Formularze z poprzednich ćwiczeń realizują obliczenia, ale tylko w przypadku wprowadzenia poprawnych danych liczbowych. Nie są odporne na błędy użytkownika i nie wyświetlają żadnych ostrzeżeń. Aby wyeliminować te problemy konieczna jest walidacja danych pochodzących od użytkownika.

#### Walidacja formularzy po stronie klienta

Walidację danych można realizować po stronie klienta i po stronie serwera. Walidacja po stronie klienta zwykle realizowana jest w JavaScript i ma na celu nie tylko wygodę użytkownika ale uniknięcie przesyłania niepoprawnych danych na serwer. Tego typu zabezpieczenia można łatwo ominąć i nie można traktować ich jako środka gwarantującego bezpieczeństwo serwisu. Wszystkie dane pochodzące od użytkownika (w tym dane przekazane za pomocą formularzy), należy traktować jako potencjalnie niebezpieczne i dokonywać ich podwójnej walidacji. Walidacja po stronie serwera realizowana jest za pomocą skryptów np. PHP.

# Ćw.1. Walidacja formularza

Zrealizuj walidację formularza z pliku *formularze.html*. Do walidacji wykorzystaj funkcje pomocnicze umieszczone w zewnętrznym pliku *walidacja.js*.

W pliku powinny się znaleźć funkcje odpowiadające za:

- sprawdzanie poprawności danych wpisanych do pola tekstowego m.in. z wykorzystaniem wyrażeń regularnych,
- sprawdzenie czy i który przycisk typu radio został zaznaczony,
- sprawdzenie czy i które przyciski typu checkbox zostały wybrane.

Formularz z pliku *formularze.html* uzupełnij o pola, w których zostaną umieszczone ewentualne komunikaty o błędach. Przykładowy kod źródłowy takiego formularza z odpowiednimi komentarzami może mieć postać przedstawioną na Listingu 1.

**Listing 1.** Formularz do walidacji uzupełniony komórkami tabeli, w których będą się wyświetlały ewentualne komunikaty o błędach

```
<form method="post" action="mailto:xxx@xxx.xx" enctype="text/plain"</pre>
    onsubmit="return sprawdz()" >
 Nazwisko: 
    <input name="nazw" size="30" id="nazw"/> 
     Wiek:
    <input name="wiek" size ="30" id="wiek"/>
     Państwo:
    <select name="kraj" id="kraj">
        <option value="pl" selected="selected">Polska</option>
        <option value="gb">Wielka Brytania</option>
       </select>
     Adres e-mail: 
    <input name="email" size ="30" id="email"/>
```

Zauważ, że w formularzu na czerwono oznaczono elementy, w których będą się pojawiały komunikaty o ewentualnych błędach. Każde z pól formularza posiada atrybut *name* a te, które będą poddawane walidacji również atrybut *id*. Funkcje walidujące, wykorzystują metody getElementById lub getElementsByName do pobrania wartości badanego pola formularza. Walidacja powinna być realizowana za pomocą pomocniczych funkcji umieszczonych w pliku *walidacja.js*, który należy dołączyć do dokumentu HTML znacznikiem <script> wewnątrz elementu <head>. Całą walidację realizuje funkcja sprawdz(), która jest wywoływana w znaczniku <form> tuż przed przekazaniem danych na serwer. Akcja wysłania danych z formularza jest rozpoczynana po kliknięciu na przycisk typu submit. Przechwycenie tego zdarzenia umożliwia atrybut onsubmit znacznika form. W zależności od wyniku zwróconego przez funkcję sprawdz(), dane z formularza zostaną przesłane na serwer lub pojawią się stosowne komunikaty o błędach. Pomocnicze funkcje pokazano na Listingu 2 a przykładowy efekt walidacji na rys.1.

### Listing 2. Plik walidacja.js

```
function sprawdzPole(pole_id,obiektRegex) {
//Funkcja sprawdza czy wartość wprowadzona do pola tekstowego
//pasuje do wzorca zdefiniowanego za pomocą wyrażenia regularnego
//Parametry funkcji:
//pole_id - id sprawdzanego pola tekstowego
//obiektRegex - wyrażenie regularne
var obiektPole = document.getElementById(pole_id);
if(!obiektRegex.test(obiektPole.value)) return (false);
else return (true);
}
function sprawdz radio(nazwa_radio){
//Funkcja sprawdza czy wybrano przycisk radio
//z grupy przycisków o nazwie nazwa radio
var obiekt=document.getElementsByName(nazwa_radio);
 for (i=0;i<obiekt.length;i++)</pre>
 { wybrany=obiekt[i].checked;
      if (wybrany) return true;
      return false;
}
```

```
function sprawdz box(box id)
{//Funkcja sprawdza czy przycisk typu checkbox
//o identyfikatorze box_id jest zaznaczony
var obiekt=document.getElementById(box_id);
if (obiekt.checked) return true;
else return false;
function sprawdz()
{ //Funkcja realizujaca sprawdzanie całego fomularza
 //wykorzystując funkcje pomocnicze
 //-----
 var ok=true; //zmienna informująca o poprawnym wypełnieniu formularza
 //Definicje odpowiednich wyrażeń regularnych dla sprawdzenia
 //poprawności danych wprowadzonych do pól tekstowych
 obiektNazw = /^[a-zA-Z]{2,20}$/; //wyrażenie regularne dla nazwiska
 obiektemail =
        /^([a-zA-Z0-9])+([.a-zA-Z0-9_-])*@([a-zA-Z0-9_-])+(.[a-zA-Z0-9_-]+)+/;
 obiektWiek=/^[1-9][0-9]{1,2}$/;
 //Sprawdzanie kolejnych pól formularza.
 //w przypadku błędu - pojawia się odpowiedni komunikat
 if (!sprawdzPole("nazw",obiektNazw))
      ok=false;
      document.getElementById("nazw error").innerHTML=
        "Wpisz poprawnie nazwisko!";
 else document.getElementById("nazw_error").innerHTML="";
 //Uzupełnij - sprawdz kolejne pola formularza
 // if ...
 return ok;
}
```

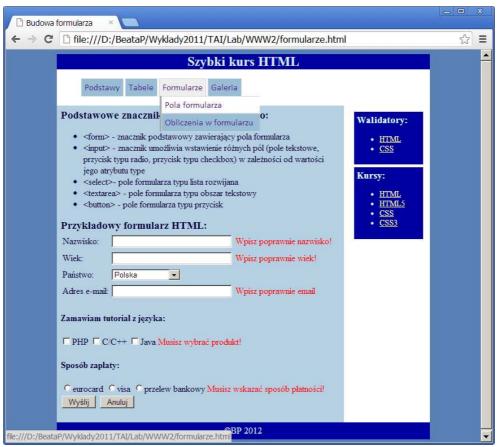
## Ćw.2. Walidacja i zebranie danych z wypełnionego formularza

Zmodyfikuj funkcję sprawdz() tak aby po sprawdzeniu danych ale przed ich wysłaniem pojawiało się okno z informacją jakie dane użytkownik wprowadził do formularza (Rys.2). Aby uniknąć sprawdzania czy wybrano przycisk z grupy radio – zaznacz za pomocą atrybutu checked="checked" jeden z nich jako wybór domyślny. Będziesz potrzebować dodatkowej funkcji JavaScript, która sprawdzi, który z przycisków grupy radio został wybrany. Rys.2 pokazuje efekt działania zmodyfikowanej funkcji sprawdz().

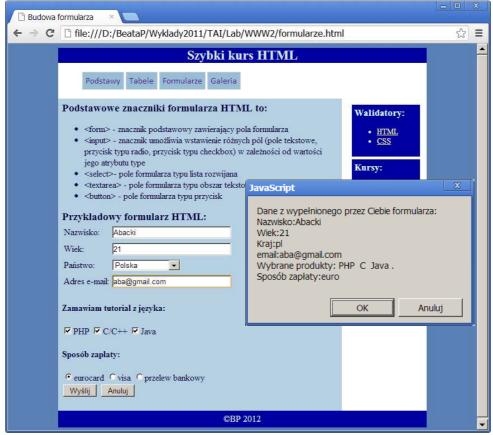
#### Wskazówka!

W funkcji sprawdz () zadeklaruj kolejną zmienną lokalną, w której będziesz zbierał dane z formularza np.:

```
var dane="Dane z wypełnionego przez Ciebie formularza:"; Dla każdego pola, po poprawnej jego walidacji, uzupełnij łańcuch z danymi: dane+="\nNazwisko: "+...
```



Rys.1. Wynik walidacji



Rys.2. Widok okna z informacją o poprawnie wprowadzonych danych