Technická zpráva k projektu IMS – Pekárna

Jan Doležel (xdolez81@stud.fit.vutbr.cz)
Adam Gajda (xgajda07@stud.fit.vutbr.cz)

12. prosince 2021

Zadání

Z důvodu toho, že by pekárna mohla být brána jako příliš jednoduché zadání, jsme při modelování zahrnuli i výrobu mouky a následný prodej dokončeného produktu.

10 SHO ve výrobě

Prostudujte technologii výroby vybraného produktu. Zaměřte se na reálný výrobní podnik. Výrobu modelujte jako SHO. Ukažte propustnost výroby a úzká místa.[PPa]

1 Úvod

V této práci se zabýváme simulací (viz. [PPc], slajd 8) modelu(viz. [PPc], slajd 7) farmy produkující pšenici a poté pšeničnou mouku, která je dále využita v pekárně pro výrobu chleba. Tento chléb je následně prodán zákazníkům pekárny. Smyslem experimentů je dosáhnout optimální produkce pšenice pro potřebu pekárny, závislou na poptávce ze strany zákazníků pekárny. Na základě modelu a simulačních experimentů bude ukázáno chování systému v ne zcela vhodných situacích.

1.1 Na projektu se podíleli, významné zdroje informací

Na práci se jako autor podílel Jan Doležel a Adam Gajda. Data která stojí za zmínku:

- Výroba chleba[Evo]
- Ingredience na chleba[Han]
- Pečení chlebu Šumava[vol]
- Průměrný poměr slámy ku zrnu v jarním období[Wal]
- Statistiky/informace o produkci pšenice[Owe]

1.2 Ověření validity modelu

Validace (viz. [PPc], slajd 37) proběhla v prostředí pekárny kde proběhla konzultace s jejím zaměstnancem a pročtení literatury týkající se řešeného problému (viz. 1.1).

2 Rozbor tématu a použitých metod/technologií

Při modelování (viz. [PPc], slajd 8) výroby mouky, pekárny, prodejny pečiva jsme došli k tomuto závěru.

Výrobna mouky bude rozdělena na dvě větve kde ta první nám produkuje slámu pro farmáře a ta druhá mouku, která se ve vozidle přiveze do pekárny. Po každoroční sklizni se za pomocí mlátičky z celé pšeničné rostliny získá pšeničné zrno a sláma v poměru 1.33[Wal]. Slámu použije každý den farmář k nakrmení dobytka. Pšeničné zrno bude semleto na pšeničnou mouku. Tato mouka bude odvezena autem do pekárny.

Funkce pekárny je rozdělena do třech částí. První část za pomoci pekařů míchače ingrediencí a děliče těsta nejprve ze surových ingrediencí umíchá dávku těsta, poté tato dávka těsta je prohnětena a nakonec je tato dávka rozdělena na kusy odpovídající jednomu chlebu. Druhá část za pomocí pekařů a tvarovacích stolů tyto kusy těsta vytvaruje do podoby chleba a připraví je na pečení. Poslední část za pomoci pekařů a pecí chleby upeče a hotové výrobky doveze na prodejnu.

Prodejna pečiva je rozdělena na zákazníky a prodavače. Po příchodu zákazníka bude zákazníkovi ihned oznámeno, že již není žádné pečivo, pokud tomu tak je tak zákazník ihned odejde. Jinak zákazník počká dokud se mu nezačne prodavač věnovat. Následně si zákazník vybere požadovaný kus pečiva a přejdou společně s prodavačem k placení. Po zaplacení odchází zákazník i s kusem pečiva z prodejny. Pokud ale požadovaný kus pečiva není na skladě, je to zákazníkovi oznámeno a ten z prodejny odchází.

2.1 Popis použitých technologií a jejich původ

- Petriho síť tvořena za pomocí programu draw.io[Ltd]
- OS: Ubuntu[Can]
- IDE: Visual Studio Code[Mic]
  - Rychlé, flexibilní a modulární IDE
- Jazyk: C/C++
- Knihovny:
  - simlib (SIMLIB)[PPb]
    - Jednoduchá simulační (viz. [PPc], slajd 8) knihovna pro C++, která umožňuje objektově orientovaný popis spojitých (viz. [PPc], slajd 224), diskrétních (viz. [PPc], slajd 119), kombinovaných (viz. [PPc], slajd 283) a různých experimentálních modelů (viz. [PPc], slajd 302-307).
  - iostream[Str]
  - getopt[Sou]

3 Koncepce

Cílem tohoto projektu je simulovat tvorbu chleba, který bude následně prodán a taktéž tvorbu mouky, která je pro tvorbu potřeba. Cílem této simulace je zjistit optimální poměr produkce mouky, chleba a jeho následný prodej a to tak, aby ani jeden z těchto elementů nezaostával. Právě proto jsme se rozhodli nemodelovat kažení surovin.

4 Architektura simulačního modelu/simulátoru

4.1 Mapování abstraktního modelu na simulační

Výroba mouky jenž je abstraktně modelována na obrázku 1 je rozdělena do čtyř párů generátor/proces.

První pár generuje nové nezpracované obilí, které dále zpracovává do slámy a pšeničného zrna.

```
class Generator_wheat_plants : public Event
{
    void Behavior()
    {
        wheat_plants += 3045;
        (new WheatProcessing)->Activate();
        Activate(Time + GEN_WHEAT + GEN_WHEAT_WAIT);
    }
};

class WheatProcessing : public Process{
    void Behavior(){
        while(wheat_plants >= 233)
```

```
        {
            wheat_plants -= 233;
            Enter(Wheat_thresher, 1);
            Wait(1 DAY);
            plant_matter += 133;
            wheat_grains += 100;
            Leave(Wheat_thresher);
        }
    }
};
```

Další tři páry fungují na stejném principu a to tak, že generátor periodicky kontroluje zda-li má dostatek surovin k dalšímu provedení procesu. A proces, který vykoná svou práci a bude ji nadále vykonávat dokud má dostatek surovin. U následujícího úseku kódu můžeme vidět, že se podívá jestli je k dispozici dostatek pšeničných zrn a pokud ano, pustí tam proces, který tato zrna semele na mouku.

```
class Generator_wheat_grains : public Event{
    void Behavior(){
        if(wheat_grains >= 50)
        {
            wheat_grains -= 50;
            (new Wheat_grains)->Activate();
        }
        Activate(Time + 1); // check every minute if there are at least 50 kg of grain
    }   // end Behavior
};  // end Generator_wheat_grains

class Wheat_grains : public Process{
    void Behavior(){
        do{
            Enter(Wheat_mill);
            Wait(1 HOUR);

            flour_to_deliver += 50;
            Leave(Wheat_mill);
        }while(wheat_grains >= 50);
    }
};
```

Proces (viz. [PPc], str. 121) vytváření dávek těsta z ingrediencí, jenž je abstraktně modelován (viz. [PPc], str. 8) na obrázku 2 je v simulačním (viz. [PPc], str. 8) modelu (viz. [PPc], str. 7) realizován pomocí třídy typu Process (viz. [PPc], str. 163). Aby mohl být tento proces vytvořen musí být dostatečné množství mouky v zásobárně. O to se stará generátor tohoto procesu, který periodicky kontroluje zda je v zásobárně dostatek mouky na dávku těsta. Výsledkem tohoto procesu jsou kusy těsta, které je dalším procesem potřeba vytvarovat do podoby chleba.

```
class BulkToPiecesGener : public Event{
    void Behavior(){
        while(flourKg > 18)
        {
            flourKg -= 18; //take 18kg of wheat flour
            (new BulkToPieces)->Activate();
        }
        Activate(Time + 1); // periodic check
    }   // end Behavior
};  // end BulkToPiecesGener

class BulkToPieces : public Process{
    void Behavior(){
    .
    .
    for(int i=0; i<60; i++)
    {
        Wait(10 SECOND); // 10s to make in piece of dough for rounding in divider
        piecesOfDoughForRounding++;
    }
    Leave(DoughDivider, 1);
```

Proces tvarování těsta do podoby chlebů, jenž je abstraktně modelován na obrázku 3 je v simulačním modelu realizován pomocí třídy typu Process. Generátor tohoto procesu pracuje s počtem kusů těsta

připravených ke tvarování do podoby chleba a periodicky toto množství kontroluje. Produktem procesu tvarování těsta jsou vytvarované chleby, které jsou dále zpracovány procesem pečení chleba.

```
class PiecesToRoundedPiecesGener : public Event{
    void Behavior(){
        if(piecesOfDoughForRounding > 0)
        {
            (new PiecesToRoundedPieces)->Activate();
        }
        Activate(Time + 1); // periodic check
    }   // end Behavior
};  // end PiecesToRoundedPiecesGener

class PiecesToRoundedPieces : public Process{
    void Behavior(){
        .
        .
        while(piecesOfDoughForRounding > 0)
        {
            piecesOfDoughForRounding--; // take on piece of dough
            Wait(Uniform(30 SECOND, 1 MINUTE)); // 30-60s Dough rounding
            piecesOfDoughForBaking++; // add rounded dough to baking pieces
        }
        .
        .
```

Proces pečení chleba, jenž je abstraktně modelován na obrázku 4 je v simulačním modelu realizován pomocí třídy typu Process. Generátor tohoto procesu pracuje s počtem kusů těsta vytvarovaného do podoby chleba připravených na pečení a periodicky toto množství kontroluje. Produktem procesu pečení chleba jsou hotové chleby umístěné na prodejně v pekárně.

```
class RoundedPiecesToBreadGener : public Event{
    void Behavior(){
        while(piecesOfDoughForBaking >= 20)     // 20 bread to fill one oven
        {
            piecesOfDoughForBaking -= 20; // take 20 rounded bread
            (new RoundedPiecesToBread)->Activate();
        }
        Activate(Time + 1);
    }   // end Behavior
};  // end RoundedPiecesToBreadGener

class RoundedPiecesToBread : public Process{
    void Behavior(){
        .
        .
        double p = Uniform(0, 100);
        if(p <= 44)
        {
            bread_rustic += 20;
        }
        else if(p > 34 && p <= 67)
        {
            bread_roll += 20;
        }
        else
        {
            bread_french += 20;
        }
        bread_counter += 20;
        .
        .
```

Prodejna chleba jenž je abstraktně modelována na obrázku 5 je rozdělena na jeden pár generátor a jeden proces. Generátor v určitém časovém intervalu generuje zákazníky.

Proces tyto nově vygenerované zákazníky zpracovává (e.i. zákazník si koupí chleba, zákazník odchází).

```
if(bread_counter < 1) // if no bread around customer will immediately leave
    return;

Enter(Shopkeeper);
```

```
double p = Uniform(0, 100);
if(p <= 40) //bread rustic
{
    if(bread_rustic < 1) //after customer finds out that no rustic bread is present
    {
        Leave(Shopkeeper);
        return;
    }

    bread_rustic--;
}
else if(p > 40 && p <= 85) //bread roll
{
    if(bread_roll < 1) //after customer finds out that no bread roll is present
    {
        Leave(Shopkeeper);
        return;
    }

    bread_roll--;
}
else //bread french
{
    if(bread_french < 1) //after customer finds out that no french bread is present
    {
        Leave(Shopkeeper);
        return;
    }

    bread_french--;
}

bread_counter--;
Wait(Uniform(1 MINUTE, 3 MINUTE));

Leave(Shopkeeper);
```

5 Podstata simulačních experimentů a jejich průběh

5.1 Postup experimentování

V našich experimentech se snažíme dosáhnout vysokého využití prodavače a nízkého počtu pečiva čekající na prodejně. Každý experiment bude trvat 23 měsíců.

Po nastavení výchozích hodnot jsme zjistili ze využití prodavače je na velmi nízké úrovni. Proto jsme začali experimentovat za pomocí zvyšování akrů, mlátiček, pekařů a tímto způsobem jsme po pár hodinách experimentování došli k přijatelnému výsledku.

5.2 Dokumentace jednotlivých experimentů

`Použití: ./bread [-a N] [-t N] [-m N] [-s N] [-b N] [-o STRING] [-d N]`

- Volitelný argument `-a N`: počet akrů pšeničného pole
- Volitelný argument `-t N`: počet mlátiček
- Volitelný argument `-m N`: počet mlýnů
- Volitelný argument `-s N`: počet prodavačů
- Volitelný argument `-b N`: počet pekařů
- Volitelný argument `-o STRING`: soubor kde se vytisknou statistiky
- Volitelný argument `-d N`: počet měsíců běhu simulace

Experiment1: ./bread -a 10 -t 1 -m 1 -b 1 -d 23 -o exp1.dat

```
+-----------------------------------------------------------+
| STORE Wheat thresher                                      |
+-----------------------------------------------------------+
|  Capacity = 1  (1 used, 0 free)                           |
|  Time interval = 0 - 9.3312e+07                           |
|  Number of Enter operations = 393                         |
|  Minimal used capacity = 0                                |
|  Maximal used capacity = 1                                |
|  Average used capacity = 0.362963                         |
+-----------------------------------------------------------+
+-----------------------------------------------------------+
| STORE Shopkeeper                                          |
+-----------------------------------------------------------+
|  Capacity = 1  (0 used, 1 free)                           |
|  Time interval = 0 - 9.3312e+07                           |
|  Number of Enter operations = 282896                      |
|  Minimal used capacity = 0                                |
|  Maximal used capacity = 1                                |
|  Average used capacity = 0.167235                         |
+-----------------------------------------------------------+
```

Experiment2: ./bread -a 20 -t 1 -m 1 -b 1 -d 23 -o exp2.dat

```
+-----------------------------------------------------------+
| STORE Wheat thresher                                      |
+-----------------------------------------------------------+
|  Capacity = 1  (0 used, 1 free)                           |
|  Time interval = 0 - 5.9616e+07                           |
|  Number of Enter operations = 522                         |
|  Minimal used capacity = 0                                |
|  Maximal used capacity = 1                                |
|  Average used capacity = 0.756522                         |
+-----------------------------------------------------------+
+-----------------------------------------------------------+
| STORE Shopkeeper                                          |
+-----------------------------------------------------------+
|  Capacity = 1  (0 used, 1 free)                           |
|  Time interval = 0 - 5.9616e+07                           |
|  Number of Enter operations = 380508                      |
|  Minimal used capacity = 0                                |
|  Maximal used capacity = 1                                |
|  Average used capacity = 0.348957                         |
+-----------------------------------------------------------+
```

```
Experiment3: ./bread -a 40 -t 2 -m 1 -b 2 -d 23 -o exp3.dat
+-----------------------------------------------------------+
| STORE Wheat thresher                                      |
+-----------------------------------------------------------+
|  Capacity = 2  (2 used, 0 free)                           |
|  Time interval = 0 - 5.9616e+07                           |
|  Number of Enter operations = 1382                        |
|  Minimal used capacity = 1                                |
|  Maximal used capacity = 2                                |
|  Average used capacity = 2                                |
+-----------------------------------------------------------+
+-----------------------------------------------------------+
| STORE Shopkeeper                                          |
+-----------------------------------------------------------+
|  Capacity = 1  (1 used, 0 free)                           |
|  Time interval = 0 - 5.9616e+07                           |
|  Number of Enter operations = 493108                      |
|  Minimal used capacity = 0                                |
|  Maximal used capacity = 1                                |
|  Average used capacity = 0.756731                         |
+-----------------------------------------------------------+
```

```
Experiment4: ./bread -a 80 -t 3 -m 1 -b 3 -d 23 -o exp4.dat
+-----------------------------------------------------------+
| STORE Wheat thresher                                      |
+-----------------------------------------------------------+
|  Capacity = 3  (3 used, 0 free)                           |
|  Time interval = 0 - 5.9616e+07                           |
|  Number of Enter operations = 2038                        |
|  Minimal used capacity = 0                                |
|  Maximal used capacity = 3                                |
|  Average used capacity = 2.94928                          |
+-----------------------------------------------------------+
+-----------------------------------------------------------+
| STORE Shopkeeper                                          |
+-----------------------------------------------------------+
|  Capacity = 1  (1 used, 0 free)                           |
|  Time interval = 0 - 5.9616e+07                           |
|  Number of Enter operations = 492442                      |
|  Minimal used capacity = 0                                |
|  Maximal used capacity = 1                                |
|  Average used capacity = 0.853792                         |
+-----------------------------------------------------------+
```

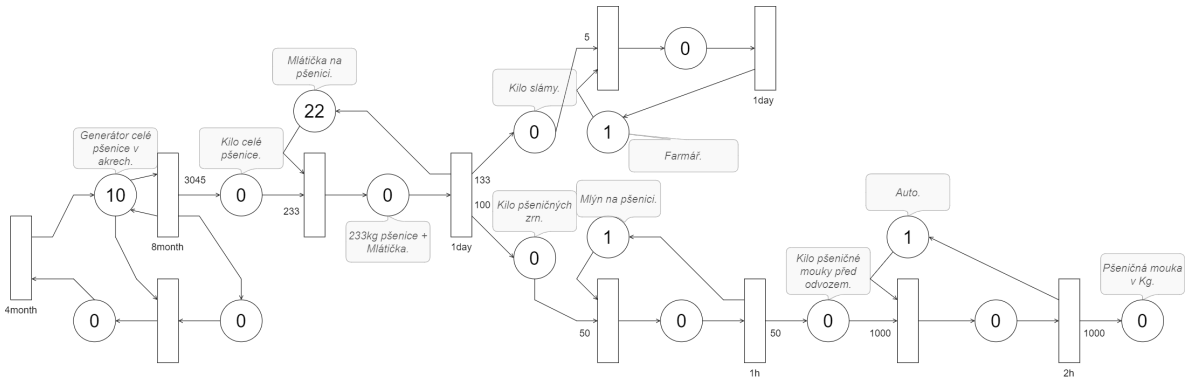
5.3 Závěry experimentů

Bylo provedeno mnoho experimentů, pomocí kterých se nám povedlo odstranit několik chyb v modelu, například bylo vyřešeno vznikání negativního počtu kg mouky.

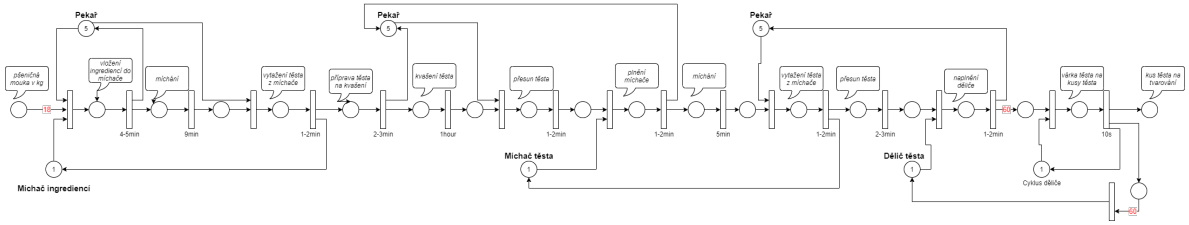
6 Shrnutí simulačních experimentů a závěr

Díky provedeným experimentům můžeme nyní konstatovat, že za pomocí navrhnutému modelu jsme byli schopni značně zlepšit efektivitu naší pekárny.

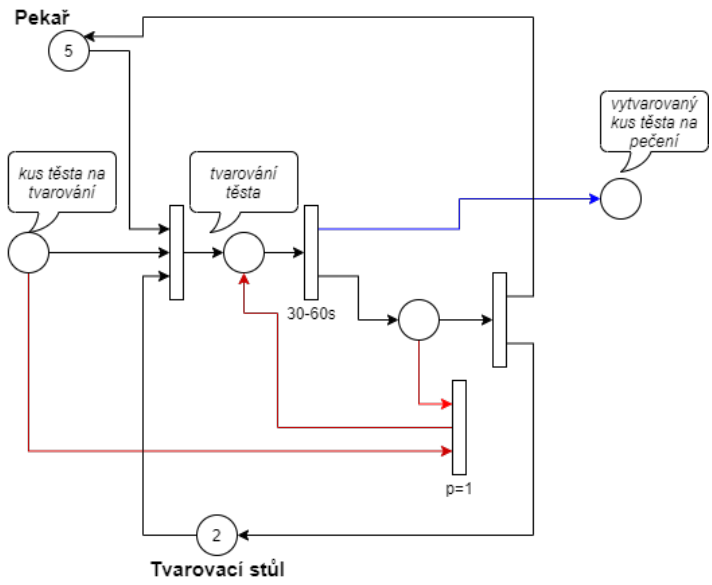
Přílohy



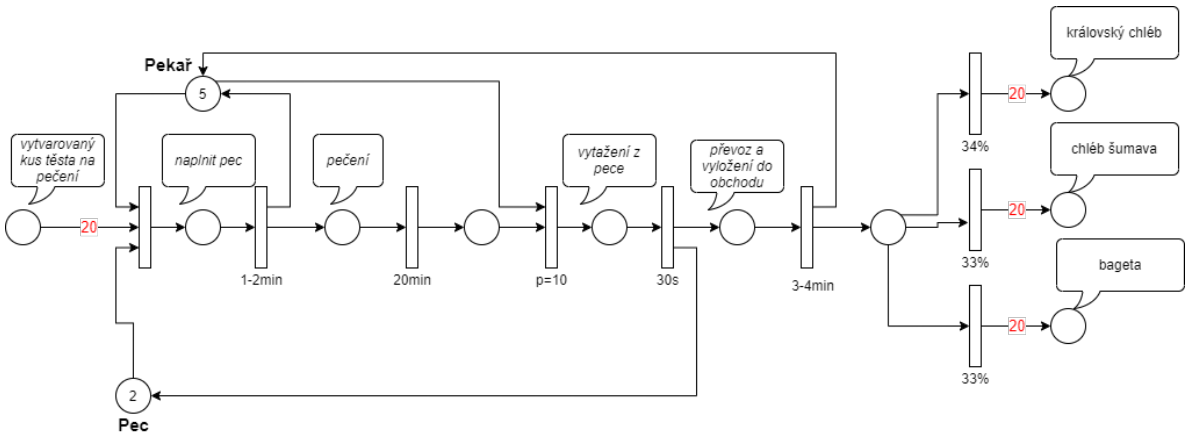
Obrázek 1: Výroba mouky



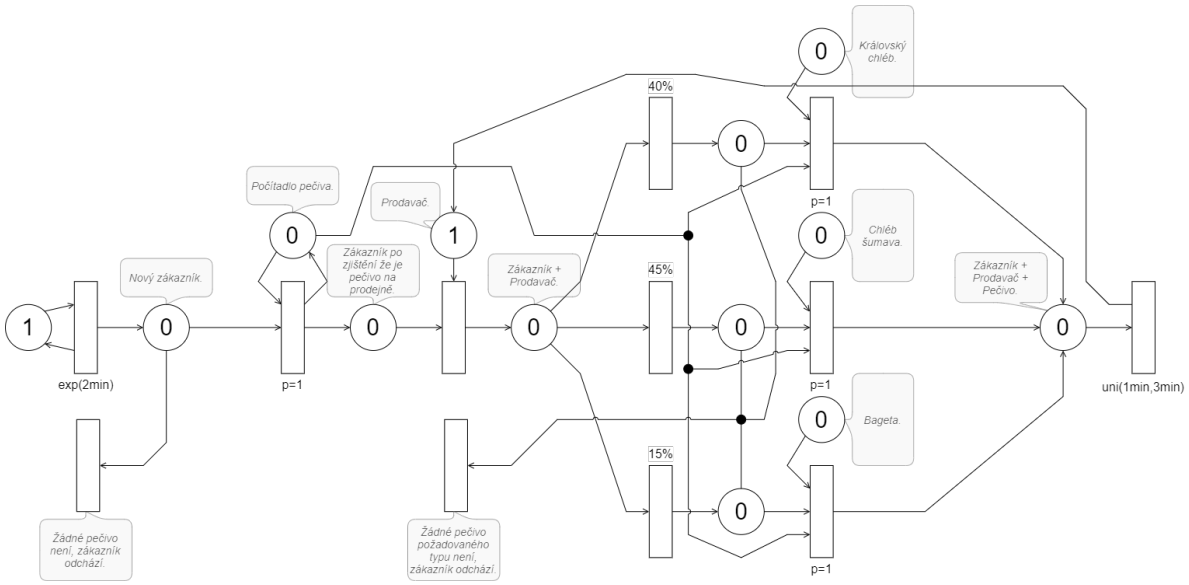
Obrázek 2: Proces přetvoření ingrediencí na kusy těsta



Obrázek 3: Proces tvarování kusů těsta



Obrázek 4: Proces pečení chleba



Obrázek 5: Prodejna

Reference

[Can] Canonical. Enterprise open source and linux — ubuntu. https://ubuntu.com/. [Online, last visited on 12.12.2021].

[Evo] Joanna Evoniuk. Bread processing also known as bread manufacture process. https://bakerpedia.com/processes/bread-processing/. [Online, last visited on 12.12.2021].

[Han] Olga Hančarová. Jak upéct chleba. https://www.skrblik.cz/navod/jak-upect-chleba/. [Online, last visited on 12.12.2021].

[Ltd] JGraph Ltd. draw.io – diagrams for confluence and jira - draw.io. https://drawio-app.com/. [Online, last visited on 12.12.2021].

[Mic] Microsoft. Visual studio code - code editing. redefined. https://code.visualstudio.com/. [Online, last visited on 12.12.2021].

[Owe] Julie Owens. Wheat facts. https://nationalfestivalofbreads.com/nutrition-education/wheat-facts. [Online, last visited on 12.12.2021].

[PPa] Dr. Ing. Peringer Petr. Témata projektů ims 2021/22 - postcovidová doba. http://perchta.fit.vutbr.cz/vyuka-ims/53. [Online, last visited on 12.12.2021].

[PPb] Dr. Ing.; David Leska; David Martinek Peringer Petr. Simulation library for c++. http://www.fit.vutbr.cz/~peringer/SIMLIB/. [Online, last visited on 12.12.2021].

[PPc] Ph.D. Peringer Petr, Dr. Ing.; Ing. Martin Hrubý. Modelování a simulace. http://www.fit.vutbr.cz/study/courses/IMS/public/prednasky/IMS.pdf. [Online, last visited on 12.12.2021].

[Sou] Open Source. Getopt (the gnu c library). https://www.gnu.org/software/libc/manual/html_node/Getopt.html. [Online, last visited on 12.12.2021].

[Str] Bjarne Stroustrup. ¡iostream¿ - c++ reference. https://www.cplusplus.com/reference/iostream/. [Online, last visited on 12.12.2021].

[vol] Babiččina volba. Chléb Šumava. https://www.babiccinavolba.cz/recepty/chleb-sumava/#null. [Online, last visited on 12.12.2021].

[Wal] Richard Engel; Dan Long; Gregg Carlson; Rosie Wallander. Estimating straw production of spring and winter wheat. https://landresources.montana.edu/fertilizerfacts/documents/FF33StrawSWWW.pdf. [Online, last visited on 12.12.2021].