

**Pytania wprowadzające:**

**1. Czym się różnią testy funkcjonalne od нефункционаlnych?**

Testy funkcjonalne dotyczą tego „co” system robi. Czy działają wszystkie funkcjonalności. Sprawdzają czy wszystkie elementy działają poprawnie. Natomiast testy нефункционаlnych sprawdzają „jak” działa system. Testują samo działanie systemu poprzez:

- testy wydajnościowe
- testy obciążeniowe
- testy przeciążeniowe
- testy użyteczności
- testy pielęgnowalności
- testy niezawodności
- testy przenaszalności

Według mnie najłatwiej można powiedzieć, że testy funkcjonalne dotyczą poprawności działania funkcjonalności systemu – poprawności działania natomiast нефункционаlnych jego wnętrza – wykorzystanych rozwiązań.

**2. Co to są ‘smoke testy’ i ‘testy regresji’? Kiedy je stosujemy?**

Smoke testy swoją nazwę zawdzięczają sposobowi testowania urządzeń, a mianowicie test polegał na włączeniu urządzenia i sprawdzeniu czy nie ma dymu lub ognia. Smoke testy przeprowadzane są przed samym oddaniem aplikacji/systemu. Sprawdzają czy krytyczne - najważniejsze funkcjonalności działają, nie analizują dokładnie wszystkich funkcjonalności.

Testy regresywne są wykonywane na już przetestowanym urządzeniu po wprowadzonych poprawkach. Ich celem jest ustalenie czy wystąpiły jakieś nowe usterki lub czy przez zmiany nie ujawniły się inne błędy. Testy regresywne muszą być powtarzalne by móc porównać wcześniejsze wyniki. Są one uniwersalne – mogą być przeprowadzane w każdym z typów testów: strukturalnych, funkcjonalnych i нефункционаlnych.

**3. Co jest celem testowania?**

- Celem testowania jest wyszukiwanie błędów w oprogramowaniu na każdym etapie jego powstawania.
- Podniesienie jakości oprogramowania dzięki poprawie ujawnionych usterek przed wprowadzeniem systemu na rynek oraz zmierzenie jakości atrybutów нефункционаlnych np. efektywności, stopień wykorzystania zasobów (np. pamięci) .
- Sprawdzenie czy działanie systemu jest zgodne z założeniami – niezbędnymi funkcjonalnościami oraz dokumentacją; czy spełnia oczekiwania użytkownika
- Określenie ryzyka związanego z oddaniem systemu w określonej chwili
- Sprawdzenie czy po wprowadzeniu poprawek do systemu nie pojawiły się kolejne usterki lub poprawki nie wywołały innych problemów.
- Określenie niezawodności systemu
- Sprawdzenie zgodności z umową i czy system jest zgodny ze wszystkimi przepisami prawnymi

- Sprawdzenie czy system działa pomimo zmiany środowiska np. upgrade

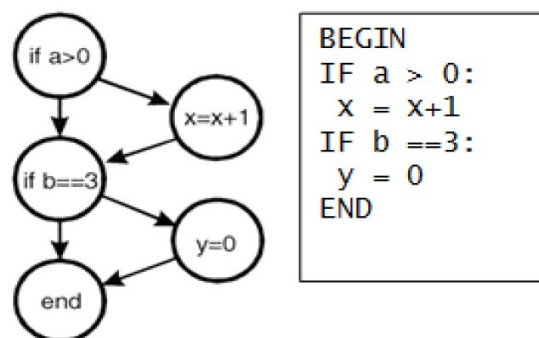
#### 4. Jak tester może się upewnić, że błąd został naprawiony?

Przeprowadzić retesty -jeszcze raz niezaliczone testy i potwierdzić usunięcie defektu - testowanie potwierdzające. Przeprowadzić testy regresywne. Sprawdzić czy po wprowadzonych poprawkach nie powstały nowe usterki.

#### 5. Testujesz aplikację termometr która wykonuje pomiar temperatury. Co byś zrobił aby przetestować zachowanie aplikacji przy skrajnych wartościach -50C i 200C ?

Przeprowadziłabym testy z ustawionymi wartościami – zasymulowanie – wprowadzenie na sztywno poprzez oprogramowanie wartości -50 lub 200 stopni ponieważ przeprowadzenie precyzyjnych testów w warunkach laboratoryjnych mogłoby być kosztowne, a uzyskanie precyzyjnych wartości szczególnie przy określeniu temperatury -50 bardzo trudne. (200 stopni można ustawić w piekarniku, ale to raczej mało profesjonalny test i nieprecyzyjny :) )

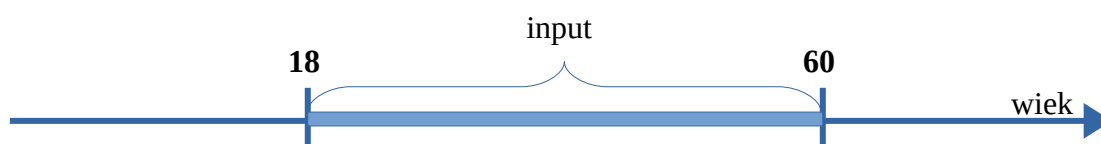
#### 6. Ile przypadków testowych potrzeba, aby pokryć wszystkie możliwości?



	a>0	a<=0
b==3		
b!=3		

Przypadki wynikające z przyjmowanych wartości a i b  
a może przyjąć wartości a>0 (true) i wejdzie do pętli lub else a<=0 nie wejdzie do pętli natomiast dla b należy sprawdzić gdy b==3 i wykona się instrukcja y=0 lub gdy b!=3 i przejdzie do end. W sumie 4 przypadki należy przeanalizować .

#### 7. Dany jest input „wiek”, który przyjmuje wartości od 18 do 60. Twoim zadaniem jest przetestować go za pomocą techniki wartości brzegowych. Jakie wartości wpisujesz do inputu? Podaj wszystkie liczby, które wpisujesz.



Wiek:

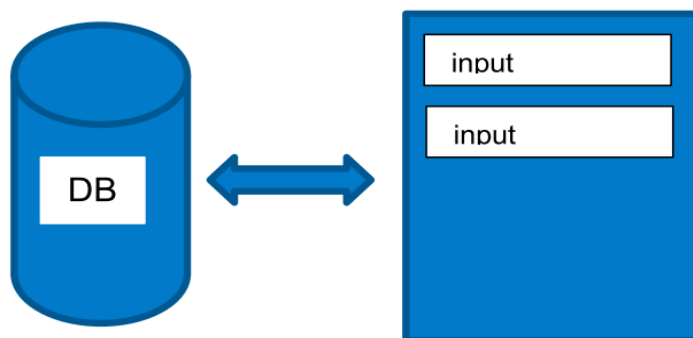
17 lat – ponieważ znajduje się na końcu (brzegu) przedziału nienależącego do dziedziny

18 i 60 – znajdują się na brzegach badanego przedziału

61 lat – ponieważ znajduje się na początku (brzegu) przedziału nienależącego do dziedziny

**8. Dołączasz do projektu w trakcie developmentu aplikacji, do której nie ma dokumentacji. Schemat logowania do aplikacji wygląda następująco:**

**Jakie pytania zadasz analitykowi, zanim przystąpisz do testów logowania?**



1. W jakim celu będzie wykorzystywana baza- jaki rodzaj danych będzie w nich znajdował?
2. W jaki sposób będą wykonywane i przechowywane kopie zapasowe?
3. Jaki jest rozmiar bazy – ilu kont użytkowników maksymalnie może znajdować się w bazie ?
4. Kto będzie miał dostęp do kont użytkowników?
5. Jaki będzie sposób rejestracji. Czy użytkownicy zakładają swoje konta czy baza będzie utworzona zewnętrznie, a użytkownicy otrzymają swój login, hasło?
6. Czy próby uwierzytelniania udane i nieudane będą przechowywane? Kto ma do nich dostęp?
7. Jakie będą wymagania dotyczące hasła?
8. Jakie dane będą znajdować się w formularzu?
9. Jaki będzie sposób weryfikacji użytkownika?

## **9. Czym się różni metoda GET od POST?**

W zasadzie przetwarzania danych złożonego formularza zależy od tego, czy jest wysyłany z *method = "get"* lub *method = "post"*. Ponieważ dane są kodowane w różny sposób, potrzebne są różne mechanizmy dekodowania. Tak więc, ogólnie rzecz biorąc, zmiany metody mogą wymagać zmian w skrypcie, który przetwarza zgłoszenia.

- **GET**- Zapotrzebowanie na dane z określonych zasobów – zapytanie niezakodowane jest wysyłane w adresie URL w postaci par nazwa=wartość. Długość jest ograniczona dlatego używa

się szczególnie, gdy parametrów niewiele. Jest mniej bezpieczny od Post – nie mogą być przekazywane za jego pomocą np. hasła ponieważ będą one widoczne w pasku przeglądarki.

- Parametry znajdują się w historii przeglądarki ponieważ są częścią adresu URL
- można się do niej odwołać – może być zakładką
- łatwiejsza do włamania – mniejsze bezpieczeństwo ponieważ dane przesyłane jest częścią adresu URL. Więc to jest zapisywane w historii przeglądarki i serwera dzienników w postaci zwykłego tekstu.
- jedynymi typami danych wpisywanymi do formularza są znaki ASCII
- metoda nie może być wykorzystywana do wysyłania haseł
- ma limity w długości – ilości informacji
- może być buforowane
- **POST**- do przekazywania parametrów wykorzystywany jest nagłówek zapytania – metoda umożliwia przekazywanie większych parametrów ponieważ nie jest ograniczona przez pasek URL oraz nie jest w nim widoczna.
  - Parametry nie są przechowywane w historii przeglądarki
  - nie można tworzyć zakładek
  - po naciśnięciu buttona wstecz przeglądarka wysyła ostrzeżenie do użytkownika i informuje że dane będą musiały być jeszcze raz wprowadzone
  - może wysyłać pliki na serwer,
  - nie ma ograniczeń związanych z typem danych – dane mogą być wysyłane również w postaci binarnej
  - trudniejszy do włamania – metoda używana podczas wysyłania haseł, danych wrażliwych

W praktyce metody pozwalają nam „rozdzielać” zapytania trafiające pod ten sam adres – np. metody GET używamy do wyświetlenia formularza, a metody POST do jego przesłania – obie rzeczy możemy realizować pod tym samym adresem url

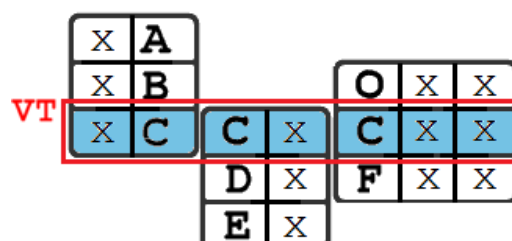
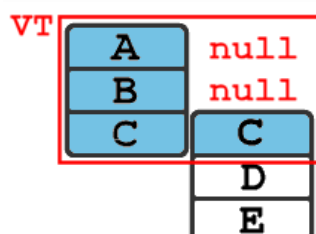
## 10. Czy HTTP jest protokołem zmiennostanowym?

Nie.

Służy on do przesyłania danych dla sieci internetowej. Dzięki niemu przesyłane są żądania udostępnienia klientowi dokumentów z sieci. Polega ona na tym, że klient wysyła zapytanie, a serwer wysyła na nie odpowiedź – dane.

*Hypertext Transfer Protocol* jest protokołem bezstanowym – nie przechowuje danych – informacji co się działo wcześniej. Zmniejsza to obciążenie serwera danymi jednakże ze względu na ich brak został wprowadzony system ciasteczek cookies, które zapamiętują dane sesji - Są to małe pliki tekstowe umieszczane na urządzeniach użytkowników końcowych (np. komputerach) za pośrednictwem przeglądarek internetowych. Pliki te pozwalają na przechowywanie określonych

## 11. Czym różni się LEFT JOIN od INNER JOIN?



**INNER JOIN** zwraca wiersze w których obie wartości w obu tabelkach się zgodziły-zwraca część wspólną zbiorów danych z którymi łączymy obie tabele.

W wyniku złączenia wewnętrznego (INNER JOIN) otrzymujemy tabelę wynikową (VT), składającą się ze wszystkich kolumn tabel wejściowych.

## Dane wejściowe

Results		Messages				
	IdPrac	Imie	Nazwisko	DzZatr	Stanowisko	IdManager
1	1	Adam	Nowak	2010-01-01	BOSS	NULL
2	2	Piotr	Kowalski	2010-03-15	Kierownik Sprzedaży	1
3	3	Michał	Pogodny	2010-04-04	Sprzedawca	2
4	4	Anna	Dymna	2010-05-15	Kierownik Marketingu	1
5	5	Jan	Maly	2011-08-01	Marketer	4
6	6	Leopold	Stuff	2011-11-15	Sprzedawca	2
7	7	Monika	Miśa	2012-03-31	Asystent	1
8	8	Joanna	Wesoła	2012-08-01	Sprzedawca	2

	NrRej	Marka	Rocznik	IdPrac
1	P0123AA	VOLVO V60	2012	4
2	P0422VX	FIAT 500	2012	6
3	P064231	VOLVO S40	2012	2
4	P0745AX	Mercedes	2013	1
5	P0841XY	FIAT 500	2013	NULL

## LEFT JOIN

	Imie	Nazwisko	Stanowisko	Marka
1	Adam	Nowak	BOSS	Mercedes
2	Piotr	Kowalski	Kierownik Sprzedaży	VOLVO S40
3	Michał	Pogodny	Sprzedawca	NULL
4	Anna	Dymna	Kierownik Marketingu	VOLVO V60
5	Jan	Maký	Marketer	NULL
6	Leopold	Stuff	Sprzedawca	FIAT 500
7	Monika	Miła	Asystent	NULL
8	Joanna	Wesoła	Sprzedawca	NULL

## INNER JOIN

	NrRej	Marka	Rocznik	IdPrac
1	P0841XY	FIAT 500	2013	NULL
2	P0745AX	Mercedes	2013	1
3	P064231	VOLVO S40	2012	2
4	P0123AA	VOLVO V60	2012	4
5	P0422VX	FIAT 500	2012	6

	IdPrac	Imie	Nazwisko	DzZatr	Stanowisko	IdManager
1	1	Adam	Nowak	2010-01-01	BOSS	NULL
2	2	Piotr	Kowalski	2010-03-15	Kierownik Sprzedaży	1
3	3	Michał	Pogodny	2010-04-04	Sprzedawca	2
4	4	Anna	Dymna	2010-05-15	Kierownik Marketingu	1
5	5	Jan	Mały	2011-08-01	Marketer	4
6	6	Leopold	Stuff	2011-11-15	Sprzedawca	2
7	7	Monika	Mika	2012-03-31	Asystent	1
8	8	Joanna	Wesoła	2012-08-01	Sprzedawca	2

	IdPrac	Imie	Nazwisko	DzZatr	Stanowisko	IdManager	NrRej	Marka	Rocznik	IdPrac
1	1	Adam	Nowak	2010-01-01	BOSS	NULL	P0745AX	Mercedes	2013	1
2	2	Piotr	Kowalski	2010-03-15	Kierownik Sprzedaży	1	P064231	VOLVO S40	2012	2
3	4	Anna	Dymna	2010-05-15	Kierownik Marketingu	1	P0123AA	VOLVO V60	2012	4
4	6	Leopold	Stuff	2011-11-15	Sprzedawca	2	P0422VX	FIAT 500	2012	6

Źródło rysunków: <http://www.sqlpedia.pl/laczenie-tabel-sql/>  
<http://cezarywalenciuk.pl/blog/programing/post/join-inner-outer-left-right>

## 12. W jakim katalogu, standardowo Linux trzyma pliki konfiguracyjne:

- a. /boot
- b. /var
- c. /etc
- d. /cfg

## 13. Jak przetestowałbyś bashową komendę cp? (argumenty funkcji można pominąć)

Komenda `cp` kopiuje ŹRÓDŁO do CELU lub wiele ŹRÓDEŁ do KATALOGU. Komenda jest bardzo rozbudowana i ma bardzo wiele zastosowań dzięki dodatkowym opcjom.

• `cp [przełącznik...] nazwa_pliku... nowa_nazwa_lub_katalog`

kopiowanie pliku określonego przez pierwszy argument pod nazwę lub do katalogu określonego drugim argumentem, np.:

`cp abc.txt xyz.txt` kopiuje plik `abc.txt` pod nową nazwę `xyz.txt` w katalogu bieżącym, jeśli plik nie istnieje to go utworzy, a jeśli istnieje zastąpi zawartość pliku kopią `abc.txt`

`cp /tmp/abc.txt ~` kopiuje plik `abc.txt` z katalogu `/tmp` do katalogu domowego użytkownika

`cp abc.txt ~/xyz.txt` kopiuje plik `abc.txt` z katalogu bieżącego pod nową nazwę `xyz.txt` w katalogu domowym użytkownika

`cp abc.txt efg.bak xyz.txt /home/kasia/mojepliki`

Spowoduje to skopiowanie plików `abc.txt`, `efg.bak`, `xyz.txt` z bieżącego katalogu do katalogu `/home/kasia/mojepliki` – w tym przypadku parametrem docelowym jest katalog

`cp /*.txt ~` kopiowanie wszystkich plików z rozszerzeniem `.txt` z katalogu bieżącego do katalogu domowego użytkownika;

`*` – zastępuje dowolny ciąg znaków (także pusty);

`?` – zastępuje dokładnie jeden dowolny znak;

### **Proponowane metody testowanie komendy `cp`:**

- skorzystanie z przełącznika `-v`, który wyświetla w czasie wykonywania operacji kopiowania nazwy plików
- sprawdzenie-porównanie wielkość kopiowanych plików i katalogów. Można podczas testowania skorzystać z polecenia `du`, które wyświetla rozmiar podanego pliku lub katalogu pozwala ona również wyświetlanie rozmiarów wielu plików i katalogów. Porównanie rozmiarów umożliwi stwierdzenie czy wszystkie elementy zostały skopiowane
- podsumowanie ilości skopiowanych plików – określenie czy w katalogu źródłowym liczba plików jest równa liczbie plików w katalogu docelowym (przy założeniu że kopiowanie dotyczy pustego katalogu). Jeśli katalog docelowy istniał przed przekopiowaniem i zawierał elementy warto określić ich ilość przed operacją oraz ich rozmiar – sumę
- sprawdzenie czy ścieżka do kopii jest odpowiednia – czy pliki przekopiowały się w odpowiednie miejsce
- porównanie formatów plików po wykonaniu kopiowania, w zależności od kopiowanych plików można wykonać wyszukiwanie, wyświetlanie, sumowanie ilości plików z odpowiednim rozszerzeniem (nazwą) - przydatne jest wykorzystanie „`*`” oraz „`?`” w celu łatwiejszego wyszukania plików np.: `*.c` – wyświetlenie plików z rozszerzeniem `c`,