

Vylepšený task manager

Předmět: Testování softwaru

Datum: 21.2.2026

Autor: Lucie Mazánková

Cílem projektu bylo rozšířit původní konzolovou aplikaci pro správu úkolů tak, aby data nebyla ukládána pouze do paměti programu, ale do relační databáze MySQL. Aplikace nyní pracuje s databází a podporuje plné CRUD operace (Create, Read, Update, Delete). Součástí řešení je také sada automatizovaných testů vytvořených pomocí knihovny pytest.

Projekt propojuje programovací jazyk Python s databázovým systémem MySQL a demonstруje práci s databázovým připojením, SQL dotazy i základní testování aplikace.

Použité technologie

- Programovací jazyk: Python 3
- Databáze: MySQL
- Knihovna pro připojení: mysql-connector-python
- Testování: pytest
- Nástroj pro správu databáze: MySQL Workbench

Nejprve je potřeba mít nainstalovaný balíček:

pip install mysql-connector-python

Databázová struktura

Byla vytvořena databáze `ukoly_db`, která obsahuje tabulku `ukoly`.

Struktura tabulky:

- id – primární klíč, AUTO_INCREMENT
- nazev – název úkolu (povinný údaj)
- popis – popis úkolu (povinný údaj)
- stav – ENUM ('Nezahájeno', 'Probíhá', 'Hotovo'), výchozí hodnota 'Nezahájeno'
- datum_vytvoreni – datum a čas vytvoření (automaticky generované)

Tabulka je vytvořena automaticky při spuštění programu, pokud ještě neexistuje.

Funkcionalita aplikace

Aplikace je rozdělena do jednotlivých funkcí:

prijeni_db()

Zajišťuje připojení k MySQL databázi. V případě chyby zobrazí chybovou hlášku.

vytvoreni_tabulky()

Vytvoří tabulku `ukoly`, pokud ještě neexistuje.

hlavni_menu()

Zobrazuje uživateli nabídku možností:

1. Přidat úkol
2. Zobrazit úkoly
3. Aktualizovat úkol
4. Odstranit úkol
5. Ukončit program

Program kontroluje neplatné vstupy a vyžaduje opakovanou volbu.

pridat_ukol()

- Kontroluje, že název i popis nejsou prázdné.
- Úkol je uložen do databáze.
- ID a datum vytvoření se generují automaticky.
- Výchozí stav je nastaven na „Nezahájeno“.

zobrazit_ukoly()

- Zobrazuje seznam úkolů.
- Obsahuje filtr – zobrazí pouze úkoly se stavem „Nezahájeno“ nebo „Probíhá“.
- Pokud nejsou žádné úkoly, zobrazí informaci o prázdném seznamu.

aktualizovat_ukol()

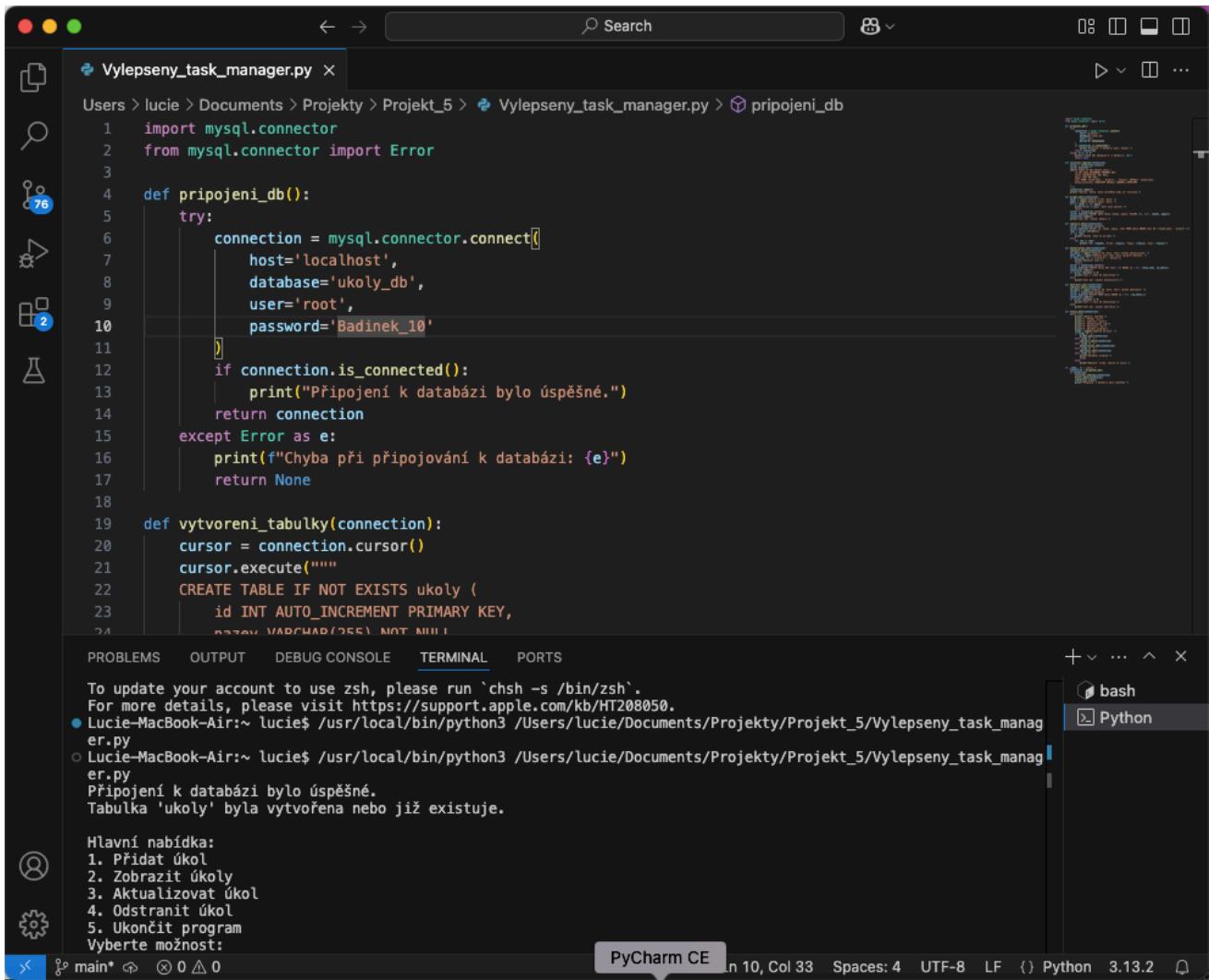
- Uživatel vybírá úkol podle ID.
- Lze změnit stav na „Probíhá“ nebo „Hotovo“.
- Program kontroluje existenci zadaného ID.

odstranit_ukol()

- Uživatel vybírá úkol podle ID.
- Úkol je trvale odstraněn z databáze.
- Při zadání neexistujícího ID je uživatel upozorněn.

Toto řešení představuje jednoduchého správce úkolů s využitím Pythonu a databáze MySQL. Program je rozdělen do jednotlivých funkcí, které zajišťují připojení k databázi, vytvoření potřebné tabulky a provádění základních CRUD operací – přidání, zobrazení, aktualizace a odstranění úkolu. Uživatel s aplikací interaguje prostřednictvím textového menu v konzoli. Data se ukládají do tabulky `ukoly`, která obsahuje sloupce pro název, popis, stav úkolu a datum vytvoření. Pro práci s databází je použit modul `mysql.connector` a připojení je řešeno přes lokální server.

Tabulka `ukoly` v databázi byla nejprve vytvořena ručně ve vizuálním nástroji MySQL Workbench, kde byla definována její struktura – sloupce, datové typy i výchozí hodnoty. Následně je program schopen tabulku znova vytvořit (pokud neexistuje), aby byl kód plně funkční i samostatně bez závislosti na předchozí ruční konfiguraci.



The screenshot shows the PyCharm Community Edition interface. The code editor displays `Vylepseny_task_manager.py` with the following content:

```
1 import mysql.connector
2 from mysql.connector import Error
3
4 def prijeni_db():
5     try:
6         connection = mysql.connector.connect(
7             host='localhost',
8             database='ukoly_db',
9             user='root',
10            password='Badinek_10'
11        )
12        if connection.is_connected():
13            print("Připojení k databázi bylo úspěšné.")
14            return connection
15        except Error as e:
16            print(f"Chyba při připojování k databázi: {e}")
17            return None
18
19    def vytvoreni_tabulky(connection):
20        cursor = connection.cursor()
21        cursor.execute("""
22            CREATE TABLE IF NOT EXISTS ukoly (
23                id INT AUTO_INCREMENT PRIMARY KEY,
24                nazev VARCHAR(255) NOT NULL,
25                popis TEXT,
26                stav ENUM('Plánování', 'V provozu', 'Hotové') NOT NULL
27            )
28        """)
29
30    connection = prijeni_db()
31    vytvoreni_tabulky(connection)
32
33    print("Tabulka 'ukoly' byla vytvořena nebo již existuje.")
```

The terminal window shows the following output:

```
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Lucie-MacBook-Air:~ lucie$ /usr/local/bin/python3 /Users/lucie/Documents/Projekty/Projekt_5/Vylepseny_task_manager.py
Připojení k databázi bylo úspěšné.
Tabulka 'ukoly' byla vytvořena nebo již existuje.
```

The status bar at the bottom indicates: PyCharm CE, In 10, Col 33, Spaces: 4, UTF-8, LF, Python 3.13.2.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree, with 'ukoly_db' selected. Under 'Tables', 'ukoly' is selected, and its 'Columns' (id, nazev, popis, stav, datum_vytvoreni) are listed. The main pane shows a query result grid for the 'ukoly' table. The query is:

```
1  SELECT * FROM ukoly_db.ukoly
```

The result grid has columns: id, nazev, popis, stav, datum_vytvoreni. The first row shows values: id (NULL), nazev (NULL), popis (NULL), stav (NULL), datum_vytvoreni (NULL). The right sidebar contains a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." The bottom status bar says "Query Completed".

Automatizované testování

Pro testování byla použita knihovna pytest.

Každá hlavní funkce (přidání, aktualizace, odstranění) obsahuje:

- 1 pozitivní test (ověření správné funkčnosti)
- 1 negativní test (ověření reakce na neplatný vstup)

Testy:

- Dynamicky vytvářejí testovací data.
- Po dokončení testu jsou testovací záznamy odstraněny.
- Nedochází k trvalé změně databáze.

Byla použita samostatná testovací databáze, aby nedošlo k ovlivnění produkčních dat.

Vytvoření testovací databáze ukoly_test_db

Důležité (aby testy prošly)

Musíš mít:

- vytvořenou databázi ukoly_test_db
- vytvořenou tabulku ukoly
- běžící MySQL server
- správné heslo k root uživateli

MySQL Workbench

Administration Schemas Query 5 ukoly SQL File 5*

SCHEMAS Filter objects

- knihovna
- sys
- ukoly_db
 - Tables
 - Views
 - Stored Procedures
 - Functions
- ukoly_test_db
 - Tables
 - ukoly
 - Views
 - Stored Procedures
 - Functions
- zamestnanci

CREATE DATABASE IF NOT EXISTS ukoly_test_db;
 USE ukoly_test_db;
 CREATE TABLE IF NOT EXISTS ukoly (
 id INT AUTO_INCREMENT PRIMARY KEY,
 nazev VARCHAR(255) NOT NULL,
 popis TEXT NOT NULL,
 stav ENUM('nezahájeno', 'probíhá', 'hotovo') DEFAULT 'nezahájeno',
 datum_vytvoreni TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

Action Output

Time	Action	Response
17-9-2018 1:11	CREATE DATABASE IF NOT EXISTS ukoly_test_db;	Success

Object Info Session

EXPLORER ...

OPEN EDITORS test_ukoly.py ~ NO FOLDER OPENED OUTLINE TIMELINE

Users > lucie > Documents > Projekty > Projekt_5 > test_ukoly.py > pripoj_test_db

```

1 import pytest
2 import mysql.connector
3 from mysql.connector import Error
4
5 # === Pomocné funkce ===
6
7 def pripoj_test_db():
8     return mysql.connector.connect(
9         host="localhost",
10        user="root",
11        password="Badinek_10",
12        database="ukoly_test_db"
13    )
14
15 def vycisti_tabulku(connection):
16     cursor = connection.cursor()
17     cursor.execute("DELETE FROM ukoly")
18     connection.commit()
19
20 # === Testy pro přidání úkolu ===
21
22 def test_pridani_ukolu_ok():
23     conn = pripoj_test_db()
24     vycisti_tabulku(conn)
25     cursor = conn.cursor()
26     cursor.execute("INSERT INTO ukoly (nazev, popis) VALUES (%s, %s)", ("Test úkol", "Popis"))
27     conn.commit()
28
29     cursor.execute("SELECT COUNT(*) FROM ukoly WHERE nazev = 'Test úkol'")
30     count = cursor.fetchone()[0]
31     assert count == 1
32     vycisti_tabulku(conn)
33     conn.close()
34
  
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash: syntax error near unexpected token `Projekt_5'
 Lucie-MacBook-Air:~ lucie\$ /usr/local/bin/python3 /Users/lucie/Documents/Projekty/Projekt_5/test_ukoly.py
 Lucie-MacBook-Air:~ lucie\$

Ln 12, Col 33 Spaces: 4 UTF-8 LF {} Python 3.13.2

```
test_ukoly.py::test_pridani_ukolu_ok PASSED
test_ukoly.py::test_pridani_ukolu_neplatny PASSED
test_ukoly.py::test_aktualizace_ukolu_ok PASSED
test_ukoly.py::test_aktualizace_neexistujiciho_ukolu PASSED
test_ukoly.py::test_odstraneni_ukolu_ok PASSED
test_ukoly.py::test_odstraneni_neexistujiciho_ukolu PASSED
```

Závěr

Projekt demonstruje praktické využití propojení Pythonu s relační databází MySQL. Byla implementována plnohodnotná CRUD aplikace s ošetřením chybových stavů a automatizovaným testováním.

Díky oddělení databázové logiky od uživatelského rozhraní je aplikace přehledná, snadno rozšiřitelná a testovatelná. Projekt přispěl k lepšímu pochopení práce s databázemi, SQL dotazy a testováním aplikací v Pythonu.