**Projekt: Task Manager**

**1. část**

# zadání č. 1

ukoly = []

# Funkce pro přidání nového úkolu

def pridat\_ukol():

    nazev = input("Zadejte název úkolu: ")

    popis = input("Zadejte popis úkolu: ")

    ukol = {"nazev": nazev, "popis": popis}

    ukoly.append(ukol)

    print(f"Úkol '{nazev}' byl přidán.\n")

# Funkce pro zobrazení všech úkolů

def zobrazit\_ukoly():

    if len(ukoly) == 0:

        print("Žádné úkoly nejsou k dispozici.\n")

    else:

        print("Seznam úkolů:")

        for i in range(len(ukoly)):

            u = ukoly[i]

            print(f"{i+1}. {u['nazev']} - {u['popis']}")

        print()

# Funkce pro odstranění úkolu

def odstranit\_ukol():

    zobrazit\_ukoly()

    if len(ukoly) == 0:

        return

    cislo = input("Zadejte číslo úkolu, který chcete odstranit: ")

    if cislo.isdigit():

        cislo = int(cislo)

        if 1 <= cislo <= len(ukoly):

            smazany = ukoly.pop(cislo - 1)

            print(f"Úkol '{smazany['nazev']}' byl odstraněn.\n")

        else:

            print("Neplatné číslo úkolu.\n")

    else:

        print("Zadejte platné číslo.\n")

# Hlavní menu programu

def hlavni\_menu():

    while True:

        print("Správce úkolů - Hlavní menu")

        print("1. Přidat nový úkol")

        print("2. Zobrazit všechny úkoly")

        print("3. Odstranit úkol")

        print("4. Konec programu")

        volba = input("Vyberte možnost (1-4): ")

        if volba == "1":

            pridat\_ukol()

        elif volba == "2":

            zobrazit\_ukoly()

        elif volba == "3":

            odstranit\_ukol()

        elif volba == "4":

            print("Program končí.")

            break

        else:

            print("Neplatná volba. Zkuste to znovu.\n")

# Spuštění programu

hlavni\_menu()

**2. část**

**Testovací případy- hlavni\_menu**

**Pozitivní testování**

**Název testovacího případu**: TC01 - Přidání úkolu s názvem a popisem

**Popis:** Tento test ověřuje, zda program správně přidá nový úkol do seznamu, pokud uživatel zadá běžný název a popis s číselnou možností 1-4. Cílem je ověřit správné uložení a následné zobrazení zadaných dat

**Vstupní podmínky:** Program zobrazuje hlavní menu

**Kroky testu**:

* Spustit program a zobrazit hlavní menu
* Vybrat možnost **1** – „Přidat nový úkol“
* Do pole **název úkolu** zadat např. „chleba“
* Do pole **popis úkolu** zadat např. „koupit chleba a máslo“
* Potvrdit zadání „enter“
* Vybrat možnost **2** – „Zobrazit všechny úkoly“ a zkontrolovat, zda se nový úkol objevil

**Očekávaný výsledek**: Po zadání názvu úkolu program úkol správně uloží do seznamu, vypíše potvrzující zprávu „úkol „**chleba**“ byl přidán“

**Skutečný výsledek**: Úkol byl do programu napsán, uložen a správně se zobrazil dle očekávání v seznamu. Vše proběhlo v pořádku.

**Stav:** Pass

**Poznámky**: Tento případ je důležitý protože ověřuje test, který potvrzuje, že systém funguje tak, jak byl navržen pro běžné používání

**Negativní testování**

**Název testovacího případu**: TC02- Zadání neplatné volby v hlavním menu

**Popis:** Tento test ověřuje, jak program reaguje, když uživatel v hlavním menu nezadá číslo 1–4, ale jiný, neplatný vstup (např. text xxxx, číslo 5 a dál). Cílem testu je prověřit odolnost programu vůči chybným vstupům a jeho schopnost zobrazit správnou výstrahu bez pádu

**Vstupní podmínky**: Program je spuštěn a hlavní menu se zobrazuje

**Kroky  testu** :

* Spusťte program
* Ověřte, že se zobrazuje hlavní menu s nabídkou voleb
* Po zobrazení hlavního menu zadat místo čísla volbu: xxxx
* Pozorovat výstup programu

**Očekávaný výsledek: Program** zobrazí zprávu **„Neplatná volba, zkuste to znovu.“** a znovu zobrazí hlavní menu se všemi možnostmi 1–4. A nabízí zadat volbu správných čísel znovu

**Skutečný výsledek**: Program vypíše zprávu **„Neplatná volba, zkuste to znovu.“** a znovu nabídne hlavní menu s nabídkou voleb. Program nezamrzne, nespadne, nezobrazí v terminálu žádný problém a pokračuje správně dál

**Stav** : Pass

**Poznámky:** Tento případ je důležitý protože testuje **negativní chování uživatele** – zadání nesprávného vstupu. Takové očekávání je v praxi běžné a aplikace by ho měla umět zvládnout bez chyby

**Název testovacího případu**: TC03- Ponechané prázdné pole v hlavním menu, potvrzené klávesou Enter

**Popis:** Tento test ověřuje, jak program reaguje, když uživatel v hlavním menu nestiskne žádné číslo (1–4), ale pouze Enter – tedy ponechá vstupní pole prázdné. Cílem je otestovat ošetření prázdného vstupu

**Vstupní podmínky:** Program je spuštěn a nachází se v hlavním menu

**Kroky testu:**

* Spustit program
* Zobrazí se hlavní menu
* **Nepřidávat žádný text, číslo**, pouze stisknout Enter (prázdný vstup)
* Pozorovat výstup

**Očekávaný výsledek**: Program vypíše zprávu **„Neplatná volba, zkuste to znovu.“** a znovu zobrazí hlavní menu

**Skutečný výsledek**: Program zobrazil zprávu **„Neplatná volba, zkuste to znovu.“** a znovu nabídl hlavní menu. Program běží dál, nedošlo k chybě, nabízí možnost znovu přidat nový úkol

**Stav**: Pass

**Poznámky**: Tento test je důležitý, protože prázdný vstup je častou chybou uživatelů. Program reagoval správně – nezamrzl ani nespadl

**Hraniční testování**

**Název testovacího případu**: TC04 – Přidání úkolu s velmi dlouhým názvem a popisem

**Popis**: Tento test ověřuje, jak se program chová při zadání extrémně dlouhého názvu a popisu úkolu. Cílem je zjistit, zda program zvládne zpracovat a uložit dlouhý text bez chyby nebo pádu, a zda se úkol správně zobrazí v seznamu

**Vstupní podmínky**: Program zobrazuje hlavní menu

**Kroky  testu**:

* Spustit program a zobrazit hlavní menu.
* Vybrat možnost číslo **1** – „Přidat nový úkol“
* Do pole **název úkolu** zadat dlouhý text, byl použit dlouhý počet nesmyslných znaků písmen a číslic, byl také vložen dlouhý text z jednoho článku
* Do pole **popis úkolu** zadat také velmi dlouhý text (např. 300 znaků). Po zadání „enter“ se zobrazí hláška, že úkol „xy byl přidán“
* Po uložení úkolu zvolit možnost **2** – „Zobrazit všechny úkoly“
* Zobrazí se seznam úkolu: 1. „A zvolený text v rámci testu „

**Očekávaný výsledek:**  
Program přijme dlouhý název a popis, přidá úkol do seznamu a správně ho vypíše bez zkrácení nebo chyby. Nezhroutí se ani nezamrzne

**Skutečný výsledek:**  
Program úspěšně přijal dlouhé texty, uložil je a zobrazil celý název i popis. Nevykázal žádné chyby

**Stav: Pass**

**Poznámky: tento případ je důležitý, protože testuje hraniční případ kdy je zadáno záměrně nebo omylem nadměrně dlouhý text. Může nám pomoct odhalit případné omezení ve vstupním poli**

**Testovací případy- pridat\_ukol**

**Pozitivní testování**

**Název testovacího případu:** TC05- Přidání úkolu se správně vyplněným názvem a popisem

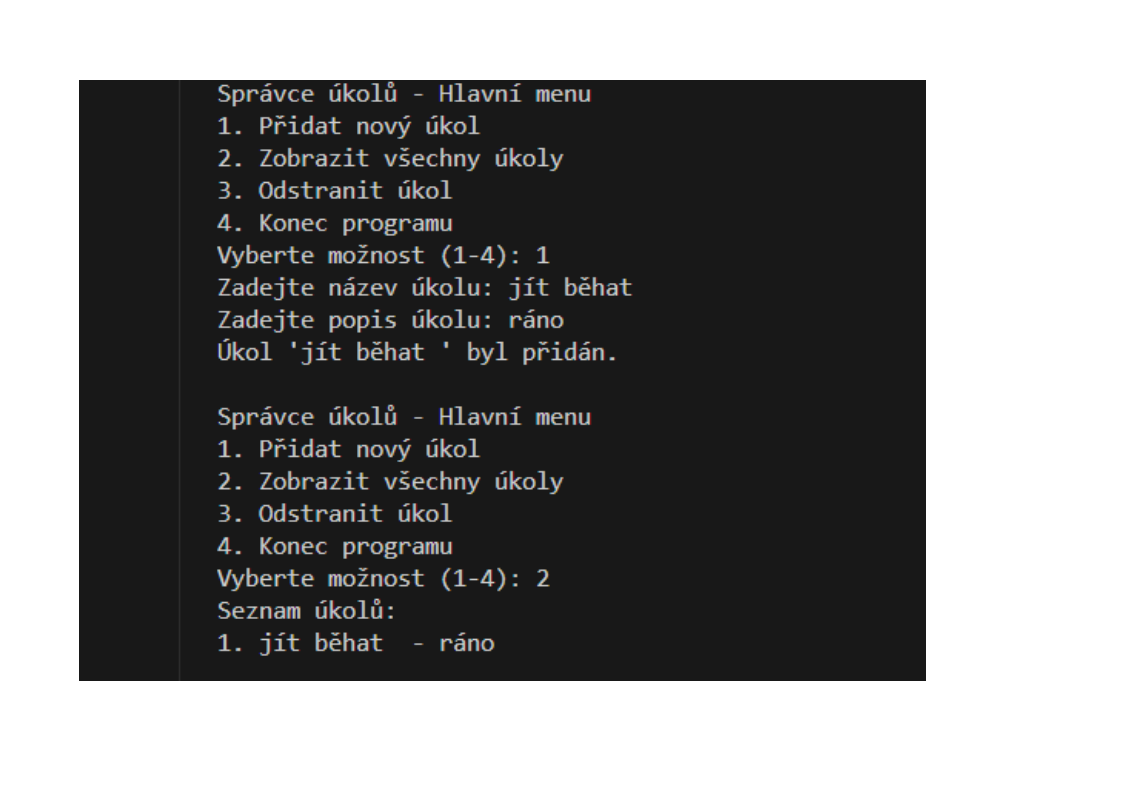
**Popis:** Ověření, že funkce „pridat\_ukol“ správně uloží úkol, když je zadán platný název a popis

**Vstupní podmínky:** Program je spuštěn zobrazuje zadané úkoly

**Kroky k testu:**

* Vyber možnost 1 – „Přidat nový úkol“
* Zadej název úkolu: jít běhat
* Zadej popis úkolu: ráno
* Enter potvrdit úkol
* Zobrazit seznam úkolů zadat možnost 2
* Zobrazí se správně zadaný úkol i s popisem

**Očekávaný výsledek**: Úkol je přidán a zobrazen v seznamu správně



**Skutečný výsledek**: Úkol se přidal a byl správně zobrazen

**Stav**: Pass

**Poznámky**: Tento případ je důležitý protože ověřuje test, který potvrzuje, že systém funguje tak, jak byl navržen pro běžné používání

**Negativní testování**

**Název testovacího případu**: TC06 – Přidání úkolu bez názvu a popisu bez názvu

**Popis:** Tento případ ověřuje, jak program zareaguje na situaci, kdy uživatel při přidávání úkolu **nechá vstupní pole prázdné a nezadá žádný název** (např. stiskne Enter) stejně tak se týká popisu úkolu, který ponechá také bez názvu. Cílem testu je zjistit, zda program tuto situaci správně ošetří nebo zda umožní přidání neúplného, prázdného úkolu

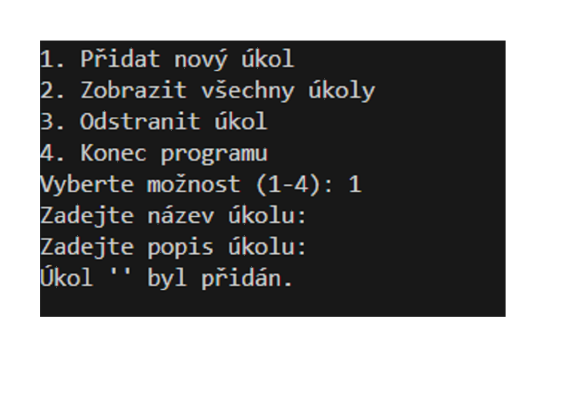
**Vstupní podmínky**: Program je spuštěn a hlavní menu se zobrazuje

**Kroky k testu:**

* Spustit program a zobrazit hlavní menu
* Zvolit možnost **1** – „Přidat nový úkol“
* Do pole **název úkolu a popis úkolu** **nezadat nic** zmáčknout Enter => potvrdit
* Zvolit možnost **2** – „Zobrazit všechny úkoly“

**Očekávaný výsledek:** Program by měl upozornit na prázdné pole a neuvedení názvu a popisu úkolu. Upozornit, že vyplnění pole je povinné

**Skutečný výsledek**: Program povolil přidání prázdného pole a zobrazuje se tam pouze znak apostrof :



**Stav**: Fail (program by měl zobrazit povinnost vyplnit pole)

**Poznámky**: Tento případ ověřuje, jak test upozorní na chybějící validaci vstupu. Uživatel může omylem přidat úkol bez názvu, což může později způsobit zmatek při jeho čtení nebo mazání. Doporučuje se přidat kontrolu, která nedovolí přidat úkol, pokud název chybí

**Hraniční testování**

**Název testovacího případu**: TC07- Bez zadání úkolu zvolena volba 3 – Smazat úkol

**Popis**: Cílem testu je ověřit, zda program správně reaguje, když se uživatel pokusí smazat úkol, aniž by byl dříve přidán jakýkoli úkol. Zkusit jak se program zachová když bude zadán pokus o smazání úkolu aniž by byl předtím zadán

**Vstupní podmínky:** Program byl právě spuštěn a nebyl přidán žádný úkol

**Kroky testu**:

* Pustit program
* Zobrazí se hlavní menu
* Zvolit možnost **3 – Odstranit úkol**
* Pozorovat, co program vypíše

**Očekávaný výsledek**: Program zobrazí hlášení **„Není co odstranit, seznam je prázdný.“** a vrátí se zpět do hlavního menu

**Skutečný výsledek**: Program zobrazí hlášení **„Není co odstranit, seznam je prázdný.“pokračoval dál ve funkci**

**Stav**: Pass

**Poznámky:** Tento test je důležitý, protože chrání program před chybami při práci s prázdným seznamem úkolů

**Testovací případy - zobrazit\_ukoly**

**Pozitivní testování**

**Název testovacího případu**: TC08 - Zobrazení všech úkolů

**Popis**: Cílem testu je ověřit zda funkce „zobrazit úkoly“ zobrazí a vypíše správně úkoly, které byly v minulosti přidány

**Vstupní podmínky**: Program byl spuštěn, byly vloženy úkoly s popisem, takže seznam obsahuje přidané úkoly

**Kroky testu**:

* Spustit program a zobrazit hlavní menu
* Vybrat možnost **1 – Přidat nový úkol**
* Zadáme název: Oslava
* Zadáme popis: koupit dárek a květiny
* Opět vybrat možnost **1 – Přidat další nový úkol**
* Zadat název: Malování
* Zadat popis: koupit barvu, štětec
* Vybrat možnost **2 – Zobrazit všechny úkoly**

**Očekávaný výsledek**: Program vypíše seznam všech úkolů ve správném pořadí

**Skutečný výsledek**: zobrazil přesně tento výstup, bez chyb

**Stav**: Pass

**Poznámky:** Test ověřuje, že se úkoly zobrazují ve stejném pořadí, v jakém byly zadány. Uživatel má přehled o zadaných úkolech v programu

**Testovací případy - odstranit\_ukol**

**Pozitivní testování**

**Název testovacího případu**: TC09 - Odstranění existujícího úkolu

**Popis**: Cílem testu je ověřit, že funkce „odstranit úkol“ správně smaže konkrétní úkol ze seznamu na základě čísla zadaného uživatelem

**Vstupní podmínky**: Program byl spuštěn, byly vloženy úkoly s popisem, takže seznam obsahuje přidané úkoly

**Kroky testu**:

* Spustit program a zobrazit hlavní menu
* Vybrat možnost **1 – Přidat nový úkol**
* Zadáme název: Oslava
* Zadáme popis: koupit dárek a květiny
* **Zobrazí se hláška, že byl úkol přidán a otevře se opět hlavní menu**
* **Zadáme** možnost **3 – Odstranit úkol**
* Zobrazí se seznam úkolů s čísly a hláška zadejte číslo úkolu, který chcete odstranit
* Zadání čísla **1** (odpovídající prvnímu úkolu)
* Potvrdit odstranění



**Očekávaný výsledek:** Program zobrazí vypsané úkoly a po vložení čísla 3 vypíše hlášku, že byl úkol odstraněn a úkol již v seznamu nebude

**Skutečný výsledek**: Úkol byl úspěšně odstraněn, program zobrazil potvrzení a úkol už není v seznamu

**Stav**: Pass

**Poznámky:** Cílem testu je ověřit, že mazání funguje správně

### Negativní testování

**Název testovacího případu: TC10 -** Pokus o smazání úkolu při prázdném seznamu úkolů

**Popis:**  
Cílem testu je ověřit, jak program reaguje na pokus o odstranění úkolu, když seznam je prázdný. Test kontroluje, zda program zobrazí vhodnou hlášku a nespadne

**Vstupní podmínky:**  
Program byl spuštěn. Seznam je prázdný – žádný úkol nebyl přidán

**Kroky testu:**

* + Spustit program „hlavní měnu“
  + V hlavním menu zvolit možnost **3** (Odstranit úkol)

**Očekávaný výsledek:**  
Program vypíše hlášku **„Žádné úkoly nejsou k dispozici.“** a vrátí se zpět do hlavního menu bez jakékoli chyby

**Skutečný výsledek:**  
Program opravdu vypíše hlášku **„úkoly nejsou k dispozici.“** a zobrazí hlavní menu znovu. Nic nespadlo, vše proběhlo hladce

**Stav:** **Pass**

**Poznámky:**  
Tento test prokázel, že uživatel nemůže omylem rozbít aplikaci tím, že se pokusí smazat něco, co neexistuje

**Hraniční testování**

**Název testovacího případu:** TC11- Zadání neexistujícího čísla při mazání úkolu

**Popis:** Tento test ověřuje, jak program reaguje, když uživatel zadá číslo úkolu, který **neexistuje**

**Vstupní podmínky:** Program byl spuštěn. V seznamu úkolů se nachází několik zadaných úkolů

**Kroky testu:**

* Spustit program a zobrazit hlavní menu
* Vybrat možnost **1 – Přidat nový úkol**
  + Zadání názvu: Nákup
  + Zadání popisu: Koupit chleba
* Vybrat možnost **3 – Odstranit úkol**
* Zobrazí se seznam úkolů:
* Zadat číslo **5** jako pokus o odstranění
* Pozorovat výstup programu

**Očekávaný výsledek**: Úkol zůstane zachován v seznamu a nic se nesmaže. Program zobrazí hlášku „ Neplatné číslo úkolu“

**Skutečný výsledek**: Program opravdu zobrazil hlášku o neplatné volbě a úkol nebyl smazán.

**Stav**: Pass

**Poznámky:** Tento test ověřuje, že program správně reaguje na vstup mimo rozsah