

Comparative analysis of Web application development on Java and PHP

Analiza porównawcza wytwarzania aplikacji internetowych na przykładzie Javy oraz PHP

Kacper Truskowski*, Maciej Pańczyk

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

This article presents a comparative analysis of two programming languages used to develop web applications. Two most popular programming languages Java and PHP were compared. The evaluated criteria were: implementation and performance, the time of performing specific operations on 100, 1000 and 10000 records was checked. A code analysis was carried out to determine in which language it is easier to implement an application with identical functionalities. Before the implementation of the application, articles, books and websites on a similar topic were reviewed. In order to test the efficiency, the Postman developer tool was used. The comparison shows that PHP is a more efficient language for developing web applications than Java.

Keywords: Java; PHP; web application

Streszczenie

Niniejszy artykuł przedstawia analizę porównawczą dwóch języków programowania używanych do wytwarzania aplikacji internetowych. Porównano dwa najbardziej popularne języki programowania Java oraz PHP. Oceniane kryteria to: implementacja oraz wydajność. Sprawdzono czas wykonywania konkretnych operacji na 100, 1000 oraz 10000 rekordach. Została przeprowadzona analiza kodu w celu ustalenia w którym języku łatwiej jest zaimplementować aplikację posiadającą identyczne funkcjonalności. Przed implementacją aplikacji, dokonano przeglądu artykułów, książek oraz stron internetowych poświęconych podobnemu zagadnieniu. W celu zbadania wydajności wykorzystano narzędzie developerskie Postman. Z porównania wynika, że PHP jest wydajniejszym językiem wytwarzania aplikacji internetowych niż Java.

Słowa kluczowe: Java; PHP; aplikacje internetowe

*Corresponding author

Email address: kacper.truskowski@pollub.edu.pl (K. Truskowski)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

W ciągu ostatnich lat wzrosło zapotrzebowanie na usługi informatyczne, rozwinęło się wiele nowych języków programowania, a wiele z nich zostało unowocześnionych. Wytworzone narzędzia mają na celu zwiększenie bezpieczeństwa oraz wydajności tworzonych aplikacji, jednocześnie ułatwiając ich implementację. Obecnie tworzone aplikacje są wdrażane w oparciu o wiele technologii, które się wzajemnie uzupełniają. Programiści mają do dyspozycji coraz nowsze i bardziej zaawansowane szkielety programistyczne. Wspomniane szkielety odpowiadają za czynności, które kiedyś były zadaniem programisty. Aplikacje internetowe powoli wypierają z naszego życia aplikacje desktopowe. Bardzo dynamicznie rozwijająca się technologia pozwala na wykorzystanie dużej elastyczności narzędzi programistycznych. Brak konieczności instalacji specjalnego oprogramowania na komputerze użytkownika czy też niezależność systemu operacyjnego składają się na dużą popularność aplikacji internetowych. W niniejszej pracy zestawiono analizę wydajnościową najpopularniejszych języków do wytwarzania aplikacji internetowych Java

oraz PHP [1]. Dzięki analizie porównawczej możliwe było wyciągnięcie wniosków o tym, jaka technologia jest wydajniejsza, w jakim scenariuszu.

2. Cel i zakres badań

Celem poniżej przedstawionego artykułu jest przeprowadzenie analizy porównawczej dwóch języków programowania stosowanych do wytwarzania aplikacji internetowych Java oraz PHP. Badaniom poddano wydajność oraz przeprowadzono analizę kodu, dzięki której można było określić w którym języku łatwiej jest zaimplementować małą aplikację internetową. Postawiono następujące tezy badawcze:

T1: Dla mniejszej aplikacji internetowej bardziej wydajnym językiem programowania będzie PHP.

T2: Dla aplikacji napisanej w języku Java aby otrzymać podobne wyniki czasowe co w przypadku języka PHP są wymagane większe zasoby sprzętowe.

T3: Implementacja aplikacji internetowej w języku PHP jest prostsza i mniej rozbudowana.

3. Przegląd literatury

Analizując obecną literaturę niełatwo znaleźć zestawienie porównujące języki Java oraz PHP. Porównanie tych języków stanowi pewne wyzwanie z powodu różnic jakie wynikają podczas implementacji aplikacji. Artykułem, który pod pewnymi kryteriami jest zbliżony do niniejszej pracy, jest artykuł pt. „Tworzenie aplikacji internetowych na platformie JEE I PHP – analiza porównawcza” [2]. Autor zestawia wyżej wymienione technologie, jednak podczas tworzenia aplikacji PHP użył szkieletu programistycznego Laravel. Oznacza to że osiągnięte wyniki będą się różnić od aplikacji napisanej w czystym PHP. Autor porusza w swojej pracy między innymi takie zależności jak:

- wydajność podczas operowania na danych z bazą danych,
- bezpieczeństwo,
- analiza kodu podczas implementacji.

Powyższe kryteria poza kryterium dotyczącym bezpieczeństwa są równoważne z zakresem badawczym niniejszego artykułu i pozwalają na wyznaczenie technologii, która będzie lepsza w analizowanych scenariuszach

4. Analizowane technologie

Badania zostały przeprowadzone na dwóch aplikacjach napisanych w różnych językach programowania Java oraz PHP. Języki te bardzo różnią się od siebie. Pierwszy z nich jest językiem obiektowym, który wymaga na urządzeniu posiadania maszyny wirtualnej (ang. Java Virtual Machine, JVM). PHP w przeciwieństwie do Javy nie wymaga maszyny wirtualnej. Jest to jednak język strukturalny, który zawiera elementy języka obiektowego [3].

4.1. Java

Java jest obiektowym językiem programowania ogólnego zastosowania. Podstawowa część składni została zaczerpnięta z języka C++. Programy wykonywane w tym języku są kompilowane do postaci binarnej, którą wykonuje maszyna wirtualna [4]. Głównymi zaletami języka są:

- obiektowość,
- niezawodność oraz bezpieczeństwo.

4.2. PHP

PHP to skryptowy język, który służy do implementowania aplikacji webowych w czasie rzeczywistym. Większość powszechnie używanych aplikacji internetowych, np. Facebook czy Wikipedia, jest napisana w tym języku. Język PHP najczęściej wykorzystuje się do obsługi formularzy na stronach internetowych [5].

5. Metoda badań

W celu przeprowadzenia badań zostały zaimplementowane dwie aplikacje napisane w języku Java oraz PHP. Zawierają one identyczne funkcjonalności. Aplikacje nie mają zaimplementowanej części wizualnej aby wydajność była badana w miarodajny sposób. Pomiarów zostały wykonane przy użyciu narzędzia Postman [6].

Dane w obu przypadkach były generowane poprzez bibliotekę Faker [7]. Biblioteka Faker umożliwiła uzupełnienie bazy losowymi danymi na których były wykonywane operacje zapisywania, wyświetlania, edycji oraz usuwania danych. Poniżej zostały przedstawione fragmenty kodu odpowiadające za generowanie danych w obu aplikacjach (Listing 1, Listing 2)

Listing 1: Fragment kodu odpowiadający za generowanie danych w aplikacji webowej napisanej w języku PHP

```
<?php
require_once "vendor/autoload.php";
$faker = Faker\Factory::create();
$db = new PDO(
    'mysql:host=127.0.0.1;dbname=
    crudoperation',
    'root',
    ''
);

array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES
utf8");
);
$inserts = array();
$queryFormat = '("%s", "%d")';
for ($i=1; $i<=1; $i++)
{
    $name = $faker->firstName();
    $rating = $faker->numberBetween($min = 1, $max = 10);
    $inserts[] = sprintf( $queryFormat, $name, $rating );
}
$query = implode( ", ", $inserts );
$db->query( 'INSERT INTO crud (name, rating) VALUES '. $query );
?>
```

Listing 2: Fragment kodu odpowiadający za generowanie danych w aplikacji webowej napisanej w języku Java

```
package com.testApplication.movieLibrary;
import com.github.javafaker.Faker;

public class FakerDemo {
    public static void main(String[] args) {
        Faker faker = new Faker();
        for(int i=0; i < 10; i++){
            String firstName = faker.name().firstName();
            int numberBetween = faker.number().numberBetween(0,10);
            System.out.println(firstName + " " + numberBetween);
        }
    }
}
```

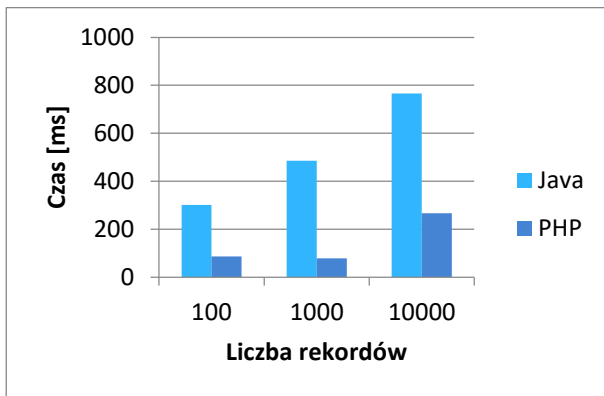
Na pojedynczych rekordach zostały wykonane operacje edycji oraz usuwania danych. W przypadku zapisywania oraz wyświetlania danych operacje były wykonywane na 1, 100, 1000 i 10000 rekordach. Wszystkie operacje zostały wykonane 5 razy w celu uzyskania miarodajnego wyniku. Podczas operacji mierzony był czas realizacji zapytania do bazy.

6. Platforma testowa i wyniki badań

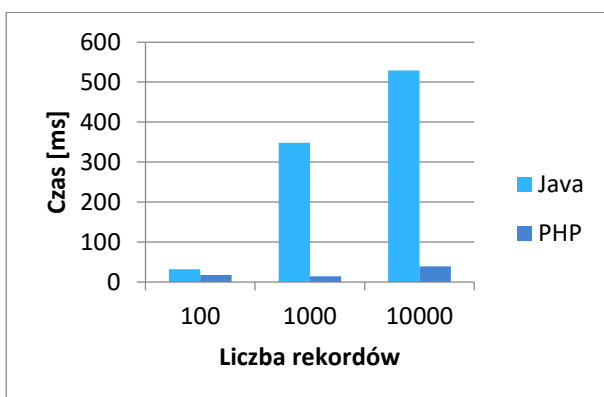
Badania zostały przeprowadzone na tym samym stanowisku przy wyłączonych wszelkich aplikacjach, które mogłyby mieć wpływ na wyniki. Dostęp do aplikacji w obu przypadkach odbywał się za pomocą usługi localhost na serwerze Apache. Do porównywania wydajności została użyta baza danych MySQL. Wykorzystywana była taka sama baza danych zatem zapytania SQL były w obu przypadkach identyczne. Ta Komputery, który został użyty do testów posiadał następujące parametry:

- system operacyjny: Windows 10 Home, 64 bitowy,
- procesor: AMD Ryzen 5 1600 3,2 GHz six core,
- pamięć Ram 16 GB,
- dysk HDD – 500 GB.
- karta graficzna: NVIDIA GeForce GTX 1050 Ti

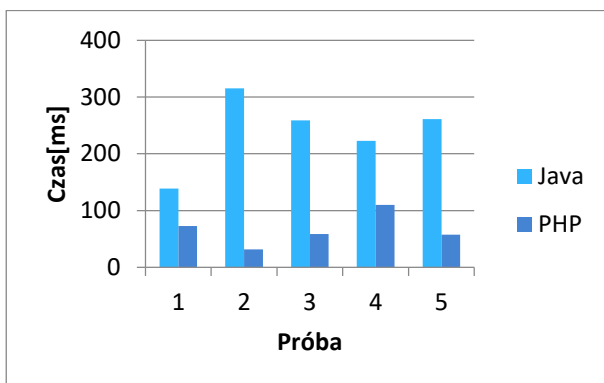
Rysunki 1- 4 prezentują średnie wyniki jakie otrzymały obie aplikacje w przypadku wykonywania poszczególnych operacji. Na wykresach nie został przedstawiony czas wykonywania zapisywania oraz wyświetlania na pojedynczym rekordzie ponieważ na tle innej ilości rekordów byłby on niewidoczny.



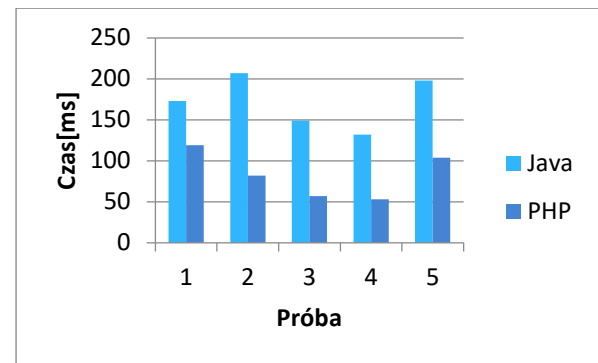
Rysunek 1: Porównanie czasów zapisu do bazy danych dla obu języków dla różnej liczby zapisywanych rekordów.



Rysunek 2: Porównanie czasów wyświetlania danych z bazy dla obu języków i różnej liczby zapisywanych rekordów.



Rysunek 3: Usuwanie pojedynczego rekordu z bazy.



Rysunek 4: Edycja pojedynczego rekordu z bazy.

7. Analiza kodu

Analiza zaimplementowanych aplikacji testowych pozwoliła na ocenę, w której technologii, Java czy PHP, początkujący programista będzie mógł łatwiej napisać aplikację internetową o identycznych funkcjach. Podczas tworzenia aplikacji w języku Java stworzenie konkretnej funkcjonalności odbywało się w trzech klasach. Najpierw została stworzona klasa ze zmiennymi, odpowiadająca poszczególnym polom w bazie (Movie). Następnie została stworzona klasa, która odpowiadała za komunikację z bazą danych (MovieRepository), a na końcu klasa, gdzie zaimplementowana metoda została wywoływana (MovieController) [8]. Podczas implementacji aplikacji w języku Java została wykorzystana biblioteka Lombok [9], która posłużyła do wygenerowania konstruktorów oraz ułatwiła implementację funkcji występujących w aplikacji. Zależności obiektów zostały wstrzyknięte przy pomocy biblioteki Spring. Adnotacje znajdujące się nad klasami (@Data, @NoArgsConstructor, @Autowired) pozwalają nam na korzystanie z nich bez dodatkowych instalacji. Dodatkowo została wykorzystana funkcja „query”, która wspomaga wymianę danych z bazą danych. W przypadku aplikacji w języku PHP wszystko odbywało się w jednym miejscu i było bardziej przejrzyste, co sprawia że początkujący programista łatwiej może się odnaleźć w kodzie. Podczas implementacji została wykorzystana funkcja odpowiadająca za połączenie z bazą danych (mysqli) oraz funkcje odpowiadające za kontakt aplikacji z bazą danych (mysqli_query, mysqli_fetch_assoc). Dla obu aplikacji do wygenerowania danych na których były testowane funkcje posłużyła biblioteka Faker. Jeśli chodzi o ilość dostępnych materiałów i dokumentację języka, Java ma pod tym względem bardzo dużą przewagę nad językiem PHP. Pamięć RAM, jakiej wymagała poszczególne aplikacje także została zmierzona narzędziem Postman.

8. Analiza wyników

Na podstawie przedstawionych wyników można zauważyć przewagę aplikacji napisanej w języku PHP. Podczas wykonywania zapisu rekordów do bazy Java wykonuje się prawie czterokrotnie wolniej niż w przypadku zapisywania danych w aplikacji napisanej w języku PHP. W aplikacji napisanej w technologii PHP różnice czasowe pomiędzy wyświetlaniem ilości

rekordów są praktycznie niewidoczne. Ilość wyświetlanych danych nie wpływała na szybkość wykonywania zapytania. Inaczej jednak było w przypadku aplikacji napisanej w języku Java. Większa ilość rekordów sprawiała, że czas wykonywania zapytania do bazy znacząco się wydłużał. W przypadku 10000 rekordów było to średnio prawie 529,2ms, natomiast dla aplikacji napisanej w języku PHP wynosiło średnio 39ms. Wykonywane operacje na pojedynczych rekordach takie jak usuwanie czy edytowanie danych również wskazują na wyższą wydajność języka PHP. Różnice czasowe nie były jednak na tyle duże aby użytkownik podczas korzystania z obu aplikacji potrafił wskazać, która aplikacja działa szybciej. Duży wpływ na otrzymane wyniki mogło mieć środowisko na którym zostały przeprowadzone badania z powodu użycia dysku HDD, który jest wolniejszy od dysku SSD. Język PHP jest językiem skryptowym, nie wymaga kompilacji. Inaczej było w przypadku aplikacji napisanej w języku Java gdzie kompilacja jest wymagana. Czas kompilacji nie został uwzględniony podczas wykonywania pomiarów, różnice pomiędzy technologiami byłyby wtedy jeszcze większe. Poniżej znajduje się Tabela 1 przedstawiająca statystyki dotyczące implementacji poszczególnych aplikacji.

Tabela 1: Statystyki aplikacji

Statystyki	Aplikacja w języku PHP	Aplikacja w języku Java
Liczba klas użytych do implementacji	6	5
Liczba bibliotek	1	4
Liczba dedykowanych funkcji	5	7
Liczba linii kodu	101	109
Zużycie pamięci RAM podczas wyświetlania pojedynczego rekordu	110 KB	204 KB
Zużycie pamięci RAM podczas dodawania pojedynczego rekordu	165 KB	270 KB
Rozmiar aplikacji	52,61 KB	110 KB

9. Wnioski

Wyniki jednoznacznie wskazują, która technologia jest wydajniejsza w przypadku aplikacji posiadającej takie same funkcjonalności. Trzeba jednak zwrócić uwagę, że język PHP służy wyłącznie do implementacji aplikacji internetowych a Java jest językiem obiektowym ogólnego zastosowania. Zatem pomimo tego iż język Java wymaga większych zasobów sprzętowych aby uzyskać podobne czasy co aplikacji napisana w języku PHP to ma ogólniejsze zastosowanie. Napisane aplikacje posiadają podstawowe funkcjonalności aplikacji webowej, nie zawierają warstwy odpowiadającej za widok. Największe dysproporcje czasowe pojawiają się podczas wyświetlania 10000 rekordów z bazy gdzie średnio język PHP osiąga dwunastokrotnie większą prędkość podczas wykonywania zapytania. Wyniki definitywnie potwierdzają tezy T1 i T2. Po przeanalizowaniu kodu pod względem implementacji, wysunięto wniosek iż implementacja w języku PHP jest prostsza dla początkującego programisty, głównie dlatego iż wszystko znajduje się w jednym miejscu i używane metody nie wymagają dodatkowych adnotacji nad klasami jak w przypadku aplikacji Java. Podczas implementacji aplikacji w języku Java trzeba było wykorzystać więcej bibliotek oraz funkcji, niż w przypadku aplikacji napisanej w języku PHP. Pomimo że dodawanie biblioteki odbywało się w bardzo prosty sposób, to jednak początkujący programista może o tym nie wiedzieć i wtedy implementacja staje się trudniejsza. Została zatem potwierdzona teza T3. Ilość linii kodu pomiędzy aplikacjami jest bardzo podobna. Jednakże większa ilość funkcji oraz bibliotek, które zostały wykorzystane w aplikacji Java w dużym stopniu zmniejszają objętość kodu. Jeśli chodzi o zużycie pamięci RAM podczas wykonywania operacji dodawania oraz wyświetlania pojedynczego rekordu więcej pamięci wymaga aplikacja napisana w języku Java.

Literatura

- [1] K. Arnold, J. Gosling, D. Holmes, The Java Programming Language, Fourth Edition, Addison Wesley Professional 2005.
- [2] S. Jędruszk, B. Jędruszk, B. Pańczyk, Tworzenie aplikacji internetowych na platformie JEE I PHP – analiza porównawcza, Journal of Computer Sciences Institute 11 (2019) 86-90
<https://doi.org/10.35784/jcsi.145>.
- [3] A. Kołtun, B. Pańczyk, Analiza porównawcza narzędzi do badania wydajności aplikacji internetowych, Journal of Computer Sciences Institute 17 (2020) 351-357
<https://doi.org/10.35784/jcsi.2209>.
- [4] J. Farrell, Java Programming Cengage Learning, 2011.

-
- [5] Dokumentacja Php, <https://www.php.net/manual/en/>, [21.04.2022].
- [6] HTTP request methods, <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>, [22.04.2022].
- [7] Dokumentacja biblioteki Faker, <https://javastart.pl/baza-wiedzy/frameworki/javafaker>, [22.04.2022].
- [8] Dokumentacja Java api: <https://docs.oracle.com/javase/7/docs/api>, [21.04.2022].
- [9] Dokumentacja biblioteki Lombok, <https://javastart.pl/baza-wiedzy/frameworki/project-lombok>, [27.07.2022].