

Poszukiwanie bibliotek o określonej funkcjonalności

lab 5

1 Wprowadzenie

Biblioteka w języku Python to zbiór modułów oferujących gotowe funkcje, klasy i narzędzia, które mogą znacząco przyspieszyć rozwój aplikacji. Wykorzystując gotowe rozwiązania, oszczędzamy czas potrzebny na implementację, a równocześnie bazujemy na przetestowanym kodzie społeczności lub profesjonalistów.

W ramach tego laboratorium należy wybrać dowolną dziedzinę (np. przetwarzanie danych, generowanie wykresów, obsługa plików PDF, sieci neuronowe, przetwarzanie języka naturalnego, web scraping) i znaleźć co najmniej dwie biblioteki w języku Python, które rozwiązują typowe problemy z tego obszaru.

2 Cel zadania

1. Poszerzenie wiedzy o ogólnodostępnych bibliotekach Python rozszerzających możliwości języka w określonej dziedzinie.
2. Umiejętnie wyszukiwanie i ocena przydatności bibliotek w rozwiązywaniu konkretnych problemów.
3. Ćwiczenie umiejętności instalacji, konfiguracji i demonstracji działania wybranych bibliotek.
4. Dokumentacja znalezionych rozwiązań w formie krótkiego raportu.

3 Zakres zadania

1. Wybór dziedziny lub obszaru zastosowań, który Cię interesuje (np. wizualizacja danych, przetwarzanie języka naturalnego, web scraping, itp.).
2. Wyszukanie co najmniej dwóch bibliotek w języku Python, które oferują interesującą Cię funkcjonalność.
3. Zapoznanie się z dokumentacją i przykładowymi zastosowaniami tych bibliotek.
4. Przygotowanie krótkich przykładów kodu (po 1–2 przykłady na bibliotekę) ilustrujących podstawowe możliwości.
5. Zamieszczenie kodu w repozytorium GitHub wraz z krótkim wyjaśnieniem.

4 Instrukcje – krok po kroku

4.1 Wybór bibliotek

1. Zastanów się, w jakim obszarze chcesz poszukiwać bibliotek (np. analiza tekstu, przetwarzanie obrazów, generowanie raportów, itd.).
2. Korzystając z wyszukiarek internetowych lub serwisów takich jak PyPI, GitHub, zidentyfikuj kilka bibliotek, które oferują funkcje potrzebne do realizacji wybranego celu.
3. Obejrzyj przykładowe projekty, dokumentację oraz popularność/wspieranie projektu (liczba gwiazdek na GitHub, częstotliwość aktualizacji).

4.2 Instalacja i pierwsze kroki

1. Zainstaluj wybrane biblioteki przy użyciu `pip` lub `conda` (np. `pip install nazwa_biblioteki`).
2. Przygotuj prosty skrypt (lub notatnik Jupyter), w którym zimportujesz biblioteki i wypróbowujesz kilka podstawowych funkcji.

4.3 Przykłady użycia

1. Dla każdej z bibliotek opracuj 1–2 demonstracyjne przykłady (np. minimalny kod prezentujący najważniejsze możliwości).
2. Umieść kod w repozytorium (np. w folderze `examples`).
3. Dodaj krótkie komentarze lub docstringi, by wyjaśnić zasadę działania.

4.4 Raport podsumowujący

1. W pliku `raport.md` lub `raport.pdf` opisz wybrane biblioteki (nazwa, przeznaczenie, główne funkcje).
2. Przedstaw kluczowe zalety każdej z nich (np. prosta konfiguracja, dobre wsparcie społeczności, bogata dokumentacja).
3. Jeśli zauważysz ograniczenia (np. brak wsparcia dla starszych wersji Pythona), również o tym wspomnij.
4. Dołącz linki do dokumentacji i repozytoriów bibliotek.

5 Przesyłanie zadania

1. Wgraj opracowany kod (folder `examples` z przykładowymi skryptami) do repozytorium na GitHub.
2. Dodaj plik raportu (`raport.md` lub `raport.pdf`).
3. Link do repozytorium oraz pliki załącz w zadaniu na Moodle.

6 Kryteria oceny

1. Liczba i jakość przedstawionych bibliotek (co najmniej dwie).
2. Poprawne działanie i spójność przygotowanych przykładów kodu.
3. Jakość raportu (jasne przedstawienie cech biblioteki, dokumentacji, zalet i ograniczeń).
4. Czytelność i organizacja repozytorium (struktura folderów, README, komentarze w kodzie).

5. (Opcjonalnie) Dodatkowe punkty za dogłębne testy, porównanie wydajności, przykłady bardziej zaawansowanego użycia. Lub większą liczbę bibliotek.