

# WP 1.F Národní monitoring oběhového hospodářství

Souhrnná výzkumná zpráva

Datová základna pro komplexní modelování stavu  
odpadového a oběhového hospodářství (OH a ObH)  
v ČR. (V1.F.1.1)

Manuál pro tvorbu PD ISOH2 (V63)

Kolektiv autorů:

Ing. Radovan Šomplák, Ph.D. (VUT)  
doc. Ing. Martin Pavlas, Ph.D. (VUT)  
Ing. Jaroslav Pluskal (VUT)  
Ing. Jiří Valta (CENIA)  
Ing. Gabriela Bulková, MBA (MŽP)

Řešitelské pracoviště:

Ústav procesního inženýrství, VUT v Brně,  
CENIA, česká informační agentura životního prostředí,  
Ministerstvo životního prostředí (MŽP)

Srpen 2023

T A  
Č R

Projekt SS02030008 Centrum environmentálního výzkumu: Odpadové a oběhové hospodářství a environmentální bezpečnost (CEVOOH) je financován se státní podporou Technologické agentury ČR a Ministerstva životního prostředí ČR v rámci Programu Prostředí pro život.

## Obsah

1	Úvod .....	3
2	Tvorba PD ISOH .....	4
2.1	Návrh struktury dat .....	4
2.2	Podlimitní producenti odpadu a „hříšníci“ .....	5
2.3	Zapojení producentů do obecního systému sběru komunálního odpadu .....	6
2.4	Kaly .....	7
2.4.1	Výsledek výpočtu .....	8
2.4.2	Pre-processing dat před výpočtem kalů.....	8
2.4.3	Výpočet sušiny.....	10
2.4.4	Popis kódu.....	11
2.5	Autovraky a baterie.....	15
2.6	Vznik odpadu mimo provozovnu.....	16
2.7	Katalogová čísla dle nové legislativy .....	17
3	Komunikace PD ISOH a výpočtového jádra .....	17

## 1 Úvod

Data o produkci a nakládání s odpady jsou ukládány do databáze Informačního systému odpadového hospodářství (ISOH). Pro následné analýzy dat je z ISOH vytvářena tzv. pracovní databáze (PD ISOH). Ta obsahuje určité přepočty, úpravy surových dat a přiřazení příznaků s ohledem na následné použití. Předkládaný dokument popisuje jednotlivé kroky tvorby PD ISOH.

Motivace pro aktualizaci postupu pro tvorbu PD ISOH v roce 2023 je vývoj nové verze ISOH, která reaguje na změny legislativy v odpadovém hospodářství a potřeby Ministerstva životního prostředí (MŽP).

PD ISOH obsahuje pouze změny, které jsou nezbytné pro následnou práci s daty, analýzy a výpočty indikátorů. Nezbytné úpravy jsou definovány tak, že pro další analýzy a výpočty nebude třeba čerpat ze surových dat z ISOH (až na výjimky zejména pro kontrolní činnost MŽP). Zároveň se v rámci tvorby PD ISOH neprovádějí žádné další přípravné úpravy dat. Cílem je mít data v PD ISOH v co nejvíce odpovídající podobě vykazovaným datům do ISOH.

Převážná většina dat z ISOH se pouze překlopí do PD ISOH bez jakéhokoli zásahu. Prováděné úpravy a doplnění dat, které s tvorbou PD ISOH souvisí, jsou popsány v **kap. 2**. Důležitým aspektem při tvorbě PD ISOH bylo zahrnutí legislativních změn, které přinesl Zákon o odpadech č. 541/2020 Sb. a Vyhláška č. 273/2021 Sb., o podrobnostech nakládání s odpady. S ohledem na návaznost jednotlivých kroků uvedených v **kap. 2** je nutné dodržovat stanovenou posloupnost úprav popsanou níže.

Posloupnost kroku při tvorbě PD ISOH:

- Zavedení geografické struktury z **kap. 2.1**
- Identifikujeme v původních hlášeních, kde je potřeba aplikovat transformace
  - Zapojení producentů do obecního systému sběru komunálního odpadu, viz **kap. 2.3**
  - Autovraky a baterie, viz **kap. 2.5**
  - Vznik odpadu mimo provozovnu, viz **kap. 2.6**
- Aplikace třech uvedených transformací
- Vygenerování umělých hlášení pro podlimitní a hříšnický, viz. **kap. 2.2**
- Výpočet sušiny u kalů, viz. **kap. 2.4**

Z důvodu komunikace PD ISOH s externími výpočetními moduly (viz **kap. 2.4** – výpočet sušiny u nakládání s kaly) byl v **kap. 3** navržen rámec této komunikace, který může být využíván i v případě implementace nových výpočtových modulů.

## 2 Tvorba PD ISOH

### 2.1 Návrh struktury dat

Databáze PD ISOH je navržena ve stejném územním detailu jako jsou data v původní databázi ISOH, tzn., že data jsou ukládána v PD ISOH na úrovni evidentů (provozoven – IČP, zařízení – IČZ a obchodníků – IČOB).

Oproti ISOH budou v PD ISOH navíc přidány následující údaje:

- Geografická data
  - Číselník základních územních jednotek (**ZÚJ**) **bude doplněn o 206 fiktivních jednotek**. Bude přidáno jedno fiktivní identifikační číslo na úrovni ZÚJ v každé obci s rozšířenou působností (ORP), celkem 205 jednotek. Dále bude vytvořena fiktivní jednotka pro hl. m. Prahu. Na vytvořené fiktivní jednotky bude registrován odpad vzniklý mimo provozovnu, viz **kap. 2.6**.
  - Ke každé fiktivní jednotce (viz bod výše) bude doplněn jeden fiktivní název pro ZÚJ v každé ORP a hl. m. Prahu.
  - Ke každému záznamu v PD ISOH bude přidáno číslo obce (podle Českého statistického úřadu (ČSÚ) uvedeno v Číselníku obcí – CISOB). Stejně jako běžnému označení ZÚJ bude i ke každému fiktivnímu ZÚJ (viz body výše) vytvořeno fiktivní číslo obce. To neplatí pouze pro hl. m. Prahu.
  - Ke každému číslu obce (včetně fiktivních) bude přidán název obce.
  - Bude doplněno jedno fiktivní ORP pro Prahu, na kterém bude registrován odpad vzniklý mimo provozovnu v Praze (mobilní odpad bude evidován jak bez rozlišení ZÚJ, tak bez rozlišení konkrétního obvodu hl. m. Prahy, viz **kap. 2.6**).
- Data z odpadového hospodářství
  - Hodnota sušiny a zvodnělý stav pro kaly (kat. č. **02 02 04, 02 03 05, 02 04 03, 02 05 02, 02 06 03, 02 07 05, 03 03 11 a 19 08 05**), viz **kap. 2.4**. Obě hodnoty budou u jednoho záznamu v samostatných sloupcích. Je vhodné zdůraznit, že nesmí dojít během zpracování dat k započítání kalu v sušině i zvodnělém stavu najednou.
  - Informace o způsobu získání dat dané **příznakem**. Bude možné rozpoznat hodnoty, které byly ohlášeny evidentem (běžné hlášení), vygenerovány systémem (např. podlimitní evidenti – **kap. 2.2**) nebo dopočítány (libovolné pomocné hodnoty, viz sušina v kalech – **kap. 2.4**). Konkrétní podoba příznaků se bude doplňovat a měnit na základě potřeb exportů a analýz dat a požadavků MŽP.

- Ostatní data
  - Počet obyvatel pro ZÚJ, obec, ORP, kraj a ČR (poskytuje ČSÚ); bude staženo z veřejně přístupných databází.
  - NACE (poskytuje ČSÚ) – pravidelné roční předání datové sady z ČSÚ na MŽP.

Pro většinu analýz prováděných uživatelem PD ISOH (tzn. CENIA a MŽP) detail ZÚJ není nutný a stačí **detail obce, tj. je nutná agregace ZÚJ v případech, kdy jedné obci náleží více ZÚJ (typicky města)**. Použití ZÚJ má za následek falešný dojem větší přesnosti, což reálně často není pravda a spíše dochází ke špatným závěrům při zpracování takových dat. Příkladem může být hl. m. Praha, kde se nachází 57 ZÚJ. Přitom např. pro SKO (kat. č. 20 03 01) je dominantní podíl produkce vykazován pouze v jednom ZÚJ (Praha 1 má téměř 99 % produkce za rok 2021). Většina požadavků na zobrazení a export dat bude směřovat na územní celek obec a ČR, proto je v novém PD ISOH informace „**obec**“ přidána.

## 2.2 Podlimitní producenti odpadu a „hříšníci“

Subjekty, které produkují nebo nakládají s relativně malým množstvím odpadu, nemusí podávat Hlášení souhrnných údajů z průběžné evidence produkce a nakládání s odpady (viz Zákon o odpadech č. 541/2020 Sb.). V takovém případě tito tzv. „**podlimitní**“ výše zmíněné hlášení nepodávají a vyskytují se pouze jako *partner* v listu 2 přílohy č. 13 vyhlášky č. 273/2021 Sb. (dříve listu 2 přílohy č. 20 vyhlášky č. 383/2001 Sb.). Druhým typem producentů, kteří nepodávají hlášení jsou tzv. „**hříšníci**“. To jsou evidenti, kteří mají vyšší produkci, než je stanovený limit v zákoně č. 541/2020 Sb., přesto ale hlášení nepodali. V listu 2 jsou uvedeny u partnera základní informace o producentovi (typ producenta, adresa atd.), takže lze vytvořit **automaticky nové hlášení** za tyto dva typy evidentů.

Dále v textu budou použity pojmy „**stará legislativa**“ (kódy dle vyhlášky č. 383/2001 Sb.) a „**nová legislativa**“ (kódy dle zákona č. 541/2021 Sb. a vyhlášky č. 273/2021 Sb.).

Výše zmíněné dva typy evidentů budou identifikovány tak, že se projdou všechny záznamy a pokud je původce evidován jako partner u převzetí odpadu pod kódem nakládání „**B00**“ (stejně v případě staré i **nové legislativy**) a zároveň sám nepodal v příslušném roce hlášení (myšleno za subjekt mající IČO), jedná se o podlimitního producenta odpadu nebo hříšníka.

Dle **nové legislativy** se analogicky bude postupovat také u odpadu z léčiva od občanů. Tento odpad převzatý zařízením pro nakládání s odpady od lékární je evidován pod kódem nakládání „**B22**“ (platí pouze pro **novou legislativu**).

Nebudou zde řešeny situace, kdy chybí údaj u předání odpadu, stará legislativa: „**XN2, XN3, XN8, XN10**“; **nová legislativa**: „**XN3, XN33**“, v případech vykázaného převzetí odpadu „**B00**“ (resp. „**B22**“), jestliže evident s chybějícím předáním odpadu podal hlášení.

Tato situace nastává poměrně velmi často a doplněním tohoto odpadu by se velmi pravděpodobně odpad uvedl duplicitně. Vinou nesrovnalostí v datech mohou být evidence předání odpadu na chybnou provozovnu atp.

V případě dat o odpadech pro **nově vytvořené evidenty** (podlimitní producenty odpadu a hřištníky) se doplní pro každý takto nalezený záznam produkce pod kódem nakládání „**A00**“ (stejně v případě staré i **nové legislativy**) o stejné hmotnosti jaká byla vykázána pod kódem nakládání „**B00**“, také se doplní „**A20**“ v případě nalezených podlimitních předávající léčiva převzatých pod kódem „**B22**“, viz výše. Dále je nutné doplnit záznam na minusové straně o předání odpadu „**AN3**“ (stejně v případě staré i **nové legislativy**) o stejné hmotnosti a jako partnera uvést původního evidenta z listu 2.

U každého nově vytvořeného evidenta (podlimitního producent odpadu nebo hřištník) je doplněno takové IČP a ZÚJ, jaké je u něj uvedeno v hlášení, kde je uveden jako partner. Pokud je podlimitní producent nebo hřištník uveden v různých hlášeních jako partner s několika různými IČP a/nebo ZÚJ, je se stejnými (různými) IČP a/nebo ZÚJ uveden v doplněných datech (systémově vytvořených záznamech) jako evident.

V případě, že nově vytvořený evident je v původních hlášeních uveden jako partner s několika různými variantami neklíčových atributů (typicky se může lišit adresa, bývá uvedena např. s/bez orientačního čísla apod.), pak se v doplněných datech (systémově vytvořených záznamech) použije nejčastější uvedená varianta, kde figuruje jako partnera v původních hlášeních.

U systémově vytvořených záznamů je nutné tuto skutečnost rozlišit příznakem viz **kap. 2.1.**

Situace, kdy by nepodal hlášení subjekt, který má IČZ nebo IČOB se zde neošetřují, protože budou vyřešeny již v předešlých kontrolách před tvorbou PD ISOH.

Tento bod lze zpracovat až když budou seskupena úplná data od všech evidentů.

### **2.3 Zapojení producentů do obecního systému sběru komunálního odpadu**

Producent (evident: právnická osoba, fyzická osoba podnikající – firma, živnostník) má obvykle možnost smluvního zapojení do obecního systému nakládání s komunálním odpadem. V takovém případě musí tuto skutečnost evident uvést do ročního Hlášení souhrnných údajů z průběžné evidence, viz list 1 z přílohy č. 13 vyhlášky č. 273/2021 Sb. (dříve list 1 přílohy č. 20 vyhlášky č. 383/2001 Sb.). V rámci listu 1 evident označí všechny druhy komunálního odpadu (odpady skupiny 20), které se smluvní dohody týkají. V takovém případě je nutné tyto odpady uvedené na listu 2, které jsou zahrnuty do hlášení příslušné obce, nezapočítat do PD ISOH podruhé u tohoto evidenta, jelikož jsou uvedeny u dané obce. Proto se tyto odpady (záznamy) nepromítnou u producentů (evidentů) do PD ISOH. To platí pro všechny evidenční kódy nakládání.

Výjimku představují chyby, kdy producent (evident) sice uvede, že s určitým katalogovým číslem je zapojen do obecního systému nakládání s komunálním odpadem, zároveň ale u tohoto katalogového čísla vykáže předání odpadu partnerovi (pod kódem „AN3“), který rovněž vykáže jeho převzetí (pod kódem „B00“). V takových případech evident patrně není zapojen do obecního systému nakládání s komunálním odpadem a údaje je nutné překlopit do PD ISOH v nezměněné formě. Nekontroluje se soulad IČP ani množství odpadu, postačuje IČO subjektu. Tyto chyby by měly být vyřešeny již během kontrol na formuláři.

Platí stejně pro starou i **novou legislativu**. Tento bod nutno řešit postupně pro každého evidenta zvlášť.

Tento bod lze zpracovat až když budou seskupena úplná data od všech evidentů.

## **2.4 Kaly**

Základní princip v této části je, že na plusové straně producent zadává množství vyprodukovaného kalu ve zvodnělém stavu v listu 2 (příloha č. 13 vyhlášky č. 273/2021 Sb.) a údaj o sušině v listu 3. PD ISOH pak pracoval s hodnotou v sušině, která byla při tvorbě záznamu vypočítána. V případě všech ostatních kódů nakládání (kromě produkce odpadu) se u dosavadní verze PD ISOH (do roku 2021) množství kalu v sušině přepočítávalo celorepublikovým průměrem. V kontextu terminologie používané níže, se tedy průměrná sušina počítala pro uzel „ČR“. Takové zjednodušení zanáší do PD ISOH velké chyby a je vhodnější průměrovat data více regionálně, resp. v co největším možném detailu, což by mělo odpovídat jednotlivým evidentům, provozovnám nebo zařízením (obecně bude v textu využíván termín „uzel“). Návrh postupu výpočtu je uveden v následujících kapitolách.

Hlavním problémem pro výpočet sušiny je, že všechny odpady se po prvním předání a jeho následném převzetí vykazují pod kódem nakládání „B00“ a další sledování celého řetězce od producenta je zašuměno dalšími odpady od jiných původců. Tento problém lze vyřešit za předpokladu, že veškerý evidovaný kal vstupující do uzlu bude ideálně promíchán a až následně s ním bude dále nakládáno.

Hmotnost v sušině se tedy vypočítá jako vážený součet produkce dle podílu sušiny ve zvodnělých kalech a vstupních proudů (převzetí) do řešeného uzlu (uzlem může být myšlen libovolný územní celek – ZÚJ, obec, ORP, kraj, ČR, ale také vykazující subjekty – IČO, IČP nebo IČZ).

**Pro účely určení sušiny pro všechny záznamy v PD ISOH se bude pracovat s maximálním detailem v evidenci a za uzel se bude dále považovat IČO, IČP nebo IČZ.**

Jedná se o stejný přístup jako doposud s tím rozdílem, že je problematika řešena ve větším detailu a průměr je vypočítán v každém uzlu. Takto jsou v konkrétní lokalitě do výsledné



sušiny započítány pouze ty kaly, které mají návaznost na daný uzel a je tím eliminováno ovlivnění odpadovými proudy z jiné části ČR. Ve výsledku tak lze každý uzel rozdělit na dvě části (evidence plus a minus), přičemž se předpokládá, že podíl sušiny je u obou částí stejný (splnění hmotnostní bilance  $VSTUP = VÝSTUP$ ). Výsledný matematický model pak představuje systém o nula stupních volnosti (pouze jedno řešení) obsahující pouze bilanční rovnici pro množství (podíl) sušiny a výsledek je dán řešením soustavy lineárních rovnic.

**Výpočet sušiny se provede pro katalogová čísla odpadů 02 02 04, 02 03 05, 02 04 03, 02 05 02, 02 06 03, 02 07 05, 03 03 11 a 19 08 05 (pro každé katalogové číslo samostatně).**

Tento bod lze zpracovat až když budou seskupena úplná data od všech evidentů.

#### **2.4.1 Výsledek výpočtu**

Oproti záznamům v ISOH se v PD ISOH za každý existující záznam u kalů vytvoří nové sloupce, aby bylo možné do databáze uložit údaje o hmotnosti jak zvodnělého kalu, tak i sušiny.

Samotný výpočet sušiny je doprovázen pomocnými parametry, které se definují ve výpočtovém jádru a nijak neovlivňují PD ISOH. Výpočtové jádro si pouze načte potřebná data a zároveň doplní ke každému řádku databáze údaj o sušině. Výpočetní modul nijak nebude upravovat původní data, tj. nebudou modifikovány údaje, odstraněny nebo přidány záznamy atp. Pouze k záznamům, kde bude chybět údaj o sušině z evidence dat, se příslušné dopočítané hodnoty doplní. Komunikace PD ISOH a výpočetního jádra je definována v **kap. 3**.

#### **2.4.2 Pre-processing dat před výpočtem kalů**

Z archivní databáze ISOH je nutné vybrat veškeré záznamy týkající se odpadu evidovaného pod konkrétním katalogovým číslem (např. 19 08 05 „Kaly z čistíren odpadních vod“). Pro výpočet bude nutné získat o každém záznamu následující informace:

- Evident
- Partner
- Kód nakládání
- Množství odpadu + (t)
- Množství odpadu – (t)
- Sušina (t)

Evident i partner záznamu musí představovat unikátní dvojici vybraných uzlů. V případě zařízení je za uzel považována unikátní kombinace IČO+IČZ, v případě provozoven je za uzel považována unikátní kombinace IČO+ZÚJ (IČP partnerů je v hlášeních zatíženo velmi



vysokou nekvalitou dat, proto IČP nelze použít jako spolehlivou součást klíče pro rozlišení provozovny partnera), pro který bude následně vypočítán podíl sušiny.

Kód nakládání slouží ke správnému zařazení záznamu do vybraných skupin vystupujících v matematickém modelu. Množství odpadu charakterizuje evidované hodnoty zvodnělého kalu, rozdělené do kategorie plus (nenulová hodnota „Množství odpadu + (t)“) či mínus (nenulová hodnota „Množství odpadu – (t)“) dle povahy uvedeného kódu nakládání. Hmotnost sušiny se u plusových kódů nakládání vypočítá podle charakteru příslušného kódu, přičemž přesný postup je vysvětlen níže v textu. U záporných kódů nakládání je množství sušiny předmětem výpočtu a v pre-processingu tak není řešeno.

Dále je nutné evidované záznamy rozdělit do několika skupin podle kódů nakládání, jelikož příprava dat a implementace jednotlivých skupin do výpočtu může mít svá specifika. Zde je nutno podotknout, že v budoucnu se může výčet kódů lišit s ohledem na změny v legislativě. Zde jsou uvedeny jak evidenční kódy nakládání dle staré legislativy, tak kódy dle **nové legislativy**. Pro matematický model jsou v Tab. 1 **obecně uvedeny všechny existující evidenční kódy daného charakteru nakládání**.

Tab. 1: Evidenční kódy nakládání podle jejich charakteru

Použité agregace nakládání pro pre-processing kalů (charakter nakládání)	Stará legislativa	Nová legislativa
Kumulace (+)	A00, BN30, AN50, BN50, AN60, BN60	A00, A10, A20, A30, A40, A60, AN50, BN50
Vyskladnění (+)	C00, CN50, CN60	C00, CN50
Import (+)	BN6, BN16	BN6, BN16
Sekundární produkce (+)	BN40	BN40, BN41, BN44
Předání (-)	XN2, XN3, XN8, XN10	XN3, XN33
Převzetí (+)	B00	B00, B22
Zpracování (-)	Ostatní kódy (tj. mínusové kódy kromě předání)	Ostatní kódy (tj. mínusové kódy kromě předání)

První skupina **kumulace** je klíčová, jelikož ohlášený podíl sušiny ve zvodnělých kalech je navázán právě na tuto skupinu. Hmotnost sušiny se získá vynásobením zvodnělého množství hodnotou sušiny z listu 3 přílohy č. 13 vyhlášky č. 273/2021 Sb. (dříve listu 3 přílohy č. 20 vyhlášky č. 383/2001 Sb.), a vydělí se 100. V případě, že není v listu pro daný subjekt uveden podíl sušiny, je pole ponecháno prázdné. Z dostupných údajů (záznamy u subjektů, kde byl ohlášen podíl sušiny) skupiny kumulace se následně vypočítá **celorepublikový průměr podílu sušiny v kalech**, tj. množství celkové sušiny je vyděleno celkovým množstvím vyprodukovaného zvodnělého kalu.

U ostatní plusových záznamů (**vyjma převzetí „B00“ a „B22“**) je opět využit **evidovaný údaj o podílu sušiny v uzlu**. Pokud není uveden, využije se vypočtený **celorepublikový průměr**. Rozdílnost tedy spočívá v tom, že skupiny mimo **kumulaci** neovlivňují průměr za ČR.

Pro správné fungování matematického modelu je nutné spárovat skupinu předání a převzetí, tj. spojit dvojici záznamů, jenž mají zaměněná pole „Evident“ a „Partner“. Tyto dvojice jsou však často v nesouladu (liší se evidovaným množstvím) a před samotným výpočtem je nutné jednoznačně určit stejnou hmotnost pro evidentu i partnera. Při výpočtu je nesoulad odstraněn s využitím průměru mezi evidovanými hodnotami, jenž je popsán následující rovnicí:

$$X_{ij} = \frac{X_{i,j}^+ + X_{i,j}^-}{2}, \quad \text{pro } \forall i, j \in I,$$

kde, symbol  $X_{i,j}^+$  představuje evidované množství převzatého odpadu (kód „**B00**“), zatímco symbol  $X_{i,j}^-$  evidované množství předaného odpadu (součet kódů stará legislativa: „**XN2, XN3, XN8 a XN10**“; **nová legislativa: „XN3, XN33**“) mezi dvěma libovolnými uzly.

### 2.4.3 Výpočet sušiny

Matematický model pro výpočet sušiny je koncipován jako lineární optimalizační model bez účelové funkce, jelikož modelovaný systém má nula stupňů volnosti (tj. existuje pouze jedno řešení). Samotný výpočet tak lze označit spíše za soustavu lineárních rovnic, jenž lze řešit různými přístupy. Do výpočtu vstupují pouze ty uzly, u kterých je evidováno předání odpadu nebo jeho zpracování (viz dále v textu). Výčet využitých symbolů pro sestavení příslušných rovnic je následující:

#### Množiny

$i, j \in I$  Množina uzlů (IČO, IČP, IČZ)

#### Parametry

$O_i$  Množství sušiny získané sekundární produkcí v uzlu  $i$  [t]  
 $P_i$  Množství sušiny získané kumulací v uzlu  $i$  [t]  
 $S_i$  Množství sušiny získané vyskladněním zásob v uzlu  $i$  [t]  
 $X_{i,j}$  Množství zvodnělého kalu transportovaného na hraně mezi uzlem  $i$  (odesílatel/předání) a uzlem  $j$  (příjemce/převzetí) [t]  
 $Z_i$  Množství zpracovaného kalu ve zvodnělé formě v uzlu  $i$  [t]

#### Proměnné

$w_i$  Průměrný podíl sušiny ve zvodnělém kalu v uzlu  $i$  [-]

Pro každý uzel v systému je definována následující rovnice popisující hmotnostní bilanci sušiny.

$$\sum_{j \in I} (w_j X_{j,i}) + P_i + S_i + M_i + O_i = w_i \left( \sum_{j \in I} X_{i,j} + Z_i \right), \quad \text{pro } \forall i \in I,$$

**Levá strana rovnice představuje vstupy do uzlu (plusové kódy nakládání)**, postupně se jedná o převzaté množství sušiny od jiných subjektů v systému  $\sum_{j \in I} (w_j X_{j,i})$ , kumulace sušiny  $P_i$ , vyskladněná sušina  $S_i$ , importovaná sušina  $M_i$  a následně sušina získaná sekundární produkcí po úpravě  $O_i$ . **Pravá strana rovnice reprezentuje výstupy**, jež představují předané  $\sum_{j \in I} X_{i,j}$  a zpracované  $Z_i$  množství kalů. Parametry na pravé straně rovnice vystupují ve formě zvodnělého kalu, jež je následně vynásobeno hledaným podílem sušiny  $w_i$  tak, aby byla **splněna rovnost VSTUP = VÝSTUP**.

Výjimku zde tvoří ty subjekty, které neevidují žádné předání odpadu ani zpracování. Pokud je evidován tok na takový subjekt, bilanční rovnice pro tento uzel není řešitelná, jelikož neexistuje žádný výstup, který by vyvážil množství sušiny na vstupu. Pro takové uzly není tedy původní bilanční rovnice definována a je podíl sušiny je nastaven na nulu.

Hledané proměnné  $w_i$  představují poměr sušiny pro všechny kódy nakládání v systému kromě plusových kódů nakládání. Poměr sušiny pro některé uzly může být i mimo očekávaný rozsah (mezi 0 a 1). Jedná se však o důsledek hmotnostních nesouladů v systému, přičemž klíčové je monitorování absolutního množství zpracované sušiny. Podíly sušiny mimo daný rozsah tak představují určitou míru kompromisu, mohou identifikovat chybné evidenty a zejména pomáhají eliminovat vliv chyb v evidenci. Pro získání absolutního množství zpracované sušiny v uzlu je tedy nutné vynásobit hodnoty zpracování zvodnělého kalu a získané podíly sušiny z výpočetního modulu. Je třeba poznamenat, že výsledek bilance udává vážený průměr všech toků u každého uzlu, tzn., že není zohledněna lokalita a způsob nakládání pro konkrétní kaly.

**Stejným způsobem se provede přepočítání pro katalogová čísla odpadů 02 02 04, 02 03 05, 02 04 03, 02 05 02, 02 06 03, 02 07 05, 03 03 11 a 19 08 05 (pro každé katalogové číslo samostatně).**

#### 2.4.4 Popis kódu

V následující kapitole bude podrobně popsána implementace výpočtu v jazyce „Python“ včetně pre-procesingu a samotné optimalizace. Zde je nutné podotknout, že kód využívá **rozhraní „Excel – Python – Excel“** a případné napojení na databázi ISOH musí být řešeno již samotným správcem databáze. Konkrétní komunikaci PD ISOH a výpočetního jádra je věnována samostatná **kap. 3**. Výpočetní modul pro výpočet sušiny v kalech byl vyvíjen s využitím následujících verzí Pythonu a knihoven:

- Python: 3.10.6
- Pandas: 1.4.3
- PuLP: 2.7.0
- Pathlib: Součást základní verze Pythonu
- Re: Součást základní verze Pythonu

## Pre-processing dat

Nejprve jsou z databáze (v testovacím případě data v MS Excel) načteny všechny relevantní záznamy týkající se řešeného katalogového čísla odpadu (např. 19 08 05). Konkrétně jsou vybrány pouze potřebné informace o jednotlivých záznamech, jenž jsou definované výše. Následně je s údaji pracováno jako se strukturou DataFrame využívající knihovnu „Pandas“. Chybějící hodnoty jsou nahrazeny hodnotou nula, aby s nimi bylo možné dále lépe pracovat.

```
1 def ImportDat():
2     Soubor = "Test.xlsx"
3     DirPath = str(pathlib.Path().resolve())
4     Database = pd.read_excel(DirPath + "\\ " + Soubor, sheet_name=0, converters={"Evident_ID": str, "Partner_ID": str})
5     Database = Database[["Evident_ID", "Partner_ID", "Kód nakládání", "Množství odpadu + (t)", "Množství odpadu - (t)", "Sušina (t)"]]
6     Database.fillna(0, inplace=True)
7     return Database
```

V dalším kroku jsou připravovány jednotlivé skupiny nakládání. Klíčovou skupinou je kumulace, která zároveň definuje průměrnou sušinu v ČR pro nahrazení chybějících údajů. Z načtených záznamů jsou vybrány ty, které obsahují příslušné kódy nakládání. Jelikož je třeba mít pro každý uzel pouze jednu hodnotu a výčet kódů je větší než jedna, je nutné dle evidenta jednotlivé hmotnosti sečíst a sloučit jak v případě zvodnělého kalu, tak i v případě sušiny. V rámci implementace v jazyce „Python“ je nutné po sloučení dodatečně odstranit nepotřebné (duplicitní) záznamy. Následně jsou vybrány ty záznamy, jenž obsahují nenulové množství sušiny a z těchto hodnot je vypočten průměr v ČR, který je dále využit pro doplnění chybějících hodnot.

```
1 Kody_Kumulace = re.compile(r"^(A00|A10|A20|A30|A40|A60|B030|N50|N60)$")
2 Kumulace = Database.loc[Database["Kód nakládání"].str.match(Kody_Kumulace), ["Evident_ID", "Množství odpadu + (t)", "Sušina (t)"]]
3 Kumulace["Množství"] = Kumulace.groupby(["Evident_ID"])["Množství odpadu + (t)"].transform("sum")
4 Kumulace["Sušina"] = Kumulace.groupby(["Evident_ID"])["Sušina (t)"].transform("sum")
5 Kumulace = Kumulace.drop(columns=["Množství odpadu + (t)", "Sušina (t)"])
6 Kumulace = Kumulace.drop_duplicates(subset=["Evident_ID"]).reset_index(drop=True)
7 PrumernaSusinaCR = Kumulace.loc[Kumulace["Sušina"] != 0, "Sušina"].sum() / Kumulace.loc[Kumulace["Sušina"] != 0, "Množství"].sum()
8 Kumulace.loc[Kumulace["Sušina"] == 0, "Sušina"] = Kumulace.loc[Kumulace["Sušina"] == 0, "Množství"] * PrumernaSusinaCR
```

Stejný postup je realizován i v případě dalších skupin plusových nakládání (vyjma převzetí). Zde je pouze využita hodnota průměrné sušiny v ČR u kumulace pro doplnění chybějících informací o podílu sušiny. Jako ukázka je zde uvedena skupina „Vyskladnění“,

téměř identický kód je v případě importu a sekundární produkce jen s filtrováním jiných kódů nakládání.

```
1 Kody_Sklad = re.compile(r"^(C00)$")
2 Sklad = Database.loc[Database["Kód nakládání"].str.match(Kody_Sklad), ["Evident_ID", "Množství odpadu + (t)", "Sušina (t)"]]
3 Sklad["Množství"] = Sklad.groupby(["Evident_ID"])["Množství odpadu + (t)"].transform("sum")
4 Sklad["Sušina"] = Sklad.groupby(["Evident_ID"])["Sušina (t)"].transform("sum")
5 Sklad = Sklad.drop(columns=["Množství odpadu + (t)", "Sušina (t)"])
6 Sklad = Sklad.drop_duplicates(subset=["Evident_ID"]).reset_index(drop=True)
7 Sklad.loc[Skld["Sušina"] == 0, "Sušina"] = Sklad.loc[Skld["Sušina"] == 0, "Množství"] * PrumernaSušinaCR
```

Odlíšnost je v případě skupiny předání a převzetí. Je nutné vytvořit pro každou takovou dvojici unikátní identifikátor „Hrana“, jenž definuje spojení mezi danými dvěma uzly. Pro každou hranu je nutné mít pouze jednu hodnotu, tj. je nutné sloučit záznamy, které mají stejnou dvojici (evident, partner). Za tímto účelem je vytvořen sloupec „Hrana“, která je spojením textových řetězců příslušných identifikátorů odděleným podtržítkem. První část představuje uzel, který odpad předává, a druhá část uzel, jenž odpad přebírá. Přes tento údaj je možné dále propojit skupiny předání a převzetí, jež jsou klíčové pro vytvoření matice obsahující informaci o toku odpadu mezi uzly. Po propojení těchto skupin jsou postupně nahrazeny chybějící hodnoty, které mohou nastat v případě, že daná hodnota „Hrana“ je pouze v jedné z propojovaných skupin (předání / převzetí). Na závěr je množství odpadu přepravovaného mezi dvěma uzly vypočteno jako průměr z evidovaných hodnot předání a převzetí.

```
1 Kody_Prevzeti = re.compile(r"^(B00|B22)$")
2 Prevzeti = Database.loc[Database["Kód nakládání"].str.match(Kody_Prevzeti), ["Evident_ID", "Partner_ID", "Množství odpadu + (t)"]]
3 Prevzeti["Hrana"] = Prevzeti["Partner_ID"] + "-" + Prevzeti["Evident_ID"]
4 Prevzeti["Množství"] = Prevzeti.groupby(["Evident_ID", "Partner_ID"])["Množství odpadu + (t)"].transform("sum")
5 Prevzeti = Prevzeti.drop(columns=["Množství odpadu + (t)"])
6 Prevzeti = Prevzeti.drop_duplicates(subset=["Evident_ID", "Partner_ID"]).reset_index(drop=True)
7
8 Kody_Predani = re.compile(r"^(wN2|N3|N8|N10|N33)$")
9 Predani = Database.loc[Database["Kód nakládání"].str.match(Kody_Predani), ["Evident_ID", "Partner_ID", "Množství odpadu - (t)"]]
10 Predani["Hrana"] = Predani["Evident_ID"] + "-" + Predani["Partner_ID"]
11 Predani["Množství"] = Predani.groupby(["Evident_ID", "Partner_ID"])["Množství odpadu - (t)"].transform("sum")
12 Predani = Predani.drop(columns=["Množství odpadu - (t)"])
13 Predani = Predani.drop_duplicates(subset=["Evident_ID", "Partner_ID"]).reset_index(drop=True)
14
15 Toky = pd.merge(Predani, Prevzeti, how="outer", on=["Hrana"])
16 Toky.Množství_x.fillna(0, inplace=True)
17 Toky.Množství_y.fillna(0, inplace=True)
18 Toky.Evident_ID_x.fillna(Toky.Partner_ID_y, inplace=True)
19 Toky.Evident_ID_y.fillna(Toky.Partner_ID_x, inplace=True)
20 Toky.Partner_ID_x.fillna(Toky.Evident_ID_y, inplace=True)
21 Toky.Partner_ID_y.fillna(Toky.Evident_ID_x, inplace=True)
22 Toky["Množství"] = (Toky["Množství_x"] + Toky["Množství_y"]) / 2
```

## Optimalizace

Využitá optimalizační knihovna „PuLP“ pro vstupní data využívá datovou strukturu slovníku označovanou jako Dict{ }. Postupně jsou tak ze struktury „DataFrame“ vytvořeny tyto slovníky u všech připravených dat pomocí funkce „makeDict“.

```

1 Uzly = Uzly.values.tolist()
2 TokyModel = makeDict([Uzly, Uzly], [[0] * len(Uzly)] * len(Uzly), 0)
3 for id, i in Toky.iterrows():
4     TokyModel[i["Evident_ID_x"]][i["Evident_ID_y"]] = i["Množství"]
5
6 Kumulace_S = makeDict([Kumulace["Evident_ID"].values.tolist(), Kumulace["Sušina"].values.tolist(), 0)
7 Sklad_S = makeDict([Sklad["Evident_ID"].values.tolist(), Sklad["Sušina"].values.tolist(), 0)
8 Import_S = makeDict([Import["Evident_ID"].values.tolist(), Import["Sušina"].values.tolist(), 0)
9 Sekundarni_S = makeDict([Sekundarni["Evident_ID"].values.tolist(), Sekundarni["Sušina"].values.tolist(), 0)
10 Kumulace_M = makeDict([Kumulace["Evident_ID"].values.tolist(), Kumulace["Množství"].values.tolist(), 0)
11 Sklad_M = makeDict([Sklad["Evident_ID"].values.tolist(), Sklad["Množství"].values.tolist(), 0)
12 Import_M = makeDict([Import["Evident_ID"].values.tolist(), Import["Množství"].values.tolist(), 0)
13 Sekundarni_M = makeDict([Sekundarni["Evident_ID"].values.tolist(), Sekundarni["Množství"].values.tolist(), 0)
14 Zpracovani_M = makeDict([Zpracovani["Evident_ID"].values.tolist(), Zpracovani["Množství"].values.tolist(), 0)

```

Dále je definován optimalizační problém a proměnné modelu. Konkrétně se jedná o hledané průměrné podíly sušiny v jednotlivých uzlech. Následně jsou tvořeny postupně bilanční omezení pro každý uzel, jenž popisuje rovnost VSTUP = VÝSTUP. Tato rovnice je pro konkrétní uzel implementována do modelu pouze v případě, že daný uzel má evidované předání nebo zpracování. V opačném případě je podíl sušiny nastaven na vážený průměr z evidovaných vstupů nebo nastaven na nulu, když vstupy nejsou evidovány.

```

1 SusinaUzly = LpVariable.dicts("WI", Uzly, cat=LpContinuous)
2 prob = LpProblem("VypocetSusiny", LpMinimize)
3
4 for i in Uzly:
5     if (Zpracovani_M[i] + sum_of_Dict(TokyModel[i], i)) > 0:
6         prob += (Kumulace_S[i] + Sklad_S[i] + Import_S[i] + Sekundarni_S[i] + lpSum(TokyModel[j][i] * SusinaUzly[j] for j in Uzly if i
7 != j)) == SusinaUzly[i] * (Zpracovani_M[i] + lpSum(TokyModel[i][j] for j in Uzly if i != j)))
8     else:
9         if (Kumulace_M[i] + Sklad_M[i] + Import_M[i] + Sekundarni_M[i]) > 0:
10             prob += SusinaUzly[i] == (Kumulace_S[i] + Sklad_S[i] + Import_S[i] + Sekundarni_S[i]) / (Kumulace_M[i] + Sklad_M[i] + Import_M[i] + Sekundarni_M[i])
11         else:
12             prob += SusinaUzly[i] == 0

```

V posledním kroku je spuštěn samotný optimalizační algoritmus. Po ukončení výpočtu je hodnota proměnných převedena na strukturu „DataFrame“, vhodná pro následující výpis do souboru MS Excel.

```

1 prob.writeLP("VypocetSusiny.lp")
2 prob.solve()
3
4 Solve_Susina = pd.DataFrame()
5 Solve_Susina["Evident_ID"] = SusinaUzly.keys()
6 for id, i in Solve_Susina.iterrows():
7     Solve_Susina.loc[id, "Průměrná sušina"] = SusinaUzly[i["Evident_ID"]].varValue

```



## 2.5 Autovraky a baterie

V rámci vyhlášky č. 273/2021 Sb. (tab. č. 1 přílohy č. 13 – evidenční kódy) proběhla změna ve vykazování sekundární produkce odpadů **pro vybraná** katalogová čísla skupiny **16 a 19**. Konkrétně se vykazuje sekundární produkce odpadů pod samostatným kódem nakládání dle **nové legislativy: „BN41“**. Proto již nehrozí v dalších analýzách umělé navýšení množství odpadu při zaevidovaných autovracích a následných odpadů vzniklých z jejich úpravy.

Při tvorbě PD ISOH (pro případ staré i **nové legislativy**) není nutné u těchto odpadů provádět žádné změny při překlápění dat z ISOH. Nicméně při tvorbě analýz a výpočtu indikátorů OH založených na datech evidovaných dle staré legislativy je nutné mít na paměti, že sekundárně produkováný odpad má stále evidenční kód nakládání **„A00“**.

Podle staré legislativy se autovraky 16 01 04\* evidují pod „BN30“. Autovraky 16 01 04\* chybně zaevidované pod **„A00“ se v PD ISOH převedou na „BN30“**, množství je zachováno, čímž se jejich produkce neztratí, ale započítá se.

V případě **nové legislativy** je nutné u kat. č. **16 01 04\*** ověřit správnost evidence u produkce, nově by měla být evidována pod evidenčním kódem nakládání **„A40“** (produkce prvním převzetím výrobků s ukončenou životností nebo odpadních dopravních prostředků z různých druhů dopravy a strojů). Jestliže by byla produkce chybně evidována pod evidenčními kódy nakládání **„A00, A10, A20, A30“**, je nutné tento kód změnit na **„A40“**. Množství je zachováno. To platí pro všechny producenty včetně firem.

Stejným způsobem je u **nové legislativy** vhodné přistupovat i k odpadům spojeným s bateriemi a akumulátory s kat. č. podskupin **16 06 a 20 01 (odpady: 16 06 01, 16 06 02, 16 06 03, 16 06 04, 16 06 05, 20 01 33, 20 01 34)**. Pouze se zde nahradí všechny chybně evidované kódy nakládání **„A00, A10, A20“** na **„A30“** (produkce prvním převzetím zpětně odebraných výrobků s ukončenou životností – elektrozařízení, baterie a akumulátory, pneumatiky). Množství je zachováno. To platí pro všechny producenty včetně firem. Stejně jak bylo provedeno u podskupiny **16 06** se provede i pro kat. č. **16 08 01, 16 08 02, 16 08 03**.

V případě **nové legislativy** u zařízení ke zpracování vozidel s ukončenou životností či zařízení ke zpracování elektrozařízení (s IČZ), které předává pneumatiky (kat. č. **16 01 03**) a automobilové či průmyslové baterie a akumulátory, či přenosné průmyslové baterie (kat. č. **16 06 01\*, 16 06 02\*, 16 06 03\*, 16 06 04, 16 06 05, 20 01 33\*, 20 01 34**) a do systému zpětného odběru uvede evidenční kód **„XN3“**, **změní se tento kód na „XN33“**.

Tento bod lze řešit postupně pro každého evidenta zvlášť.



## 2.6 Vznik odpadu mimo provozovnu

Odpad může vzniknout mimo provozovnu ve dvou případech: 1) prostřednictvím mobilního zařízení k úpravě, využití nebo odstranění odpadu a 2) v případě provádění nějaké služby nebo práce mimo provozovnu. Příkladem bodu 2) můžou být např. stavební práce, při kterých vzniká stavební odpad. V případě vzniku takového odpadu (bod 1 i 2) se eviduje produkce odpadu na správní obvod ORP (SO ORP), ve kterém vznikla bez rozlišení ZÚJ (tyto záznamy jsou přepojeny na fiktivní ZÚJ pro příslušné ORP, resp. na fiktivní ZÚJ pro hl. m. Prahu bez ohledu na to, jaké ZÚJ bylo původně uvedeno v hlášení). V případě, že evident vyprodukoval tímto způsobem odpad ve více ORP, pak podává za každé ORP samostatné hlášení.

Taková produkce je specificky označena v rámci evidence v listu 1 vyhlášky 273/2021 Sb., příloha 13 (**nová legislativa**). Konkrétně se jedná o kolonku „Typ hlášení“:

- Vznik odpadu mimo provozovnu – činnost na území SO ORP / SOP.
- Mobilní zařízení k úpravě / využití / odstranění odpadu (viz § 94 bod 1 v zákonu 541/2020 Sb.).

V případě **staré legislativy** je nutné tyto informace hledat v Registru zařízení. Konkrétně to platí pro IČZ s druhem zařízení popsany v tab. 2.

Tab. 2: Typy zařízení týkající se vzniku odpadu mimo provozovnu

Kód	Název
MobBio01	Mobilní biologické procesy
MobFyzChe01	Mobilní fyzikálně – chemické procesy
MobChemZprac	Mobilní fyzikálně chemické zpracování odpadu (využití / odstranění / úprava odpadu)
MobMechZprac	Mobilní mechanické zpracování odpadu (využití / odstranění / úprava odpadu)
MobMech01	Mobilní mechanické procesy
MobOst01	Mobilní ostatní procesy
MobZpracVoz	Mobilní zpracování vozidel z různých druhů dopravy, kromě silniční (kolejová, letecká, lodní)

Dále je třeba prověřit informace u IČP. Pokud obsahuje název IČP kód pro územní jednotku, kterou lze najít v číselníku ORP, pak se předpokládá, že se jedná o vznik odpadu mimo provozovnu v daném ORP, pro které je kód pro územní jednotku v názvu IČP. Tyto záznamy se vykáží na fiktivní ZÚJ/obec v příslušném SO ORP, viz popis v **kap. 2.1**.

Následně je třeba zajistit uvedení fiktivního ZÚJ i pro případy automaticky vygenerovaných nových hlášení za podlimitní producenty a hřištníky (viz **kap. 2.2**). Zde se jedná o situace, kdy je kód ORP uveden u IČP partnera (podlimitního producenta nebo hřištníka). I tento typ odpadu je třeba vykazovat na fiktivní ZÚJ/obec v příslušném SO ORP.

Tímto způsobem je třeba evidovat produkci i nakládání s odpady (všechny kódy nakládání), aby i v případě fiktivních ZÚJ platily hmotnostní bilance.

Tento bod lze řešit postupně pro každého evidenta zvlášť.

## 2.7 Katalogová čísla dle nové legislativy

Změny evidence katalogových čísel odpadu definuje vyhláška č. 8/2021 Sb. o Katalogu odpadů a posuzování vlastností odpadů (Katalog odpadů). Oproti původní podobě zde došlo pouze k možnosti evidence odpadu ve větším detailu. Doposud obsahovalo katalogové číslo vždy šest číslic. Nyní je možné v některých případech zařadit odpad ve větším detailu pod označením s osmi číslicemi. Např. pro 20 01 01 (papír a lepenka), je nyní možné rozšířit evidenci na 20 01 01 01 (kompozitní a nápojové kartony).

Odpad pod osmičíselným kat. č. se z databáze ISOH přepíše běžným způsobem do PD ISOH. V rámci exportu dat je třeba zajistit, aby toto množství odpadu nechybělo i v nadřazených skupinách, podskupinách a družích odpadů (dvoučíselné, čtyřčíselné, šestičíselné) obdobně jako export podskupin (např. 20 01) nebo skupin (např. 20).

Tento bod lze řešit postupně pro každého evidenta zvlášť.

## 3 Komunikace PD ISOH a výpočtového jádra

Z důvodu neustálého vývoje matematických nástrojů pro potřeby analýz, rekonstrukcí a vyhodnocení odpadového hospodářství je nutné i do budoucna definovat obecné rozhraní mezi výpočtovým jádrem a PD ISOH. Výpočetní moduly založené na pokročilých metodách není možné implementovat do databázových systémů, které neumožňují integraci složitějších algoritmů. Dílčí výpočtové operace tak budou implementovány ve formě externích modulů, které zároveň umožní přehledně realizovat dílčí vyhodnocení dle aktuálních potřeb. Externí moduly budou vyvíjeny v jazyce Python.

Jelikož vývojář nemusí mít přístup ke kompletní databázi PD ISOH, bude mu při vývoji poskytnuta příslušná část datové sady ve formě MS Excel s požadovanými parametry. Vývoj externího modulu tak bude probíhat v rozhraní MS Excel => Python => MS Excel. Import a export dat z prostředí MS Excel bude realizován pomocí funkcí z dostupných knihoven, přičemž vstupní či výstupní data budou převedena na strukturu „Dataframe“ s využitím knihovny „Pandas“. Povinností vývojáře je také využívat formát identifikátorů jednotlivých subjektů (IČP/IČZ) dle správce databáze.

Každý externí modul pro PD ISOH bude rozdělen do 4 částí, které jsou následující:

- Definice využitých knihoven.
- Import dat s převedením na strukturu „Dataframe“.
- Realizace výpočtu – pre-processing dat, matematické programování, převedení výstupů na strukturu „Dataframe“.

- Export dat.

Správce databáze ISOH (resp. PD ISOH) následně implementuje na PC/servery Python knihovny potřebné pro běh externího modulu. Dále funkce poskytnuté vývojářem externích modulů v bodu 2 a 4 budou nahrazeny SQL příkazem pro propojení modulu s kompletní sadou dat včetně převedení na strukturu „Dataframe“.

Následující ukázka struktury externího modulu popisuje výpočet sušiny v kalech z čištění odpadních vod na úrovni jednotlivých zařízení. Úvodní 4 řádky definují potřebné knihovny. Řádek 7 až 13 popisuje import dat ze souboru MS Excel, přičemž tato funkce bude nahrazena správcem databáze SQL dotazem „SELECT \* FROM“ z „faktové tabulky“ nebo „view“ obsahující pouze nezbytná data z evidence pro výpočet. Následuje vlastní kód výpočtu skládající se ze tří samostatných funkcí, které mohou využívat libovolný formát dat, jenž je však nutné na konci výpočtu opět převést na strukturu „Dataframe“. Předposlední blok kódu se týká exportu dat, který opět bude doplněn či nahrazen SQL kódem, pomocí které budou výsledky uloženy do příslušné části databáze dle povahy vypočtených dat (návaznost na subjekty, regiony, rekonstruovaná databáze, souhrnné indikátory atd.). Řádky 30 a 31 již zajišťují pouze spuštění výpočtového modulu.

#### Nahrazení importu dat:

```
host = '*****'
database = '*****'
username = '*****'
password = '*****'
port='*****'

url = 'mssql+pyodbc://{user}:{passwd}@{host}:{port}/{db}?driver=SQL+Server'
.format(user=username, passwd=password, host=host, port=port, db=database)

engine = sa.create_engine(url)

query = "select * from [ISOH2REPORT].[Isoh2API].[kaly]"
```

**Výpočetní modul:**

```
1 import pandas as pd
2 import re
3 import pathlib
4 from pulp import *
5
6
7 def ImportDat():
8     Soubor = "Test.xlsx"
9     DirPath = str(pathlib.Path().resolve())
10    Database = pd.read_excel(DirPath + "\\ " + Soubor, sheet_name=0, converters={"Evident_ID": str, "Partner_ID": str})
11    Database = Database[["Evident_ID", "Partner_ID", "Kód nakládání", "Množství odpadu + (t)", "Množství odpadu - (t)", "Sušina (t)"]]
12    Database.fillna(0, inplace=True)
13    return Database
14
15
16 def sum_of_Dict(Dict, Node):
17
18
19 def PreprocessingDat():
20
21
22 def Optimalizace():
23
24
25 def ExportDat(Output):
26     with pd.ExcelWriter("Vystup_Python.xlsx") as writer:
27         Output.to_excel(writer, sheet_name="Sušina", index=False, encoding="utf-8-sig", header=True)
28
29
30 Vysledek = Optimalizace()
31 ExportDat(Vysledek)
```

**Nahrazení exportu dat:**

```
def ExportDatSql(Output):
    conn = sa.create_engine(url, use_setinputsizes=False)
    Output.to_sql('Susina', conn, schema='Isoh2API', if_exists='replace')
```

```
Vysledek = Optimalizace()
#ExportDat(Vysledek)
ExportDatSql(Vysledek)
```

Po provedení kalkulací ve výpočtovém jádře je výsledek vrácen do samostatné tabulky, odkud se s ním dále pracuje ve standardních strukturách PD ISOH. Předchozí verze dat je nahrazena nově vytvořenou.