



PROGRAMFEJLESZTÉS DOKUMENTÁCIÓ

Raktározási felület tervezése és futtatása

Raktározási program, mely egy grafikus felületen jelenik meg segédlettel, és adat beviteli lehetőségekkel, python programozási nyelvben megvalósítva.

Általános információk

- **Munkafolyamat neve:** Raktározási felület tervezése és futtatása
- **Cél:** Python programozási nyelvben, egy tetszőleges program megírása, mely egy felületet jelenít meg, raktározási céllal.
- **Fejlesztési környezet:** Pycharm, sqlite3

Munkafolyamat leírása

Raktározási program, python programozási nyelvben megvalósítva. Mely egy grafikus felületen jelenik meg:

segédlettel,
adat beviteli lehetőségekkel,
különböző gombokkal ellátva.

Működési logika

A program egy raktározási felületet biztosít, amelyben termékek adatait lehet felvinni, lekérdezni és exportálni CSV fájlba. Az alkalmazás egy grafikus felületen (GUI) keresztül működik, amelyet a `tkinter` könyvtár segítségével valósítottak meg. Az adatokat egy adatbázisban tárolják, amelyhez kapcsolódó függvényeket a `csoportmunka\_adat` modul biztosít.

1. Indítás és inicializáció:

Az alkalmazás indításakor létrejön a fő ablak (`win`), amelynek címe "Raktározási felület" és mérete 1920x1080 pixel. A háttérszürke, és az ablak átméretezhető.

A program betölti a logo képet, amely a felület bal felső sarkában jelenik meg.

Egy üzenetdoboz figyelmezteti a felhasználót, hogy ez egy raktározást segítő adatbáziskezelő program, és ajánlja a "Segédlet" elolvasását.

2. Gombok és funkcióik:

Új adatok felvétele a rendszerbe: Ez a gomb a `db\_insert` függvényt hívja meg, amely a felhasználó által bevitt adatokat (terméknév, darabszám, leltári szám, behelyezési dátum, beszállító, származási hely) beszúrja az adatbázisba.

Adatok lekérdezése a rendszerből: Ez a gomb a `db\_query` függvényt hívja meg, amely lekéri az adatbázisban tárolt adatokat, és megjeleníti őket a `ttk.Treeview` táblázatban.

Kiírás CSV fájlba: Ez a gomb a `db\_export\_csv` függvényt hívja meg, amely az adatbázisban tárolt adatokat CSV fájlba exportálja.

Segédlet: Ez a gomb egy üzenetdobozban megjeleníti a program használati útmutatóját.

Kilépés a rendszerből:** Ez a gomb bezárja az alkalmazást.

3. Adatbeviteli mezők:

A felületen több beviteli mező található, ahol a felhasználó megadhatja a termék nevét, darabszámát, leltári számát, behelyezési dátumát, beszállítóját és származási helyét. Ezeket az adatokat a program az adatbázisba menti.

4. Táblázat megjelenítése:

A ``ttk.Treeview`` widget segítségével egy táblázat jelenik meg, amelyben a termékek adatai láthatók. A táblázat oszlopai: "Termék név", "Darabszám", "Leltári szám", "Behelyezési dátum", "Beszállító cég", "Származási hely".

5. Adatbázis műveletek:

Az adatbázis létrehozása a ``db_create`` függvény hívásával történik, amely az alkalmazás indításakor fut le.

Az adatbázis bezárása a ``db_close`` függvény hívásával történik, amely az alkalmazás bezárásakor fut le.

6. Segédlet:

A segédlet gombra kattintva egy üzenetdoboz jelenik meg, amelyben a program használati útmutatója olvasható. Az útmutató kiemeli, hogy minden adatmező kitöltése kötelező, és a leltári szám nem ismétlődhet.

7. Főbb függvények:

``db_insert``: Beszúrja az új adatokat az adatbázisba.

``db_query``: Lekéri és megjeleníti az adatbázisban tárolt adatokat.

``db_export_csv``: Exportálja az adatbázis adatait CSV fájlba.

``db_create``: Létrehozza az adatbázist, ha az még nem létezik.

``db_close``: Bezárja az adatbázis kapcsolatot.

Ez a program egy egyszerű, de hatékony eszköz a raktári adatok kezelésére, amely lehetővé teszi az adatok könnyű bevételét, lekérdezését és exportálását.

Hibakezelés és validáció

A program nem tartalmaz explicit hibakezelést vagy adatvalidációt, de a felhasználót figyelmezteti, hogy minden mező kitöltése kötelező, és a leltári szám nem ismétlődhet. Az adatbázis szintjén lehetőség van egyedi leltári számok kikényszerítésére.

Használati esetek

1. Új termék adatainak felvétele

Leírás: A felhasználó bevitelez egy új termék adatait a rendszerbe.

Lépések:

1. A felhasználó kitölti az összes adatmezőt (Termék név, Darabszám, Leltári szám, Behelyezési dátum, Beszállító cég, Származási hely).
2. A felhasználó rákattint az „Új adatok felvétele a rendszerbe” gombra.
3. A program ellenőrzi, hogy minden mező kitöltve van-e, és hogy a leltári szám egyedi-e.
4. Ha minden adat helyes, a program beszúrja az adatokat az adatbázisba.
Kimenet: Az új termék adatai hozzáadódnak az adatbázishoz.

2. Termék adatainak lekérdezése

Leírás : A felhasználó lekérdezi a rendszerben tárolt termékek adatait.

Lépések:

1. A felhasználó rákattint az „Adatok lekérdezése a rendszerből” gombra.
2. A program lekéri az adatbázisban tárolt összes termék adatát.
3. Az adatok megjelennek a ttk.Treeview táblázatban.

Kimenet: A táblázatban megjelennek a termékek neve, darabszáma, leltári száma, behelyezési dátuma, beszállítója és származási helye.

3. Adatok exportálása CSV fájlba

Leírás: A felhasználó exportálja a rendszerben tárolt adatokat egy CSV fájlba.

Lépések:

1. A felhasználó rákattint az „Kiírás CSV fájlba” gombra.
2. A program elkészíti a CSV fájlt az adatbázisban tárolt adatokkal.
3. A CSV fájl mentésre kerül a felhasználó által megadott helyre.

Kimenet: Egy CSV fájl készül, amely tartalmazza a termékek adatait.

4. Segédlet megtekintése

Leírás: A felhasználó megtekinti a program használati útmutatóját.

Lépések:

1. A felhasználó rákattint az „Segédlet!” gombra.
2. Megjelenik egy üzenetdoboz, amelyben a program használati útmutatója olvasható.

Kimenet: A felhasználó értesül a program használatának szabályairól (pl. minden mező kitöltése kötelező , leltári szám egyedisége).

5. Kilépés a rendszerből

Leírás: A felhasználó bezárja az alkalmazást.

Lépések:

1. A felhasználó rákattint a „Kilépés a rendszerből” gombra.
2. A program bezárja az adatbázis kapcsolatot (db_close függvény hívása).
3. Az alkalmazás ablak bezárul.

Kimenet: A program leáll, és az adatbázis kapcsolat biztonságosan lezárul.

6. Hibás adatbevitel kezelése

Leírás: A felhasználó hibás adatokat próbál bevinni (pl. üres mező, ismétlődő leltári szám).

Lépések:

1. A felhasználó nem tölti ki valamelyik mezőt, vagy olyan leltári számot ad meg, amely már létezik az adatbázisban.
2. A felhasználó rákattint az „Új adatok felvétele a rendszerbe” gombra.
3. A program hibát jelez.

Kimenet: Az adatokat nem szűrja be az adatbázisba, és a felhasználó értesül a hibáról.

7. Adatbázis inicializálása

Leírás: Az adatbázis létrehozása, ha az még nem létezik.

Lépések:

A program indításakor a db_create függvény meghívásra kerül.

Ha az adatbázis még nem létezik, a program létrehozza a szükséges táblákat és struktúrákat.

Kimenet: Az adatbázis készen áll az adatok tárolására.

8. Adatbázis kapcsolat lezárása

Leírás: Az adatbázis kapcsolat biztonságos lezárása a program bezárásakor.

Lépések:

A felhasználó rákattint a "Kilépés a rendszerből" gombra.

A program meghívja a db_close függvényt, amely bezárja az adatbázis kapcsolatot.

Kimenet: Az adatbázis kapcsolat lezárul, és az alkalmazás bezárul.

9. Logo és felület megjelenítése

Leírás: A felhasználó megtekinti a program grafikus felületét, amely tartalmazza a logo képet és az adatbeviteli mezőket.

Lépések:

A program indításakor megjelenik a fő ablak, amely tartalmazza a logo képet, a gombokat és az adatbeviteli mezőket.

Kimenet: A felhasználó látja a program felületét, és elkezdheti használni.

10. Adatok frissítése a táblázatban

Leírás: A felhasználó frissíti a táblázatban megjelenő adatokat.

Lépések:

1. A felhasználó rákattint az "Adatok lekérdezése a rendszerből" gombra.
2. A program lekéri a legfrissebb adatokat az adatbázisból.
3. A táblázat frissül, és megjeleníti a legújabb adatokat.

Kimenet: A táblázatban megjelennek a legfrissebb termékadatok.

Program blokkok ismertetése

```
from tkinter import *
import tkinter as tk
from tkinter import ttk
from csoportmunka_adat import *
import tkinter as tk
import random
from tkinter import messagebox
```

Használatos függvények és
segédletek importálása

```
win=Tk()
win.title("Raktározási felület")
win.geometry("1920x1080")
win.config(bg="gray")
win.resizable(True,True)
```

Képernyő beállítása

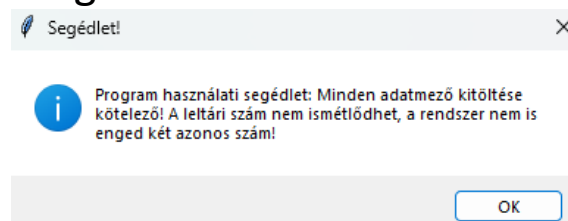
```
logo = tk.Canvas(win, width="222",
height="240", bg="gray", bd="0")
logo.grid(row=1, rowspan=4,
column=2, columnspan=4, padx=3,
pady=3)
photo=
tk.PhotoImage(file='kep/logo.gif')
item = logo.create_image(113, 122,
image=photo)
```

Logo beállítása és elérési út megadása



```
messagebox.showinfo("Kezelő
felület", 'Ez egy raktározást
segítő adatbáziskezelő program!
Adatok bevitele előtt olvasd el a
"Segédlet"-et')
```

Segédlet „messagebox” megadása



```
Button(win, text="Új adatok felvétele a
rendszerbe", bg="black", fg="white",
command=lambda:db_insert(termeknev,
darabszam, leltariszam, behelyezesdatum,
beszallito, szarmazasihely),
font="Helvetica 12 bold").grid(row=2,
column=11, padx=5, pady=5)

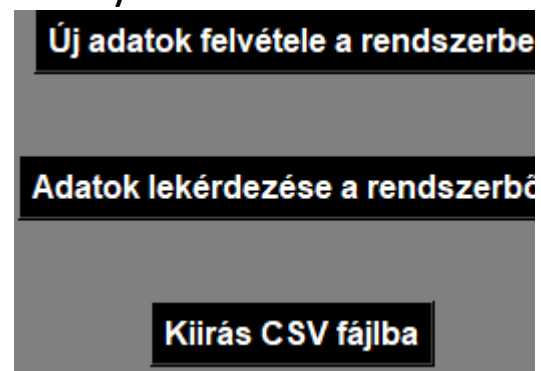
Button(win, text="Adatok lekérdezése a
rendszerből", bg="black", fg="white",
command=lambda:db_query(table), font="Hel
vetia 12 bold").grid(row=3, column=11,
padx=5, pady=5)

Button(win, text="Kiírás CSV
fájlba", bg="black", fg="white",
command=lambda:db_export_csv, font="Helve
tia 12 bold").grid(row=4, column=11,
padx=5, pady=5)

Button(win,
text="Segédlet!", bg="black", fg="white",
command=segedlet, font="Helvetica 12
bold").grid(row=19, column=20, padx=5,
pady=5)

Button(win, text="Kilépés a
rendszerből", bg="#FF0000",
fg="#FFFFFF", font="Helvetica 12 bold",
command=win.destroy).grid(row=20,
column=20)
```

Gombok és azoknak a funkciójának beállítása és a grafikus panelen való elhelyezésük



```

Label(win, text="Termék
név",bg="gray",
font="Helvetica 12
bold").grid(row=5, column=1)
termeknev=Entry(win,
font="Helvetica 20 bold")
termeknev.grid(row=5,
column=3, padx=5, pady=5)

Label(win,
text="Darabszám",bg="gray",
font="Helvetica 12
bold").grid(row=6, column=1)
darabszam=Entry(win,
font="Helvetica 20 bold")
darabszam.grid(row=6,
column=3, padx=5, pady=5)

Label(win, text="Leltári
szám",bg="gray",
font="Helvetica 12
bold").grid(row=7, column=1)
leltariszam=Entry(win,
font="Helvetica 20 bold")
leltariszam.grid(row=7,
column=3, padx=5, pady=5)

Label(win, text="Behelyezési
dátum",bg="gray",
font="Helvetica 12
bold").grid(row=8, column=1)
behelyezesdatum=Entry(win,
font="Helvetica 20 bold")
behelyezesdatum.grid(row=8,
column=3, padx=5, pady=5)

Label(win, text="Beszállító
cég",bg="gray",
font="Helvetica 12
bold").grid(row=9, column=1)
beszallito=Entry(win,
font="Helvetica 20 bold")
beszallito.grid(row=9,
column=3, padx=5, pady=5)

Label(win, text="Származási
hely",bg="gray",
font="Helvetica 12
bold").grid(row=10,
column=1)
szarmazasihely=Entry(win,
font="Helvetica 20 bold")
szarmazasihely.grid(row=10,
column=3, padx=5, pady=5)

```

Grafikus felületen megjelenítendő beviteli mezők konfigurálása és elhelyezése

Termék név	
Darabszám	
Leltári szám	
Behelyezési dátum	
Beszállító cég	
Származási hely	

Az elkészült grafikus felület

Raktári felület



Új adatok felvétele a rendszerbe

Adatok lekérdezése a rendszerből

Kiírás CSV fájlba

Termék név

Darabszám

Leltári szám

Behelyezési dátum

Beszállító cég

Származási hely

Termék név	Darabszám	Leltári szám	Behelyezési dátum	Beszállító cég	Származási hely
------------	-----------	--------------	-------------------	----------------	-----------------

Segédlet!

Kilépés a rendszerből!

11:11
2023.03.18.

Tartalom

Általános információk.....	1
Munkafolyamat leírása	1
Működési logika.....	1
1. Indítás és inicializáció:	1
2. Gombok és funkcióik:	1
3. Adatbeviteli mezők:	2
4. Táblázat megjelenítése:	2
5. Adatbázis műveletek:	2
6. Segédlet:	2
7. Főbb függvények:	2
Használati esetek.....	3
1. Új termék adatainak felvétele	3
2. Termék adatainak lekérdezése	3
3. Adatok exportálása CSV fájlba	3
4. Segédlet megtekintése	3
5. Kilépés a rendszerből.....	4
6. Hibás adatbevitel kezelése	4
7. Adatbázis inicializálása	4
8. Adatbázis kapcsolat lezárása	4
9. Logo és felület megjelenítése	5
10. Adatok frissítése a táblázatban.....	5
Program blokkok ismertetése.....	5
Az elkészült grafikus felület	8

2025. 03. 18.

