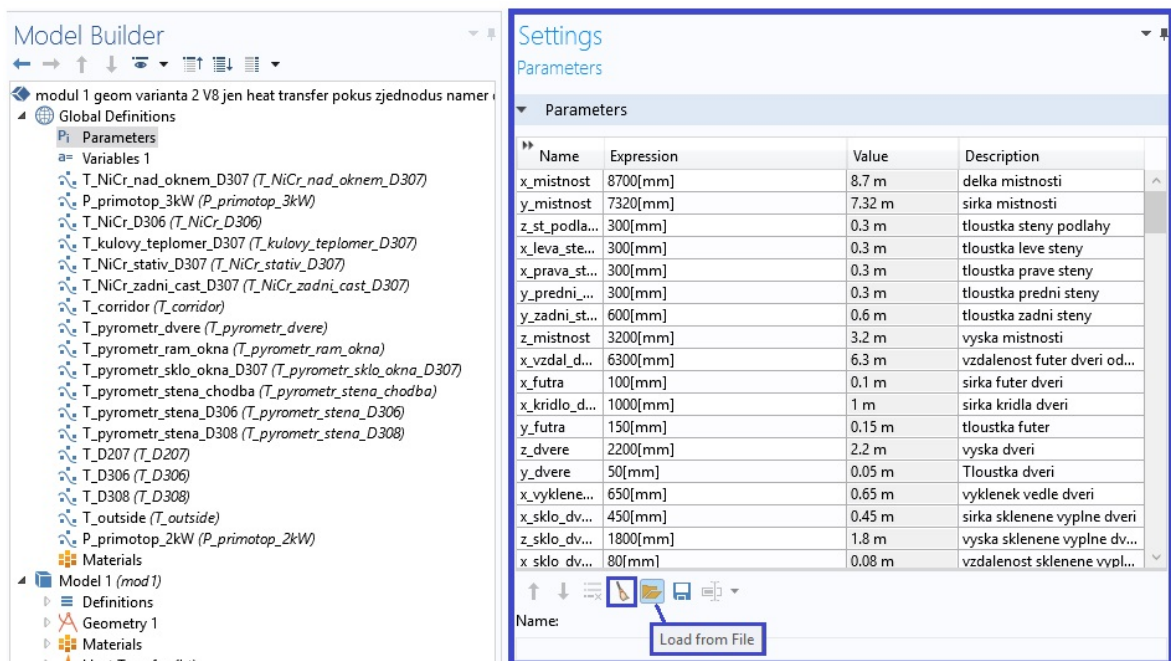
```
identifikátor,hodnota[jednotka],slovní popis hodnoty
```

Ukázka formátu vstupních dat pro nastavení části základních prvků konstrukce místnosti:

```
x_mistnost 8700 [mm] "Délka místnosti"  
y_mistnost 7320 [mm] "Šířka místnosti"  
z_mistnost 3000 [mm] "Výška místnosti"
```

### 3.3 Import textového souboru se vstupními parametry do studovaného modelu

V COMSOLu je seznam parametrizovaných vstupních parametrů zobrazen formou tabulky. Všechny vstupní parametry lze modifikovat přímo zde a nebo je možnost je importovat ze vstupního souboru. Před samotným importem je důležité všechny dosavadní vstupní parametry smazat, protože jinak by po importu došlo k jejich duplikaci a COMSOL by hlásil chybu. Samotný import je velmi jednoduchý a je zobrazen na obrázku 3.3. Po úspěšném importu je možné pustit simulaci s novými vstupními parametry.

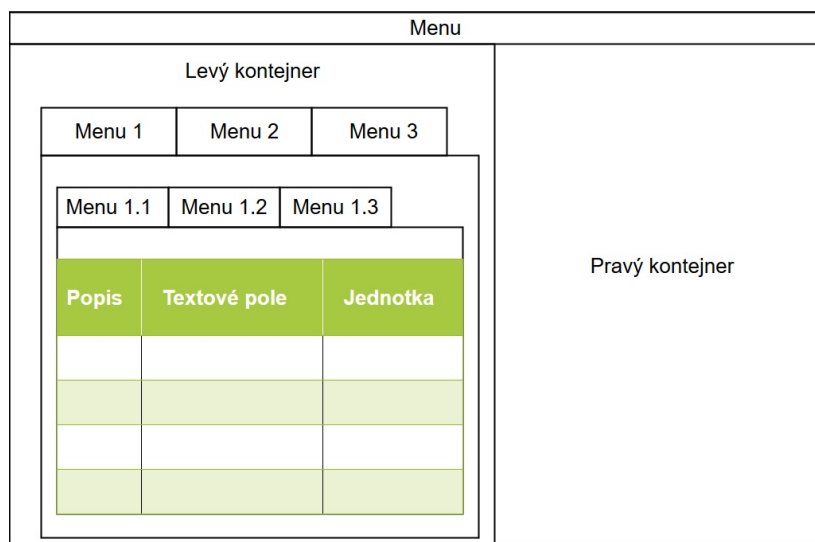


Obr. 3.3 Ukázka importu textového souboru se vstupními parametry do studovaného modelu

## III. PROJEKTOVÁ ČÁST

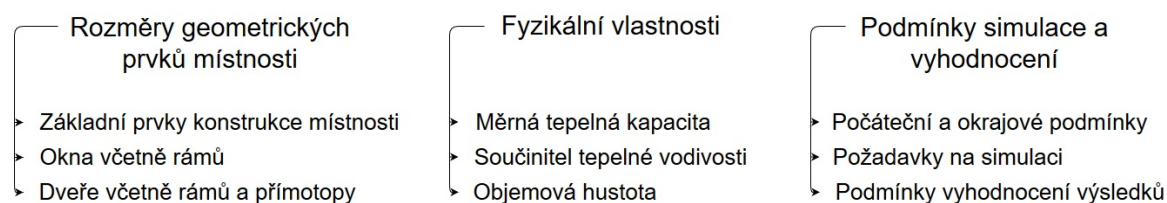
#### 4 Návrh aplikace

Hlavní menu aplikace bude ponecháno na standardním místě v horní části obrazovky a zbylá část okna bude rozdělena na dvě části. V levém kontejneru bude zobrazen formulář pro zadávání parametrizovaných hodnot a v pravém kontejneru se budou zobrazovat doplňující informace k právě zadávanému poli. Návrh aplikace lze vidět na obrázku 4.1.



Obr. 4.1 Návrh rozvržení prvků aplikace

Při návrhu je nutné počítat s tím, že počet parametrizovaných vstupů je vysoký a není možné je všechny zobrazit najednou. Tyto parametry je možné rozdělit do logických kategorií a podkategorií viz obrázek 6.3, jak to bylo naznačeno v sekci 3. V levém kontejneru budou ovládací prvky pro přepínání mezi jednotlivými kategoriemi a podkategoriemi. Vstupní parametry vztahující se k právě vybrané kategorii/podkategorii budou zobrazeny formou tabulky.



Obr. 4.2 Kategorizace vstupních parametrizovaných parametrů

## 5 Použité technologie

Při vývoji JavaFX aplikace jsem využíval volně dostupné neplacené programy a technologie zmíněné v této kapitole.

### 5.1 Netbeans

Netbeans je vývojové prostředí pro tvorbu, překlad a ladění aplikací. Jedná se o otevřený software se spoustou dostupných rozšiřujících modulů a podporuje širokou paletu programovacích jazyků. Při vývoji aplikace jsem použil Netbeans verze 8.1.

### 5.2 Java Development Kit

Pro tvorbu programů a appletů je nutné mít nainstalovaný Java Development Kit (JDK), který obsahuje soubor základních nástrojů pro vývoj aplikací. Jeho součástí je Java Runtime Environment (JRE), pomocí kterého se spouští aplikace vytvořené v Java. JavaFX 2 je aktuálně také součástí JDK, ale v budoucnosti by měla být vyčleněna do vlastního modulu. Při tvorbě aplikace jsem využíval verzi JDK 1.8.073.

### 5.3 Maven

Nástroj pro správu projektů a jejich závislostí, který usnadňuje proces buildování, je založen na objektovém modelu projektu reprezentovaným souborem pom.xml (Project Object Model-POM). Používal jsem verzi Apache Maven 3.3.9.

## 6 Implementace

Pro tvorbu aplikace jsem využíval vývojové prostředí Netbeans, které bylo popsáno v kapitole 5. Aplikace byla vytvořena jako Maven JavaFX projekt, díky čemuž má specifickou adresářovou strukturu, pro jednodušší proces správy závislostí a vytváření buildů.

Jako první jsem začal s implementací hlavního kontejneru okna obsahujícího formulář pro zadávání parametrizovaných hodnot, což je popsáno v kapitole 6.1. Dále jsem vytvořil kontroléry a funkcionalitu exportu do textového souboru, který je obsahem kapitoly 6.3. V kapitole 6.2 popisují implementaci zobrazení 3D modelu studované místnosti v aplikaci. Dále proběhlo naprogramování pravého kontejneru pro zobrazování pomocných doplňujících informací.

### 6.1 Vytváření grafického uživatelského rozhraní pro zadávání vstupních parametrů

Při implementaci jsem se rozhodl využívat pro tvorbu prvků uživatelského rozhraní zejména FXML, které bylo rozebráno v sekci 2.2. Jako cíl jsem si stanovil, aby veškeré informace týkající se parametrizovaných vstupních hodnot, jako například popis pole a výchozí hodnota, byly pouze v FXML šablonách a kontroléry by sloužily pouze pro zpracování hodnot. Výhoda tohoto řešení je, že pokud by byla potřeba přidat nový vstupní parametr, či upravit stávající, tak by došlo k úpravě pouze v FXML šabloně. Navíc tuto úpravu by mohl udělat i člověk, který s danou problematikou nemá příliš zkušeností, protože struktura FXML je relativně intuitivní a je podobná formátu XML, který je velmi rozšířený.

Každá kategorie, která je zobrazená na obrázku 6.3, má vlastní šablonu a kontrolér, který slouží například pro zpracování událostí vzniklých na základě akce uživatele. Ve výsledku jsou tedy tři kontroléry, které jsou potomky abstraktní třídy *AbstractController*, která umožňuje sdílení společné funkcionality. Uživatel může přepínat mezi jednotlivými kategoriemi a podkategoriemi viz obrázek 6.1 a při každém kliknutí dochází ke změně zobrazených vstupních parametrizovaných hodnot v levé části kontejneru. Parametrizované vstupy jsou uloženy v tabulce realizované pomocí *GridPane*. Na pravé straně tabulky je rolovací prvek, jelikož některé podkategorie obsahují mnoho vstupních hodnot a přesahují tak maximální možnou výšku okna.

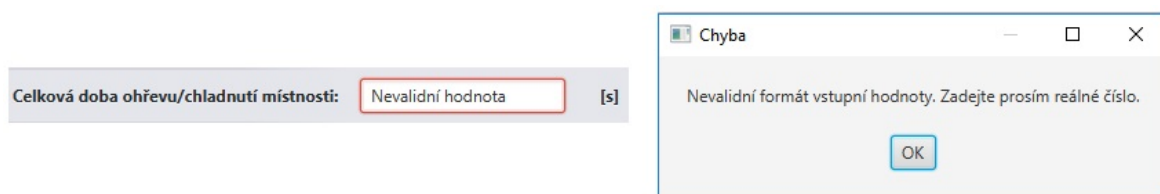
Při zadávání vstupních hodnot uživatelem dochází k validaci, jestli se jedná o reálné číslo a nebo kladné reálné číslo podle toho, co dává pro daný parametr smysl. Pokud tato podmínka není splněna, tak je zobrazeno vyskakovací modální okno s chybovou hláškou a dané pole je červeně zvýrazněno viz obrázek 6.2, dokud nedojde k opravě. Jestliže by byla potřeba striktnější validace, stačí vytvořit novou třídu implementující



The screenshot shows a software interface with a top navigation bar containing four tabs: 'Rozměry geom. prvků místnosti', 'Fyzikální vlastnosti', 'Podmínky simulace a vyhodnocení', and '3D model'. Below this, there are three sub-sections: 'Základní prvky konstrukce místnosti', 'Okna včetně rámu', and 'Dveře včetně rámu a přímotopy'. The 'Základní prvky konstrukce místnosti' section contains three input fields: 'Délka místnosti:' with the value '8700' and unit '[mm]', 'Šířka místnosti:' with the value '7320' and unit '[mm]', and 'Výška místnosti:' with the value '3000' and unit '[mm]'.

Obr. 6.1 Zobrazení detailu kategorií/podkategorií geometrických a termofyzikálních vstupních parametrů ve vytvořené aplikaci

rozhraní *ValidateTextField* a kontrolér poté bude automaticky nad těmito prvky provádět definovanou validaci. Zároveň v pravém kontejneru se zobrazují obrázky ukazující zvýrazněný prvek ve 3D modelu, který uživatel právě mění ve formuláři. Například pokud uživatel začne pracovat s textovým polem pro tloušťku stropu, tak se zobrazí obrázek modelu studované místnosti pro tepelnou stabilitu se zvýrazněným stropem.



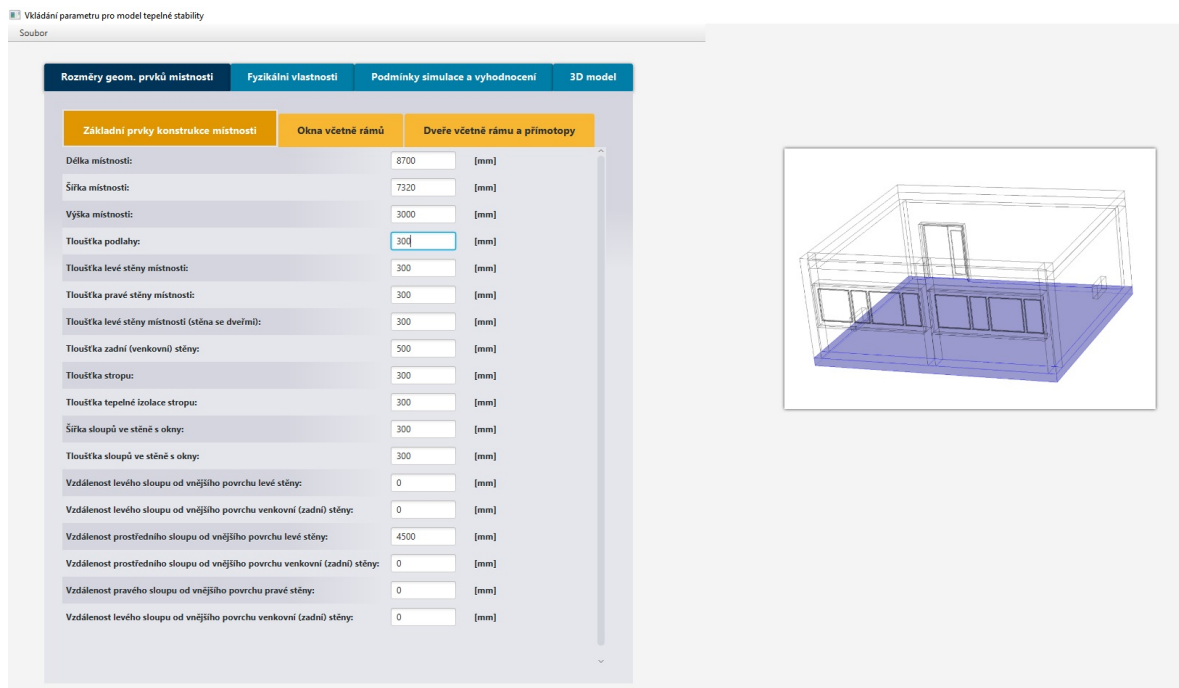
Obr. 6.2 Upozornění uživatele při zadání nevalidní hodnoty v aplikaci

## 6.2 Tvorba a zobrazení 3D modelu studované místnosti

Jelikož počet vstupních parametrů je poměrně vysoký, rozhodl jsem se v rámci aplikace provést zobrazení studovaného 3D modelu pro tepelnou stabilitu místnosti, aby se uživatel jednodušeji zorientoval.

Při počátečním návrhu jsem chtěl, aby 3D model byl co nejvíce interaktivní. Tedy aby umožňoval uživateli nejenom prohlížení modelu a jeho otáčení, ale také změnu velikosti či pozice jednotlivých geometrických prvků na základě změny vstupních parametrů, které uživatel může pomocí formuláře měnit. Například, aby při navýšení velikosti šířky místnosti, byla tato změna v reálném čase promítnuta i v 3D modelu. Další chtěná funkcionalita byla, aby se po kliknutí na některé vstupní pole zvýraznila i odpovídající část v 3D modelu. Tím je myšleno, že pokud by uživatel kliknul či zadával hodnotu například u pole pro rozměry stropu, tak aby došlo i v modelu ke zvýraznění stropu oproti ostatním prvkům modelu. Cílem bylo, aby uživatel měl co největší přehled o tom, jaké hodnoty právě mění a jaký vliv to má na studovaný model.

Při realizaci tohoto návrhu jsem nicméně narazil na několik problémů. Jedna z prv-



Obr. 6.3 Zobrazení celého okna programovaná aplikace

ních možností byla exportovat 3D model studované místnosti přímo z COMSOLu a poté ho načíst aplikací. Exportovaný soubor obsahuje všechny informace pro vykreslení 3D modelu, ale dochází zde ke ztrátě informace o tom, co která křivka znamená. Není tedy možné přesně identifikovat které křivky odpovídají například rámu dveří.

Další uvažovanou variantou bylo vytvořit kompletně celý model znovu přímo v JavaFX pomocí struktury MashView, která slouží pro tvorbu libovolných tvarů. Tohle řešení nicméně zvyšuje riziko pro nekonzistentnosti 3D modelu místností mezi aplikací JavaFX a modelem v COMSOLu. Pokud by došlo ke změně 3D modelu v COMSOLu, tak by také musela být upravena aplikace, což nemusí být rychlé a jednoduché změny a klade to zbytečné nároky z hlediska udržitelnosti aplikace. Po několika experimentech jsem i tuto variantu zavrhl i v důsledku výše zmíněných důvodů.

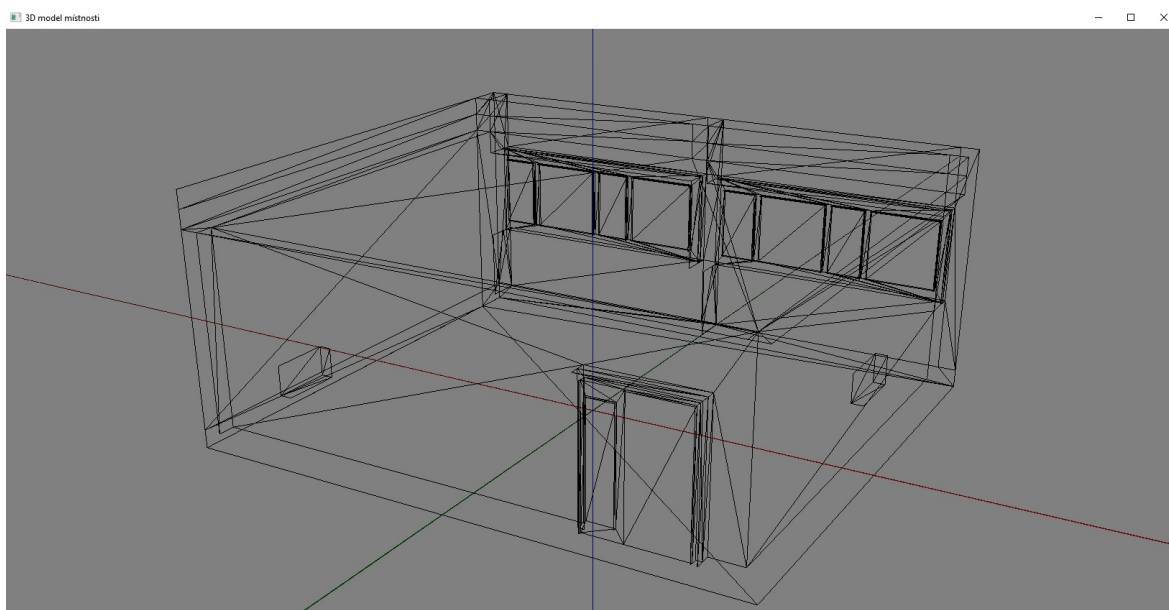
Pro finální řešení jsem se rozhodl využít exportovaný model studované místnosti a dále jeho zobrazení v aplikaci. COMSOL umožňuje exportovat 3D model v několika různých formátech (COMSOL Multiphysics binary, Parasolid binary, STL binary a další). Samotná JavaFX však neumožňuje import ani jednoho tohoto formátu, a proto jsem využil externí knihovnu, která dokáže importovat 3D model ve formátu STL a následně načíst křivky do třídy *MeshView*. Na základě načtených hodnot se vytvoří instance třídy *Group*, která již může být umístěna jako prvek scény. U modelu nejsou nastaveny žádné textury a je průhledný, protože při nastavení textur by uživatel nemohl vidět všechny prvky modelu, ale pouze ty vnější. Vidět tedy jsou pouze obrysy objektů a lze zde pozorovat, jak jsou jednotlivé objekty sestavené z trojúhelníků stejně jako



u zmíněného příkladu rozloženého povrchu krychle viz obrázek 2.4.

Pokud bychom vykreslili tento objekt, tak byl by vidět statický 3D model, který nelze otáčet, a proto je nutné ještě přidat perspektivní kameru. U kamery se nastavuje její počáteční pozice v kartézské soustavě a nejbližší/nejvzdálenější pozice od kamery, kterou rendruje. Pro zpracování informace od uživatele o operaci, která se má provést (otočení, přiblížení, atd.) se využívá reakce na události. Na každou událost je navázána obslužná rutina, která zabezpečuje, že například při posunu kolečka na myši se zároveň změní i pozice kamery v požadovaném směru. Kamera umožňuje rotaci, přiblížení a oddálení 3D modelu. Pro lepší orientaci jsem přidal i zobrazení jednotlivých os kartézského souřadnicového systému.

Pro výsledný model je vytvořen nový *stage*, který byl popsán v kapitole 2.1.1, a je zobrazen ve vlastním okně, jak lze pozorovat na obrázku 6.4. U prvního návrhu formulář pro vstupní parametrizované hodnoty a 3D model sdílely stejný *stage* a docházelo pouze k přepínání scén. Pokud uživatel zavřel okno aplikace, když si prohlížel 3D model, tak mohl nechtěně ztratit i zadané hodnoty ve formuláři. Jestliže dojde ke znovuotevření 3D modelu, tak model zůstane stejně natočený, jako při posledním zavření.



Obr. 6.4 Zobrazení studovaného 3D modelu ve vytvořené aplikaci

### 6.3 Export zadaných vstupních parametrů do souboru

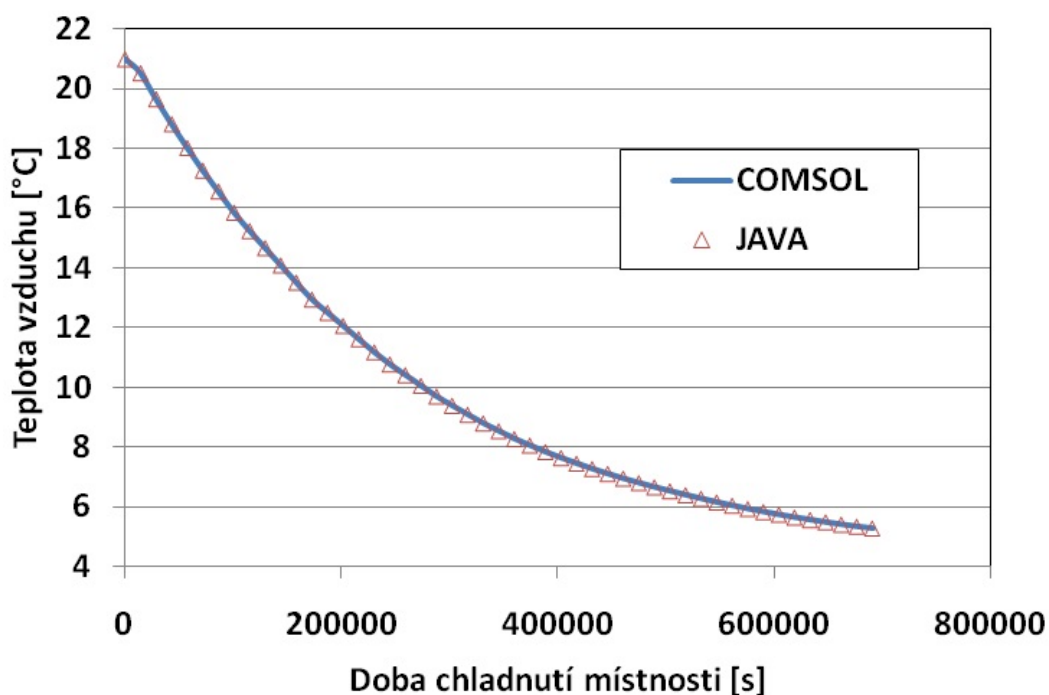
Tlačítko pro export zadaných parametrizovaných vstupů je v horním menu aplikace. Uživateli se zobrazí okno s možností vybrat si lokaci pro uložení exportovaného souboru. Implementována byla varianta pro export do textového formátu, který byl definován v kapitole 3.2, třídou *TxtFileExport*, ale aplikace lze jednoduše rozšířit o další pod-

porované formáty. Stačí vytvořit novou implementaci třídy *AbstractFileExport*, která je odpovědná za agregaci všech potřebných informací o parametrizovaných vstupech pomocí jednotlivých kontrolérů.

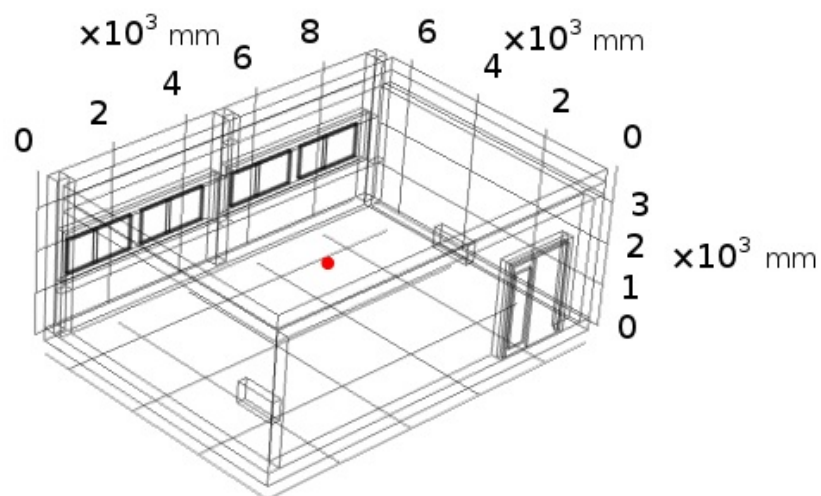
## 7 Testování funkčnosti programované aplikace

Ověření funkčnosti modelu bylo provedeno porovnáním simulace se vstupními daty vytvořenými JAVA aplikací a importovanými do programu COMSOL v podobě TXT souborů a simulací provedenou za stejných podmínek s daty vloženými do modelu přímo v uživatelském prostředí programu COMSOL. V testovaném příkladu bylo uvažováno vychládání místnosti o počáteční teplotě 21 °C v zimním období.

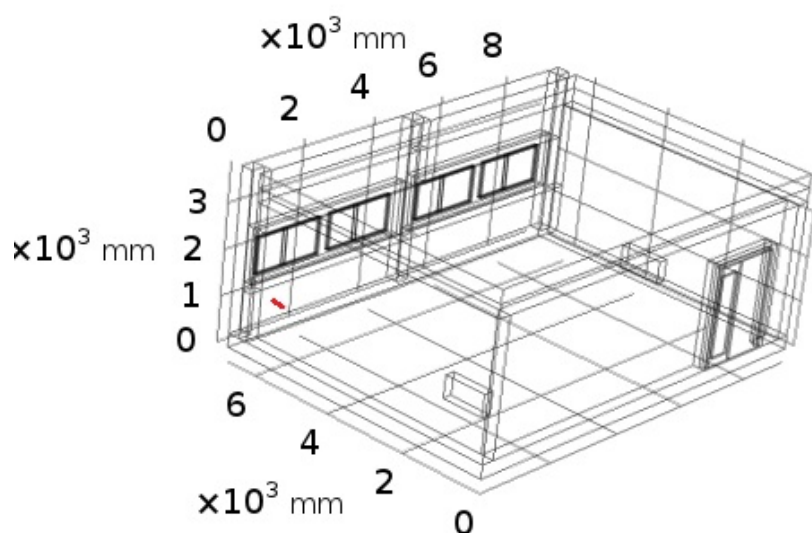
Výsledky simulací jsou uvedeny v obrázcích 7.1 a 7.4. Na obrázku 7.1 jsou zobrazeny časové průběhy teploty vzduchu ve středu místnosti, na obrázku 7.4 jsou zobrazeny časové průběhy teploty ve venkovní stěně místnosti. Místa odečítání teploty jsou zobrazena na obrázcích 7.2 a 7.3. Výsledky potvrdily shodu simulací pro oba postupy vkládání vstupních dat do modelu studované místnosti.



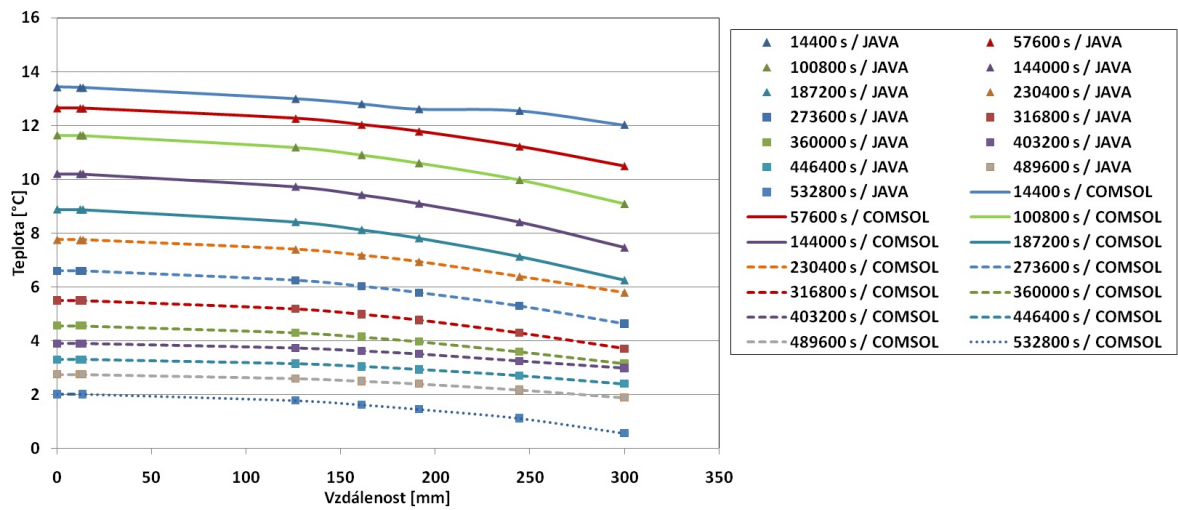
Obr. 7.1 Časový průběh teploty vzduchu ve středu místnosti při jejím vychládání - výsledky simulací se vstupními daty importovanými do modelu z JAVA aplikace a s daty vloženými do modelu přímo v uživatelském prostředí programu COMSOL



Obr. 7.2 Model s vyznačením místa středu místnosti (červený bod), v němž byla odečítána teplota v průběhu simulace vychládání místnosti.



Obr. 7.3 Model s vyznačením místa ve venkovní stěně (červená křivka), v němž byla odečítána teplota v průběhu simulace vychládání místnosti.



Obr. 7.4 Teplota venkovní stěny místnosti při jejím vychládání - výsledky simulací se vstupními daty importovanými do modelu z JAVA aplikace a s daty vloženými do modelu přímo v uživatelském prostředí programu COMSOL.

## ZÁVĚR

V bakalářské práci jsem se zabýval tvorbou pomocné Java aplikace pro vkládání vstupních dat do parametrizovaného modelu určeného pro testování tepelné stability místnosti, připraveného v programu COMSOL Multiphysics. Cílem práce bylo navrhnout a vytvořit uživatelské rozhraní umožňující zadání všech požadovaných vstupních hodnot a následný export do formátu podporovaného programem COMSOL.

Pro základní pochopení fyzikální podstaty studovaného modelu jsem se v teoretické části práce seznámil se základními mechanismy sdílení tepla probíhajícími ve studovaném modelu místnosti. Za tímto účelem jsem v teoretické části bakalářské práce popsal základní principy a zákony popisující časově ustálené vedení tepla ve vícevrstvé rovině stěně, přestup tepla a sálání tepla. Následně jsem se zabýval popisem sdílení tepla prostupem jednovrstvou a vícevrstvou rovinou stěnou.

Za účelem získání znalosti ohledně tvorby uživatelského grafického rozhraní v JavaFX jsem v teoretické části dále popisoval základní přehled funkcionality JavaFX včetně tvorby 3D objektů.

Pro nabytí informací důležitých pro návrh aplikace jsem popsal a kategorizoval jednotlivé vstupní parametrizované hodnoty a na základě toho jsem je rozdělil do odpovídajících sekcí v grafickém rozhraní aplikace. Podle formátu podporovaného programem COMSOL jsem vytvořil funkcionalitu exportu parametrizovaných hodnot do TXT souboru. Pro zjednodušení a zpříjemnění práce uživatele s aplikací jsem do pravé části aplikace přidal pomocné obrázky se zvýrazněnými částmi modelu a také interaktivní 3D model studované místnosti.

Funkčnost aplikace jsem na studovaném modelu ověřil porovnáním simulací se vstupními daty vytvořenými pomocí Java aplikace a importovanými do programu COMSOL v podobě TXT souborů a simulací provedenou za stejných podmínek s daty vloženými do modelu přímo v uživatelském prostředí programu COMSOL.

Jako námět pro další práci bych navrhnul například ukládání rozpracovaného formuláře se vstupními parametry a poté jeho znovu načtení. Rovněž by bylo možné provést úpravy, které by umožňovaly vkládat do aplikace další okrajové podmínky, jako je příkon přímotopu, teplota okolních místností apod. Tyto hodnoty se však nyní získávají v podobě experimentálně stanovených dat, ukládanými již příslušnými měřicími zařízeními do samostatných souborů, které lze do modelu v COMSOLu přímo importovat. Proto není potřeba provádět jejich vkládání do JAVA aplikace.

**SEZNAM POUŽITÉ LITERATURY**

- [1] VAVERKA, Jiří. Stavební tepelná technika a energetika budov. Brno: VUTIUM, 2006. ISBN 80-214-2910-0.
- [2] HEJZLAR, Radko. Sdílení tepla. Vyd. 4. V Praze: Vydavatelství ČVUT, 2004. ISBN 8001029743.
- [3] PŘÍHODA, Miroslav a Miroslav RĚDR. Sdílení tepla a proudění. 2., přeprac. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2008. ISBN 978-80-248-1748-4.
- [4] JÍCHA, Miroslav. Přenos tepla a látky. Brno: CERM, 2001. Učební texty vysokých škol. ISBN 8021420294.
- [5] JANÁČOVÁ, Dagmar a Hana CHARVÁTOVÁ. Procesy v technice budov. Prostup tepla cvičení. Zlín: Univerzita Tomáše Bati, 2013.
- [6] BOŠOVÁ, Daniela a František KULHÁNEK. Stavební fyzika II: stavební tepelná technika. 6., přeprac. vyd. V Praze: České vysoké učení technické, 2014. ISBN 9788001056455.
- [7] ČSN 73 0540: Tepelná ochrana budov. Úřad pro normalizaci, metrologii a zkušebnictví, Praha, 2011.
- [8] SHARAN, Kishori. Learn JavaFX 8: building user experience and interfaces with Java 8. New York: Apress, 2015. Expert's voice in Java. ISBN 978-1-4842-1143-4.
- [9] VOS, Johan. Pro JavaFX 8: a definitive guide to building desktop, mobile, and embedded Java clients / Johan Vos, Weiqi Gao, Stephen Chin, Dean Iverson, James Weaver. Berkeley, CA: Apress, 2014. Expert's voice in Java. ISBN 1430265744.
- [10] ŘEHÁNEK, Jaroslav. Tepelná akumulace budov. Praha: ČKAIT, 2002. ISBN 80-86364-59-3.
- [11] KULHÁNEK, František. Stavební fyzika II. Stavební tepelná technika. 1. vyd. Praha: ČVUT Praha, 2013. ISBN 978-80-01-04957-0.
- [12] SCHILDT, Herbert. Java: the complete reference. Ninth edition. New York: McGraw-Hill Education, 2014. ISBN 0071808558.
- [13] CHARVÁTOVÁ, Hana, ZÁLEŠÁK, Martin, SEHNÁLEK, Stanislav. Parameterization of module for testing of thermal stability in the room. In Latest Trends on Systems. Volume II. Rhodes : Europment, 2014, s. 647-650. ISSN 1790-5117. ISBN 978-1-61804-244-6.

- 
- [14] Introduction to COMSOL Multiphysics [online]. 2017 [cit. 2017-11-29]. Dostupná z: <http://www.humusoft.cz>.
- [15] HORSTMANN, Cay S. Core Java. Tenth edition. Boston: Prentice Hall, 2016. ISBN 0-13-417730-4.
- [16] MACHÁČKOVÁ, Adéla. Sdílení tepla a proudění: učební text. Ostrava: Vysoká škola báňská - Technická univerzita, 2012. ISBN 978-80-248-2576-2.
- [17] CHARVÁTOVÁ, Hana, ZÁLEŠÁK, Martin, SEHNÁLEK, Stanislav, JANÁČOVÁ, Dagmar, VAŠEK, Vladimír, DRGA, Rudolf. Computer simulation of parameterized module for testing of thermal stability in the room. *International Journal of Mathematics and Computers in Simulations*, 2015, roč. 9, s. 53-59. ISSN 1998-0159.
- [18] CHARVÁTOVÁ, Hana, ZÁLEŠÁK, Martin. Programování modulů pro simulaci rozložení teploty při vytápění a chladnutí místnosti. *iDB Journal*, 2015, roč. Ne-  
veden, č. 2, s. 40-43. ISSN 1338-3337.



## SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

$a$	teplotní vodivost	$[\text{m}^2 \cdot \text{s}^{-1}]$
$A$	pohltivost tělesa při sálání	[1]
$c$	měrná tepelná kapacita	$[\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}]$
$C$	konstanta Nusseltova kritéria	[1]
$c_p$	měrná tepelná kapacita	$[\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}]$
$C_c$	součinitel sálání absolutně černého tělesa	$C_c = [\text{Wm}^{-2} \cdot \text{K}^{-4}]$
$D$	propustnost tělesa při sálání	[1]
$g$	gravitační zrychlení	$[\text{m} \cdot \text{s}^{-2}]$
$q$	hustota tepelného toku	$[\text{W} \cdot \text{m}^{-2}]$
$q_c$	hustota tepelného sálavého toku	$[\text{W} \cdot \text{m}^{-2}]$
$\dot{Q}$	tepelný tok	[W]
Gr	Grashofovo kritérium	[1]
$l$	charakteristický rozměr	[m]
Nu	Nusseltovo kritérium	[1]
Pe	Pécletovo kritérium	[1]
Pr	Prandtlovo kritérium	[1]
$R$	odrazivost tělesa při sálání	[1]
Re	Reynoldsovo kritérium	[1]
$S$	plocha	$[\text{m}^2]$
St	Stantonovo kritérium	[1]
$t$	teplota	$[\text{°C}]$
$t_p$	teplota povrchu stěny	$[\text{°C}]$
$T$	termodynamická teplota	[K]
$x, y, z$	souřadnice v kartézské soustavě	
$v$	rychlost	$[\text{m} \cdot \text{s}^{-1}]$
$\alpha$	součinitel přestupu tepla	$[\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}]$
$\beta$	koeficient teplotní objemové roztažnosti	$[\text{K}^{-1}]$
$\epsilon$	emisivita, poměrná zářivost	[1]
$\eta$	dynamická viskozita	[Pa.s]
$\delta$	tloušťka	[m]
$\lambda$	součinitel tepelné vodivosti	$[\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}]$
$\rho$	hustota	$[\text{kg} \cdot \text{m}^{-3}]$
$\sigma$	Stefan-Boltzmannova konstanta	$\sigma = 5,669 \cdot 10^{-8} [\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-4}]$
$\tau$	čas	[s]
$\nu$	kinematická viskozita	$[\text{m}^2 \cdot \text{s}^{-1}]$

## SEZNAM OBRÁZKŮ

Obr. 1.1	Gradient teplotního pole [4] . . . . .	12
Obr. 1.2	Vedení tepla složenou stěnou [5] . . . . .	14
Obr. 1.3	Rozdělení sálavého toku na jeho jednotlivé složky [6] . . . . .	17
Obr. 1.4	Prostup tepla složenou rovinnou stěnou [5] . . . . .	20
Obr. 2.1	Struktura hlavních komponent JavaFX [8] . . . . .	23
Obr. 2.2	Hierarchie dědičnosti třídy uzal [8] . . . . .	24
Obr. 2.3	3D krychle se souřadnicemi [8] . . . . .	27
Obr. 2.4	Povrch rozložené krychle na textury [8] . . . . .	28
Obr. 3.1	Ukázka studovaného modelu místnosti . . . . .	31
Obr. 3.2	Zobrazení teploty v řezech modelem po ukončení simulace . . . . .	32
Obr. 3.3	Ukázka importu textového souboru se vstupními parametry do studovaného modelu . . . . .	34
Obr. 4.1	Návrh rozvržení prvků aplikace . . . . .	36
Obr. 4.2	Kategorizace vstupních parametrizovaných parametrů . . . . .	36
Obr. 6.1	Zobrazení detailu kategorií/podkategorií geometrických a termofyzikálních vstupních parametrů ve vytvořené aplikaci . . . . .	39
Obr. 6.2	Upozornění uživatele při zadání nevalidní hodnoty v aplikaci . . . . .	39
Obr. 6.3	Zobrazení celého okna programovaná aplikace . . . . .	40
Obr. 6.4	Zobrazení studovaného 3D modelu ve vytvořené aplikaci . . . . .	41
Obr. 7.1	Časový průběh teploty vzduchu ve středu místnosti při jejím vychládání - výsledky simulací se vstupními daty importovanými do modelu z JAVA aplikace a s daty vloženými do modelu přímo v uživatelském prostředí programu COMSOL . . . . .	43
Obr. 7.2	Model s vyznačením místa středu místnosti (červený bod), v němž byla odečítána teplota v průběhu simulace vychládání místnosti. . .	44
Obr. 7.3	Model s vyznačením místa ve venkovní stěně (červená křivka), v němž byla odečítána teplota v průběhu simulace vychládání místnosti. . .	44
Obr. 7.4	Teplota venkovní stěny místnosti při jejím vychládání - výsledky simulací se vstupními daty importovanými do modelu z JAVA aplikace a s daty vloženými do modelu přímo v uživatelském prostředí programu COMSOL. . . . .	45