FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Bakalářská práce

Generátor technické dokumentace pro systémy SAP

Studijní program: B0613A140005 – Informační technologie
Studijní obor: B0613A140005AI – Aplikovaná informatika
Autor práce: **Petr Kaiser**
Vedoucí práce: Ing. Jana Vitvarová, Ph.D.

Liberec 2023

FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Zadání bakalářské práce

Generátor technické dokumentace pro systémy SAP

Jméno a příjmení:	**Petr Kaiser**
Osobní číslo:	M19000019
Studijní program:	B0613A140005 Informační technologie
Specializace:	Aplikovaná informatika
Zadávající katedra:	Ústav mechatroniky a technické informatiky
Akademický rok:	2022/2023

Zásady pro vypracování:

1. Seznamte se se systémem SAP z pohledu vývoje systémových rozšíření a jejich dokumentace.
2. Identifikujte SAP objekty vhodné pro dokumentování.
3. Proveďte rešerši způsobů a možností generování dokumentace v SAPu a v jiných podobných systémech.
4. Navrhněte vlastní nástroj pro generování dokumentace včetně exportu do vybraných formátů a šablon.
5. Nástroj v prostředí SAP implementujte a otestujte.

Rozsah grafických prací:	dle potřeby dokumentace
Rozsah pracovní zprávy:	30-40 stran
Forma zpracování práce:	tištěná/elektronická
Jazyk práce:	Čeština

Seznam odborné literatury:

[1] ANDERSON, George W. Naučte se SAP za 24 hodin. Brno: Computer Press, 2012. ISBN 978-80-251-3685-0.

[2] BANDARI, Kiran. ABAP: The Comprehensive Guide to SAP ABAP 7.52 and 1909. 2nd edition. SAP Press, 2019. ISBN 978-1-4831-8401-2.

[3] HARAMUNDANIS, Katherine. The Art of Technical Documentation. Digital Press, 1992. ISBN 9781483184012.

Vedoucí práce:	Ing. Jana Vitvarová, Ph.D. Ústav mechatroniky a technické informatiky
Datum zadání práce:	12. října 2022
Předpokládaný termín odevzdání:	15. května 2023

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

doc. Ing. Josef Černohorský, Ph.D.
vedoucí ústavu

V Liberci dne 12. října 2022

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

11. 5. 2023 Petr Kaiser

Poděkování

Rád bych touto formou poděkoval svému konzultantovi Bc. Matějovi Haase za poskytnutí úvodního školení do SAPu a zřízení přístupu do SAP systému, ve kterém byl program vyvíjen. Nadále děkuji své vedoucí práce Ing. Janě Vitvarové, Ph.D. za poskytnutí cenných rad k obsahu práce.

Generátor technické dokumentace pro systémy SAP

Abstrakt

Bakalářská práce je zaměřena na generování dokumentace v systému SAP do externích souborů.

V první části se práce zabývá popisem vybraných částí a principů fungování systému SAP z pohledu vývojáře.

Následně popisuje problematiku vytváření programátorské dokumentace v různých systémech a programovacích jazycích.

Dále se zabývá možnostmi vytváření dokumentace v systému SAP a porovnává již existující nástroje pro tvorbu externí dokumentace.

V druhé části se zaměřuje na návrh a implementaci vlastního nástroje, který je schopen generovat dokumentaci do externích souborů ve vybraných formátech.

Výsledný nástroj je vytvořený v jazyce ABAP, jenž slouží k vývoji aplikací v SAPu.

Klíčová slova: SAP, ABAP, generování dokumentace, šablona, XML, MS Word

Technical documentation generator for SAP systems

Abstract

The bachelor's thesis is focused on the generation of documentation in the SAP system to external files.

The first part describes selected parts and principles of the SAP system from the developer's point of view.

Subsequently, it describes the issue of creating programming documentation in various systems and programming languages.

It also describes the options for creating documentation in the SAP system and compares existing tools for creating external documentation.

The second part focuses on the design and implementation of a custom tool that will generate documentation into external files in selected formats.

The resulting tool is created in the ABAP language which is used to develop applications in SAP.

Keywords: SAP, ABAP, documentation generation, template, XML, MS Word

Seznam zkratek

ERP	Enterprise Resource Planning (plánování podnikových zdrojů)
SAP	Systeme, Anwendungen, Produkte in der Datenverarbeitung, (Systémy - Aplikace - Produkty ve zpracování dat)
ABAP	Advanced Business Application Programming
GUI	Graphic User Interface (grafické uživatelské rozhraní)
IDE	Integrated Development Environment (vývojové prostředí)
XML	Extensible Markup Language (rozšiřitelný značkovací jazyk)
MS	Microsoft
FP	Funkční požadavek

Obsah

1 Úvod

1.1 Motivace

Téma vychází z požadavku z praxe, kdy zákazník po firmě požaduje k odevzdanému softwaru i dokumentaci. Pro tuto dokumentaci dodává zákazník vlastní šablonové soubory ve formátu MS Word a vývojář následně musí dokumentaci do těchto šablon ručně vyplnit. Tento proces je ovšem časově neefektivní a při ručním vyplňování dokumentace může dojít k chybám ze strany vývojáře.

1.2 Cíle práce

Hlavním cílem práce je navrhnout a implementovat nástroj v systému SAP, který bude celý proces generování dokumentace automatizovat. Jedním z hlavních požadavků je možnost využívat zákazníkovy šablonové soubory, se kterými po modifikaci ze strany vývojáře dovede program pracovat.

Cíle práce je možné shrnout do následujících bodů:

- Seznámit se s vývojářskou částí SAPu.
- Provést analýzu tvorby dokumentace v SAPu a vyzkoušet externí nástroje pro tvorbu dokumentace.
- Provést analýzu možností tvorby dokumentace v ostatních programovacích jazycích a srovnat tyto možnosti se SAPem.
- Navrhnout a implementovat vlastní nástroj pro tvorbu dokumentace.

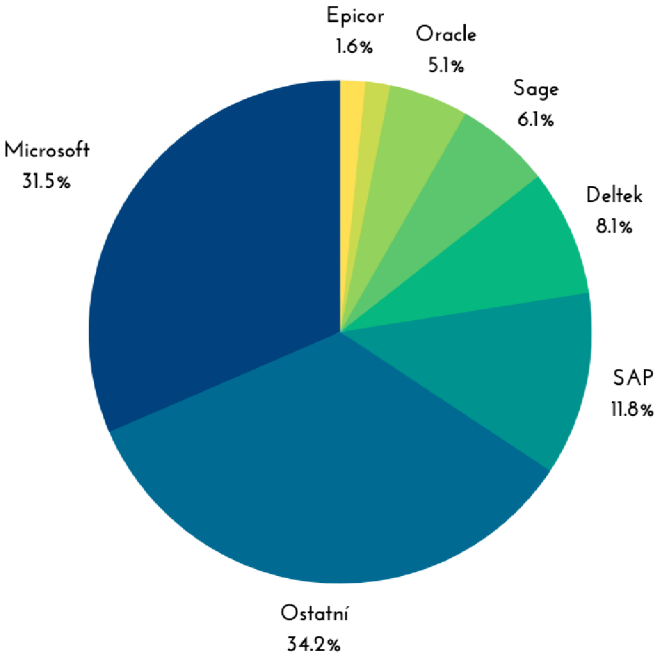
2 Systém SAP

2.1 Úvod do systému SAP

Systém SAP je produkt stejnojmenné německé společnosti a řadí se mezi ERP systémy. Tyto systémy, též nazývány jako podnikové informační systémy, slouží k řízení celého podniku. Každé oddělení může mít v systému vlastní aplikaci uzpůsobenou svým potřebám, jež navíc dokáže komunikovat s aplikacemi ostatních oddělení. Jako příklad je možné uvést aplikaci "Timesheet", která slouží k vykazování odpracovaných hodin jednotlivých zaměstnanců a tato aplikace komunikuje s účetním oddělením pro výpočet mezd zaměstnanců.

SAP je první firma, která přišla na trh s ERP softwarem a v současné době je SAP jedním z nejprodávanějších a nejpoužívanějších ERP systémů. [1]

Obrázek 2.1 zobrazuje podíl dodavatelů ERP systému pro rok 2022. Největší podíl na trhu má Microsoft s hlavním produktem Dynamics 365 Business Central, druhým v pořadí je firma SAP se svými produkty.



Obrázek 2.1: Podíl dodavatelů ERP systémů na trhu pro rok 2022 [2]

Svými vlastnostmi je systém SAP vhodný pro použití ve větších firmách, ve světě jej používá například BMW, Coca Cola, IBM, Pfizer, DHL a mnohé další. [3]

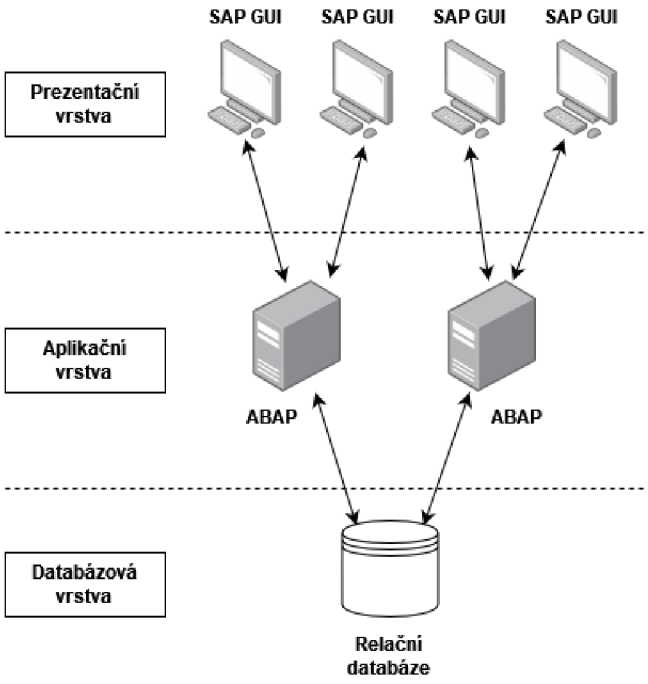
2.2 Architektura systému SAP

Informace v této kapitole pocházejí z literaty [4].

Z technického pohledu se systém SAP dělí do tří vrstev:

- Prezentační vrstva
- Aplikační vrstva
- Databázová vrstva

Toto rozdělení vytváří architekturu **klient-server**. Obrázek 2.2 toto rozdělení znázorňuje spolu s komunikací mezi jednotlivými vrstvami.



Obrázek 2.2: Architektura klient-server

Prezentační vrstva

Slouží ke komunikaci s uživatelem za pomoci dialogových oken a obrazovek. Tuto funkcionalitu zajišťuje programový balík pro grafické uživatelské rozhraní SAP GUI. Data z prezentační vrstvy jsou posílána do aplikační vrstvy. Bez SAP GUI nelze se systémem SAP komunikovat.

Aplikační vrstva

Obsahuje aplikační server, na kterém jsou zpracovávána data a běží na něm programy (v jazyce ABAP). S klientem komunikuje pomocí SAP GUI, ke komunikaci s databází používá systém pro správu databází (v závislosti na fyzické databázi se způsob komunikace mění).

Databázová vrstva

Obsahuje databázový server, na kterém jsou uložena všechna data. V závislosti na množství dat a dalších faktorech mohou být aplikační a datové servery umístěny na jednom nebo více počítačích.

2.2.1 Transport objektů do jiných SAP systémů

Kromě technického pohledu na architekturu klient-server lze prostředí SAP rozdělit na tři oblasti z hlediska vývoje:

- Systém pro testování a vývoj
- Konsolidační systém
- Produkční systém

Systém pro testování a vývoj

Systém pro testování a vývoj ukládá zkušební data. Používá se pro systémová nastavení pro programování a testování. V systému pro testování a vývoj jsou nejprve uložena nastavení systému SAP pro daného zákazníka, která jsou přizpůsobena potřebám konkrétního zákazníka. Proces přípravy těchto nastavení se nazývá cutomizing. Tento systém často obsahuje sandbox tj. oblast, ve které mohou uživatelé simulovat a testovat podnikové procesy.

Konsolidační systém

Objekty ze systému pro testování a vývoj jsou přenášeny do konsolidačního systému (nebo také systému zajišťujícího kvalitu). Tento systém bývá často používán na akceptační testy s reálnými daty.

Produkční systém

V případě, že program projde akceptačními testy, tak je nasazen do produkce. V tomto systému nelze manipulovat s jeho parametry. Dále platí, že přímo v produkčním systému se nikdy neprovádí vývoj a nelze měnit nastavení customizingu.

Transportní požadavek

K přenosu dat mezi jednotlivými SAP systémy slouží transportní požadavek. Každý požadavek musí mít přiřazené unikátní číslo požadavku.

2.3 Programovací jazyk ABAP

ABAP je objektový programovací jazyk vytvořený společností SAP pro vývoj aplikací v systému SAP. Vedle standardních programovacích příkazů obsahuje ABAP i sadu SQL příkazů, nazývaných jako Open SQL, pro práci s databází. Všechny uživatelem vytvořené objekty jazyka ABAP musí začínat písmenem Z nebo Y.

Více o jazyce ABAP je možné se dozvědět v jeho oficiální dokumentaci [5].

2.3.1 Základní typy programů

ABAP rozlišuje 2 typy spustitelných programů:

- report,
- dynamický program.

Report

Reportem se rozumí spustitelný program. Report neumožňuje uživateli zasahovat do běhu programu. Na základě vstupních parametrů zadaných uživatelem se provede běh programu.

Dynamický program

Též zvaný jako "dynpro", umožňuje zásah uživatele během provádění programu. Využívá dialogové obrazovky, pomocí kterých uživatel rozhoduje o dalším průběhu programu.

2.4 Základní objekty jazyka ABAP

Základní stavební prvky jazyka ABAP se nazývají objekty a jsou uchovány v SAP Repository.

Ve starších verzích neumožňoval ABAP objektově orientované programování a obsahoval pouze funkční skupiny a funkční moduly. Ve verzi 4.0, vydané v roce 1999, byly přidány třídy a metody, čímž se ABAP stal plně objektově orientovaným jazykem.

Zde jsou uvedeny nejčastěji používané objekty jazyka ABAP:

- **Report** - spustitelný program (viz 2.3.1).
- **Třída** - základní stavební prvek každého objektově orientovaného jazyka, obsahuje atributy a metody, umožňuje vytvářet instance.
- **Metoda** - kód plnící určitou funkci, nejedná se o spustitelný program, ale můžeme jej využít jako součást jiných programů, slouží k přepoužitelnosti kódu.

- **Funkční modul** - kód plnící určitou funkci, slouží k přepoužitelnosti kódu.
- **Funkční skupina** - zapouzdřuje funkční moduly.
- **Include** - používá se pro zahrnutí kódu z jiného programu do aktuálního programu, často slouží k zahrnutí deklarací proměnných, konstant, typů a struktur.
- **Paket** - zapouzdřuje spolu související objekty.
- **Databázová tabulka** - základ celé databázové vrstvy, má předem danou strukturu a uchovává data.

3 Tvorba programátorské dokumentace

Dokumentace k produktu je poměrně široký pojem, může mít různé formy a obsah. V rámci této práce je kladen důraz pouze na programátorskou dokumentaci, to znamená popis produktu pro jiného programátora, který je z dané dokumentace schopen pochopit strukturu programu, aniž by musel studovat zdrojový kód. Zároveň by měl být schopen části kódu správně používat.

3.1 Obsah programátorské dokumentace

Dokumentace k programu se skládá z:

- popisu programu,
- komentářů v kódu,
- metadat.

3.1.1 Popis programu

Popis programu slouží k:

- popisu problematiky, kterou daný program řeší,
- popisu použitých technik a algoritmů k řešení problému,
- definování systémových požadavků na spuštění programu,
- popisu procesu instalace a spuštění programu,

Příkladem popisu programu může být "README" soubor.

3.1.2 Komentáře v kódu

V případě dokumentování zdrojového kódu se dokumentace píše přímo do kódu k příslušnému prvku pomocí tzv. dokumentačních komentářů. Obsah těchto komentářů se pak odvíjí od dokumentovaného objektu. Dokumentace by měla být co nejstručnější a nejvýstižnější.

Blok s dokumentačními komentáři

Blok s dokumentačními komentáři se umísťuje před deklaraci daného objektu a měl by obsahovat základní informace o daném objektu, tj. krátký popis objektu, například u třídy to je seznam atributů a metod, v případě metody pak seznam parametrů a jejich krátký popis.

Tento blok má předepsaný formát, který je nutné dodržet. Každý programovací jazyk si může definovat vlastní formát komentářů. Komentáře obsahují klíčová slova, anotovaná znakem "@", která slouží k popisu parametrů, atributů, návratových hodnot, výjimek, apod.

Obrázek 3.1 ukazuje příklad bloku dokumentačních komentářů (začínají sekvencí "!) umístěný před deklarací metody v jazyce ABAP. Komentáře obsahují účel metody a výpis parametrů včetně jejich významu.

```
"! Vrací url šablonového souboru z konfigurační tabulky zdocu_tmplt
"! @parameter is_object   | dokumentovaný objekt
"! @parameter rv_url      | url šablony
  class-methods get_template_url
    importing
      value(is_object) type ty_object_list
    returning
      value(rv_url) type zdocu_tmplt_url .
```

Obrázek 3.1: Příklad bloku dokumentačních komentářů v jazyce ABAP

Standardní komentáře

Standardní komentáře slouží například k popisu složitějších částí kódu. Konstanty a proměnné by vedle své deklarace také měly obsahovat i stručný popis jejich účelu.

Na obrázku 3.2 jsou znázorněny komentáře v jazyce ABAP (začínají znakem "). Komentáře slouží k lepšímu porozumění kroků procesu filtrování objektů obsažených v paketu.

```
"loop pres vsechny objekty nalezene v paketu
loop at lt_objectlist into data(ls_object).
  "kontrola, zda-li se jedna o programem podporovany typ objektu
  if line_exists( gt_cust[ objtype = ls_object-obj_type ] ).

    "dodatecna kontrola objektu nalezenych v paketu
    lv_exists = ''.
    read table gt_cust into data(wa_cust) with key objtype = ls_object-obj_type.

    if ls_object-obj_type = 'TABL'.
      select single tabclass from dd02l into lv_tabclass where tabname = ls_object-obj_name.

      "struktury jsou definovane jako TABL, musime preklopit na STRC
      if lv_tabclass = 'INTTAB'. "Struktura
        ls_object-obj_type = 'STRC'.
      endif.
    endif.
```

Obrázek 3.2: Komentáře v kódu v jazyce ABAP

3.1.3 Metadata

Metadata jsou informace o datech, které popisují nebo dodávají kontext k samotným datům. Jsou to tedy data o datech.

Metadata se často ukládají v samostatném souboru nebo v komentářích v kódu. V některých případech mohou být metadata také uložena v samotných datech, nebo v hlavičce souboru.

Použití metadat je velmi široké a může se lišit v závislosti na konkrétním programovacím jazyce nebo aplikaci.

Jako příklad si vezměme třídu v jazyce Java, tam se metadata ukládají do zvláštních souborů s příponou ".class", které obsahují kromě zkompilovaného kódu i informace o třídě v binární podobě.

Metadata třídy se často používají k automatizaci různých procesů, jako je například generování dokumentace, správa verzí nebo automatické získávání závislostí. Jejich použití tedy může usnadnit práci s danou třídou a zvýšit její přehlednost a uživatelský komfort.

3.2 Grafické formy dokumentace

3.2.1 UML

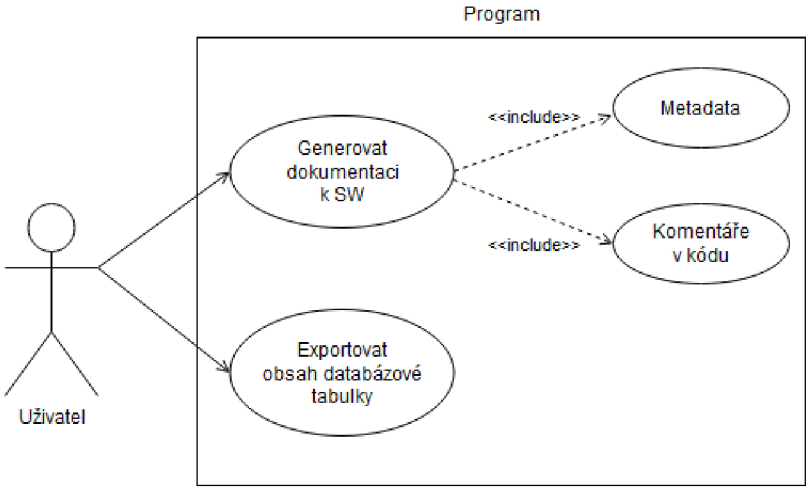
UML (Unified Modeling Language) je standardizovaný jazyk pro tvorbu modelů, který se používá při návrhu a analýze softwarových systémů. UML diagramy slouží ke grafickému znázornění různých aspektů softwarového systému, jako jsou třídy, objekty, interakce a chování.

Existuje mnoho různých typů UML diagramů, které lze použít k modelování různých aspektů softwarového systému. Mezi nejčastěji používané patří:

- Use Case diagram: popisuje interakci mezi uživatelem a systémem, používá se pro modelování požadavků na systém.
- Diagram tříd: zobrazuje třídy a jejich vztahy v systému.
- Diagram akcí: popisuje posloupnost akcí.
- Diagram stavů: popisuje chování objektů v různých stavech a přechody mezi nimi.

UML diagramy jsou velmi užitečné při návrhu a analýze softwarových systémů, protože umožňují zobrazit složité systémy jasně a srozumitelně. Můžou být také použity k vyjádření požadavků na systém a k popisu toho, jak má systém fungovat. [6]

Obrázek 3.3 ukazuje jednoduchý Use Case diagram, kdy uživatel chce po programu buď generovat dokumentaci k softwaru, která se skládá z metadat a z komentářů v kódu, nebo chce exportovat obsah databázové tabulky.



Obrázek 3.3: Ukázka Use Case diagramu

3.3 Nástroje pro tvorbu dokumentace

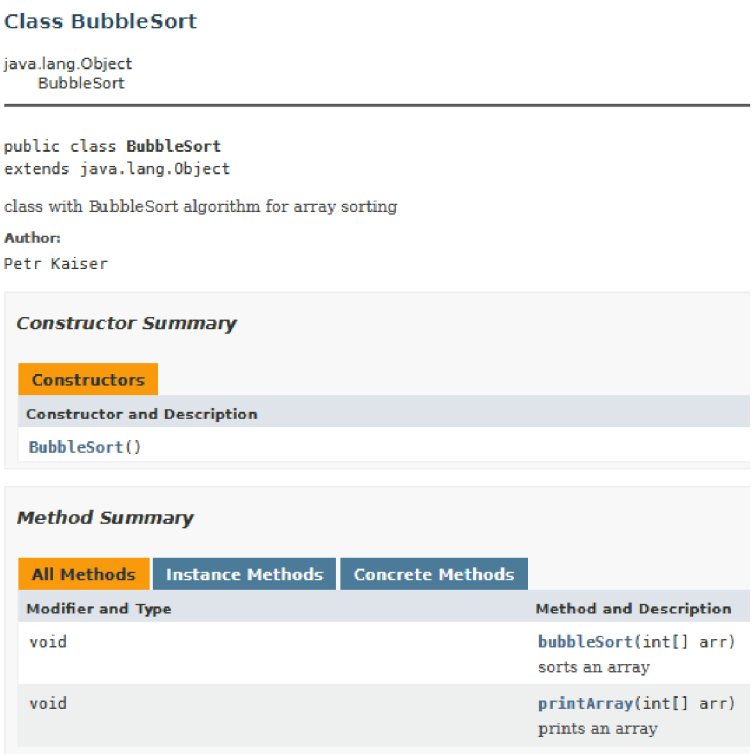
3.3.1 Javadoc

Javadoc je nástroj pro automatické generování dokumentace z kódu v programovacím jazyce Java.

Využívá metadata a bloky dokumentačních komentářů k vytvoření dokumentace ve formátu HTML, XML nebo RTF.

Javadoc je standardní nástroj pro dokumentování Java kódu a je součástí většiny vývojových prostředí pro jazyk Java.

Obrázek 3.4 ukazuje část dokumentace třídy ”BubbleSort”, jež byla vytvořena Javadocem. Konkrétně je vidět popisek třídy, konstruktor a následuje výčet metod třídy.



Class BubbleSort

java.lang.Object
BubbleSort

public class **BubbleSort**
extends java.lang.Object

class with BubbleSort algorithm for array sorting

Author:
Petr Kaiser

Constructor Summary

Constructors

Constructor and Description
BubbleSort()

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

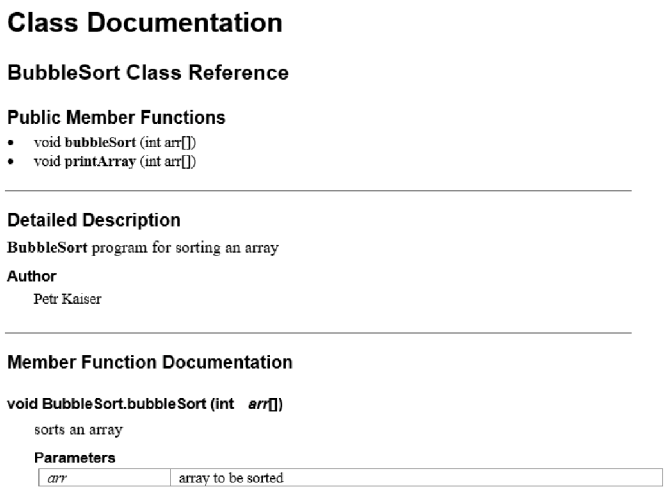
Modifier and Type	Method and Description
void	bubbleSort(int[] arr) sorts an array
void	printArray(int[] arr) prints an array

Obrázek 3.4: Výňatek z dokumentace třídy BubbleSort vytvořené pomocí Javadocu ve formátu HTML

3.3.2 Doxygen

Doxygen je univerzální nástroj pro tvorbu dokumentace v řadě jazyků (C, C++, Java, C#, Python, PHP, atd.). Obdobně jako Javadoc vygeneruje dokumentaci z metadat a komentářů v kódu. Oproti Javadocu nabízí více výstupních formátů (HTML, Latex, RTF, XML, PDF, atd.). Jedná se o open-source nástroj. [7]

Na obrázku 3.5 je vidět část dokumentace třídy "BubbleSort", jež byla vygenerována Doxygenem. Konkrétně výčet veřejných metod, popisek třídy a následuje detailní dokumentace metody "bubbleSort".



Class Documentation

BubbleSort Class Reference

Public Member Functions

- void **bubbleSort** (int arr[])
- void **printArray** (int arr[])

Detailed Description

BubbleSort program for sorting an array

Author

Petr Kaiser

Member Function Documentation

void BubbleSort.bubbleSort (int *arr*[])

sorts an array

Parameters

arr	array to be sorted

Obrázek 3.5: Výňatek z dokumentace třídy BubbleSort vytvořené pomocí Doxygenu ve formátu RTF

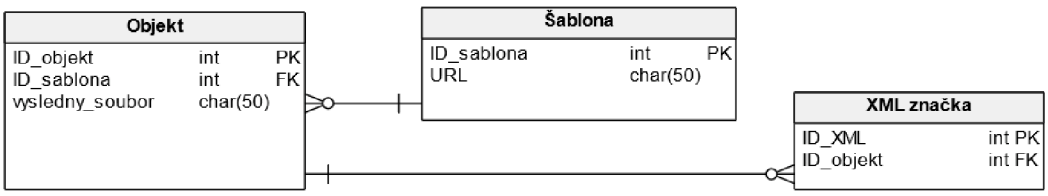
3.4 Dokumentace v databázových systémech

Dokumentace v databázových systémech by měla popisovat strukturu databáze, včetně všech relací, atributů a vazeb mezi nimi. Měla by také obsahovat informace o tom, jak se jednotlivé relace používají a jakým způsobem se data v nich ukládají a získávají.

Součástí dokumentace relačních databází obvykle bývá i relační schéma, které graficky znázorňuje jednotlivé relace, jejich atributy a vazby mezi relacemi.

K vytváření relačního schématu by mělo docházet už při návrhu databáze, to znamená ještě před jejím vytvořením, avšak vybrané programy pro správu databáze (MySQL Workbench, Microsoft SQL Server Management Studio, pHpMyAdmin, apod.) jsou schopny toto schéma vygenerovat zpětně i z produkční databáze.

Na obrázku 3.6 je zobrazeno relační schéma databáze, kterou využívá generátor dokumentace. Databáze se skládá ze tří relací: Objekt, Šablona, XML značka. Každý objekt musí mít šablonu, šablona může sloužit pro více objektů. Každý objekt může mít více XML značek, každá XML značka náleží právě jednomu objektu.



Obrázek 3.6: Relační schéma databáze

4 Dokumentace v systému SAP

4.1 Základní možnosti tvorby dokumentace

SAP nabízí následující možnosti tvorby dokumentace:

- komentáře v kódu,
- popisky objektů,
- ABAP dokumentace.

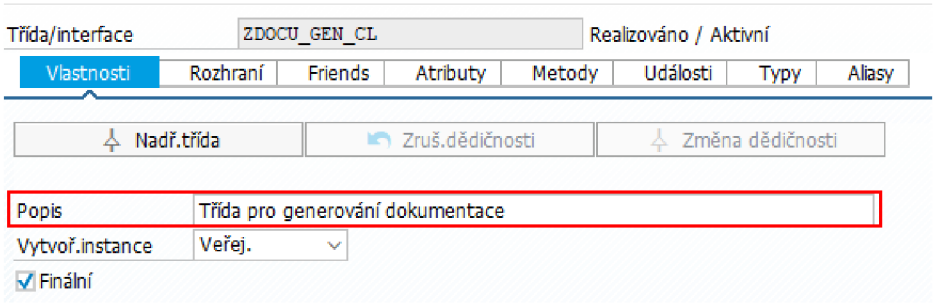
4.1.1 Komentáře v kódu

Jazyk ABAP má vyhrazené 2 znaky, které označují komentáře:

- * - nachází se na začátku řádku a označí celý řádek jako komentář,
- " - může být umístěn kdekoliv na řádku a označí všechen obsah řádku nacházející se za tímto znakem jako komentář.

4.1.2 Popisky objektů

Při vytváření nového objektu je možné vyplnit krátký popisek objektu, který ve stručnosti popisuje jeho funkci. Na obrázku 4.1 je znázorněno vyplňování popisku při vytváření třídy.

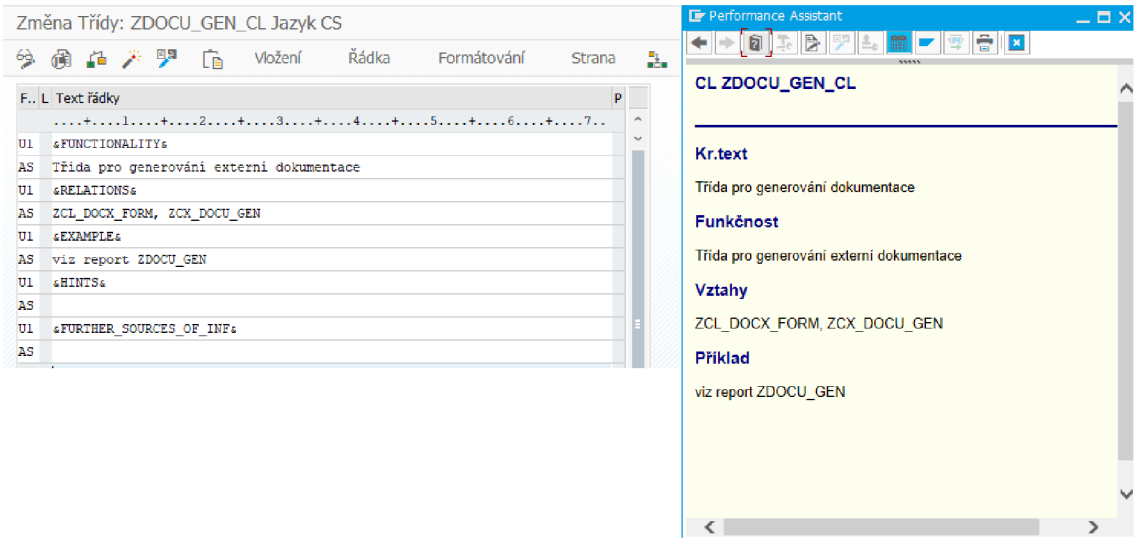


Obrázek 4.1: Krátký popisek třídy

4.1.3 ABAP dokumentace

ABAP dokumentací se rozumí dokumentační soubor, který musí uživatel pro daný objekt vytvořit a ručně vyplnit jeho obsah. Hodí se spíše pro obecný popis objektu. ABAP dokumentace je závislá na jazykové lokalizaci SAP GUI, pro každou jazykovou mutaci je potřeba sestavit tuto dokumentaci zvlášť.

Obrázek 4.2 nalevo zobrazuje textový editor, ve kterém je dokumentace vytvářena a napravo je poté vidět zobrazitelný výsledek. Takto vytvořenou dokumentaci lze exportovat do formátů RTF, ASCII, ITF a HTML.



Obrázek 4.2: ABAP dokumentace v režimu změny a zobrazení

4.2 Generátor diagramů tříd UML

SAP obsahuje systémový program **UML_CLASS_DIAGRAM**, který dovede generovat UML diagramy tříd (viz 3.2.1). Potřebuje k tomu mít nainstalovaný plugin JNet, jehož instalace není zcela triviální.

V případě, že JNet není v systému k dispozici, tak program dovede vyexportovat potřebná data k sestavení UML diagramu do souboru typu **XMI** (XML Metadata Interchange). To je formát, jenž je založen na formátu XML a slouží pro přenos metadat mezi programy. Tento soubor lze pak použít v různých programech, které slouží k tvorbě UML diagramů (např. Enterprise Architect) a pomocí něj si požadovaný UML diagram vytvořit.

Vedle generování UML dovede program také exportovat interaktivní dokumentaci ve formátu HTML. Výslednou formou se tato dokumentace velmi podobá Javadocu.

Obrázek 4.3 obsahuje ukázku dokumentace metody, vytvořenou tímto programem. V dokumentaci je možné vidět: základní vlastnosti metody (statická, privátní),

popisek metody, seznam parametrů včetně jejich datových prvků a popisků.

Operation: GET_TEMPLATE_URL

Properties	**Property**	**Value**	
	visibility	private	
	is static	true	
Parameter	**name**	**direction**	**type**
	IS_OBJECT	in	*TY_OBJECT_LIST*
	RV_URL	return	*ZDOCU_TMPLT_URL*
Documentation	**AbapDoc**		
	Vrací url šablonového souboru z konfigurační tabulky zdocu_tmplt		
	Parameter		
	IS_OBJECT		dokumentovaný objekt
	RV_URL		url šablony

Obrázek 4.3: Dokumentace metody vytvořená systémovým programem UML_CLASS_DIAGRAM

4.3 Obsah tabulky

K zobrazení obsahu databázových tabulek slouží Data Browser.Ten používá pokročilý ALV framework, který umožňuje přímou práci s obsahem databázové tabulky, aniž by uživatel musel psát selecty, inserty, apod. do databáze.

Data Browser také umožňuje exportovat obsah tabulky do textových formátů (.txt, .rtf, .html) a do MS Word nebo MS Excel.

Data Browser: Tabulka ZDOCU_OBJ 8 nal.objektů

Kontrolní tabulka...

MANDT	PGMID	OBJTYPE	ACTIVE	FILENAME	FILETYPE	TEMPLATE
001	LIMU	FUNC	X	ZE02_FUNC_<OBJ_NAME>	DOCX	
001	LIMU	METH	X	ZE02_METH_<OBJ_NAME>	DOCX	
001	R3TR	CLAS	X	ZE02_CLAS_<OBJ_NAME>	DOCX	ZE02_CLASS
001	R3TR	FUGR	X	ZE02_FUGR_<OBJ_NAME>	DOCX	

	A	B	C	D	E	F	G
1	MANDT	PGMID	OBJTYPE	ACTIVE	FILENAME	FILETYPE	TEMPLATE
2	001	LIMU	FUNC	X	ZE02_FUNC_<OBJ_NAME	DOCX	
3	001	LIMU	METH	X	ZE02_METH_<OBJ_NAME	DOCX	
4	001	R3TR	CLAS	X	ZE02_CLAS_<OBJ_NAME	DOCX	ZE02_CLASS
5	001	R3TR	FUGR	X	ZE02_FUGR_<OBJ_NAME	DOCX	

Obrázek 4.4: Obsah tabulky zobrazený v Data Browseru a vyexportovaný do MS Excel

5 Návrh vlastního nástroje pro generování dokumentace

5.1 Specifikace funkčních požadavků

5.1.1 Obecné

- **FP1** - Program bude generovat externí dokumentaci pro vybrané typy objektů.
  - Obsah dokumentace pro jednotlivé typy objektů je definován v podkapitole 5.1.3.
- **FP2** - Program bude ke generování dokumentace používat šablonové soubory ve formátu .docx.
- **FP3** - Uživatel může vytvářet a používat vlastní šablonové soubory.
- **FP4** - Program bude exportovat celý obsah databázové tabulky do externího souboru ve formátu .xlsx.

5.1.2 Grafické uživatelské rozhraní

- **FP5** - Uživatel si může vybrat z režimů pro generování dokumentace a pro export obsahu tabulky.
- **FP6** - Uživatel vybere dokumentovaný objekt a výstupní adresář.
- **FP7** - Uživatel může nahrát vlastní šablonu a použít ji.
- **FP8** - Uživatel může generovat dokumentaci pro více objektů stejného typu najednou.
- **FP9** - Uživatel může exportovat obsah více tabulek najednou.
- **FP10** - Uživatel může zvolit, zda-li chce dokumentovat třídu nebo funkční skupinu včetně detailní dokumentace metod nebo funkčních modulů.
- **FP11** - Uživatel může vybrat typy objektů obsažené v paketu nebo v transportním požadavku, pro které chce generovat dokumentaci.

5.1.3 Obsah dokumentace pro jednotlivé typy objektů

Dokumentace bude sestavena z metadat, popisků, ABAP dokumentace a u vybraných typů objektů i z komentářů v kódu daného objektu.

Typ objektu	Hlavička	Atributy	Metody/ funkce	Parametry	Datové prvky	Popisek	ABAP dok.	Vybrané komentáře
Třída	X	X	X		X	X	X	
Metoda	X			X	X	X	X	X
Funkční skupina	X		X			X	X	
Funkční modul	X			X	X	X	X	X
Program	X					X	X	X
Tabulka	X	X			X	X	X	

Tabulka 5.1: Obsah dokumentace pro jednotlivé typy objektů

Doplňující poznámky k obsahu dokumentace

- **Hlavička** bude obsahovat základní informace o dokumentovaném objektu (ID, název, typ, apod.).
- Atributy tříd a tabulek spolu s parametry metod a funkčních skupin mohou být určitého referenčního typu. Pokud je tento referenční typ **datovým prvkem**, tak program bude dokumentovat i tento datový prvek.

5.1.4 Export obsahu tabulky

Obsah databázové tabulky lze exportovat do souboru typu MS Excel pomocí Data Browseru (viz 4.3). Ovšem uživatel se k této možnosti musí v Data Browseru složitě proklikat.

Z praxe vychází, že zákazník mnohdy vedle dokumentace k programu požaduje i obsahy databázových tabulek. Cílem je tedy uživateli nabídnou rychlou možnost exportu obsahu celé databázové tabulky ve stejném programu, kde generuje dokumentaci a tím mu ušetřit čas.

5.2 Externí nástroje pro tvorbu dokumentace

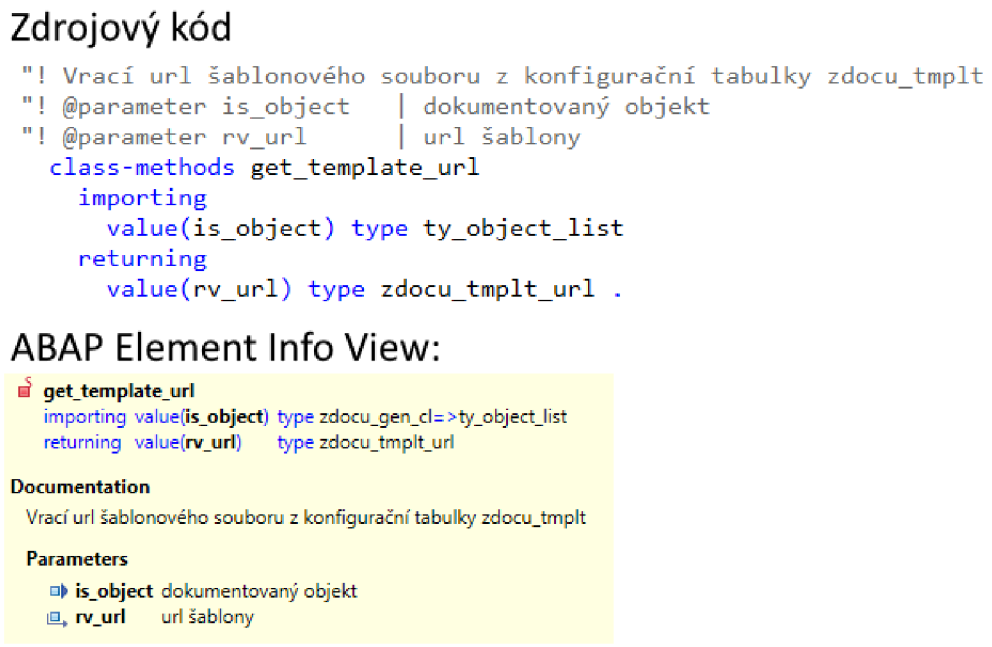
5.2.1 Vývojové prostředí Eclipse

Eclipse je open source vývojové prostředí primárně určené pro jazyk Java. Pomocí pluginů lze toto prostředí rozšířit i pro práci s jinými jazyky jako například C++, PHP, nebo právě SAP. [8]

Práce se SAPem v tomto prostředí se více podobá práci s klasickými programovacími jazyky, avšak zdaleka neobsahuje všechny možnosti, které nabízí standardní SAP GUI.

Oproti standardnímu SAP GUI nabízí Eclipse možnost práce s bloky dokumentačních komentářů (viz 3.1.2). Dokumentace se tedy vpisuje přímo do kódu k definici příslušného objektu za pomoci speciálně uvozených komentářů. Formát bloku komentářů je definován rozšířením SAPu do Eclipse. Tato dokumentace funguje v kódu pouze jako kontextová nápověda (obdobně jako Javadoc).

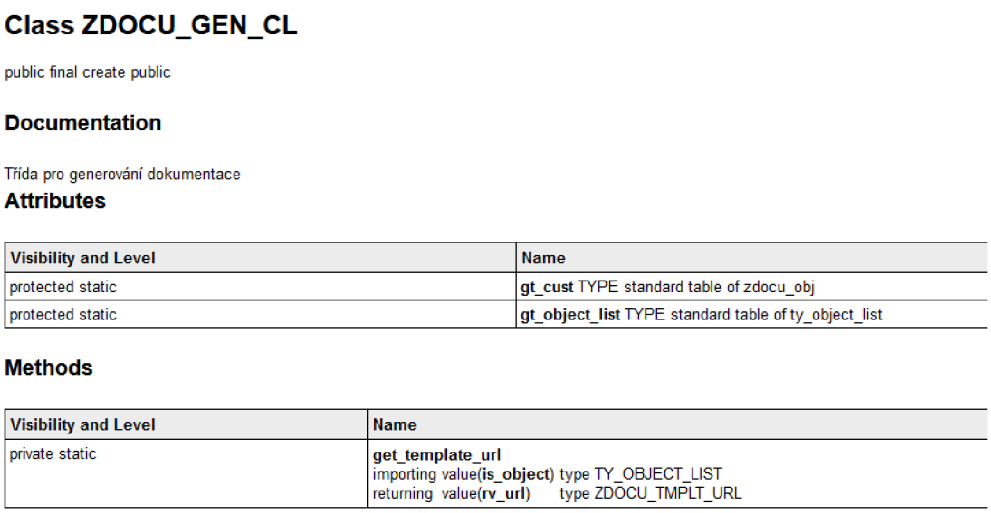
Obrázek 5.1 ukazuje dokumentaci metody v kódu a obsah této dokumentace zobrazený pomocí "ABAP Element Info View".



Obrázek 5.1: Dokumentace metody v prostředí Eclipse

Na úrovni tříd dovede Eclipse vyexportovat dokumentaci do formátu HTML. Pro každou třídu tak vznikne jeden statický HTML soubor (tj. bez křížových odkazů) s výčtem atributů a metod.

Výňatek z této dokumentace je zobrazen na obrázku 5.2. Ve spodní části obrázku v sekci "Methods" je vidět dokumentace metody popisovaná na obrázku 5.1.



Class ZDOCU_GEN_CL

public final create public

Documentation

Třída pro generování dokumentace

Attributes

Visibility and Level	Name
protected static	**gt_cust** TYPE standard table of zdocu_obj
protected static	**gt_object_list** TYPE standard table of ty_object_list

Methods

Visibility and Level	Name
private static	**get_template_url** importing value(**is_object**) type TY_OBJECT_LIST returning value(**rv_url**) type ZDOCU_TMPLT_URL

Obrázek 5.2: Výňatek z dokumentace třídy vytvořené prostředím Eclipse ve formátu HTML

5.2.2 PIKON - SAP ABAP Documentation Generator

Jedná se o placený produkt německé softwarové firmy PIKON, která se specializuje na SAP.

Funkcionality:

- Dokumentace ve formátu DOCX nebo XML.
- Vícejazyčné uživatelské rozhraní.
- Definování obsahu dokumentace (výběr typů objektů, pro které bude dokumentace sestavena).
- Generování dokumentace pro objekty paketu a transportního požadavku.

Firma neposkytuje žádnou veřejnou ukázku programu, seznam funkcionalit je převzat z popisu programu na jejich webových stránkách. [9]

Firma byla několikrát emailem oslovena o poskytnutí bližších informací o programu pro účely porovnání v této práci. Žádná odpověď od firmy bohužel nepřišla.

5.2.3 KCT Data ABAPDoc

Komplexní a zdarma dostupné řešení nabízí česká firma KCT Data.

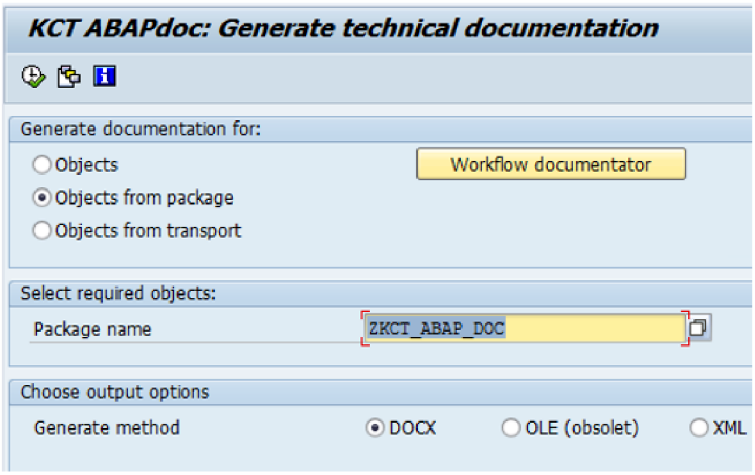
Funkcionality:

- Dokumentace ve formátu DOCX nebo XML.

- Definování obsahu dokumentace v programu (výběr typů objektů, pro které bude dokumentace sestavena).
- Generování dokumentace pro objekty paketu a transportního požadavku.
  - Program zobrazí celý obsah paketu/transportu a uživatel z něj vybere konkrétní objekty.
- V GUI lze definovat úvodní sekvenci znaků komentářů, jenž program vyexportuje.

Informace byly převzaty z [10].

Obrázek 5.3 zobrazuje část GUI programu, kde je vybráno generování dokumentace pro objekty paketu "ZKCT_ABAP_DOC" do formátu DOCX.



Obrázek 5.3: Výňatek z GUI programu KCT Data ABAPDoc

Obrázek 5.4 ukazuje dokumentaci metody "RENDER_ADD_INFO", jež byla vytvořena tímto programem. Dokumentace obsahuje popisek metody, tabulku s parametry a komentáře obsažené v kódu.

2.3.14 Method RENDER_ADD_INFO

Description: Creates Word document object

Method interface

Parameter id	Decription	Type	Pass value	Optional
IS_OBJECT_ALV	IS_OBJECT_ALV	Importing	No	No
IS_OUTPUT_OPTIONS	IS_OUTPUT_OPTIONS	Importing	No	No
IO_RENDER	IO_RENDER	Importing	No	No
EV_TEXT_ERROR	EV_TEXT_ERROR	Exporting	Yes	Yes
EF_RESULT	EF_RESULT	Exporting	Yes	Yes

Code comment

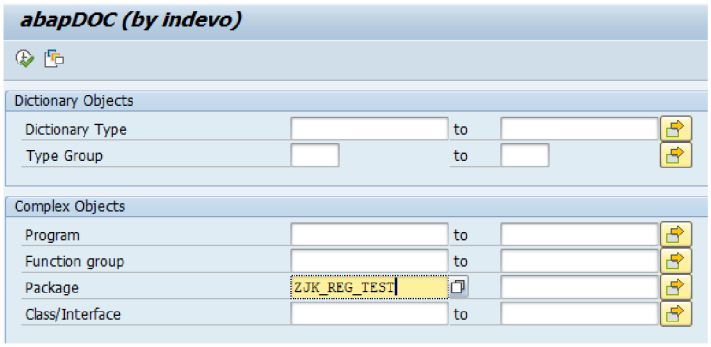
```
*& ---------------------------------------------------------------------
*&   Method RENDER_ADD_INFO
*& ---------------------------------------------------------------------
*&  Outputs the description, documentation,  superclass, attributes,
*&    method headers, method interfaces and code comments.
*&
*&  For inherited attribs and methods, we must climb to the superclass where
they are defined.
*&
*&  Uses following flags in is_output_options:
*&  - tech_docu (output content of "goto/documentation")
*&  - module_interfaces (output list of method parameters)
*&  - print_empty_par (output empty paragraphs in docu and in code comments)
*& ---------------------------------------------------------------------
```

Obrázek 5.4: Dokumentace metody ve formátu DOCX vytvořená tímto programem

5.2.4 indevo abapDOC

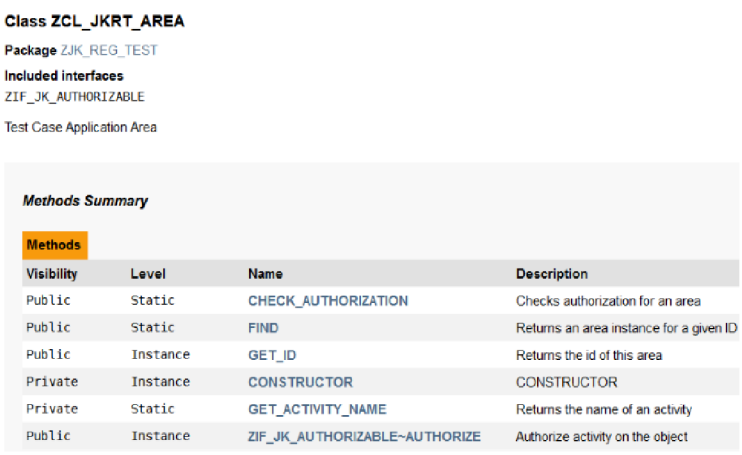
Zdarma dostupné řešení polské firmy indevo nabízí generování dokumentace ve formátu HTML a strukturou výsledného dokumentu připomíná dokumentaci vytvořenou Javadocem. [11]

Obrázek 5.5 ukazuje část GUI programu, kde uživatel vyplňuje názvy objektů, pro které chce generovat dokumentaci. Zde je konkrétně zadáno generování dokumentace pro paket ”ZJK_REG_TEST”.



Obrázek 5.5: GUI programu indevo abapDOC

Obrázek 5.6 zobrazuje výňatek z dokumentace třídy ”ZCL_JKRT_AREA” vytvořené tímto programem. Konkrétně je možné vidět paket, kterému třída náleží, popisek třídy a výčet metod třídy.



Class ZCL_JKRT_AREA

Package ZJK_REG_TEST

Included interfaces

ZIF_JK_AUTHORIZABLE

Test Case Application Area

Methods Summary

Methods

Visibility	Level	Name	Description
Public	Static	CHECK_AUTHORIZATION	Checks authorization for an area
Public	Static	FIND	Returns an area instance for a given ID
Public	Instance	GET_ID	Returns the id of this area
Private	Instance	CONSTRUCTOR	CONSTRUCTOR
Private	Static	GET_ACTIVITY_NAME	Returns the name of an activity
Public	Instance	ZIF_JK_AUTHORIZABLE~AUTHORIZE	Authorize activity on the object

Obrázek 5.6: Výňatek z dokumentace třídy vytvořené programem indevo abapDOC ve formátu HTML

5.2.5 SAP Customizing Documentation Generation Tool

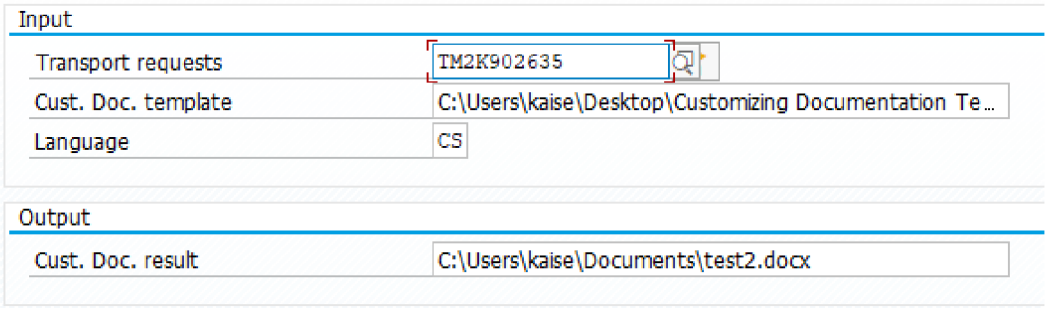
Zdarma dostupný nástroj, který generuje dokumentaci k tabulkám a exportuje jejich obsah ve formátu DOCX. Nástroj je omezen pouze na tabulky obsažené v transportním požadavku.

Program dovede pracovat s šablonou, která je k programu přiložena. V té si uživatel může upravit hlavičku s názvem firmy a logem. Vlastní šablonu uživatel vytvořit a použít nemůže.

Od ostatních nástrojů se liší možností vybrat jazykovou mutaci, ve které se vyexportují popisky.

Autor uvádí, že by program měl fungovat ve všech verzích SAPu od verze 7.0. [12] Z testování prováděného v této práci ovšem vyplynulo, že program není funkční ve verzi SAPu 7.4, v aktuální verzi 7.5 ovšem funguje.

Na obrázku 5.7 je vidět GUI tohoto programu, kde je v jednotlivých řádcích postupně zadáno číslo transportního požadavku, umístění šablonového souboru, jazykový klíč a umístění výsledného souboru.



Obrázek 5.7: GUI programu SAP Customizing Documentation Generation Tool

Obrázek 5.8 ukazuje obsah tabulky ”TFO05”, jenž byl vyexportován tímto programem. Pod názvy jednotlivých atributů jsou obsaženy i jejich popisky.

Table TFO05:

Table TFO05: SAP bar codes

TDBARCODE Název čárkového kódu SAP	TDTEXT Krátký text	TDBCMINC Minimální počet znaků v čárkovém kódu	TDBCMAXC Maximální počet znaků v čárkovém kódu	TDBCSIZE Šířka čárk.kódu	TDBCSIZEU Měrná jednotka pro šířku čárk.kódu	TDBCHEIGT Výška čárk.kódu	TDBCHEIGTU Měrná jednotka výšky čárkového kódu	TDBCUPWARD Zarovnání tisku čárk.kódu nahoru: ano/ne	TDBARCTYPE Technický typ čárkového kódu (evtl.skupiny čárk.kódů)
BARCLVS	Test Barcode im LVS	01	20	5.00	CM	2.00	CM		
BC_93	Code 93	01	40	7.00	CM	1.30	CM		C93
BC_C128B	Code 128 B, n.txt,h=13mm	01	40	9.00	CM	1.30	CM		

Obrázek 5.8: Výňatek z obsahu tabulky vyexportovaného nástrojem SAP Customizing Documentation Generation Tool ve formátu .docx

5.3 Porovnání specifikace vlastního programu s již existujícími řešeními

5.3.1 Use case

Uživatel chce:

U1 Generovat dokumentaci pro jednotlivé objekty zvlášť.

U2 Generovat dokumentaci pro všechny objekty paketu.

U3 Generovat dokumentaci pro objekty paketu dle vlastního výběru.

U4 Generovat dokumentaci pro všechny objektu transportního požadavku.

U5 Generovat dokumentaci pro objekty transportního požadavku dle vlastního výběru.

U6 Generovat dokumentaci pro více objektů stejného typu najednou.

U7 Generovat dokumentaci pro více objektů různých typů najednou.

U8 Vybrat konkrétní části obsahu výsledné dokumentace.

U9 Mít v dokumentaci obsažené jen vybrané komentáře z kódu.

U10 Generovat dokumentaci do externího souboru určitého formátu.

U11 Exportovat obsah tabulky.

U12 Použít vlastní šablonu.

Use case	**U1**	**U2**	**U3**	**U4**	**U5**	**U6**
Vlastní program	ano	ano	výběr podle typů	ano	výběr podle typů	ano
UML_CLASS_DIAGRAM	pouze třídy	pouze třídy	ne	ne	ne	ne
Eclipse	pouze třídy	pouze třídy	ne	ne	ne	ne
PIKON	ano	ano	nelze posoudit	ano	nelze posoudit	nelze posoudit
KCT data	ano	ano	výběr podle názvů	ano	výběr podle názvů	ano
indevo	ano	ano	ne	ano	ne	ano
SAP Customizing Documentation Generation Tool	ne	ne	ne	ano	ne	ne

Tabulka 5.2: Porovnání nástrojů na tvorbu externí dokumentace 1. část

Use case	**U7**	**U8**	**U9**	**U10**	**U11**	**U12**
Vlastní program	ne	ano (v šabloně)	ano (předdefinové typy)	.docx	ano	ano
UML_CLASS_DIAGRAM	ne	lze vybrat typy metod (private, protected, public)	ne	.html	ne	ne
Eclipse	ne	lze vybrat typy metod (private, protected, public)	ne	.html	ne	ne
PIKON	nelze posoudit	nelze posoudit	nelze posoudit	.docx, .xml	ne	ne
KCT data	ano	ano (v GUI)	ano (výběr vlastní úvodní sekvence v GUI)	.docx, .xml	ne	ne
indevo	ano	ne	ne	.html	ne	ne
SAP Customizing Documentation Generation Tool	ne	ne	ne	.docx	ne	částečně (lze upravit k programu přiloženou šablonu)

Tabulka 5.3: Porovnání nástrojů na tvorbu externí dokumentace 2. část

5.3.2 Výsledky porovnání

Z porovnání vychází, že nejblíže z hlediska funkcionalit má k programu, vyvíjenému v této práci, nástroj od KCT Data. Jeho hlavní výhodou je možnost výběru konkrétních objektů z paketu/transportu.

Největší výhodou programu, vyvíjeného v této práci, je možnost používat a vytvářet vlastní šablony.

Velmi užitečný výstup také nabízí systémový program UML_CLASS_DIAGRAM. Výsledný HTML soubor se velmi podobá výstupům z Javadocku. Nevýhodou je absence výběru konkrétních typů objektů.

Vzhledem k malému množství veřejně dostupných informací o nástroji od firmy PIKON nemá výsledek jeho porovnání s ostatními příliš vypovídající hodnotu. Jelikož se jedná o placený nástroj firmy, jež se specializuje na SAP, tak lze předpokládat, že se jedná o profesionální řešení s řadou funkcionalit.

6 Implementace vlastního nástroje pro generování dokumentace

Jednou z hlavních vlastností implementovaného programu je možnost používat vlastní šablonové soubory. Je tedy nutné zvolit vhodnou technologii jejich tvorby, aby program do nich dovedl správně vložit požadované data.

6.1 Výběr technologie tvorby šablon

Základem šablony jsou značky, jenž značí místa v dokumentu, která mají být nahrazena požadovaným obsahem. Existují dva způsoby, jak v MS Word vytvořit šablonu se značkami tak, aby s ní SAP dovedl pracovat:

- tvorba značek přímo v dokumentu,
- použití externího XML souboru se značkami.

6.1.1 Tvorba značek přímo v dokumentu

Uživatel vytváří značku přímo na požadovaném místě v dokumentu. Problematická je v tomto přístupu validace šablon, neboť značky, které uživatel v dokumentu vytvoří nemusí být kompatibilní se značkami, které program podporuje.

6.1.2 Použití externího XML souboru se značkami

XML

XML je standardizovaný jazyk pro přenos dat mezi programy. Zabývá se strukturou dat a jejich obsahem. Nezabývá se vzhledem, ten lze definovat například pomocí kaskádových stylů. Pomocí XSLT transformací je možné XML soubor transformovat do jiného typu souboru (např. HTML). [13]

XML soubor obsahuje hlavičku, definující verzi XML a kódování, pak následují jednotlivé elementy.

Obrázek 6.1 zobrazuje část obsahu XML souboru, konkrétně hlavičku a značku pro nadpis, ID , název a typ objektu a datum vytvoření dokumentačního souboru.

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="http://www.sap.com/SAPForm/0.5">

        <TITLE>%TITLE%</TITLE>

        <OBJ_ID>%OBJ_ID%</OBJ_ID>
        <OBJ_NAME>%OBJ_NAME%</OBJ_NAME>
        <OBJ_TYPE>%OBJ_TYPE%</OBJ_TYPE>
        <OBJ_DATE>%DATE%</OBJ_DATE>

</data>
```

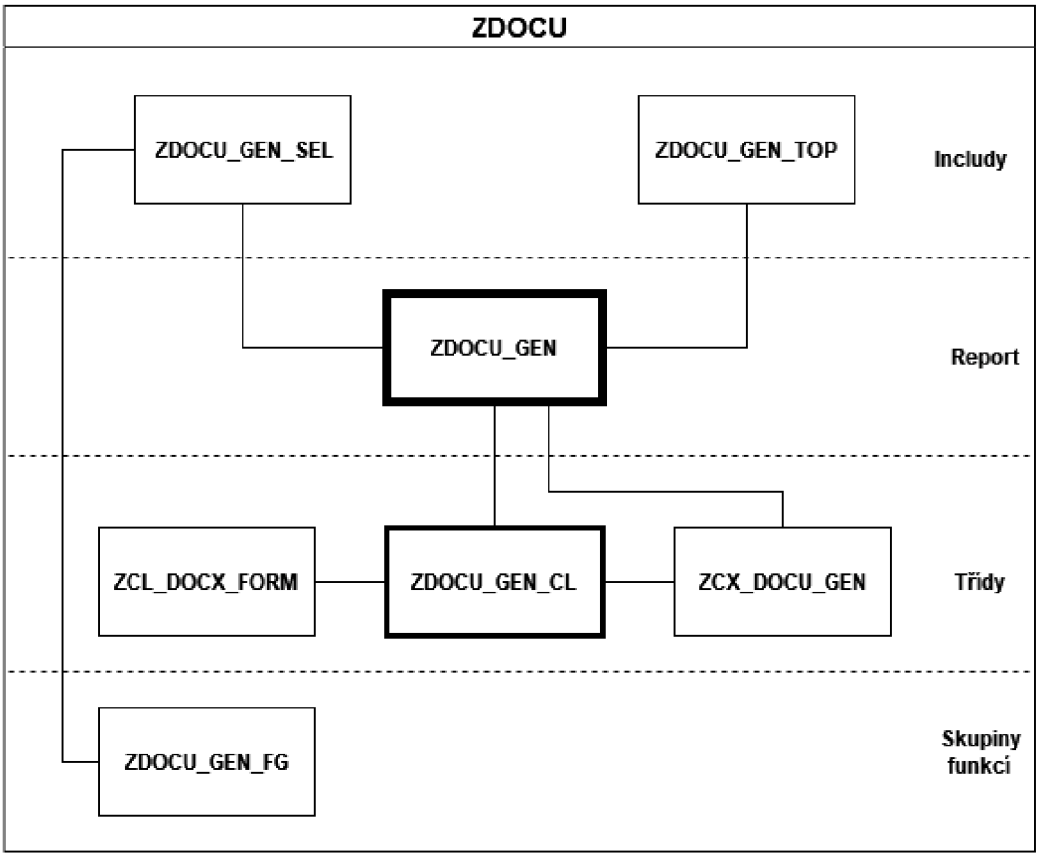
Obrázek 6.1: Obsah XML souboru

Uživatel si musí vytvořit značky v XML souboru, ten následně v režimu vývojáře v MS Word do šablonového dokumentu nahraje a vkládá do něj požadované XML značky.

Program využívá právě tento přístup, jelikož nabízí způsob, jak vytvářet validní a s programem kompatibilní šablony. K programu je dodána sada XML souborů s validními značkami, pokud uživatel použije značky pouze z těchto přiložených souborů, tak vždy vytvoří validní šablonu. Návod na tvorbu šablon je obsažen v příloze.

6.2 Objektový model

Celý program je zapouzdřen v paketu **ZDOCU**, jenž obsahuje 3 třídy, 1 skupinu funkcí, 1 report a 2 includy. Tento paket lze přidat do transportního požadavku a tím program distribuovat do dalších SAP systémů.



Obrázek 6.2: Objektový model programu

6.2.1 Třídy

- **ZCL_DOCX_FORM** - Upravená systémová třída CL_DOCX_FORM, jež obsahuje metody pro práci s dokumenty typu DOCX a XML. Základní systémovou třídu použít nelze, neboť ta neumožňuje práci s uživatelskými objekty (objekty v názvu začínající písmenem Z).
- **ZDOCU_GEN_CL** - Hlavní třída celého programu, obsahuje metody pro:
  - získání dat o objektech,
  - zpracování dat,
  - přidružení XML značek ke zpracovaným datům,
  - vypsání výsledné statistiky.

  Volá metody třídy ZCL_DOCX_FORM pro:
  - vložení dat do šablony,
  - vygenerování výsledného dokumentu.

- **ZCX_DOCU_GEN** - Specifická třída obsahující výjimky dle standardu SAPu, metody dědí ze standardní systémové třídy pro výjimky CX_ROOT.

6.2.2 Skupiny funkcí

Skupina funkcí **ZDOCU_GEN_FG** obsahuje funkční modul, který slouží k nahrání nové šablony do MIME repozitáře.

6.2.3 Includy

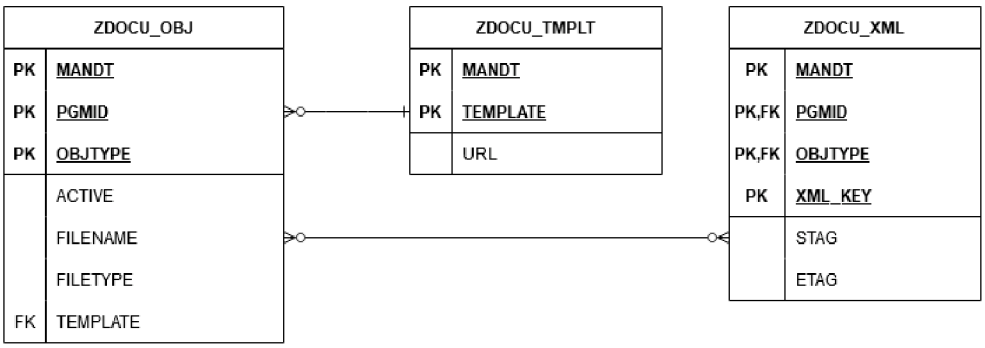
- **ZDOCU_GEN_TOP** - obsahuje deklarace globálních proměnných programu.
- **ZDOCU_GEN_SEL** - definuje výběrové prvky GUI (parametry, select-options), definuje obsah globálních proměnných programu na základě výběru objektů v GUI uživatelem, pro nahrání šablony využívá funkční modul ze skupiny funkcí ZDOCU_GEN_FG.

6.2.4 Report

Report **ZDOCU_GEN** je spustitelná část programu. Obsahuje includy ZDOCU_GEN_TOP a ZDOCU_GEN_SEL. Volá metody třídy ZDOCU_GEN_CL, které zajišťují průběh programu (viz 6.4).

6.3 Datový model

K uchování interních dat používá program 3 databázové tabulky.



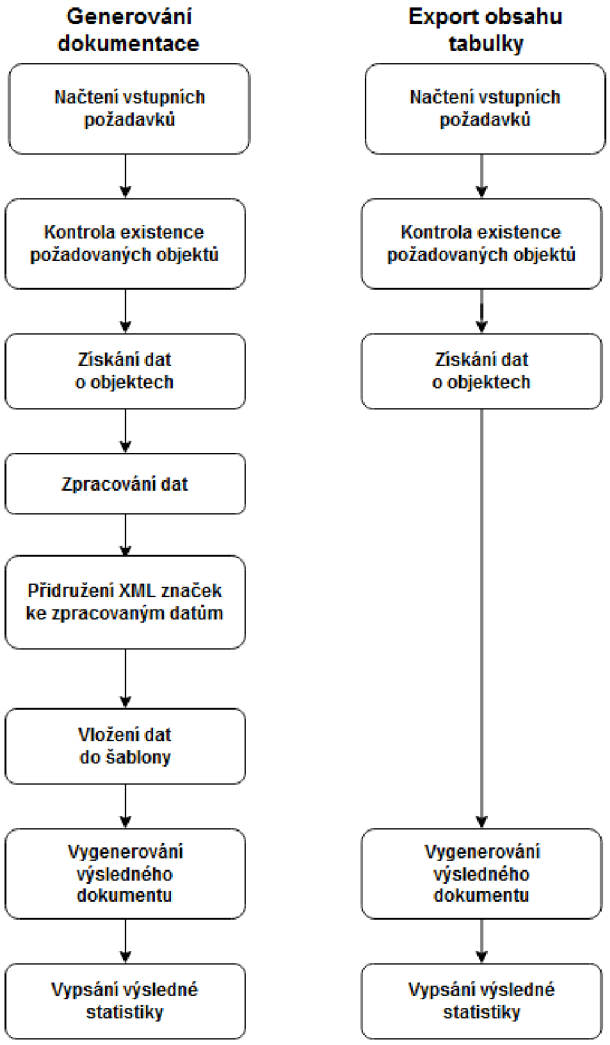
Obrázek 6.3: Datový model programu

- **ZDOCU_OBJ** - Tabulka, která slouží k uchování konkrétního nastavení generování pro jednotlivé typy objektů (viz 6.5.1).

- **ZDOCU_TMPLT** - Tabulka obsahující odkazy na šablony v MIME repozitáři. Každý typ objektu musí obsahovat jednu šablonu, jedna šablona může sloužit pro více typů objektů.
- **ZDOCU_XML** - Databázová tabulka XML značek, každý typ objektu může obsahovat více XML značek, každá značka může náležet více typům objektů.

6.4 Průběh programu

Průběh programu se liší pro režim generování dokumentace a pro režim exportu obsahu tabulky. Obrázek 6.4 zobrazuje vykonávané kroky jednotlivých režimů. Detailní průběh jednotlivých kroků je popsán dále v této podkapitole.



Obrázek 6.4: Průběh programu pro jednotlivé režimy

6.4.1 Načtení vstupních požadavků

Program načte vstupní parametry z GUI a uloží je do globálních proměnných programu.

6.4.2 Kontrola existence požadovaných objektů

Ve chvíli, kdy program získá názvy požadovaných objektů, je nutné zkontrolovat, zda-li tyto objekty v systému vůbec existují. Ke kontrole existence elementárních objektů slouží funkční modul `TR_CHECK_EXIST` . Ke kontrole existence paketu slouží funkční modul `TR_DEVCLASS_GET` . Ke kontrole existence transportního požadavku slouží tabulka `E070` , jenž obsahuje informace o všech transportních požadavcích uložených v systému.

6.4.3 Načtení objektů z paketu a z transportního požadavku

V případě paketu a transportního požadavku je po kontrole jejich existence ještě potřeba načíst jejich objekty. Program načte pouze ty typy objektů, které má uživatel v nastavení označené jako "aktivní".

K načtení objektů z paketu slouží funkční modul `RS_GET_OBJECTS_OF_DEVCLASS` .

K načtení objektů z transportního požadavku slouží funkční modul `TR_GET_OBJECTS_OF_REQ_AN_TASKS` .

6.4.4 Zisk dat o objektech

Daty se rozumí ta data o objektech, ze kterých je vytvářena dokumentace. Konkrétně:

- metadata,
- popisky,
- ABAP dokumentace,
- zdrojový kód pro zisk vybraných komentářů.

Data o objektech si SAP ukládá v systému. Cílem je tato data vyhledat a získat. Pro jednotlivé typy objektů jsou data ukládána na různých místech systému a jejich způsob zisku se proto liší. Data lze získat:

- přímo ze systémových databázových tabulek,
- pomocí systémových příkazů,
- pomocí systémových funkčních modulů,
- pomocí systémových tříd a jejich metod.

Třída

K zisku dat o třídě slouží:

Získaná informace	Třída	Metoda	Funkční modul	Tabulka
Atributy				SEOCOMPODF
Metody	CL_ABAP_OBJECTDESCR	DESCRIBE_BY_NAME		
Datové prvky				DD04L
Zdrojový kód	CL_RECA_RS_SERVICES	GET_SOURCE		
Popisek			SEO_CLASS_READ	
ABAP dokumentace			DOCU_READ	
Popisky atributů				SEOCOMPOTX
Popisky datových prvků				DD04T

Tabulka 6.1: Způsoby zisku informací o třídě

Způsoby zisku dat o dalších typech objektů jsou popsány v příloze.

Obsah tabulky

K získání celého obsahu tabulky se využívá SQL příkaz `SELECT * FROM <tab>` .

Popisky objektů a ABAP dokumentace jsou závislé na konkrétní jazykové mutaci. To znamená, že určitý objekt může obsahovat pro každý jazyk jiný popisek nebo ABAP dokumentaci. To je také důvod, proč jsou popisky objektů uložené v jiné databázové tabulce, než zbylé informace o objektu. Program tato data načítá v takovém jazyce, v jakém má uživatel spuštěného SAP klienta.

6.4.5 Zpracování dat

Z vybraných získaných dat(zdrojový kód objektu, ABAP dokumentace) je potřeba vybrat pouze tu část, která se hodí do výsledné dokumentace.

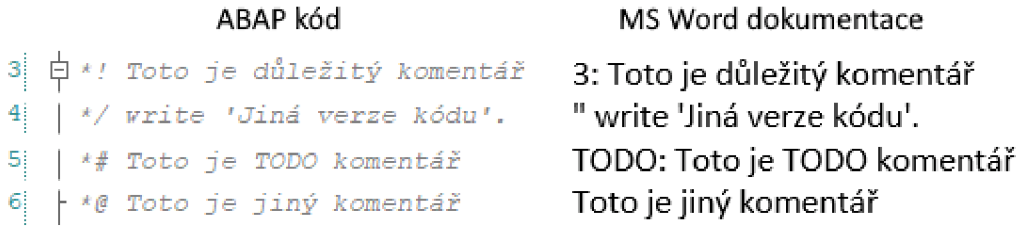
Komentáře v kódu

Ze zdrojového kódu vybraných objektů(metoda, funkční modul, report) dochází k vyparsování komentářů. Jazyk ABAP má rezervované 2 znaky pro uvození komentáře viz. 4.1.1. Tento program rozeznává 4 vlastní typy komentářů uvozené vybranými sekvencemi 2 znaků:

- `*!` **Důležitý komentář** - výsledná dokumentace obsahuje číslo řádku komentáře v kódu a jeho text
- `*/` **Jiná verze kódu** - možnost obsáhnout zakomentovanou část kódu do dokumentace, v dokumentaci je tento typ komentáře uvozen znakem ”, neboť

tento znak také uvozuje komentář a při vložení této sekce z dokumentace zpět do kódu se bude zakomentovaný kód opět chovat jako komentář a pomocí klávesové zkratky pro odkomentování lze tuto sekci jednoduše odkomentovat

- `*#` **TODO komentář** - ve výsledné dokumentaci je před textem komentáře obsažena sekvence *TODO:*
- `*@` **Jiný komentář** - výsledná dokumentace obsahuje pouze text komentáře



Obrázek 6.5: Typy komentářů

Pokud programátor chce, aby jeho komentář byl obsažen ve výsledné dokumentaci, tak jej musí uvést právě jednou z těchto sekvencí, na jinak uvozené komentáře program nereaguje. Díky tomuto přístupu má uživatel mnohem lepší kontrolu nad obsahem dokumentace a dokumentace tak obsahuje skutečně jen důležité informace.

Program nepotřebuje k sestavení dokumentace bloky s dokumentačními komentáři (viz 3.1.2), neboť si požadovaná data o objektech dohledá v databázi.

ABAP dokumentace

Z obsahu ABAP dokumentace je potřeba odstranit znaky, jenž ABAP do dokumentace vkládá, aby ji v režimu zobrazení zobrazoval v čitelném formátu.

Speciální znaky jazyka XML

Při zpracování dat je také potřeba v textových datech ošetřit (escapovat) speciální znaky jazyka XML. Při neošetření by nedošlo ke správnému přenosu dat do MS Word, neboť XML parser by považoval daný speciální znak jako XML značku, ne jako součást textové informace. Toto ošetření znaků řeší v ABAPu systémový příkaz `escape` .

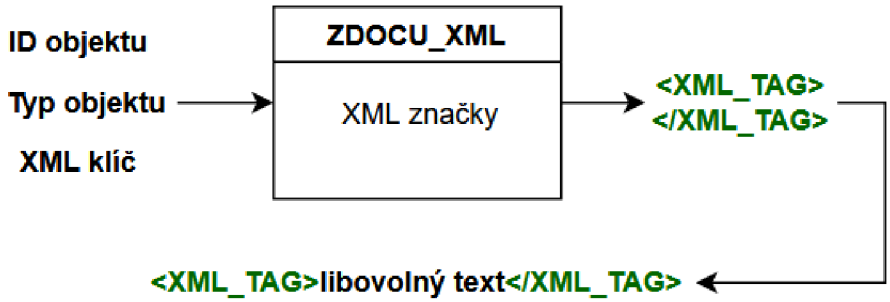
Speciální znak	Escape sekvence
< (menší než)	< nebo <
> (větší než)	> nebo >
& (ampersand)	&
’ (jednoduchá uvozovka)	'
” (dvojitá uvozovka)	"

Tabulka 6.2: Speciální znaky jazyka XML [14]

6.4.6 Přidružení XML značek ke zpracovaným datům

Ke zpracovaným datům je potřeba přidružit XML značky, jenž jsou uloženy v databázové tabulce `ZDOCU_XML` , aby se data vložila do šablony na správná místa. Program se na základě id objektu, typu objektu a XML klíče dané XML značky dotazuje do databáze, ta programu navrátí odpovídající otevírací a uzavírací XML značku.

Otevírací XML značka je poté vložena před text, uzavírací značka je vložena za text. Tento proces znázorňuje obrázek 6.6. Takto upravený text je možné vložit do šablony.

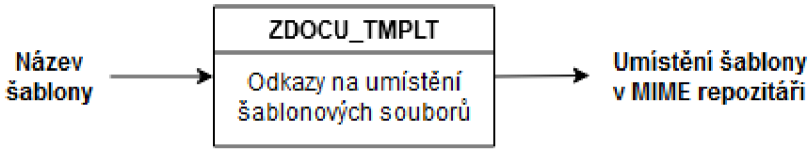


Obrázek 6.6: Zisk XML značek z databáze a jejich přidružení k textu

6.4.7 Vložení dat do šablony

Šablonové soubory jsou uloženy v MIME repozitáři, jenž v SAPu slouží jako úložiště různých souborů. Databázová tabulka `ZDOCU_TMPLT` obsahuje názvy šablon a odkazy na jejich umístění v MIME repozitáři.

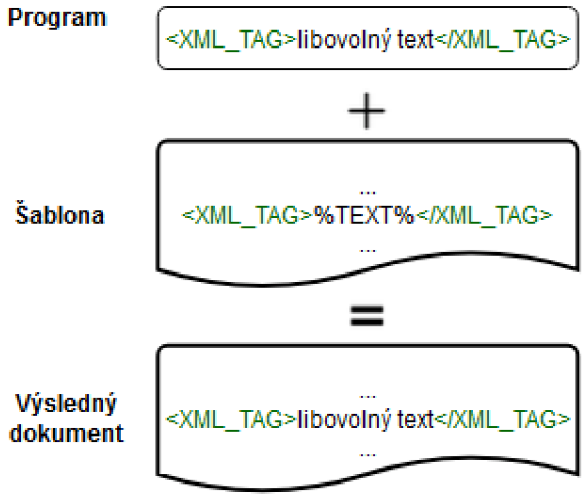
Program se na základě názvu šablony, jež je aktuálně přidělena konkrétnímu typu objektu dotáže na umístění šablonového souboru.



Obrázek 6.7: Zisk umístění šablony v MIME repozitáři

Tento šablonový soubor si pomocí systémové třídy `CL_DOCX_DOCUMENT` a její metody `LOAD_DOCUMENT` načte. Pomocí metody `MERGE_DATA` třídy `ZCL_DOCX_FORM` dojde k vložení dat, získaných v předchozím kroku, do šablony.

Spojení šablonového souboru s konkrétním obsahem znázorňuje obrázek 6.8.



Obrázek 6.8: Vložení výsledného obsahu do šablony

6.4.8 Vygenerování výsledného dokumentu

K vytvoření výsledného souboru je nejprve potřeba převést textová data do binární podoby. K tomu slouží funkční modul `SCMS_XSTRING_TO_BINARY` . Následně metoda `GUI_DOWNLOAD` systémové třídy `CL_GUI_FRONTEND_SERVICES` vytvoří z těchto binárních dat výsledný soubor požadovaného formátu. K vytvoření výsledného dokumentu typu MS Word slouží metody třídy `ZCL_DOCX_FORM`

Výsledný dokument je vygenerován do adresáře, jenž uživatel vybral na výběrové obrazovce.

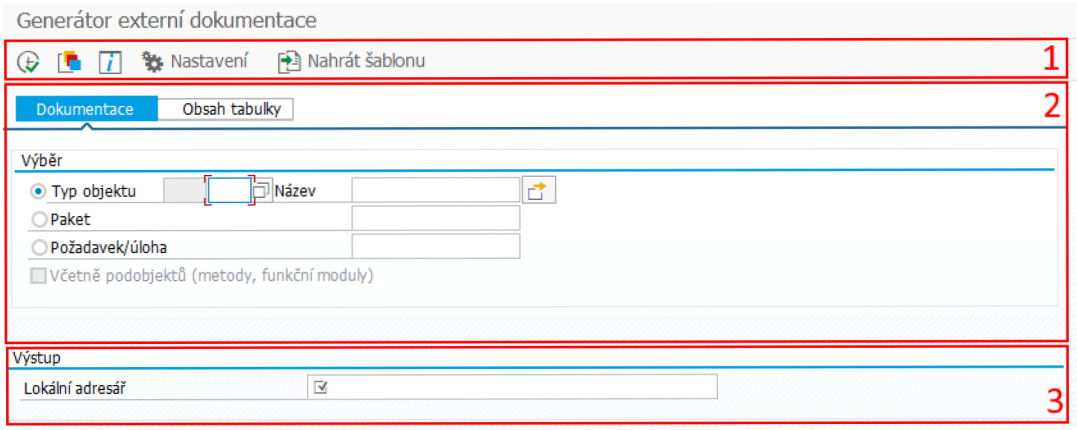
6.4.9 Výpis výsledné statistiky

Po vytvoření všech výsledných souborů ve uživateli zobrazí výsledná statistika se seznamem vytvořených souborů (viz 6.5.1).

6.5 Grafické uživatelské rozhraní

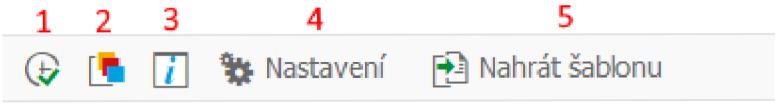
Grafické uživatelské rozhraní se skládá ze tří oblastí:

1. Hlavička - obsahuje tlačítka pro správu programu.
2. Aktuální režim - generování dokumentace nebo export obsahu tabulky.
3. Výběr výstupního adresáře.



Obrázek 6.9: Základní oblasti GUI

6.5.1 Hlavička



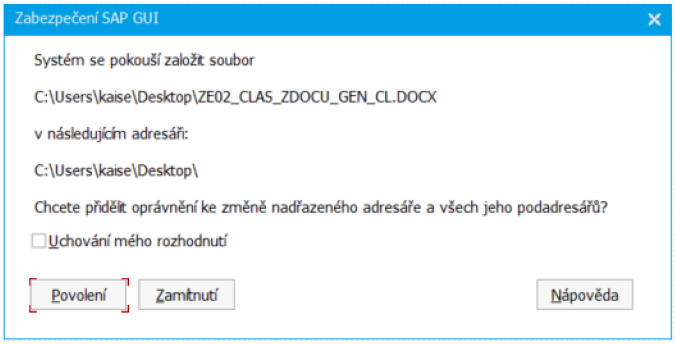
Obrázek 6.10: Hlavička GUI

1. Provedení programu
2. Vyvolání varianty
3. Zobrazení programové dokumentace
4. Nastavení programu
5. Nahrání šablony

Provedení programu

Provedením programu se rozumí spuštění procesu vytváření výsledného souboru nebo souborů na základě aktuálního režimu a vstupních parametrů.

Při tomto kroku zobrazí SAP uživateli okno, ve kterém ho žádá o povolení vytvořit výsledný soubor na disku (viz obrázek 6.11). Tento krok je nutné odsouhlasit stiskem tlačítka "Povolení". V jiném případě program vyhodí chybovou hlášku a výsledný soubor se nevytvoří.



Obrázek 6.11: Žádost o povolení vytvořit výsledný soubor

Po vytvoření všech výsledných souborů ve uživateli vypíše výsledná statistika. Obrázek 6.12 znázorňuje výslednou statistiky pro objekty paketu "ZDOCU".

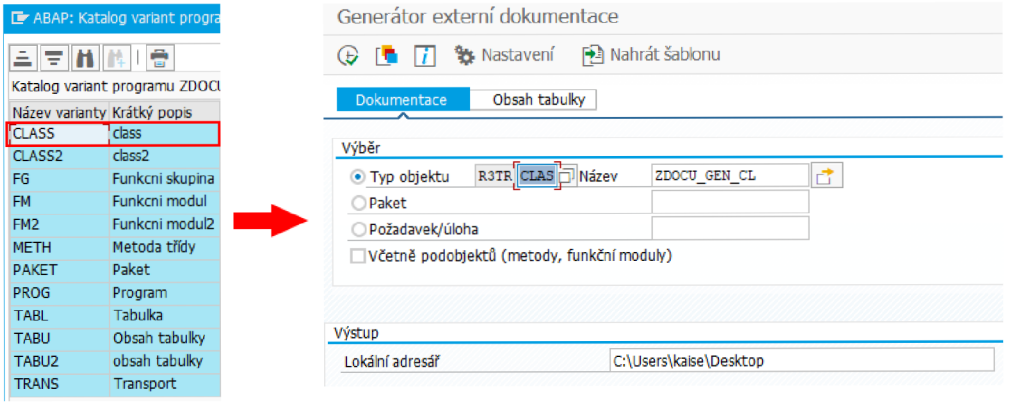
```
Výsledný seznam objektů:

R3TR CLAS ZCL_DOCX_FORM                 \ZE02_CLAS_ZCL_DOCX_FORM.DOCX
R3TR CLAS ZCX_DOCU_GEN                  \ZE02_CLAS_ZCX_DOCU_GEN.DOCX
R3TR CLAS ZDOCU_GEN_CL                  \ZE02_CLAS_ZDOCU_GEN_CL.DOCX
R3TR CLAS ZDOCU_GEN_CL_ORIG             \ZE02_CLAS_ZDOCU_GEN_CL_ORIG.DOCX
R3TR FUGR ZDOCU_GEN_FG                  \ZE02_FUGR_ZDOCU_GEN_FG.DOCX
R3TR FUGR ZDOCU_MAINT                   \ZE02_FUGR_ZDOCU_MAINT.DOCX
R3TR PROG ZDOCU_GEN                     \ZE02_PROG_ZDOCU_GEN.DOCX
R3TR PROG ZDOCU_GEN_CL_GENERATE_MS_WORD \ZE02_PROG_ZDOCU_GEN_CL_GENERATE_MS_WORD.DOCX
R3TR PROG ZDOCU_GEN_GENERATE_MS_WORD    \ZE02_PROG_ZDOCU_GEN_GENERATE_MS_WORD.DOCX
R3TR PROG ZDOCU_GEN_SEL                 \ZE02_PROG_ZDOCU_GEN_SEL.DOCX
R3TR PROG ZDOCU_GEN_TOP                 \ZE02_PROG_ZDOCU_GEN_TOP.DOCX
R3TR PROG ZDOCX_XML_TEST                \ZE02_PROG_ZDOCX_XML_TEST.DOCX
R3TR TABL ZDOCU_MAP                     \ZE02_TABL_ZDOCU_MAP.DOCX
R3TR TABL ZDOCU_OBJ                     \ZE02_TABL_ZDOCU_OBJ.DOCX
R3TR TABL ZDOCU_TMPLT                   \ZE02_TABL_ZDOCU_TMPLT.DOCX
R3TR STRC ZINTFSTRUC                    \ZE02_STRC_ZINTFSTRUC.DOCX
R3TR STRC ZSCTSOBJECT                   \ZE02_STRC_ZSCTSOBJECT.DOCX
```

Obrázek 6.12: Výpis objektů obsažených v paketu a k nim vytvořené soubory

Vyvolání varianty

Nastavení vstupních požadavků je možné uložit v podobě varianty. Tuto variantu lze při dalších spuštěních programu vyvolávat a GUI se uživateli vyplní.



Obrázek 6.13: Vyvolání uložené varianty pro třídu

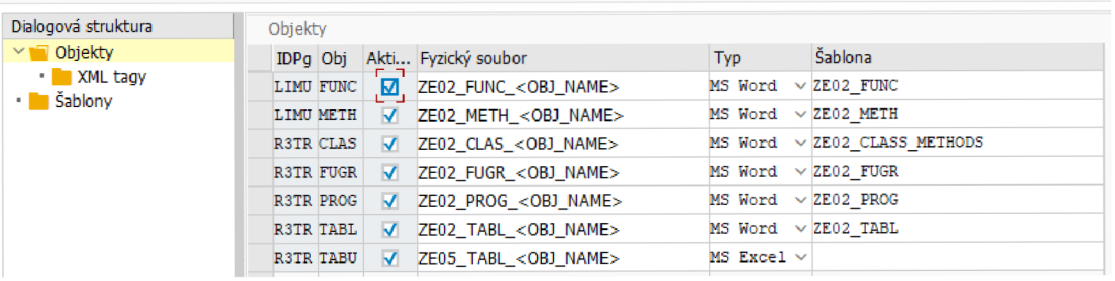
Zobrazení programové dokumentace

Program uživateli zobrazí jeho ABAP dokumentaci (viz 4.1.3).

Nastavení programu

Program se skládá ze 3 databázových tabulek (viz 6.3). Do těchto databázových tabulek je v nastavení zřízen pohled pro práci s jejich obsahem.

- **Objekty** - slouží ke správě obsahu databázové tabulky `ZDOCU_OBJ`



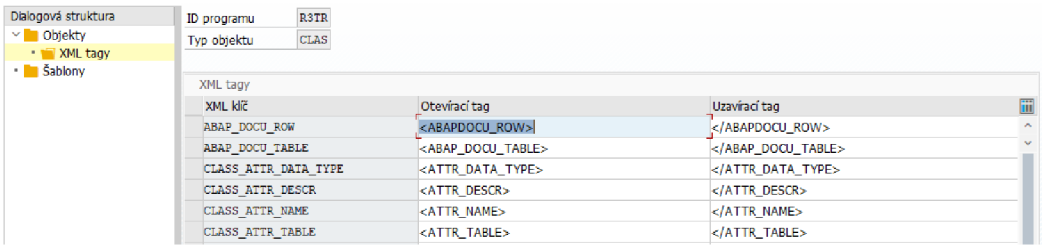
Obrázek 6.14: Konfigurační tabulka pro jednotlivé typy objektů

Pro jednotlivé typy objektů se zde nastavuje:

- **IDPg** - ID programu v transportním požadavku.
- **Obj** - typ objektu.
- **Aktivní** - rozhodnutí o tom, zda-li tento typ objektu bude zahrnut v seznamu generovaných objektů.
- **Fyzický soubor** - název výsledného souboru lze zvolit libovolný, pro obsažení názvu objektu v názvu souboru je potřeba do názvu souboru zahrnout <OBJ_NAME>.

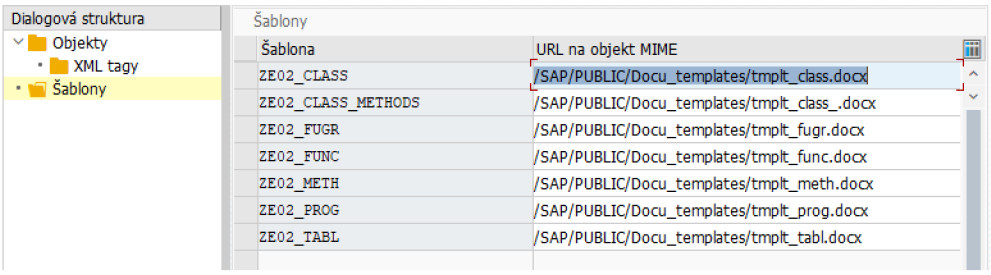
  - **Typ** - typ výsledného souboru (MS Word/MS Excel).
  - **Šablona** - název šablony, výběr je omezen pouze na šablony obsažené v konfigurační tabulce šablon.

- **XML tagy** - slouží ke správě obsahu databázové tabulky `ZDOCU_XML` , každý typ objektu má vlastní sadu XML značek, když uživatel vybere v nastavení objektů určitý typ objektu (označí jeho řádek), tak si může zobrazit sadu jeho XML značek.



Obrázek 6.15: Výňatek z XML značek třídy

- **Šablony** - slouží ke správě obsahu databázové tabulky `ZDOCU_TMPLT` , po nahrání nové šablony do programu se provede záznam do této databázové tabulky a uživatel si zde může provedené změny ověřit.



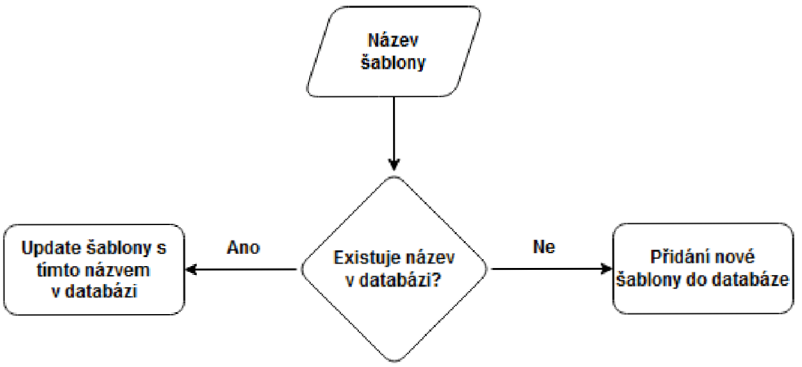
Obrázek 6.16: Tabulka šablon

Nahrání šablony

Nahrání šablony se skládá ze dvou kroků:

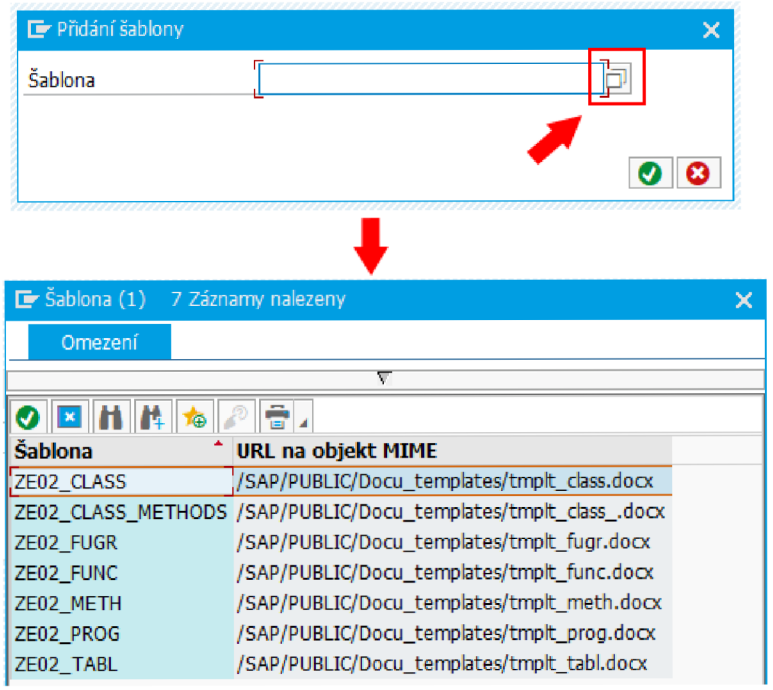
1. Výběr šablonového souboru - uživateli se zobrazí průzkumník souborů, v němž vybere požadovaný šablonový soubor a ten se nahraje do MIME repozitáře.

2. Zadání názvu šablony - uživateli se zobrazí okno, do kterého vyplní název šablony a dále se proces řídí následujícím diagramem:



Obrázek 6.17: Přidání/update záznamu v databázi šablon

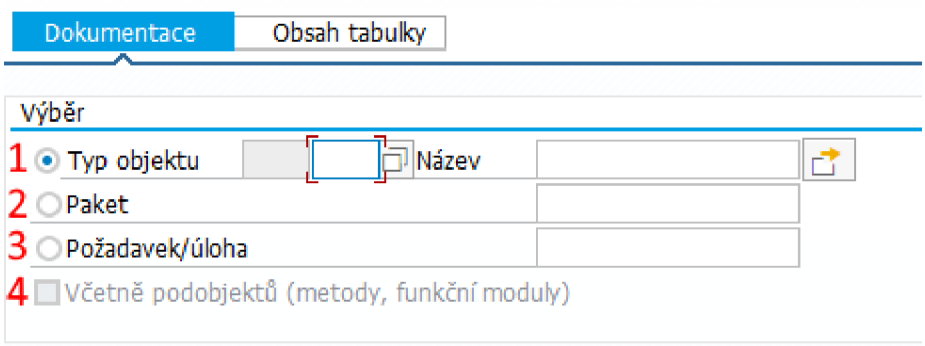
Při zadávání názvu šablony může uživatel využít tlačítko pro nápovědu vyhledávání, jež mu zobrazí seznam všech již existujících šablon. Z nich může uživatel vybrat existující šablonu pro update v databázi.



Obrázek 6.18: Zobrazení existujících šablon

6.5.2 Režimy GUI

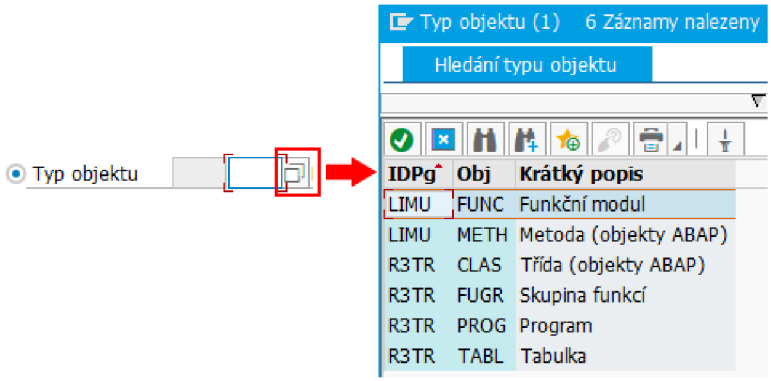
Generování dokumentace



Obrázek 6.19: Režim pro generování dokumentace

Generování dokumentace pro:

1. jednotlivé objekty - uživatel vybere typ objektu, může si zobrazit seznam podporovaných objektů a z něj vybrat typ objektu,



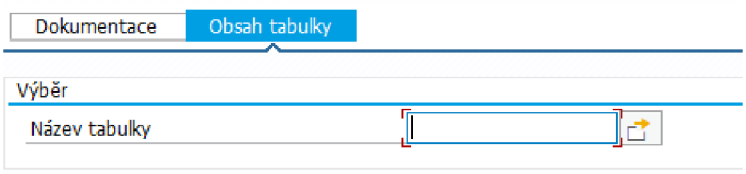
Obrázek 6.20: Výběr z dostupných typů objektů

   následně zadá název objektu, zde má možnost vícenásobného výběru pro zadání více objektu stejného typu,

2. paket - uživatel zadá název paketu,
3. transportní požadavek - uživatel zadá číslo transportního požadavku,
4. třídu včetně metod nebo funkční skupinu včetně funkčních modulů.

Export obsahu tabulky

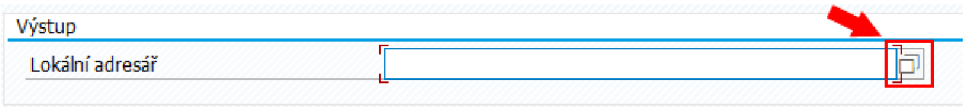
Pro export obsahu tabulky musí uživatel zadat název databázové tabulky. Pro export obsahů více tabulek najednou může využít tlačítko pro vícenásobný výběr.



Obrázek 6.21: Zadání názvu tabulky

6.5.3 Výběr výstupního adresáře

Uživatel vybere lokální adresář, do kterého se budou vytvářet veškeré soubory. Při nevyplnění tohoto pole se použije výchozí adresář SAP GUI. V případě stisku červeně vyznačeného tlačítka se zobrazí průzkumník souborů pro výběr výstupního adresáře.



Obrázek 6.22: Výběr výstupního adresáře

6.6 Příklad výsledné dokumentace

Jako příklad dokumentace, která byla vytvořena tímto programem si vezměme dokumentaci metody ”GET_TEMPLATE_URL”. Dokumentace obsahuje:

- název metody,
- základní vlastnosti metody(privátní, statická),
- krátký popis metody,
- tabulku s výčtem parametrů,
- tabulku s globálními datovými prvky parametrů,
- ABAP dokumentaci - pro tuto metodu ABAP dokumentace v češtině neexistuje.

GET_TEMPLATE_URL

Viditelnost
Private
Typ
Static
Krátký popis

Zisk URL odkazu na šablonu v MIME repozitáři

Parametry

Název	Typ	Druh typizace	Referenční typ	Volitelný	Popisek
IS_OBJECT	Importing	TYPE	TY_OBJECT_LIST		Objekt
RV_URL	Returning	TYPE	ZDOCU_TMPLT_URL		URL na objekt MIME

Datové prvky

Název	Dat. typ	Délka	Des. m.	Doména	Popisek
ZDOCU_TMPLT_URL	CHAR	100	0	TEXT100	URL na objekt MIME

ABAP dokumentace

Neexistuje v jazyce CS

Obrázek 6.23: Příklad dokumentace metody ve formátu DOCX

7 Testování

7.0.1 Automatizované testy

K automatizovanému testování slouží v SAPu tzv. unit testy.

Pro tyto potřeby byl tvořen testovací include, který obsahuje různé testovací scénáře pro jednotlivé třídy, metody apod.

7.0.2 Ruční kontrola

Hlavním výstupem programu jsou externí soubory s různým obsahem. Proces jejich vytváření není možné automatizovaně testovat, a proto je nutné jejich obsah ručně zkontrolovat.

7.0.3 Zátěžové testy

V případě menšího množství objektů (nižší desítky) trvá provedení programu řádově jednotky, maximálně desítky sekund. Vždy záleží na množství dat v jednotlivých objektech.

Program byl testován i na rozsáhlých třídách a paketech obsahujících vyšší desítky až stovky objektů. V případě několika desítek objektů může zpracování trvat až jednotky minut. V případě stovek objektů nemusí dojít ke zpracování všech dat v časovém limitu (ten má SAP nastaven na 600 sekund) a program tak ukončí svou činnost chybou. Tento problém by se dal řešit lepší optimalizací způsobů zisku a zpracování dat.

8 Nasazení do produkce

Program byl vyvíjen na systému určeném pro vývoj. Do produkčního prostředí ho lze buď transportovat, nebo importovat pomocí abapGitu.

8.1 Transport

Obecným popisem procesu nasazování programů do produkce pomocí transportů se zabývá kapitola 2.2.1.

8.1.1 Transportní cesta

Pokud je mezi dvěma SAP systémy nadefinována transportní cesta, tak je možné provést transport přímo. Tímto způsobem byl program zkušebně nahrán do jiného SAP systému.

8.1.2 Lokální soubor

Pokud transportní cesta neexistuje, tak je možné transportní požadavek vyexportovat do dvou lokálních souborů. Tyto soubory je poté nutno ručně nahrát do jiného SAP systému a tím dokončit transport.

Program byl vyexportován do dvou externích souborů **K902637** a **R902637**, které jsou k práci přiloženy. Do SAPu je lze nahrát například pomocí transakce CG3Z.

8.2 abapGit

abapGit je open source rozšíření do SAPu vyvíjené komunitou vývojářů. Umožňuje online propojení s GitHubem i vytváření offline repozitářů na lokálním počítači.

abapGit umožňuje distribuci objektů, které náleží danému paketu. Problém nastává s obsahem databázových tabulek (viz 6.3), jejich obsah paketu nenáleží a proto je nutné jejich obsah po instalaci programu ručně vyplnit.

Detailní návod na instalaci programu pomocí abapGitu je obsažen v příloze.

9 Závěr

V rámci bakalářské práce byl navržen a implementován vlastní nástroj, který slouží k tvorbě externí dokumentace v systému SAP.

Před samotným návrhem programu byla provedena rešerše stávajících možností tvorby dokumentace v systému SAP a byly otestovány existující externí nástroje k tvorbě dokumentace.

Při návrhu programu bylo nutné zohlednit požadavek, aby program dovedl pracovat s různými, uživatelem vytvořenými šablonami. Tuto funkci existující nástroje neumožňují a tím se od nich program odlišuje.

Program byl vyvinut v jazyce ABAP. Šablony a výsledné dokumentační soubory jsou ve formátu MS Word, obsah tabulky je ve formátu MS Excel. K tvorbě šablon a k přenosu dat ze SAPu do MS Word slouží jazyk XML. Program byl vyvíjen a testován na SAPu ve verzi 7.4 a zkušebně byl nasazen do jiného SAP systému ve verzi 7.5.

Během vývoje byl kladen důraz na univerzálnost kódu, aby bylo možné v budoucnu program rozšířit o další funkce nebo typy objektů, pro něž je možné generovat dokumentaci. Při hledání dat o objektech bylo nutné ponořit se hluboko do systému SAP, aby bylo možné pochopit způsoby, jakými si SAP data o objektech uchovává.

Výsledkem je program s jednoduchým, intuitivním GUI, jenž je připraven k použití v praxi.

Do budoucna by mohl program být ještě rozšířen o více výstupních formátů (např. HTML), dále by mohl umožňovat export části obsahu tabulky (výsledek selectu), nebo by program mohl být lépe optimalizován, aby nedocházelo k vypršení časového limitu při velkém množství dat. GUI programu je optimalizováno pro češtinu, dále by jej bylo možné lokalizovat pro více jazyků (angličtina, němčina).

Bibliografie

1. *Plánování podnikových zdrojů* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-12-22]. Dostupné z: https://cs.wikipedia.org/wiki/Pl%5C%C3%5C%A1nov%5C%C3%5C%A1n%5C%C3%5C%AD_podnikov%5C%C3%5C%BDch_zdroj%5C%C5%5C%AF.
2. *ERP Market Share, Size, and Trends Report for 2022* [online]. [cit. 2022-12-22]. Dostupné z: https://softwareconnect.com/erp/erp-market/.
3. *Companies That Use SAP* [online]. [cit. 2022-12-22]. Dostupné z: https://www.thomsondata.com/customer-base/sap.php.
4. KÜHNHAUSER, Karl-Heinz. *ABAP Výukový kurz*. Brno: Computer Press, 2009. ISBN 978-80-251-2117-7.
5. *ABAP - Keyword Documentation* [online]. [cit. 2023-04-27]. Dostupné z: https://help.sap.com/doc/abapdocu_752_index_htm/7.52/en-us/abenabap.htm.
6. *Unified Modeling Language* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-12-22]. Dostupné z: https://cs.wikipedia.org/wiki/Unified_Modeling_Language.
7. *Doxygen* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-12-22]. Dostupné z: https://cs.wikipedia.org/wiki/Doxygen.
8. *Eclipse (vývojové prostředí)* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-12-22]. Dostupné z: https://cs.wikipedia.org/wiki/Eclipse_(v%5C%C3%5C%BDvojov%5C%C3%5C%A9_prost%5C%C5%5C%99ed%5C%C3%5C%AD).
9. *SAP ABAP Documentation Generator* [online]. [cit. 2022-12-19]. Dostupné z: https://www.pikon.com/en/services/sap-add-ons/add-on-sap-abap-documentation-generator/.
10. *AbapDoc* [online]. [cit. 2022-12-22]. Dostupné z: https://github.com/kctdata/abapDoc.
11. *AbapDOC – Free Documentation Generator for ABAP Repository Objects* [online]. [cit. 2022-12-22]. Dostupné z: https://indevo.pl/abapdoc-documentation-generator-for-abap-repository-objects/.
12. FRYDLIAND, Dmitry. *SAP Customizing Documentation Generation Tool* [online]. [cit. 2022-12-20]. Dostupné z: https://blogs.sap.com/2017/11/30/sap-customizing-documentation-generation-tool/.

13. *Extensible Markup Language* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-12-22]. Dostupné z: https://cs.wikipedia.org/wiki/Extensible_Markup_Language.

14. *Using Special Characters in XML* [online]. [cit. 2022-11-03]. Dostupné z: https://docs.oracle.com/cd/A97335_02/apps.102/bc4j/developing_bc_projects/obcCustomXml.htm.

10 Příloha

10.1 Způsoby získávání potřebných informací o objektech

Třída

Získaná informace	Třída	Metoda	Funkční modul	Tabulka
Atributy				SEOCOMPODF
Metody	CL_ABAP_OBJECTDESCR	DESCRIBE_BY_NAME		
Datové prvky				DD04L
Zdrojový kód	CL_RECA_RS_SERVICES	GET_SOURCE		
Popisek			SEO_CLASS_READ	
ABAP dokumentace			DOCU_READ	
Popisky atributů				SEOCOMPOTX
Popisky datových prvků				DD04T

Tabulka 10.1: Způsoby zisku informací o třídě

10.1.1 Metoda

Získaná informace	Funkční modul	Tabulka
Statická/ instanční	SEO_METHOD_GET_DETAIL	
Parametry		SEOSUBCODF
Datové prvky		DD04L
Zdrojový kód	SEO_METHOD_GET_SOURCE	
Popisek	SEO_METHOD_GET_DETAIL	
ABAP dokumentace	DOCU_READ	
Popisky parametrů		SEOSUBCOTX
Popisky datových prvků		DD04T

Tabulka 10.2: Způsoby zisku informací o metodě

10.1.2 Funkční skupina

Získaná informace	Funkční modul	Tabulka	Příkaz
Funkční moduly		ENLFDIR	
Zdrojový kód			READ REPORT
Popisek		AREAT	
ABAP dokumentace	DOCU_READ		
Popisky funkčních modulů		TFTIT	

Tabulka 10.3: Způsoby zisku informací o funkční skupině

10.1.3 Funkční modul

Získaná informace	Funkční modul	Tabulka
Funkční skupina		ENLFDIR
Parametry	FUPARAREF	
Zdrojový kód	RPY_FUNCTIONMODULE_READ	
Datové prvky		DD04L
Popisek		TFTIT
ABAP dokumentace	DOCU_READ	
Popisky datových prvků		DD04T

Tabulka 10.4: Způsoby zisku informací o funkčním modulu

10.1.4 Report

Získaná informace	Funkční modul	Tabulka	Příkaz
Zdrojový kód			READ REPORT
Popisek		TRDIRT	
ABAP dokumentace	DOCU_READ		

Tabulka 10.5: Způsoby zisku informací o reportu

10.1.5 Tabulka

Získaná informace	Funkční modul	Tabulka
Atributy		DD03L
Datové prvky		DD03L
Popisek		DD02T
ABAP dokumentace	DOCU_READ	
Popisky datových prvků		DD04T

Tabulka 10.6: Způsoby zisku informací o tabulce

10.2 Tvorba vlastních šablon

Pro tvorbu šablon je potřeba povolit v MS Word režim vývojáře. Tento režim umožňuje vkládání XML značek do dokumentu. Tvorba šablony se skládá z následujících kroků:

1. Nahrání XML souboru k MS Word dokumentu.
2. Vložení XML značek na požadovaná místo v dokumentu.

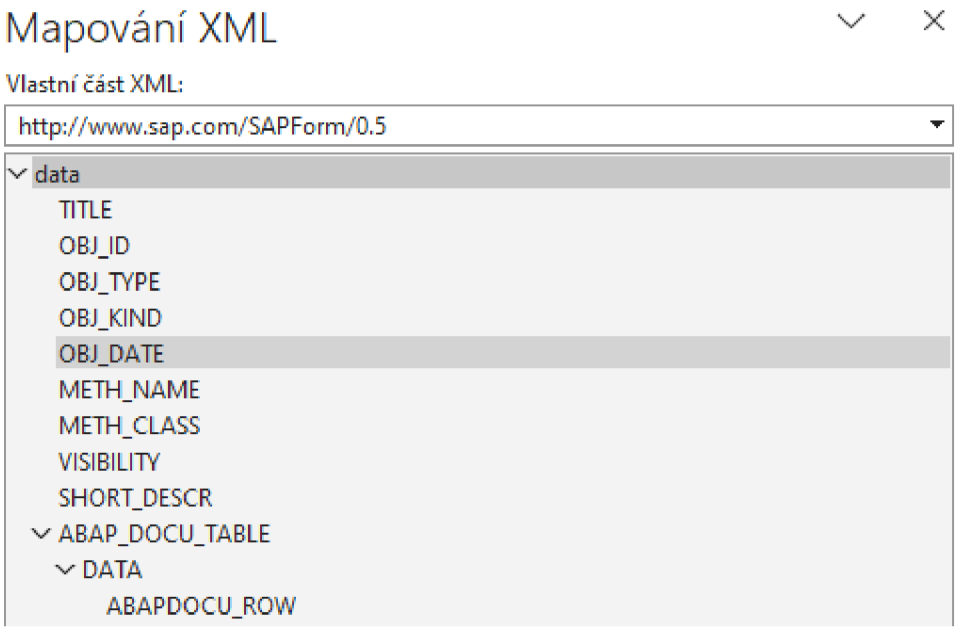
10.2.1 Nahrání XML souboru k MS Word dokumentu

Každý typ objektu má vlastní XML soubor s podporovanými XML značkami.

Typ objektu	XML soubor
Třída	xml_tags_class.xml
Metoda	xml_tags_meth.xml
Report	xml_tags_prog.xml
Funkční skupina	xml_tags_fugr.xml
Funkční modul	xml_tags_func.xml
Tabulka	xml_tags_tabl.xml

Tabulka 10.7: XML soubory pro jednotlivé typy objektů

Po nahrání tohoto souboru uživatel ve Wordu vidí všechny tyto značky a může z nich vybírat.

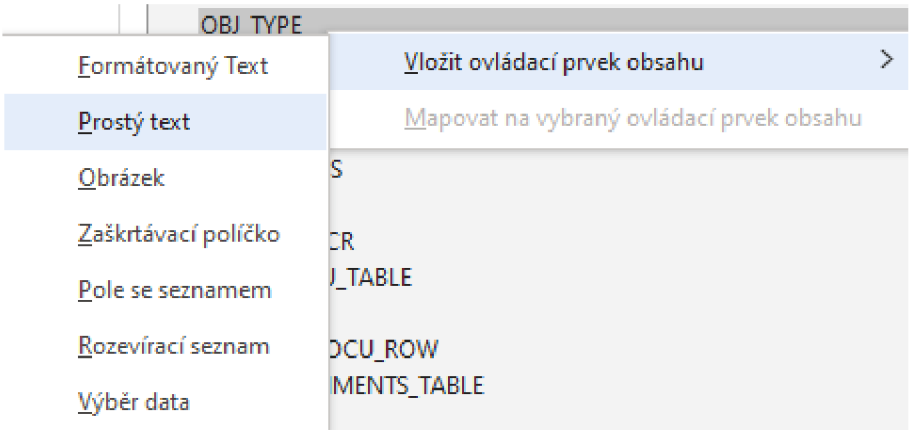


Obrázek 10.1: Výňatek z výběru XML značek v MS Word

10.2.2 Vložení XML značek na požadovaná místo v dokumentu

Každá XML značka reprezentuje část generovaného obsahu. Vkládáním značek do do dokumentu si uživatel **definuje obsah** jeho dokumentace.

Vložení probíhá tak, že uživatel vybere značku ze seznamu, klikne na ni pravým tlačítkem myši a vybere možnost **"Vložit ovládací prvek obsahu → Prostý text"** Značka se vloží na to místo v dokumentu, kde se nachází uživatelův kurzor.

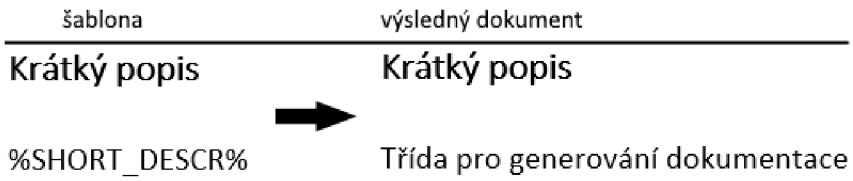


Obrázek 10.2: Vložení značky do dokumentu

Dané značce jde ve Wordu nastavit libovolný styl, jenž bude po vygenerování výsledného souboru dodržen.

Text

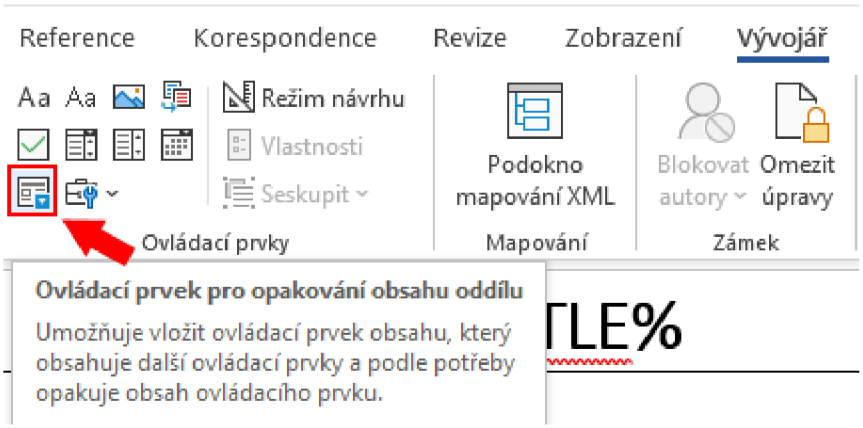
Pro generování bloku textu je potřeba vložit značku na požadované místo.



Obrázek 10.3: Text v dokumentu

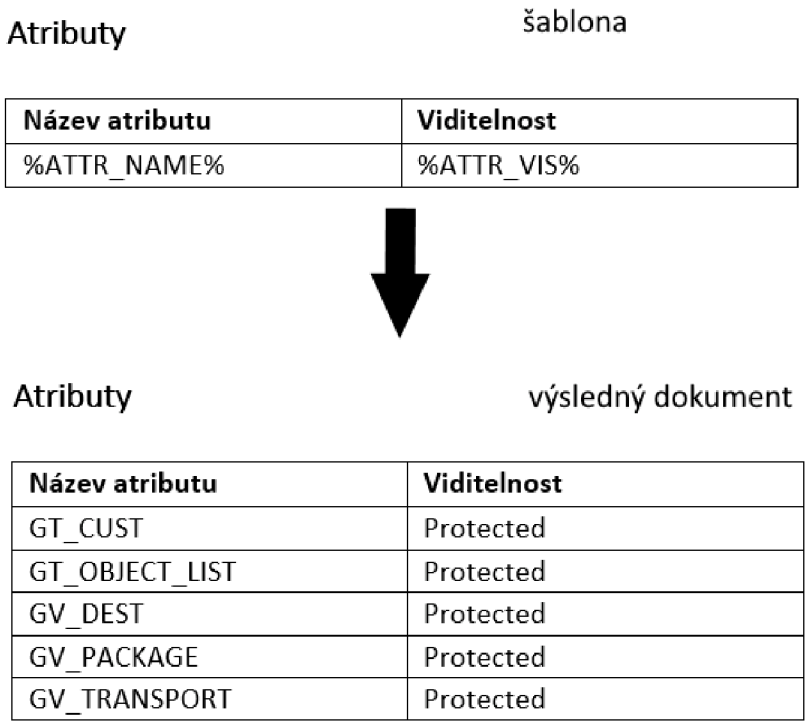
Tabulky

Při generování tabulky s větším počtem řádků je potřeba nadefinovat jeden řádek tabulky a na ten nastavit opakování. Opakování se nastavuje tak, že uživatel označí jeden řádek tabulky a ve vývojářském menu vybere možnost **"Ovládací prvek pro opakování obsahu oddílu"**.



Obrázek 10.4: Nastavení opakování řádku tabulky

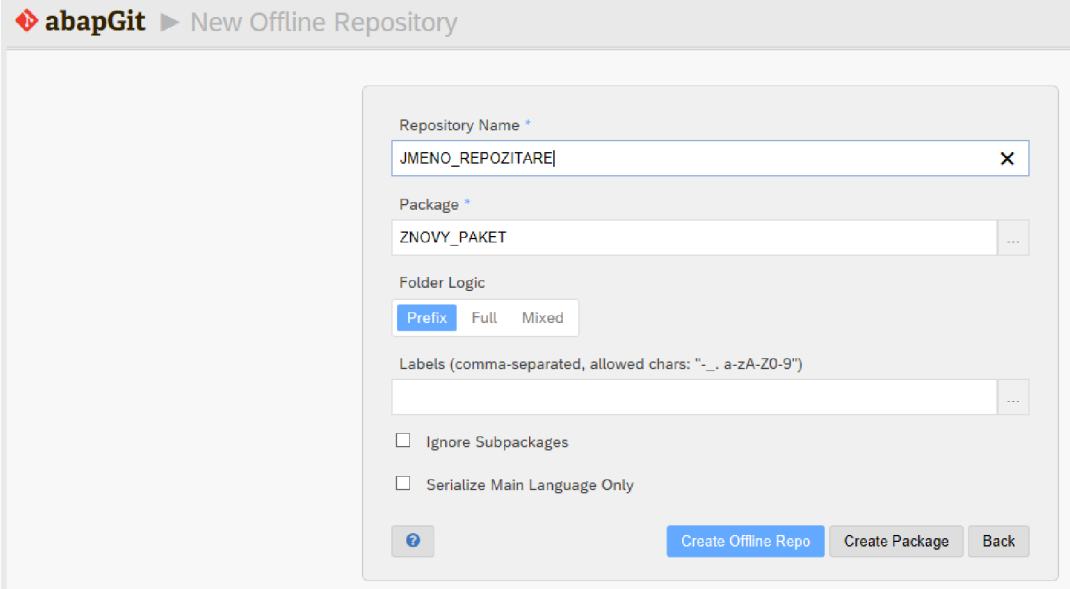
Tabulka si pak sama dotvoří požadovaný počet řádků dle generovaného obsahu.



Obrázek 10.5: Tabulka v dokumentu

10.3 Návod na instalaci pomocí abapGitu

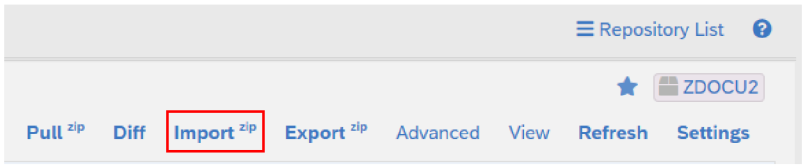
1. Nainstalujeme abapGit. Standalone verze bude stačit.
2. Vytvoříme nový **offline** repozitář.



Obrázek 10.6: Tvorba nového offline repozitáře

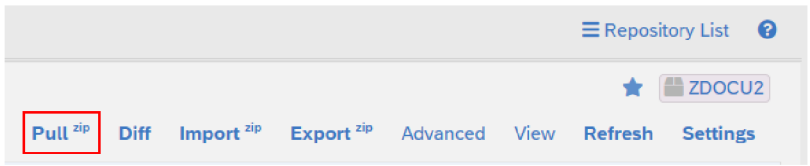
- Název repozitáře lze zvolit libovolný.
- Při výběru paketu je doporučeno zvolit nový, ještě neexistující paket. Systém si ho vytvoří.

3. Do repozitáře importujeme tlačítkem **Import** k práci přiložený soubor **ZDOCU.zip**.



Obrázek 10.7: Import .zip souboru

4. Tlačítkem **Pull** nahrajeme obsah repozitáře do paketu.

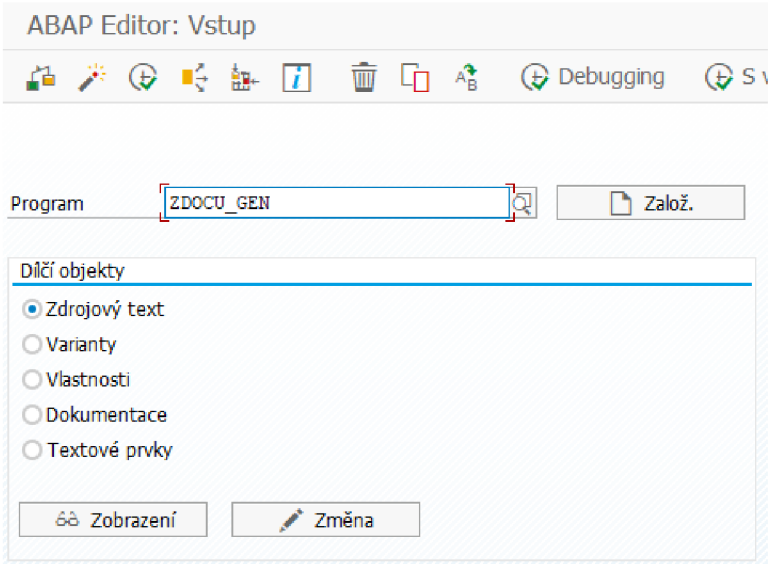


Obrázek 10.8: Nahrání souborů z repozitáře do paketu

Pokud nedojde k nahrání všech souborů z repozitáře do paketu, tak abapGit vyhodí chybu, v takovém případě tento krok opakujeme, dokud nedojde k úspěšnému nahrání všech souborů.

Při aktivaci souborů může SAP hlásit problémy s aktivací souborů, opět se jedná o chybu abapGitu, nikoliv instalovaného programu. V tomto případě provedeme aktivaci i přes veškerá upozornění.

5. Tím je instalace dokončena. Program můžeme spustit například pomocí ABAP Editoru (transakce SE38). Jako název programu zadáme **ZDOCU_GEN**.



Obrázek 10.9: Spuštění programu v ABAP Editoru

AbapGit neslouží k distribuci obsahů tabulky, konfigurační tabulky si musí uživatel sám ručně vyplnit v nastavení programu. Nastavení, na kterém byl program testován je obsaženo v přiložených souborech, uživatel si jej může zkopírovat.