

Dokumentacja do projektu „System zarządzania magazynem”

System umożliwia operowanie na produktach magazynu (dodawanie, usuwanie, aktualizowanie stanu, wyszukiwanie wraz z sortowaniem po wynikach) oraz generowanie raportów na temat stanu magazynu. System posiada interfejs graficzny, w którym można wykonywać wspomniane operacje.

Plik main:

Rozpoczyna działanie całego systemu tworząc obiekt typu WarehouseApp i uruchamiając w nim interfejs graficzny, na którym może działać użytkownik.

Klasa Item:

Pozwala na tworzenie obiektów przedstawiających produkty w magazynie. Posiada gettery i settery do wszystkich atrybutów tej klasy.

Klasa WarehouseApp:

Jest odpowiedzialny za interfejs graficzny i wywoływanie odpowiednich metod w zależności od tego co zrobi użytkownik.

Metody:

- `create_widgets()` – tworzy elementy GUI i wiąże z poszczególnymi elementami metody, które mają wywołać
- `search()` – sprawdza, czy jest wybrana opcja szukania po ID, czy po nazwie i w zależności od wyboru wyświetlane są produkty na podstawie wpisanego przez użytkownika tekstu
- `list_to_tree()` – usuwa istniejące elementy z tabeli produktów wyświetlanej użytkownikowi i dodaje nowe elementy do wyświetlenia. W wypadku braku istnienia jakichkolwiek nowych elementów zgłaszany jest błąd użytkownikowi
- `refresh_tree()` – odświeża tabelę produktów i wyświetla wszystkie istniejące w bazie
- `add()` – tworzy obiekt typu Item i przekazuje go do bazy w celu dodania
- `edit()` – pozwala na wybranie jednego elementu i zaktualizowanie go. Jeżeli więcej niż jeden lub żaden produkt nie jest wybrany, jest zgłaszany błąd użytkownikowi. Wyświetlane jest okno ze stworzonym obiektem ItemDialog, służącym do aktualizowania produktów. Nowy stworzony obiekt typu Item jest przekazywany bazie w celu nadpisania istniejącego obiektu
- `delete()` – pozwala na wybranie jednego lub więcej elementów i usunięcie ich. Jeżeli żaden produkt nie został wybrany, jest zgłaszany błąd użytkownikowi. Selekcja elementów jest przekazywana bazie w celu faktycznego usunięcia
- `report_choice()` – wywołuje konstruktor klasy ReportDialog służący do generowania raportów
- `sort_tree()` – sortuje produkty po wybranej kolumnie. Jeżeli elementy są już posortowane w danej kolejności to odwraca kolejność sortowania

Klasa DatabaseManager:

Jest odpowiedzialny za stworzenie i połączenie się z bazą produktów magazynu oraz wykonywanie operacji bezpośrednio na bazie.

Metody:

- `init_db()` – tworzy bazę i dodaje do niego przykładowe produkty
- `fetch_all_items()` – zwraca wszystkie elementy bazy
- `search_by_id()` – zwraca elementy bazy z wartością Id zgodną z kryterium użytkownika
- `search_by_name()` – zwraca elementy bazy z wartością Name podobnym do wyszukania użytkownika
- `insert_item()` – dodaje do bazy element
- `edit_item()` – aktualizuje w bazie element o wybranym Id
- `delete_items()` – usuwa z bazy element o wskazanych Id
- `close()` – zamyka połączenie z bazą
- `fetch_items_by_price()` – zwraca elementy z bazy z ceną pomiędzy wskazanym minimum i maximum
- `fetch_items_by_amount()` – zwraca element z bazy z dostępną ilością pomiędzy wskazanym minimum i maximum

Plik `digit_check`:

Zapewnia, że wpisana wartość przez użytkownika jest liczbą całkowitą (w metodzie `digit_input_check()`), albo liczbą wymierną (w metodzie `digit_dot_input_check()`)

Klasa `ItemDialog`:

Jest odpowiedzialny za interfejs graficzny okna aktualizowania produktu i aktualizacji atrybutów dostarczonego mu obiektu typu `Item`

Metody:

- `create_entry()` – tworzy przycisk oznaczający chęć aktualizacji atrybutu oraz pole tekstowe do nadpisania istniejącej wartości
- `toggle_entry()` – włącza lub wyłącza możliwość nadpisania danego atrybutu
- `submit()` – nadpisuje wybrane atrybuty obiektu typu `Item` i zwraca zaktualizowany obiekt

Klasa `ReportDialog`:

Jest odpowiedzialny za interfejs graficzny okna generowania raportów i stworzenie samego raportu.

Metody:

- `update_entries()` – włącza lub wyłącza możliwość wpisania minimalnej i maksymalnej poszukiwanej wartości ceny lub ilości produktu w zależności od wybranej opcji
- `submit()` – wywołuje odpowiednią metodę do generowania raportu w zależności od wybranej opcji. Jeżeli nie została wybrana ścieżka do zapisania pliku zostaje zgłoszony błąd użytkownikowi
- `generate_report()` – tworzy plik we wskazanym miejscu z zapisanymi wskazanymi elementami bazy
- `generate_report_default()` – generuje raport ze wszystkimi produktami
- `generate_report_by_price()` – generuje raport z produktami o cenie między wskazaną minimalną i maksymalną ceną
- `generate_report_by_amount()` – generuje raport z produktami o dostępnej ilości między wskazaną minimalną i maksymalną ilością