



# Coder's Lab

pierwsza w Polsce Szkoła Programistów

## Kurs PHP - Warsztaty OOP

Celem warsztatów jest napisanie pełnej i funkcjonalnej aplikacji w stylu Todois.

Aplikacja ma implementować następujące funkcjonalności:

- **Zadania:** Na stronie ma być możliwość tworzenia zadań do zrobienia. Zadania mają być trzymane w sesji.

Aplikacja musi mieć następujące strony:

- **Strona główna:** Strona która wyświetla wszystkie zadania jakie ma do zrobienia użytkownik.
- **Strona dodania zadania:** Strona to ma służyć do stworzenia nowego zadania. Wyświetlany formularz ma być różny w zależności od typu zadania .

Podstawowy layout stron głównych zostanie zaprezentowany na zajęciach.

Pod koniec zajęć wyślij wykładowcy informację z adresem repozytorium na którym znajduje się twój kod i listą zaimplementowanych funkcjonalności.

### Zadanie 1. Przygotowanie.

Przygotuj folder pod aplikację. Załóż nowe repozytorium git na GitHubie. Pamiętaj o tworzeniu commitów (najlepiej co ćwiczenie).

Stwórz plik .gitignore i dodaj do niego wszystkie podstawowe rzeczy (pliki \*.\*~, katalog .idea etc).

### Zadanie 2. Podstawowe zadanie.

Stwórz klasę Task. Ma ona zawierać dwie zmienne typu string. Jeden ma opisywać nazwę zadania a drugi jego treść.

Napisz konstruktor przyjmujący dwa napisy, setery i getery do obu zmiennych. Stwórz też funkcję która wyświetla zadanie na stronie.

Stwórz skrypt z 2-3 przykładowymi zadaniami w tabelce i używając pętli foreach wyświetl informację o nich na stronie.

### Zadanie 3. Strona główna.

Stwórz stronę główną. Na razie ma ona wyświetlać guzik przekierowujący do strony tworzenia zadania.

### Zadanie 4. Tworzenie zadania.

Stwórz formularz który pozwoli na stworzenie zadania. Formularz ten ma przekierowywać do strony głównej.

Na stronie głównej dopisz kod który zidentyfikuje czy do strony zostały przekazane dane zapytanie POST. Jeżeli tak to je poprawnie wczytaj i korzystając z nich stwórz nowy obiekt klasy zadanie.

Dodaj go do sesji używając funkcji serialize

(<http://php.net/manual/en/function.serialize.php>). Najwygodniejszy sposób to stworzenie tablicy zserializowanych obiektów i dodanie jej do sesji.

### Zadanie 5. Wyświetlanie zadań.

Na stronie głównej wczytaj tablicę zserializowanych obiektów i przeiteruj po niej tworząc obiekty (też wkładając je do tablicy). Użyj do tego funkcji unserialize. Następnie przeiteruj po tablicy z obiektami i wyświetl je wszystkie.

### Zadanie 6. Ukończenie zadania.

Do klasy zadanie dodaj zmienną typu boolean. Będzie ona reprezentować czy zadanie zostało już ukończone przez użytkownika czy nie. W konstruktorze nastaw ją zawsze na false. Dodaj funkcję get i funkcję finishTask która przestawi tą zmienną na true.

Zmień funkcję wyświetlającą zadanie żeby wyświetlała tekst przekreślony jeżeli zadanie jest skończone. Przy każdym zadaniu powinien też wyświetlać się guzik który przeniesie będzie służył ukończeniu zadania.

### Zadanie 7. Kończenie zadań.

Zmodyfikuj guzik służący do kończenia zadania tak żeby przesyłał użytkownika do strony głównej wysyłając też informacje o zadaniu metodą get (np. nazwę zadania).

Zmodyfikuj stronę główną w ten sposób żeby sprawdzała czy są wysyłane dane metodą GET. Jeżeli tak to przeiteruj po tabelce z zadaniami i „konczyła” wybrane zadanie.

### Zadanie 8. Możliwe rozszerzenia.

Dodaj do rozwiązania następujące typy zadań:

- Zadanie z priorytetem,
- Zadanie z datą (wyświetl tekst na czerwono jeżeli data już minęła).

Jeżeli coś nie jest jasne to nie bój się zapytać prowadzącego zajęcia.