

Laboratorium 8 –Programowanie GUI w Python

Języki skryptowe

Cele dydaktyczne

1. Zapoznanie się z programowaniem graficznych interfejsów użytkownika w języku Python.

Wprowadzenie

Zadanie z niniejszej listy dotyczy opracowania aplikacji z graficznym interfejsem użytkownika służącej do przeglądania logów, która będzie oparta o wzorzec master-detail.

Master-detail to wzorzec projektowy, który opisuje związek między dwoma zbiorami danych, w którym jeden z nich (master) jest głównym źródłem danych, a drugi (detail) jest powiązany z nim i zawiera bardziej szczegółowe informacje. W kontekście aplikacji z interfejsem użytkownika, master-detail jest często stosowanym rozwiązaniem, w ramach którego lista głównych elementów jest wyświetlana w widoku master, a szczegółowe informacje na temat każdego z tych elementów są wyświetlane w widoku detail.

Zadania

Korzystając z wybranego [zestawu narzędzi do programowania GUI](#) oraz programów z poprzednich list, napisz program z interfejsem graficznym do przeglądania logów serwera HTTP lub SSH. Program powinien być wyposażony w następujące funkcjonalności:

1. Program powinien pozwolić na wczytanie pliku z logami.
2. Po wczytaniu pliku, program powinien wyświetlić wiersze zapisane w pliku w formie listy.
3. Program powinien implementować wzorzec master-detail w zakresie przeglądania listy logów oraz wyświetlania szczegółów dotyczących konkretnego loga.
 - a. Zbiór danych *master* stanowi lista logów, wyświetlana może być surowa treść loga, np. ucięta do 30 znaków i zakończona wielokropkiem.

- b. Zbiór danych *detail* stanowią atrybuty dotyczące konkretnego loga (np. adres hosta, data, czas, kod statusu, etc.).
 - c. Wybranie przez użytkownika loga z listy powinno uaktualnić treści w komponentach wyświetlających szczegóły dotyczące konkretnego loga.
4. Program powinien umożliwiać filtrowanie listy logów ze względu na wybrany przedział czasowy.
5. Program powinien być wyposażony w przyciski "Następny" i "Poprzedni" pozwalające przeglądać kolejne logi. Zachowanie powiązane z kliknięciem tych przycisków powinno być równoważne wybraniu przez użytkownika kolejnego/poprzedniego loga. W przypadku pierwszego/ostatniego loga, przyciski powinny być nieaktywne.

Podczas projektowania aplikacji, zadbaj o to, aby dobrać odpowiednie widżety do wyświetlania potrzebnych informacji (etykiety/pola tekstowe dla danych tekstowych, pola numeryczne dla danych liczbowych, pola wyboru daty dla dat. itd.).

Zadbaj o to, aby klasy i funkcje związane z przetwarzaniem logiki aplikacji były odseparowane od klas i funkcji związanych z interfejsem użytkownika.

Za luźną referencję może posłużyć poniższy szkic:

Log browser

/home/student/NASA **Open**

From **2019-01-22** To **2019-01-22**

66.111.54.249 - - [22/Jan/2019:03:56:51 +0330] "GET /image/509..."
69.122.54.139 - - [22/Jan/2019:03:57:51 +0330] "GET /index.html..."
178.253.33.51 - - [22/Jan/2019:03:56:20 +0330] "GET /settings/logo..."
54.36.149.70 - - [22/Jan/2019:03:56:22 +0330] "GET /filter/b2..."
178.253.33.51 - - [22/Jan/2019:03:56:23 +0330] "GET /image/32574?"

Remote host **66.111.54.249**

Date **2019-01-22**

Time **03:56:51** Timezone: **Asia/Tehran**

Status code: **200** Method: **GET**

Resource: **/image/50995/productModel/200x200**

Size: **2244 Bytes**

Previous **Next**