

Zastosowanie Programowania Obiektowego  
**Informatyka III Semestr**

# Piłka nożna - Katalog (Aplikacja PHP)

**Jakub Achtelik, Oliwier Budnik**



Politechnika Koszalińska  
Koszalin, 10 Grudnia 2023

## Spis Treści

<b>1</b>	<b>Koncepcja i funkcjonalność aplikacji</b>	<b>2</b>
1.1	Założenia przyjęte dla tworzenia aplikacji:	2
1.1.1	Środowisko uruchomieniowe:	2
1.2	Założenia przyjęte dla tworzenia aplikacji	3
1.2.1	Cechy	3
1.2.2	Ograniczenia:	3
1.3	Narzędzia programistyczne:	3
1.4	Wykaz funkcjonalności aplikacji	4
1.5	Prototyp	5
<b>2</b>	<b>Podział na klasy</b>	<b>7</b>
2.1	index.php	7
2.2	Projekt.php	8

# 1 Koncepcja i funkcjonalność aplikacji

## 1.1 Założenia przyjęte dla tworzenia aplikacji:

**Rodzaj aplikacji:** Aplikacja internetowa (architektura klient-serwer)  
Aplikacja uruchamiana lokalnie na komputerach oraz zdalnie na serwerze VPS.  
Aplikacja jest dostosowana również do urządzeń mobilnych.

**Środowisko lokalne:** Windows 10, Ubuntu 22.04 LTS

**Środowisko produkcyjne (serwer):** VPS Linux (Ubuntu 20.04 LTS):

### 1.1.1 Środowisko uruchomieniowe:

#### Dla Windows:

- XAMPP 8.2.4 (serwer HTTP Apache, Serwer bazy danych MariaDB, interpreter PHP)

#### Dla Ubuntu:

- PHP 8.1.2-1ubuntu2.14 Development Server (serwer HTTP + interpreter PHP)
- mysql-server ver 8.0.34-0ubuntu0.22.04.1 (serwer bazy danych MySQL)

#### Dla VPS:

- Serwer HTTP Apache2
- Serwer MySQL – MariaDB
- Interpreter PHP 8+ ...

## 1.2 Założenia przyjęte dla tworzenia aplikacji

### 1.2.1 Cechy

1. Zastosowanie statycznego typowania (zmiennych, funkcji, metod, pól klasy), podobnie jak w językach C/C++, Java, C#. Jest to bardziej przewidywalne i pozwala narzucić określony typ np. zwracanej zmiennej, aby uniknąć wielu błędów. Domyślnie PHP nie wymaga statycznego typowania.
2. Podział projektu na wiele plików według struktury MVC Model-View-Controller (pol. Model-Widok-Kontroler):
  - Model jest pewną reprezentacją problemu bądź logiki aplikacji.
  - Widok opisuje, jak wyświetlić pewną część modelu w ramach interfejsu użytkownika.
  - Kontroler przyjmuje dane wejściowe od użytkownika i reaguje na jego poczynania
3. Logika aplikacji będzie zawarta w sposób obiektowy w klasach, każda klasa to osobny plik.

### 1.2.2 Ograniczenia:

- PHP jest podatny na pewne rodzaje ataków, takich jak na przykład wstrzykiwanie SQL, dlatego bezpieczeństwo aplikacji nie jest na najwyższym możliwym poziomie i szczegółowa konfiguracja zabezpieczeń nie jest łatwa do wdrożenia w krótkim czasie
- PHP jest językiem interpretowanym dlatego wydajność w stosunku do języków kompilowanych jest niższa
- PHP nie posiada wszystkich elementów obiektowych znanych z innych języków
- Ograniczony czas, przez co nie można zawrzeć wszystkich celów w wzorcowy sposób zgodny w 100% z dokumentacją
- Ograniczenie aktualnej wiedzy, przez co niektóre elementy projektu mogą stanowić wyzwanie

## 1.3 Narzędzia programistyczne:

Język: PHP 8+ OOP

Dodatkowe biblioteki: mysqli (łączenie się z bazą danych)

Dodatkowe technologie: HTML, CSS, JavaScript, MySQL, FontAwesone(ikonki)

IDE: Visual Studio Code + PHP Code Extensions

GIT – System Kontroli Wersji

Figma – Prototypowanie wyglądu aplikacji

Trello – zarządzanie zadaniami w zespole

Przeglądarka internetowa – Posiadająca narzędzia Chrome DevTools

Pakiet make – automatyzacja poleceń w terminalu

FileZilla – klient FTP

## **1.4 Wykaz funkcjonalności aplikacji**

Interfejs webowy, zarządzanie bazą danych z poziomu przeglądarki internetowej:

- edycje, usuwanie, dodawanie nowego piłkarza,
- sortowanie oraz wyświetlanie zdjęć,
- wyszukiwanie po nazwisku, imieniu itp.
- filtrowanie szczegółowe po np. kraju, pozycji itp.
- logowanie oraz autoryzacja użytkownika przeglądającego aplikację
- dodawanie/edycje, zdjęcia piłkarza

Użytkownik może za pomocą przeglądarki internetowej:

- połączyć się z serwerem na którym hostowana jest aplikacja
- zalogować się do panelu poprzez formularz logowania, uzyskać autoryzację
- Panel umożliwia przeglądanie katalogu piłkarzy w przystępnej formie oraz inne operacje (edycja, usuwaniem, filtrowanie itp.).
- Użytkownik końcowy (klient) nie musi posiadać znajomości obsługi relacyjnej bazy danych aby w intuicyjny sposób zarządzać aplikacją.

## 1.5 Prototyp

Prototyp graficzny interfejsu użytkownika. Wykonane w programie Figma.



Rysunek 1: Widok strony głównej



Rysunek 2: Widok ekranu logowania

The screenshot shows the 'Dodawanie' (Adding) form in the 'Piłka Nożna Katalog' application. The form is titled 'Dodawanie' and contains several input fields for user information: 'imię' (name), 'nazwisko' (surname), 'kraj' (country), 'pozycja' (position), and 'wzrost' (height). Below these fields is a 'dodaj' (add) button. The navigation bar at the top includes 'Strona Główna', 'Dodaj', 'Szukaj', and 'Wyloguj'.

Rysunek 3: Widok strony głównej

The screenshot shows the 'Edycja' (Editing) form in the 'Piłka Nożna Katalog' application. The form is titled 'Edycja' and contains several input fields for user information: 'Lionel', 'Messi', 'Argentyna', 'napastnik', and '1.62m'. Below these fields is an 'edycja' (edit) button. The navigation bar at the top includes 'Strona Główna', 'Dodaj', 'Szukaj', and 'Wyloguj'.

Rysunek 4: Widok strony głównej

## 2 Podział na klasy

Poniżej znajdują się fragmenty kodu, które są wytłumaczone.

### 2.1 index.php

Ten plik łączy wszystkie pliki i uruchamia projekt:

---

```
1 <?php
2 use Pilkanozna\Projekt;
3
4
5 include_once './classes/FileLoader.php';
6 include_once './classes/Projekt.php';
7
8 // kolejnosc ladowania - nie zmieniac
9
10 Fl::Controllers("PilkarzPost");
11 Fl::Models("BazaDanych");
12 Fl::Helpers("BazaDanychHelper");
13 Fl::Models("FiltrowanieSql");
14 Fl::Models("ZapytaniaSql");
15 Fl::Helpers("FormularzHelper");
16 Fl::Models("Aplikacja");
17
18 Fl::Controllers("KontrolerDanych");
19 Fl::Controllers("Autoryzacja");
20 Fl::Controllers("KontrolerStrony");
21
22
23 Fl::Models("PobieraczObrazowWikipedia");
24 Fl::Views("SzablonHtml");
25 Fl::Views("StronaHtml");
26
27
28 Projekt::Uruchom();
```

---



## 2.2 Projekt.php

Ta klasa jest finalna

---

```
1  <?php
2  // klasa do uruchamnia projektu
3
4  namespace Pilkanozna;
5
6  use Pilkanozna\Controller\KontrolerStrony;
7  use Pilkanozna\Views\StronaHtml;
8
9
10 final class Projekt
11 {
12     public static function Uruchom(): void
13     {
14         $Aplikacja = new KontrolerStrony;
15         $Strona = new StronaHtml;
16
17
18         $Strona->Head();
19         $Strona->Header();
20         $Aplikacja->Routing();
21         $Strona->Footer();
22     }
23
24
25 }
```

---