
Knut - Aplikacja do Tworzenia Testów Documentation

Wydanie 0.1

Wiktor Idzikowski

April 28 2010

Spis treści

1	Knut - Główna klasa z logiką programu	3
2	QuestionFrame - Klasa ramki pytania	9
3	AnswerFrame - Klasa ramki w której znajdują się możliwe odpowiedzi	11
4	Opis klas bazy danych	13
4.1	Klasa opisująca tabelę testów	13
4.2	Klasa opisująca tabelę elementów testu	13
4.3	Klasa opisująca tabelę pytań	14
4.4	Klasa opisująca tabelę możliwych odpowiedzi	14
5	Indeksy i tabele	15
	Indeks modułów	17
	Indeks	19

Jest to dokumentacja techniczna przeznaczona dla programistów, którzy chcą zrozumieć jak działa Knut - edytor testów. Składa się ona z opisu klas, atrybutów i metod.

Program został napisany w języku [Python](#), przy użyciu wrappera [PyGTK](#) biblioteki [GTK+](#). Do części okienek wykorzystywany był kreator interfejsów użytkownika [Glade](#).

Zawartość:

- *Knut* - Główna klasa z logiką programu
- *QuestionFrame* - Klasa ramki pytania
- *AnswerFrame* - Klasa ramki w której znajdują się możliwe odpowiedzi
- *Opis klas bazy danych*
 - Klasa opisująca tabelę testów
 - Klasa opisująca tabelę elementów testu
 - Klasa opisująca tabelę pytań
 - Klasa opisująca tabelę możliwych odpowiedzi

Knut - Główna klasa z logiką programu

class Knut ()

Knut - Główna klasa

Uruchamia okno główne i wczytuje konfiguracje glade

`__init__()`

Inicjalizacja okna głównego i bazy danych

TEST_CATEGORIES

Lista dostępnych kategorii testów:

- Różności
- Matematyka
- Informatyka
- Geografia
- Historia

add_or_update_item ()

Aktualizuje lub tworzy aktualnie widoczne pytanie w bazie danych

answers_download (widget, data)

Pobieranie odpowiedzi danego ucznia

Argumenty:

- widget - widżet, który wywołał metodę
- data - dodatkowe dane zdefiniowane przy łączeniu

checkdb ()

Sprawdza czy istnieje baza danych

- jeśli istnieje to przygotowuje połączenie
- jeśli bazy nie ma to tworzy tabele i łączy z bazą

clearMainVbox ()

Usuwa widżety z głównego okna programu

destroy_main_window (*widget, data=None*)

Metoda uruchamiana przy zdarzeniu zamknięcia programu

Zapisuje ustawienia serwera do pliku settings.txt

Argumenty

- widget - widżet, który wywołał metodę
- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

get_current_test (*offset*)

Zapisuje aktualnie zaznaczony test w zmiennej self.test

Argumenty

- offset - offset przy stronicowaniu np. na drugiej stronie wybieramy 3 test czyli 13 w bazie danych

get_item_type (*id*)

Zamienia id typu odpowiedzi na zmienną tekstową

Argumenty

- id - liczba całkowita:

Wartości zwracane

- 1 -> 'one' - pytanie jednokrotnego wyboru
- 2 -> 'mul' - pytanie wielokrotnego wyboru
- 3 -> 't/f' - pytanie typu prawda/fałsz

get_selected_results (*offset*)

Zapisuje id aktualnie zaznaczonego ucznia w zmiennej self.result_username

Argumenty

- offset - offset przy stronicowaniu np. na drugiej stronie wybieramy 3 wyniki czyli 13 w bazie danych

load_server_conf ()

Wczytuje konfigurację serwera z pliku settings.txt do zmiennej self.server_conf

next_btn_clicked (*widget, data=None*)

Zmienia aktualnie wyświetlane pytanie na następne

Argumenty

- widget - widżet, który wywołał metodę
- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

post_files (*tar_file_path, answers_file_path*)

Wysyła archiwum z testem i plik xml z odpowiedziami na serwer

Argumenty

- tar_file_path - zmienna tekstowa, ścieżka do archiwum testu
- answers_file_path - zmienna tekstowa, ścieżka do pliku z odpowiedziami

prepare_img (*img_path, dir_path, prefix*)

Kopiuje obrazek do katalogu dla danego testu i pytania

Argumenty

- img_path - zmienna tekstowa, ścieżka do obrazka

- `dir_path` - zmienna tekstowa, ścieżka do katalogu, do którego obrazek będzie kopiowany
- `prefix` - prefix nazwy pliku obrazka w celu uniknięcia nadpisania obrazków o tej samej nazwie

prev_btn_clicked (*widget, data=None*)

Zmienia aktualnie wyświetlane pytanie na poprzednie

Argumenty

- `widget` - widżet, który wywołał metodę
- `data` - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

read_config (*existingConfig, warning=None*)

Otwiera okno dialogowe, w którym podaje się ustawienia testu

Argumenty

- `existingConfig` - obiekt testu, danymi z tego testu uzupełniane są pola okna dialogowego
- `warning` - opcjonalnie, zmienna logiczna, jeśli `warning=True` to wyświetla ostrzeżenie o nie wypełnieniu wszystkich pól

save_error (*text*)

W przypadku błędu w komunikacji z serwerem zapisuje odpowiedź serwera do pliku `error.html`

Argumenty

- `text` - zmienna tekstowa, treść błędu

show_about_window (*widget*)

Otwiera okno z podstawowymi informacjami o programie

Argumenty

- `widget` - widżet, który wywołał metodę

show_item (*item=None*)

Wyświetla bieżące pytanie w oknie głównym programu

Argumenty

- `item` - obiekt bieżącego pytania

show_msg (*msg*)

Otwiera okno dialogowe

Argumenty

- `msg` - zmienna tekstowa, wiadomość do wyświetlenia w oknie

show_server_config_window (*warning=None*)

Otwiera okno z danymi serwera

Argumenty

- `warning` - opcjonalnie, zmienna logiczna - jeśli `waning=True` pokazuje ostrzeżenie, że wszystkie pola muszą być wypełnione

test_browse (*widget=None, data=0*)

Przeglądanie testów z bazy danych

Argumenty

- `widget` - opcjonalnie, widżet, który wywołał metodę
- `data` - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

test_browse_from_server (*widget=None, data=0, title="", type=""*)

Wyświetla dane pobrane z serwera w formie tabelki

Argumenty

- widget - opcjonalnie, widżet, który wywołał metodę
- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu
- title - opcjonalnie, tytuł wyświetlany nad tabelką
- type - opcjonalnie, typ tabelki (wyniki, odpowiedzi bądź testy)

test_delete (*widget=None, data=None*)

Otwiera okno z potwierdzeniem usunięcia testu

Po potwierdzeniu usuwa z bazy i dysku wybrany test

Argumenty

- widget - opcjonalnie, widżet, który wywołał metodę
- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

test_delete_from_server (*widget, data*)

Wysyła żądanie usunięcia testu do serwera

Po otrzymaniu odpowiedzi wyświetla komunikat o statusie żądania

Argumenty

- widget - widżet, który wywołał metodę
- data - dodatkowe dane zdefiniowane przy łączeniu

test_download (*widget, data*)

Pobiera z serwera test i zapisuje go do bazy danych i na dysku

Argumenty

- widget - widżet, który wywołał metodę
- data - dodatkowe dane zdefiniowane przy łączeniu

test_download_results_list (*widget, data*)

Pobranie listy wyników dla danego ucznia

Argumenty

- widget - widżet, który wywołał metodę
- data - dodatkowe dane zdefiniowane przy łączeniu

test_edit_settings (*widget=None, data=None*)

Otwiera okno umożliwiające edycję ustawień testu

Argumenty

- widget - opcjonalnie, widżet, który wywołał metodę
- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

test_list_download (*widget=None, data=None*)

Pobiera listę testów z serwera

Argumenty

- widget - opcjonalnie, widżet, który wywołał metodę

- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

test_new (*widget=None, data=None*)

Tworzy nowy test

Argumenty

- widget - opcjonalnie, widżet, który wywołał metodę
- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

test_open (*widget=None, data=None*)

Otwiera wybrany test

Argumenty

- widget - opcjonalnie, widżet, który wywołał metodę
- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

test_print (*widget=None, data=None*)

Drukuje wybrany test

Argumenty

- widget - opcjonalnie, widżet, który wywołał metodę
- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

test_upload (*widget=None, data=None*)

Zapisuje wybrany test do pliku xml, kompresuje test i oddzielnie odpowiedzi

Przygotowuje plik do wysłania na serwer

Argumenty

- widget - opcjonalnie, widżet, który wywołał metodę
- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

validate_input ()

Sprawdza czy aktualnie edytowane pytanie ma wypełnione wszystkie wymagane pola

QuestionFrame - Klasa ramki pytania

class QuestionFrame (*item*)

Klasa ramki pytania

__init__ (*item*)

Inicjalizuje okno pytania, uzupełnia pytanie danymi ze zmiennej *item*

Argumenty

- *item* - obiekt bieżącego pytania, jeśli brak to wyświetla puste pola do edycji

add_img (*img_filename*)

Zmienia lub wyświetla nowy obrazek

Argumenty

- *img_filename* - ścieżka do obrazka

img_button_clicked (*widget=None, data=None*)

Otwiera okno wyboru pliku i wyświetla wybrany obrazek w sekcji pytania

Argumenty

- *widget* - opcjonalnie, widżet, który wywołał metodę
- *data* - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

remove_img (*widget=None*)

Usuwa obrazek z pytania

Argumenty

- *widget* - opcjonalnie, widżet, który wywołał metodę

AnswerFrame - Klasa ramki w której znajdują się możliwe odpowiedzi

class AnswerFrame (*item=None*)

Klasa ramki w której znajdują się możliwe odpowiedzi

__init__(*item=None*)

Inicjalizuje ramkę odpowiedzi, uzupełnia pola danymi ze zmiennej *item*

Argumenty

- *item* - opcjonalnie, obiekt bierzącego pytania, jeśli brak to wyświetla puste pola do edycji

add_img (*img_filename, index*)

Dodaje lub zamienia obrazek przy danej odpowiedzi

Argumenty

- *img_filename* - ścieżka do pliku obrazka
- *index* - indeks odpowiedzi, w której zmienia się obrazek

answer_type_combo_changed (*widget=None, data=None*)

Wyświetla pola do edycji dla danego typu odpowiedzi

Argumenty

- *widget* - opcjonalnie, widżet, który wywołał metodę
- *data* - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

get_type_index (*type*)

Zamienia tekst typu odpowiedzi na reprezentację liczbową

Argumenty

- *type* - zmienna tekstowa:

Wartości zwracane

- 'one' -> 1 - pytanie jednokrotnego wyboru
- 'mul' -> 2 - pytanie wielokrotnego wyboru
- 't/f' -> 3 - pytanie typu prawda/fałsz

img_button_clicked (*widget=None, data=None*)

Otwiera okno wyboru obrazka do danej odpowiedzi

Argumenty

- widget - opcjonalnie, widżet, który wywołał metodę
- data - opcjonalnie, dodatkowe dane zdefiniowane przy łączeniu

remove_img (*widget, index*)

Usuwa wybrany obrazek

Argumenty

- widget - widżet, który wywołał metodę
- index - indeks odpowiedzi, w której usuwa obrazek

Opis klas bazy danych

4.1 Klasa opisująca tabelę testów

```
class Test (**kwargs)
    Klasa opisująca tabelę testów
    Opis pól:
    author
        Imię i nazwisko autora
    category
        Kategoria w jakiej test się znajduje
    instructions
        Instrukcje przydatne przy rozwiązywaniu testu
    item
        Relacje jeden do wielu z tabelą elementów testu
    password
        Hasło
    time
        Czas na ukończenie testu
    title
        Tytuł testu
    version
        Wersja testu
```

4.2 Klasa opisująca tabelę elementów testu

```
class Item (**kwargs)
    Klasa opisująca tabelę elementów testu
    option
        Relacja jeden do jednego z tabelą możliwych odpowiedzi
```

order
Kolejność elementu

question
Relacja jeden do jednego z tabelą pytań

test
Relacja z wiele do jednego z testem

type
Typ elementu

4.3 Klasa opisująca tabelę pytań

```
class Question (**kwargs)
    Klasa opisująca tabelę pytań

    img
        Nazwa obrazka

    item
        Relacja wiele do jednego z tabelą elementów testu

    text
        Tekst pytania
```

4.4 Klasa opisująca tabelę możliwych odpowiedzi

```
class Option (**kwargs)
    Klasa opisująca tabelę możliwych odpowiedzi

    correct
        Poprawność odpowiedzi

    img
        Nazwa obrazka

    item
        Relacja wiele do jednego z tabelą elementów testu

    text
        Tekst odpowiedzi
```

Indeksy i tabele

- *Index*
- *Module Index*
- *Search Page*

Indeks modułów

A

`AnswerFrame`, 11

D

`dbmodel`, 13

K

`Knut`, 3

Q

`QuestionFrame`, 9

Indeks

A

add_img() (AnswerFrame.AnswerFrame metoda), 11
add_img() (QuestionFrame.QuestionFrame metoda), 9
add_or_update_item() (Knut.Knut metoda), 3
answer_type_combo_changed() (AnswerFrame.AnswerFrame metoda), 11
AnswerFrame (moduł), 11
AnswerFrame (w klasie AnswerFrame), 11
answers_download() (Knut.Knut metoda), 3
author (dbmodel.Test atrybut), 13

C

category (dbmodel.Test atrybut), 13
checkdb() (Knut.Knut metoda), 3
clearMainVbox() (Knut.Knut metoda), 3
correct (dbmodel.Option atrybut), 14

D

dbmodel (moduł), 13
destroy_main_window() (Knut.Knut metoda), 3

G

get_current_test() (Knut.Knut metoda), 4
get_item_type() (Knut.Knut metoda), 4
get_selected_results() (Knut.Knut metoda), 4
get_type_index() (AnswerFrame.AnswerFrame metoda), 11

I

img (dbmodel.Option atrybut), 14
img (dbmodel.Question atrybut), 14
img_button_clicked() (AnswerFrame.AnswerFrame metoda), 11
img_button_clicked() (QuestionFrame.QuestionFrame metoda), 9
instructions (dbmodel.Test atrybut), 13
item (dbmodel.Option atrybut), 14

item (dbmodel.Question atrybut), 14
item (dbmodel.Test atrybut), 13
Item (w klasie dbmodel), 13

K

Knut (moduł), 3
Knut (w klasie Knut), 3

L

load_server_conf() (Knut.Knut metoda), 4

N

next_btn_clicked() (Knut.Knut metoda), 4

O

option (dbmodel.Item atrybut), 13
Option (w klasie dbmodel), 14
order (dbmodel.Item atrybut), 13

P

password (dbmodel.Test atrybut), 13
post_files() (Knut.Knut metoda), 4
prepare_img() (Knut.Knut metoda), 4
prev_btn_clicked() (Knut.Knut metoda), 5

Q

question (dbmodel.Item atrybut), 14
Question (w klasie dbmodel), 14
QuestionFrame (moduł), 9
QuestionFrame (w klasie QuestionFrame), 9

R

read_config() (Knut.Knut metoda), 5
remove_img() (AnswerFrame.AnswerFrame metoda), 12
remove_img() (QuestionFrame.QuestionFrame metoda), 9

S

save_error() (Knut.Knut metoda), 5
show_about_window() (Knut.Knut metoda), 5
show_item() (Knut.Knut metoda), 5
show_msg() (Knut.Knut metoda), 5
show_server_config_window() (Knut.Knut metoda), 5

T

test (dbmodel.Item atrybut), 14
Test (w klasie dbmodel), 13
test_browse() (Knut.Knut metoda), 5
test_browse_from_server() (Knut.Knut metoda), 5
TEST_CATEGORIES (Knut.Knut atrybut), 3
test_delete() (Knut.Knut metoda), 6
test_delete_from_server() (Knut.Knut metoda), 6
test_download() (Knut.Knut metoda), 6
test_download_results_list() (Knut.Knut metoda), 6
test_edit_settings() (Knut.Knut metoda), 6
test_list_download() (Knut.Knut metoda), 6
test_new() (Knut.Knut metoda), 7
test_open() (Knut.Knut metoda), 7
test_print() (Knut.Knut metoda), 7
test_upload() (Knut.Knut metoda), 7
text (dbmodel.Option atrybut), 14
text (dbmodel.Question atrybut), 14
time (dbmodel.Test atrybut), 13
title (dbmodel.Test atrybut), 13
type (dbmodel.Item atrybut), 14

V

validate_input() (Knut.Knut metoda), 7
version (dbmodel.Test atrybut), 13