

1. **Jakie problemy rozwiązać musi warstwa sprzętowa? W jaki sposób formułuje się adresy sprzętowe?**
 - Podział strumienia danych na pakiety (ramki): istotne dla koordynacji i podziału czasu (wspólne medium!);
 - Wykrywanie kolizji;
 - Kontrola poprawności transmisji: bity parzystości, sumy kontrolne lub CRC (Cyclic Redundancy Check): szybki algorytm (indeksy przesuwające, bramki XOR, sprzężenie zwrotne) => wielomian komunikatu; dość skuteczny przy identyfikowaniu błędów; łatwość sprawdzenia przez odbiorcę;
 - Identyfikacja adresata (sygnał dociera do wszystkich maszyn...): adresy sprzętowe;
 - Interpretacja komunikatu: identyfikatory typów ramek
 - Zwane MAC (*Media Access Control*), tj. adres dostępu do nośnika; Ethernet: 48 bitów.
 - Interfejs sieciowy filtruje ramki nie obciążając „cudzymi” procesora;
 - Sposoby określania: statyczne, konfigurowalne, dynamiczne;
2. **Przedstaw 7-warstwowy model interfejsów OSI. Jakie role pełnią poszczególne warstwy?**
 7. Aplikacji: usługi realizowane przez aplikacje.
 6. Prezentacji danych: format wymiany danych, ich ewentualne szyfrowanie. (zwykle nie występuje)
 5. Sesji: nawiązywanie i zamykanie połączeń (związana z dużymi systemami wielodostępnymi – straciła na znaczeniu)
 4. Transportu: podział danych, gwarancja dostarczenia, weryfikacja.
 3. Sieciowa: logiczne adresy zamienia na sprzętowe, pakiety logiczne dzieli na ramki danych.
 2. Łącza: interpretowanie przesyłanych bitów jako ramek.
 1. Fizyczna: przesyła nieprzetworzone bity danych przez nośnik.
3. **Omów podział na warstwy stosowany w protokołach internetowych.**
 - Warstwa **aplikacji** (Telnet, FTP, e-mail...). Także SMNP (Simple Network Management Protocol).
 - Transportowa** (TCP, UDP). TCP zapewnia wiarygodny (*reliable*) przepływ danych (porcjowanie, *timeouts*). UDP przesyła datagramy nie gwarantując ich dotarcia.
 - Sieciowa** (IP, ICMP, IGMP). Albo warstwa internetowa. Obsługuje ruch pakietów poprzez sieć. Między innymi zachodzi tu rutowanie pakietów. *Internet Protocol, Internet Control Message Protocol, Internet Group Management Protocol.*
 - Połączeniowa** (sterowniki urządzenia i karta interfejsu sieciowego; np. Ethernet, token ring, użycie ARP i RARP) Albo warstwa interfejsu sieciowego. Wszelkie szczegóły fizycznej komunikacji.
4. **Wyjaśnij, na czym polega warstwowy model protokołów internetowych. Dlaczego wyróżnia się poszczególne warstwy?**
 - Złożonością zadania;
 - Heterogenicznością istniejących sieci składowych;
 - Specjalizacją poszczególnych protokołów do zastosowań.
5. **W jakim celu wprowadzono system adresów logicznych?**

Identyfikuje połączenie komputera z daną siecią.
6. **Objaśnij podział odpowiedzialności pomiędzy protokołami TCP oraz IP.**
7. **Omów tradycyjny podział adresów IP na 4 klasy. Jakie problemy wiążą się z takim podejściem i jak się im zapobiega?**
 - a. A: do 127:
 - b. B: 128-191:
 - c. C: 192-223:
 - d. D: 224-239:
 - e. E: 240-255:

marnotrawstwo adresów;
duże tablice tras (zob. dalej).

IPv6: 128-bitowe adresy => ogromna pula. Ponadto:

 - Zmieniony, rozszerzalny format nagłówka: pole „typ następnego nagłówka”;
 - Wsparcie dla obsługi dźwięku i obrazu;
 - Trzy rodzaje adresów: jednostkowy, rozsyłania grupowego, „grono” -> służy np. realizacji replik usługi.
 - Sposób zapisu: notacja szesnastkowa z dwukropkami; możliwość pomijania segmentu zer: FF0C:0:0:0:0:0:B1 -> FF0C::B1
8. **Objaśnij mechanizm enkapsulacji (*encapsulation*) datagramu IP.**
 - Zapakowanie wewnątrz ramki sieciowej; zawartość nie jest interpretowana przez niższą warstwę.
 - Datagram IP jako zawartość takiej ramki jest identyfikowany odpowiednim typem ramki sieciowej. Oczywiście ramka sieciowa otaczająca datagram musi mieć wyspecyfikowany adres sprzętowy odbiorcy.
 - Każdy etap na drodze datagramu wiąże się z rozpakowaniem ramki sieciowej a następnie wysłaniem go w nowej ramce.

-MTU (*Maximum Transfer Unit*): ogranicza wielkość danych w ramce. Jeżeli dla danej sieci rozmiar datagramu przekracza MTU, to wykonywana jest fragmentacja. Składanie datagramu odbywa się dopiero u ostatecznego odbiorcy. Odbiór datagramu: wszystko albo nic.

9. **Objasnij mechanizm odwzorowywania adresów logicznych na adresy sprzętowe.**

-Aby dostarczyć dane, należy na podstawie adresu protokołowego ustalić adres sprzętowy.

-Metody:

- a. Tablicowe (zwykle w sieciach rozległych);
- b. Obliczeniowe (gdy adres sprzętowy konfigurowalny);
- c. Sieciowe: pytanie-odpowiedź (zwykle w lokalnych ze statycznymi adresami).

-Adres sprzętowy można określić tylko wtedy, jeśli adresat jest w tej samej sieci fizycznej.

-Pakiety dla adresatów nielokalnych przejmują **ruter**.

-Ruter przynależy do co najmniej dwóch sieci (posiadając tyleż adresów logicznych i sprzętowych).

10. **Jakiego rodzaju urządzenia aktywne są stosowane w budowie sieci komputerowych?**

-Repeater (**wzmacniak**), oraz hub -> warstwa fizyczna (1 w OSI).

Propagują również zakłócenia...

-**Most**: podział na dwie sieci, mogące posiadać różne media transmisyjne -> warstwa łącza (2); pracuje na adresach MAC.

-**Switch** -> warstwa łącza danych (2). Symuluje sieć lokalną z mostami, gdzie w każdym segmencie jest tylko jeden komputer.

-**Ruter** -> podział na więcej podsieci IP -> warstwa łącza (2) i sieciowa (3). Łączenie sieci heterogenicznych. Podobnie jak most = specjalizowany komputer: procesor, pamięć, interfejsy sieciowe. Różne media transmisyjne, mechanizmy adresowania sprzętowego i formaty ramek.

-**Zapora** sieciowa -> ogranicza dostępu z zewnątrz. Rodzaje filtrów:

- a. bramki aplikacji: pakiety są odpowiednio tłumaczone; wymaga skonfigurowania oprogramowania proxy dla każdej usługi sieciowej na każdej z maszyn w sieci;
- b. filtry pakietów: blokowanie pakietów w oparciu o adres pochodzenia, adres docelowy, protokół, nr portu czy nawet zawartość.

11. **Omów właściwości i zadania protokołu TCP.**

Obsługuje następujące poważne problemy transmisji:

- a. przywracanie właściwej kolejności odebranych pakietów: wykorzystanie numeru porządkowego pakietów;
- b. unikanie duplikatów – j.w.;
- c. retransmisja zgubionych pakietów: nadawca wysyła ponownie, jeśli w zadanym czasie nie otrzymał potwierdzenia (*acknowledgment*);
- d. unikanie powtórzeń spowodowanych nadmiernym opóźnieniem: numery pakietów dodatkowo kwalifikowane identyfikatorami sesji;
- e. kontrola przepływu dla uniknięcia zalewu danych: metoda przesuwającego się okna.

12. **Wyjaśnij mechanizm enkapsulacji pomiędzy warstwami protokołów internetowych. Na czym polega demultiplexing?**

-Każda warstwa dołącza swe nagłówki przed i niekiedy również (np. Ethernet trailer) za przekazywanym pakietem danych.

-Jednostka danych wysyłanych przez protokół TCP = *segment*;

-Jednostka wysyłana przez protokół IP = *datagram* (tak naprawdę – *packet*: tj. datagram IP albo fragment datagramu IP);

-Jednostka w ramach Ethernetu = *frame (ramka)* – o określonym typie

-8-bitowy identyfikator w nagłówku IP identyfikuje protokół datagramu: ICMP, IGMP, TCP, UDP.

-Analogicznie TCP i UDP używają w nagłówku 16-bitowych numerów portów do zidentyfikowania aplikacji: numery źródłowego i docelowego portu.

-Analogicznie (np. rozróżnienie pomiędzy IP, ARP, RARP), nagłówek Ethernetowy zawiera 16-bitowe *pole typu ramki*.

-Demultiplexing: po otrzymaniu ramki kolejne warstwy „rozpakowują” dane i w oparciu o identyfikatory w znanych sobie nagłówkach przekazują odpowiedniemu protokołowi warstwy wyższej.

13. **Czemu służą numery portów? Jakie założenia przyjęto przy ich przydzielaniu?**

-Przydziałem numerów od 1 do 1023 zajmuje się *Internet Assigned Numbers Authority (IANA)*. Unix: porty zarezerwowane => tylko superuser może przydzielić taki port.

-Porty klienckie zwane efemerycznymi (*ephemeral ports*), gdyż używane są na czas korzystania danego klienta z usługi. Zwykle alokowane pomiędzy 1024 a 5000.

14. **Co to jest interfejs gniazd i jakie oferuje możliwości?**

-Najpopularniejsze jednak – udostępnianie jednolitego interfejsu tzw. **gniazd** (*sockets*). Powstał jako część BSD; zwany też *Berkeley sockets*.

-Mogą być dostępne jako wywołania systemu operacyjnego, lub w postaci funkcji bibliotecznych.

15. Objaśnij organizację i funkcjonowanie DNS (Domain Name System).

-Zapewnia przyjazność adresów Webu dla człowieka: umożliwia lokalizację maszyny (adres logiczny) na podstawie znakowej nazwy;

-Pozycja w bazie danych DNS zawiera następujące elementy:

- a. nazwa dziedziny;
- b. typ rekordu;
- c. wartość.

-Występują różne typy rekordów – oto najważniejsze:

- d. Wiązania: nazwa -> adres IP => wiązania typu **A**;
- e. Wiązania: nazwa dziedziny poczty elektronicznej -> adres IP => wiązania typu **MX** (Mail eXchanger).

16. W jaki sposób realizowana jest obsługa zapytań DNS?

-Podstawowe założenie projektowe: rozproszenie bazy poprzez hierarchiczną dekompozycję.

-Zapewnia to autonomię definiowania nazw lokalnych przez poszczególne organizacje. Hierarchia jest zorganizowana następująco:

- a. każdy serwer bierze odpowiedzialność za część hierarchii nazw;
- b. na szczycie – serwer główny, posiadający listę serwerów kontrolujących poszczególne dziedziny poziomu głównego (np. .com);
- c. pojedynczy komputer musi być odpowiedzialny za wszystkie komputery o danym zakończeniu nazwy. Innymi słowy, danego wierzchołka drzewa nie można podzielić.
- d. poszczególne organizacje mogą zatem uruchamiać dla swoich domen jeden lub hierarchię serwerów, albo oddelegować to zadanie usługodawcy;

17. Czemu służy standard MIME i jak organizuje obsługiwaną zawartość?

MIME = *Multipurpose Internet Mail Extensions*:

- a. Standaryzuje dodatkowe wiersze nagłówkowe;
- b. Otwarty system definiowania różnego typu zawartości;

Dopuszcza:

- c. wiele odrębnych obiektów przesyłanych jako pojedynczy list;
- d. nieograniczone długości wiersza oraz całego tekstu;
- e. zbiory znaków inne niż ASCII;
- f. wiele różnych fontów w jednej wiadomości;
- g. różnego rodzaju załączniki multimedialne;

18. Omów zasady działania systemu WWW.

-Tzw. interfejs *point and click*. Określany jako system *hipermedialny*, tzn. rozszerzenie (obecność innych niż tekst postaci informacji) systemu *hipertekstu*.

-Dokumenty administrowane niezależnie => niska stabilność odsyłaczy.

-Scenariusz działania serwera jest dość prosty: udostępnianie w odpowiedzi na komunikat klienta żądanych przezeń dokumentów.

19. Przedstaw zasadnicze kategorie dokumentów WWW wraz z ich zaletami i wadami.

Statyczne: zawartość określona w momencie ich napisania przez autora:

- + prostota, łatwość implementacji;
- słaba elastyczność.

Dynamiczne: brak predefiniowanej postaci; generowany na nowo w odpowiedzi na każde wezwanie:

- + możliwość prezentowania aktualizowanej często informacji;
- + zapewnienie interakcyjności bez dodatkowych wymogów dla oprogramowania przeglądarki (z punktu widzenia jej mechanizmów strony statyczne nieodróżnialne od dynamicznych);
- po pobraniu takiej strony ulec ona może dezaktualizacji;
- większe wymagania dla wydajności łącza i serwera; większy narzut czasowy (konieczne generowanie strony);
- wyższe koszty opracowania i przetestowania.

Aktywne: dokumenty zawierające w sobie kopię programu uruchamialnego lokalnie w środowisku przeglądarki:

- + może sięgać do źródeł na serwerach celem bieżącego aktualizowania informacji;
- potencjalne luki w bezpieczeństwie;
- dodatkowe koszty tworzenia i działania;
- wymaga bardziej zaawansowanego, zgodnego oprogramowania przeglądarki (w tym – problem sprawdzenia wersji danej przeglądarki) i odpowiednich zasobów po stronie klienta.

20. Scharakteryzuj protokół HTTP oraz budowę jego komunikatów.

-Przeglądanie dokumentów hipermedialnych nie wykazuje istotnej lokalności odwołań. => Inny wzorec odwołań niż dla pozostałych rodzajów programów użytkowych!

-Protokół zaprojektowany jako **bezstanowy** (brak pojęcia sesji grupującej interakcje). Interakcja przeglądarki z serwerem WWW odbywa się według modelu **bezpoleźniowego**:

- a. żądanie jest wysyłane przez klienta;
- b. Serwer przekazuje (zawsze z inicjatywy klienta) żądane zasoby lub informacja o ich niedostępności;
- c. połączenie zostaje zamknięte.

-Protokół określa format żądania oraz odpowiedzi.

-Domyślny numer portu: 80.

21. Omów najważniejsze rodzaje komunikatów HTTP stanowiących żądanie ze strony klienta.

-GET:

- a. Stosowany przy specyfikowaniu wymaganego zasobu;
- b. Parametry dołączane do URL (po znaku zapytania);
- c. Ograniczona długość przekazywanych parametrów;

-POST:

- d. Parametry wysyłane w ciele komunikatu;
- e. Opisane dodatkowo nagłówkami **Content-Type:** (np. *application/x-www-form-urlencoded*) oraz **Content-Length:**
- f. wołana lokalizacja (URI) wskazuje zwykle na program obsługujący, nie zaś na pobierany zasób;
- g. odpowiedź jest zwykle dokumentem generowanym, nie zaś statycznym.

-HEAD:

- h. jak GET, ale służy jedynie sprawdzeniu dostępności zasobu: zwracany w odpowiedzi jest komunikat nie posiadający ciała;

22. Jakie udoskonalenia wprowadza specyfikacja HTTP w wersji 1.1 i jakich dodatkowych elementów komunikatów wobec tego wymaga?

-Wprowadzenie trwałych połączeń (persistent connection), umożliwiających wykonanie wielu transakcji w oparciu o to samo połączenie TCP

=> sprawniejsze działanie.

-Wsparcie dla cache.

-Wsparcie dla tzw. *chunked encoding*, pozwalające nadawanie odpowiedzi jeszcze przed ustaleniem jej ostatecznego rozmiaru. Zapewnia sprawniejszą odpowiedź dla dynamicznie generowanych stron.

-Umożliwienie obsługi wielu domen z tego samego adresu IP -> efektywniejsze wykorzystanie puli adresów IP.

23. Wskaż najważniejsze ograniczenia tradycyjnych dokumentów WWW.

-Bezstanowość protokołu HTTP;

-Stacyczny charakter dokumentów;

-Ścisłe splecione zawartość, opis struktury oraz opis sposobu wizualizacji dokumentu;

=> niemożność efektywnej personalizacji serwisu;

=> bardzo ograniczone możliwości autoryzacji użytkownika;

=> utrudnione zarządzanie stylem wizualizacji w przypadku większych dokumentów;

24. Czym jest język JavaScript i do jakich typowych zadań bywa stosowany w dokumentach WWW?

Interpretowany język skryptowy o luźnej kontroli typów.

Od początku planowany jako rozszerzenie możliwości stron WWW, przystosowane do użycia przez nieprofesjonalistów. Ważnym pierwotnie zakładanym zastosowaniem miała być interakcja z apletami Javy.

- Umożliwienie modyfikowania wyglądu dokumentu, czy też np. nawigowania w historii bieżącego okna;
- Wykonywanie operacji zależnych od platformy (przeglądarka, system operacyjny);
- Wykonywanie prostych obliczeń po stronie klienta;
- Kontrola poprawności wypełnienia formularza;
- Określanie aktualnej daty i czasu;
- Generowanie lokalnie dokumentów HTML.

25. Scharakteryzuj pokrótce składnię języka JavaScript oraz sposób jego łączenia z HTML.

-Wyróżnione typy danych: "number", "string", "boolean", "function", "object", "undefined". Możliwość sprawdzenia typu: funkcja `string typeof(zmienna_lub_stala)`.

-Operatory logiczne, arytmetyczne i bitowe -> składnia podobnie jak w Javie;

-Instrukcje warunkowe i pętle -> podobnie jak w Javie;

-Możliwość deklarowania w kodzie zmiennych i tablic zmiennych; w dowolnym miejscu

26. Czym są aplety i jakie cechy odróżniają je od JavaScript?

-Trzy zasadnicze komponenty tej technologii:

- a. język programowania;
- b. system wykonawczy;
- c. biblioteka klas;

- Możliwość wykonywania operacji na plikach;
- Możliwość cyfrowego podpisania apletu celem udostępnienia mu operacji uprzywilejowanych;
- Większa skalowalność, możliwość tworzenia bardziej wyrafinowanego kodu;
- Udostępnianie apletu:
 - d. bezpośrednie wskazanie w URL pliku .class zawierającego aplet;
 - e. osadzenie apletu w dokumencie HTML:


```
<APPLET CODEBASE="serwer/scieżka CODE="nazwaApletu.class">
```

27. Czym są *cookies*? Jakie są ich możliwości oraz ograniczenia?

-Plik tekstowy przechowywany po stronie klienta. Tradycyjny termin informatyczny oznaczający nieprzejrzysty element danych przechowywany przez pośrednika.

-Zastosowanie:

- a. kto odwiedza stronę? Ilu jest użytkowników, ilu nowych?
- b. personalizacja strony;
- c. zachowanie stanu współpracy ze stroną;
- d. umieszczenie identyfikatora "koszyka";

-Ciasteczka stają się użyteczne dopiero przy wykorzystaniu skryptów (po stronie klienta lub po stronie serwera)

-Ciasteczko może być założone, usunięte lub zmienione na skutek odpowiedzi serwera, zaś zawarta w nim informacja zostanie przekazana przy następnym wezwaniu.

-Stąd dopiero przy następnym wezwaniu jesteśmy w stanie zorientować się, że użytkownik nie przyjmuje ciasteczek.

-Serwer może zażądać tylko ciasteczka, które sam zapisał u klienta (zgodność domeny w żądaniach).

28. Czym są Kaskadowe Arkusze Stylów (CSS)? Jak mogą być związane z dokumentem oraz na czym polega właściwość ich „kaskadowości”?

-Umożliwienie separacji stylów względem struktury i treści dokumentu.

-Bardziej precyzyjna kontrola nad układami, fontami, kolorami, tłem, efektami typograficznymi;

-Możliwość modyfikowania większego zbioru stron poprzez edytowanie pojedynczego dokumentu; b. ważne dla pielęgnacji!

-Zapewnienie zgodności pomiędzy przeglądarkami; zapewnienie sprawnego ładowania stron.

Definicje stylów posiadają określone (dość intuicyjne priorytety). Przypominają nieco przesłanianie definicji zmiennych w lokalnych podprogramach:

1. Style specyfikowane specjalnie dla danego tagu
2. Style osadzone w nagłówku
3. Style połączone referencją
4. Style zaimportowane
5. Style domyślne przeglądarki

29. Scharakteryzuj technologię Dynamic HTML.

-Nie jest to wyodrębniona technologia jak np. JavaScript czy ActiveX.

-Definicja: zawartość HTML mogąca zmieniać swój wygląd już po załadowaniu do przeglądarki.

-Umożliwia manipulację elementami dokumentu WWW wcześniej traktowanymi jako niezmiennie. Na ogół pisane w JavaScript lub VBScript. Wyróżnik – dostęp skryptu do właściwości DHTML dostępnych w przeglądarkach 4 generacji, choć wówczas rozbieżności były frustrująco duże.

30. Czym jest DOM (*Dynamic Object Model*)? Jakie podstawowe rodzaje węzłów obsługuje?

DOM udostępnia kompletny i ustandaryzowany model dokumentu jako strukturę zarządzaną przez przeglądarkę. Wprowadza również wsparcie dla zdarzeń myszy i klawiatury względem wszystkich elementów dokumentu.

Pozwala na manipulowanie obiektami dokumentu:

posiadającymi właściwościami;

mogącymi zawierać inne obiekty.

-możliwość nawigowania w tej hierarchii, albo dostępu poprzez identyfikator (atrybut ID obiektu).

Rodzaje węzłów:

-**Element**: odpowiada tagowi lub parze tagów; może zawierać zagnieżdżone elementy (typu *element*) lub węzły tekstowe (typu *text*).

-**Text**: odpowiada danym znakowym. Nie posiada elementów potomnych.

-**Attribute**: określają atrybuty elementów. Wartością atrybutu jest zawsze string. Również nie zawiera elementów pochodnych.

-**Document**: specjalizacja typu *element*. Stanowi pojedynczy korzeń dokumentu i udostępnia specyficzne operacje dla globalnego przeszukiwania.

31. Jakie możliwości wnosi stosowanie dokumentów dynamicznych?

-aplikacje oparte na bazie danych;

-autoryzacja użytkownika i personalizacja; koszyk;

-forum; poczta przez WWW oraz listy rozsyłkowe;
-zarządzanie treścią; generowanie dokumentów.

32. Omów podstawowy mechanizm autentykacji dostępny w protokole HTTP.

-**Autentykacja**: użytkownik jest rzeczywiście tym, za kogo się podaje (najpowszechniejsza metoda = sprawdzenie hasła).

-**Autoryzacja**: sprawdzenie, czy dany użytkownik jest uprawniony do danego dostępu.

-**Kontrola dostępu**: inne, ujęcie zagadnienia. Poza zidentyfikowaniem użytkownika możliwe są jeszcze inne kryteria udostępnienia zasobu (np. adres, spod którego nadchodzi żądanie).

Najprostsza metoda, tzw. *basic*:

-uwzględniona w protokole HTTP;

-wymaga ustawień konfiguracyjnych serwera WWW;

-hasło nie jest kodowane;

Przebieg interakcji:

-w odpowiedzi na żądanie serwer zwraca zamiast zasobu nagłówek: 401 Authentication Required; oraz nazwę chronionego obszaru (tzw. *realm*)

-przeglądarka wyświetla dialog (użytkownik i hasło);

-wprowadzone dane są wykorzystywane przez przeglądarkę w kolejnych żądaniach dotyczących danego obszaru;

33. Scharakteryzuj technologię interfejsu CGI. Jakie posiada ograniczenia?

-Protokół uruchamiania programów na serwerze WWW. Ustandaryzowanie takiego interfejsu zapewnia niezależność od platformy tj. OS oraz serwera webowego.

-Pozwala na udostępnianie dynamicznej informacji. Nie ogranicza się do dokumentów HTML.

-CGI nie jest językiem programowania i nie determinuje języka programowania. Można stosować nawet skrypty powłoki. Często stosowany Perl w związku z dostępnością modułów dla Webu oraz bardzo dobrymi udogodnieniami dla przetwarzania tekstu.

-Ograniczona sprawność działania: konieczność załadowania procesu przy obsłudze każdego żądania stanowi znaczne obciążenie;

-Problem współdzielenia zasobów nie jest łatwy do rozwiązania;

-Możliwość wywoływania poleceń po stronie serwera tworzy istotne zagrożenie dla bezpieczeństwa;

34. Jakie możliwości oferuje technologia SSI? Czym różni się od CGI?

-Jak sugeruje nazwa, w tym wypadku kod jest wstawiany w strukturę dokumentu HTML.

-Odpowiednie fragmenty dokumentu są ponownie generowane przy każdym odwołaniu.

-Dynamika dokumentu bez konieczności stosowania CGI; generowane są tylko wymagane fragmenty dokumentu.

-Możliwe zastosowania:

a. Licznik odwołań;

b. Data i czas; data ostatniej modyfikacji dokumentu;

c. Wkomponowywanie innego pliku lub rezultatu wywołania programu CGI.

-Rozszerzenia specyfikacji obejmują m. in.:

d. Instrukcje warunkowe (if ... else);

e. Wysyłanie poczty.

35. Scharakteryzuj technologię ASP. Jakie rodzaje zmiennych są dostępne w jej skryptach?

-Zalety:

a. Dynamiczna zawartość webowa;

b. Obsługa danych przekazanych poprzez formularze;

c. Udostępnianie źródeł danych znajdujących się na serwerze;

d. W stosunku do CGI przewagą jest prostota i szybkość tworzenia;

e. Kod ASP nie jest widoczny po stronie przeglądarki;

f. Udostępnia zwykły HTML -> brak specjalnych wymogów wobec przeglądarki;

g. Przy odpowiednim programowaniu stron pozwala ograniczyć ruch w sieci.

-**Lokalne** (wewnątrz procedur).

-**Globalne** strony (poza ciałem procedury):

dostępne dla każdego skryptu na danej stronie.

-**Sesji**: Dostępne ze wszystkich stron danej aplikacji – tworzone i wypełniane na potrzeby współpracy z konkretnym użytkownikiem. Przechowują np. imię, identyfikator, preferencje.

-**Aplikacyjne**: Również dostępne dla wszystkich stron. Przechowują jednak właściwości obowiązujące wszystkich użytkowników.

36. Scharakteryzuj technologię PHP.

-Również zaliczany do gatunku SSI.

-Bardzo szeroko dostępny; Obsługa PHP na danej platformie dostępna jako moduł serwera albo jako program CGI.

-Zastosowania:

- a. skrypty po stronie serwera;
- b. skrypty uruchamiane z linii komend;
- c. okienkowe programy działające po stronie użytkownika;

37. Scharakteryzuj technologię servletów. Jak implementuje się zachowanie servletu?

-Servlet funkcjonuje jako element oprogramowania odpowiadający na żądania (zwykle) HTTP.

Model działania nawiązuje więc po części do CGI, ma jednak w stosunku do niego istotne zalety:

- a. Przystosowane do wykonania wielowątkowego.
- b. Dzięki rozbudowanemu API Javy zwykle nie ma potrzeby wywoływania poleceń systemu operacyjnego – lepsze bezpieczeństwo.
- c. Możliwość określenia precyzyjnej kontroli dostępu do zasobów (*ACL – Java Access Control List*).
- d. Mechanizm obsługi wyjątków – potencjalnie lepsza stabilność.

-Servlet jest ładowany przy pierwszym odwołaniu doń lub przy starcie engine-u. Pozostaje w pamięci do momentu jawnego usunięcia lub zatrzymania serwera.

-Można wskazać trzy metody servletu wyznaczające jego cykl życiowy:

-void init(ServletConfig config) - Uruchamiana przy załadowaniu servletu.

-void service(ServletRequest request, ServletResponse response) - Metoda uruchamiana przy każdym żądaniu skierowanym do servletu. Implementacja musi uwzględniać wielowątkowe wykonanie.

-void destroy() - Uruchamiana przy wyłączaniu servletu.

38. Omów składnię JSP oraz stosunek tej technologii do technologii servletów.

-Umożliwiają osadzanie kodu wewnątrz struktury HTML. Wewnątrz struktury dokumentu HTML kod Javy jest umieszczany wewnątrz znaczników: `<% ... %>`.

-Kod strony, opatrzony rozszerzeniem .jsp jest następnie przetwarzany przez silnik JSP do postaci servletu i kompilowany. Wytworzony kod opiera się na pakiecie *javax.servlet.jsp* i stanowi rozszerzenie klasy *javax.servlet.http.HttpServlet*, implementujące interfejs *javax.servlet.jsp.HttpJspPage*.

39. Przedstaw założenia i zastosowanie języka SGML.

-Język znaczników (*markup language*) – zbiór konwencji znaczników stosowanych do opisywania (kodowania) tekstu. Określa:

- a. znaczniki dozwolone i wymagane;
- b. zapewnia niezawodny sposób ich wyróżniania ze zwykłego tekstu;
- c. [definiuje znaczenie poszczególnych znaczników].

-SGML = standard opisu tekstu za pomocą znaczników. Określany jako metajęzyk (?) (wrócimy jeszcze do tego pojęcia).

-Charakter **opisowy** a nie proceduralny: w tekście umieszczone są znaczniki określające interpretację poszczególnych elementów, a ew. elementy proceduralne, opisujące wymagane czynności np. sformatowania tekstu, istnieją odrębnie – w postaci osobnych zasobów.

-SGML wprowadza pojęcie **typu dokumentu**, specyfikując DTD (*Dynamic Type Definitions*).

-Typ dokumentu jest formalnie zdefiniowany w terminach jego składowych i ich struktury.

-Dla znanych typów dokumentów (tj. dla których istnieją definicje typu), można sprawdzić zgodność dokumentu ze specyfikacją typu. Program dokonujący takiego sprawdzenia nazywamy *parserem*.

-Znając typ dokumentu można w bardziej efektywny i wyspecjalizowany sposób przetwarzać wszelkie dokumenty zgodne z tym typem. Jeśli wykonywane jest przetwarzanie dokumentu, to odpowiedni program nazywamy z kolei *procesorem*.

40. W jaki sposób możliwości oferowane przez DTD w języku SGML rozbudowują podstawowy, czysto hierarchiczny model dokumentu?

-Definicje Typu Dokumentu stanowią cechę szczególną XML a zarazem integralną część specyfikacji języka.

-Definicja typu pozwala sprawdzić strukturalną poprawność dokumentu. Można je postrzegać jako język opisu dokumentu.

-Definicje typu mogą występować w pliku dokumentu, lub w plikach zewnętrznych.

-W przeciwieństwie do SGML, obecność definicji typu nie jest obowiązkowa.

-Powstały język HTML pozostał jednak silnie związany ze swym protoplastą:

-HTML stanowi istotnie okrojony podzbiór SGML;

-Składnia HTML zdefiniowana jest przy użyciu środków definiowania struktury dokumentów SGML (istnieje definicja HTML w postaci DTD);

41. Wyjaśnij pojęcie metajęzyka. W jakim zakresie stosuje się ono do SGML i XML?

-Pojęcie meta-poziomu oznacza zwykle definicję rozpatrywanego przez nas zbioru elementów, z którą ten ostatni jest związany relacją „jest wystąpieniem”.

-Pojęcie dotyczy często danych i metadanych (schemat jest poziomem „meta” w stosunku do danych gromadzonych w bazie) albo języków programowania (mechanizmu pozwalający kontrolować zachowanie

rozpatrywanego przez nas poziomu; np. wołanie procedury). Przykładem może być klasa, jako definicja narzucająca ograniczenie na strukturę jej obiektów oraz definiująca metody czy aktywne reguły dla tych obiektów. Zob. też – metamodel w UML.

-Pojęcie metajęzyka sugerowałoby zatem, że SGML oraz XML są językami służącymi definiowaniu innych języków. Termin zatem pasowałby raczej do DTD. Stąd określanie bez bliższego komentarza SGML oraz XML „metajęzykami” może być nieco mylące...

42. Scharakteryzuj język XML. Jakie założenia legły u podstaw jego specyfikacji?

- Użyteczny bezpośrednio w Internecie.
- Uniwersalny wobec dziedzin zastosowań.
- Zgodny SGML.
- Zapewnienie wsparcia dla tworzenia programów przetwarzających dokumenty XML.
- Minimalizacja elementów opcjonalnych w specyfikacji.
- Klarowność i czytelność dla człowieka.
- Konieczność szybkiego zaprojektowania języka.
- Formalny i spójny styl projektu.
- Łatwość tworzenia dokumentów XML.
- Brak wymogów odnośnie zwięzłości składni.
- Język tekstowy;
- Rozszerzalny: zestaw znaczników nie stanowi zamkniętej listy;
- Elastyczna struktura;
- Niezależność od platformy;
- Możliwość definiowania reguł poprawności dla struktury dokumentu (definicja typu dokumentu);
- Stosunkowo rygorystyczna składnia;
- Opisuje klasę obiektów danych, zwaną **dokumentami XML**;
- Częściowo określa zachowanie programów, które je przetwarzają;
- Zgodny z SGML: określany jako profil zastosowań SGML albo ograniczony SGML.

43. Przedstaw podstawowe reguły poprawności składniowej dokumentów XML.

- Zgodny z definicją produkcji „*document*” (tj. m.in. zawiera deklarację XML);
- Element główny („korzeń”) występuje raz w każdym z dokumentów i nie jest zawarty w żadnym innym elemencie. Pozostałe elementy muszą być poprawnie zagnieżdżone. W oparciu o takie zagnieżdżenie mówi się o relacji rodzic-dziecko.
- Spełnienie wszystkich ograniczeń poprawnościowych ujętych w specyfikacji;
- Wszystkie parsowalne encje do których odwołuje się dokument spełniają warunek poprawności;

44. Jakie rodzaje zawartości mogą znaleźć się w dokumentach XML?

- elementy puste (pojedynczy znacznik zakończony „/”);
- elementy zawierające tylko tekst;
- elementy złożone, tj. posiadające podelementy;
- elementy o zawartości mieszanej (tj. zarówno tekst jak i podelementy – mogą być na różne sposoby przeplecione);

45. Na czym polega poprawność strukturalna dokumentu XML i w jakim celu wprowadzono to kryterium?

- Dokument poprawny strukturalnie (*valid*): zgodność z definicją typu dokumentu (DTD).
- Definicje Typu Dokumentu stanowią cechę szczególną XML a zarazem integralną część specyfikacji języka.
- Definicja typu pozwala sprawdzić strukturalną poprawność dokumentu. Można je postrzegać jako język opisu dokumentu.

46. W jaki sposób definicje typu dokumentu mogą być łączone z opisywanym dokumentem?

- Definicje typu mogą występować w pliku dokumentu, lub w plikach zewnętrznych.
- W przeciwieństwie do SGML, obecność definicji typu nie jest obowiązkowa.

47. W jaki sposób definiuje się elementy w DTD? Jakie rodzaje zawartości elementu wyróżniamy?

- W najprostszej postaci, jest to nazwa elementu zawieranego (podelementu) ujęta w nawias;
- Oznacza to dokładnie jedno wystąpienie takiego podelementu;
- Tę licznosc można modyfikować symbolami: „+” oznacza [1..*], „?” oznacza [0..1], „*” oznacza [0..*].

- Nazwy podelementów (wraz z określeniami ich licznosci) można grupować za pomocą:
 - symbolu przecinka => oznacza sekwencję,
 - symbolu „|” => oznacza wybór dokładnie jednego z rozdzielonych tak elementów;
- W oparciu o powyższe konstrukcje można tworzyć złożone deklaracje używając nawiasów:
 - <!ELEMENT firma ((oddział)+ | przedstawicielstwo)>
 - <!ELEMENT oddział(dyrektor, handlowiec*, osoba_kontaktowa?)>

48. Na czym polega zawartość mieszana i jakim podlega ograniczeniom jej deklaracja?

- Może stanowić albo sam tekst (co sprowadza się do zawartości tekstowej),

... albo dowolna liczba powtórzeń tekstu i podelementów;

Istnieje istotne ograniczenia na postać deklaracji. Efektywnie jedyna dozwolona postać deklaracji to: `<!ELEMENT nazwaMieszanego (#PCDATA | element_1 | element_2)*>`

49. Jakie niedogodności wykazuje sposób określania zawartości elementowej w DTD?

Podstawową trudnością jest zadeklarowanie jako możliwej zawartości elementowej **podzbioru** z jakiegoś repertuaru dokumentów.

Np. *konieczność umieszczenia dwóch dokumentów tożsamości, spośród kilku zdefiniowanych rodzajów*

=> potrzeba zadeklarować osobno każdą dwuelementową kombinację:

```
<!ELEMENT tozsamosc ((dowod_osobisty | prawo_jazdy) | (dowod_osobisty, paszport) | (prawo_jazdy, paszport) | (dowod_osobisty, indeks) | (prawo_jazdy, indeks) | (paszport, indeks))>
```

50. Omów sposób deklarowania atrybutów w DTD. Jakie najważniejsze typy są dostępne?

-Atrybuty występują w postaci par nazwa="wartość" w otwierających znacznikach XML.

-Deklaracje atrybutów dla danego rodzaju elementu zawarte są w deklaracji ATTLIST odnoszącej się do nazwy opisywanego elementu.

51. Czemu służą przestrzenie nazwowe i jaką rolę odgrywa w ich definicji URI?

- Umożliwiają bezkolizyjne korzystanie z niezależnie powstałych definicji elementów i ich atrybutów.
- Deklarowane w postaci:

```
xmlns:NASZ_PREFIKS="http://uri.do.naszego/prefiksu"
```

- URI nie musi wskazywać na plik z definicją typu dokumentu; jedynie unikalnie identyfikuje daną przestrzeń nazwową. Zaleca się, aby w tym celu nie stosować względnych adresów URL.
- Deklaracja zwykle jest zlokalizowana jako atrybut w korzeniu dokumentu, jako że obowiązuje wówczas dla wszystkich elementów podrzędnych.
- Użycie: NASZ_PREFIKS:nazwaElementu
- Prefiks xml jest zarezerwowany: predefiniowane atrybuty (xml:lang, xml:space, xml:base) -> zob. dalej.

52. Czym są w XML encje (entities)? Jakich ich rodzaje wyróżniamy?

-Umożliwiają wyodrębnianie i odwoływanie się do nazwanych elementów XML.

-Encje **ogólne przetwarzane**: odwołania w postaci `&nazwaEncji`;

wewnętrzna: treść zawarta w deklaracji encji; Np.: `<!ENTITY pjwstk "Polsko-Japońska Wyższa Szkoła Technik Komputerowych">`

zewewnętrzna: treść zawarta w zewnętrznej dokumentacji;

```
<!ENTITY tekstNaglowka SYSTEM "naglowek.txt">
```

(*mechanizm jednostek zewnętrznych rozwijany jest jako XML Inclusions (Xinclud)*)

-**Ogólna zewnętrzna nieprzetwarzana**: inna postać odwołania – nazwa encji jako wartość atrybutu.

Zawiera etykietę NDATA a za nią nazwę formatu. Np. `<!ENTITY logo SYSTEM "NaszeLogo.jpg"NDATA obraz_jpg>`

-**Parametryczna**: oznaczana symbolem %. Odwołania do niej mogą występować tylko w ramach tego samego DTD. Służą określeniu powtarzających się w danym DTD treści.

53. Jakich niepożądanych ograniczeń narzuca definicja DTD?

-Określenie zawartości elementu jako tekstowej powoduje pełną dowolność co do jego zawartości;

-Elementy nie mogą określać wartości domyślnych;

-Brak ograniczenia na unikalność wartości atrybutu;

-Brak ograniczenia na unikalność zawartości elementu;

-Słabe możliwości kontroli typów;

-Brak możliwości odrębnego definiowania typów wykorzystywanych w różnych miejscach; brak dziedziczenia;

-Niemożność ustalenia kolejności atrybutów;

-Niemożność zmiany kolejności elementów;

-Przestrzenie nazwowe nie są wspierane w DTD;

-Rozbieżności składniowe w stosunku do zwykłego XML.

54. Jakich zalet posiada XML Schema w stosunku do DTD?

-W przeciwieństwie do DTD, XML Schema jest osobnym językiem, więc należy wyspecyfikować jego przestrzeń nazwową. Przypomnijmy, że przewidziano dlań prefiks *xsd*.

-Poza predefiniowanymi typami (znaczny repertuar) istnieje możliwość deklarowania własnych typów.

-Ogólnie zawartość schematu może być następująca:

-deklaracja XML;

-element-korzeń schema;

-definicje typów;

-deklaracje atrybutów i elementów.

55. Wyjaśnij różnicę pomiędzy definicją typu złożonego a deklaracją elementu złożonego w XML Schema.

-Definicje typów tworzą nowe typy dostępne do wykorzystania;

-Deklaracje elementów określają nazwy i typy elementów i atrybutów, które mogą pojawić się w odpowiednich miejscach dokumentu.

56. Omów metodę definiowania typów pochodnych XML Schema poprzez ograniczenia (*restrictions*)? Jakiego rodzaju ograniczenie mogą wystąpić?

-Tworzeniu pochodnych typów prostych poprzez nakładanie ograniczeń służy element `xsd:restriction`, umieszczany wewnątrz elementu `xsd:simpleType`.

-Element `restriction` zawiera atrybut `base`, określający typ bazowy definicji. Może zawierać szereg podelementów, określanych jako *constraining facets* (aspekty ograniczające).

57. Czemu służą w XML Schema tzw. grupy elementów oraz grupy atrybutów. Jaka konstrukcja DTD jest ich odpowiednikiem?

-Ograniczają nadmiarowość definicji schematu.

-Zastosowanie (ograniczenie nadmiarowości i uczynienie schematów bardziej przejrzystymi) odpowiada encjom parametrycznym z DTD.

-Do zadeklarowania grupy stosowane są elementy odpowiednio: `<xsd:group> ... </xsd:group>` oraz `<xsd:attributeGroup> ... </xsd:attributeGroup>`, posiadające atrybut `name`.

58. Omów możliwości XML Schema w zakresie nakładania ograniczeń unikalności i integralności referencyjnej.

-**Integralność referencyjna** – określa zależność klucza obcego. Przedmiotem odwołania musi być wartość zadeklarowana jako klucz.

-Deklarowane za pomocą elementu `keyref`: `<xsd:keyref refer="nazwaKlucza"> ... </xsd:keyref>`

-Atrybut `refer` określa, do jakiego klucza odwołują się referencje.

59. Objasnij możliwe style tworzenie definicji schematu.

-**Poprzez zagnieżdżanie**: definicje potrzebnych typów tworzone anonimowo na potrzeby poszczególnych deklaracji.

-**Plaski katalog elementów**: elementy tworzące hierarchię są deklarowane globalnie, zaś w ich docelowych miejscach występowania w schemacie stosuje się odwołania do tych globalnych deklaracji.

-**Definiowanie typów**: globalnie są definiowane nazwane typy, wykorzystywane następnie w deklaracjach elementów i atrybutów. Metoda najskuteczniej ogranicza nadmiarowość, choć wprowadza pewien narzut, wobec czego nadaje się szczególnie dla bardziej rozbudowanych schematów.

60. Scharakteryzuj język XPath i wyjaśnij pojęcie kroku w tym języku.

-Język deklaratywny służący wskazywaniu elementów, atrybutów, lub całych fragmentów dokumentu XML.

-Posiada zwięzłą nie-XML-ową składnię, przyjętą w celu umożliwienia umieszczania wyrażeń XPath w wartościach atrybutów oraz w URI.

-Typy zwracanych wartości:

- a. boolean;
- b. number (liczba zmiennoprzecinkowa);
- c. string;
- d. node-set (zbiór węzłów).

-Ścieżka (ciąg określający) XPath jest zbudowana z tzw. kroków (*step*), oddzielonych symbolem „/”.

-Krok może reprezentować: element, atrybut lub funkcję. Poszczególne kroki zawężają obszar przeszukiwania.

61. Omów budowę kroku w języku XPath.

-Składnia pojedynczego kroku jest następująca:

kierunek (*axis*);

podwójny dwukropek: “::”

symbol „*” albo nazwa elementu, albo funkcja:

`node()`

`text()`

`comment()`

`processing-instruction()` <- może zawierać nazwę szukanej instrukcji przetwarzania;

dowolna liczba *predykatów* umieszczonych w nawiasach kwadratowych: „[]”

62. Jaką rolę pełnią w XPath predykaty? Jakie środki udostępniono do ich tworzenia?

-Predykaty zmieniają *pozycję kontekstu* i *rozmiar kontekstu*.

-Występują opcjonalnie, umieszczane wewnątrz nawiasów kwadratowych celem zawężenia zwróconego zbioru elementów.

-Muszą zwracać wartość logiczną.

63. W jaki sposób język XPointer rozszerza funkcjonalność dostarczaną przez XPath?

-Stanowi rozszerzenie języka XPath.

-Uzupełnia system adresów URI używanych w XLink (zob. dalej).

-Pozwala oznaczać i wskazywać fragmenty dokumentów XML.

-Może traktować dokument zarówno jako drzewo węzłów (jak XPath), lub jako ciąg znaków (w treści elementów), pozwalając wskazywać jego podciągi.

-Wprowadza dwa specjalne typy wskaźników:

- a. punkt (*point*) w dokumencie XML;
- b. zakres (*range*) pomiędzy punktami końcowymi (mogącymi się znajdować w różnych elementach);

-Rozróżnia się także (używając środków XPath) węzły (*node*) i elementy (*element*).

64. Scharakteryzuj język XSL. Czemu służą dokumenty XSLT?

-Określa zakresy przeszukiwania; wskazania zawartości do przetworzenia za pomocą XPath.

-Definiuje sposób przekształcenia poszczególnych rodzajów wyszukanej zawartości.

-Zakładane przez XML oczyszczenie informacji z zawartości związanej z prezentacją pozostawia otwartym problem wizualizacji takich dokumentów.

-W celu realizacji niezbędnych przekształceń powstał język XSL (*eXtensible Stylesheet Language*). Swego rodzaju protoplastą tego mechanizmu były kaskadowe arkusze stylów (CSS), jednak realizacja języka XSL jest odmienna.

66. XLink

Pokonuje następujące ograniczenia odsyłaczy HTML:

-brak możliwości oznaczenia charakteru odnośnika;

-mają charakter binarny (łączy jedynie pary dokumentów: źródłowy i docelowy);

-są zawarte w dokumencie źródłowym (nie mogą być przechowywane zewnętrznie);

-problemem jest aktualizacja;

-mogą wskazywać dokumenty lub punkty w dokumentach, ale nie np. elementy dokumentu XML.

65. Objasnij, jakie style budowy szablonów XSLT można wyróżnić.

66. Objasnij różnicę pomiędzy dokumentami kierowanymi treścią a dokumentami kierowanymi układem.

67.

68. Czemu służy specyfikacja XSL – Formatting Objects? W jaki sposób udoskonala istniejące wcześniej w tym obszarze zastosowania?

69. Przedstaw (w oparciu o poznaną budowę dokumentu XSL-FO), najważniejsze możliwości oferowane przez ten język.

70. Podaj definicję przestrzeni nazwowej. W jakim celu wprowadzono to pojęcie?

71. Objasnij sposób deklarowania przestrzeni nazwowych (w tym przestrzeni domyślnej).

72. Scharakteryzuj interfejs DOM (model dokumentu, sposób działania, dostępna funkcjonalność).

73. Wskaż najważniejsze pojęcia wprowadzone w DOM dla reprezentowania zawartości dokumentu.

74. Czemu służą moduły DOM Traversal i DOM Range?

75. Scharakteryzuj model działania interfejsu SAX.

76. Jakie zdarzenia występują w interfejsie SAX i jakie środki zdefiniowano dla ich obsługi?

77. Porównaj właściwości (zalety i wady) interfejsów DOM i SAX.

78. W jakim celu zdefiniowano interfejs JAXP? Jakie interfejsy on wspiera?

-Wspólny standardowy interfejs tworzący podstawę dla wykorzystania interfejsów SAX, DOM oraz XSLT z poziomu języka Java, zapewniając niezależność wykorzystującego go kodu od konkretnych implementacji.

-Podstawowe interfejsy zawarte w pakiecie javax.xml.parsers. Pozwalają na podłączania dowolnych zgodnych parserów.

-W zakresie interfejsów SAX i DOM udostępnia fabryki obiektów realizujących oba te rodzaje przetwarzania.

-W zakresie wsparcia XSLT, JAXP dostarcza interfejsów zdefiniowanych w pakiecie javax.xml.transform.

79. Jakie zalety oraz jakie ograniczenia posiada technologia JAXB w stosunku do innych interfejsów dla obsługi XML?

80. Objasnij koncepcję i składniki technologii JAXB.

81.

82. Przedstaw scenariusz budowy aplikacji w oparciu o JAXB.

83. Jakie rozwinięcia interfejsu DOM wprowadzono specjalnie dla Javy i czym są motywowane?

-Narzędzie udostępniające zaprojektowany dla Javy model dokumentu XML. Jak sugeruje nazwa, inspirowany DOM; stanowi rozwiązanie alternatywne, łatwiejsze w użyciu, lecz nieprzenośne na inne języki programowania.

-Rozwijane jako Java Community Process (JSR 102).

-Możliwości integracji z DOM i SAX: zarówno wejście jak i wyjście przy przetwarzaniu z użyciem JDOM może przybrać postać: plików XML, drzew DOM lub zdarzeń SAX.

-Istnieje możliwość wykorzystania XSLT i XPath do struktur JDOM.

-Cecha szczególna: API JDOM jest oparte na klasach konkretnych Javy, nie zaś na interfejsach. Zapewnia to prostotę i zwięzłość (używamy konstruktorów zamiast pośrednictwa „fabryk”) kosztem elastyczności implementacji.

84. Jakie specyficzne właściwości posiadają z punktu widzenia języków zapytań dokumenty XML jako dane półstrukturalne?
85. Czym jest XML Information Set i w jakim celu został wprowadzony?
86. Objasnij wzajemny stosunek technologii XSLT, XPath, XQuery oraz XML Information Set.
87. Scharakteryzuj język XQuery. Jaką rolę odgrywa w nim język XPath?
88. Objasnij pojęcie kontekstu wyrażenia XPath.
89. Objasnij budowę wyrażenia „FLWOR” w XQuery.
90. Jakie możliwości oferuje XQuery w zakresie użycia funkcji oraz definiowania ich sygnatur?
91. Jaką rolę pełni „prolog” zapytania XQuery.
92. Omów zalety i wady języka XQuery.
- 93. Dokonaj przeglądu właściwości technologii CORBA, DCOM i RMI. Jakie problemy dla współdziałania systemów się z nimi wiążą?**
- CORBA**
- Specyfikacja neutralna w stosunku do producentów, niezależna od konkretnego języka programowania: wspiera szereg wiązań.
 - Definiuje własny protokół wymiany komunikatów IIOP (*Internet Inter-ORB Protocol*).
 - Współdziałanie jest realizowane poprzez działające w każdej części rozproszonej aplikacji oprogramowanie pośredniczące zwane brokerem żądań obiektów (ORB).
 - W porównaniu z innymi technologiami, odpowiednio zaprojektowane oprogramowanie oparte na standardzie CORBA zapewnia szczególnie wysoką wydajność rozproszonej komunikacji.
 - Wady: ekspresyjność środków programistycznych jest ograniczona dążeniem do wyodrębnienia wspólnego mianownika wszystkich wspieranych języków programowania.
- RMI**
- Technologia oparta na języku Java. Stąd nie jest tu potrzebny abstrakcyjny język interfejsów, jak to ma miejsce w standardzie CORBA. Zamiast tego, do opisanego interfejsów do odległych obiektów stosuje się bezpośrednio deklaracje interfejsów Javy, z zachowaniem odpowiednich reguł (m.in. serializowalność przekazywanych do odległej metody parametrów).
 - Podobnie jak w wypadku CORBA, celem sformułowania komunikatu do określonego obiektu (odpowiadającego wywołaniu jego metody), aplikacja musi wcześniej dysponować jego referencją. Jednym ze sposobów uzyskania takowej jest odwołanie się do usługi nazwowej.
 - Używa *Java Remote Method Protocol* (JRMP).
- DCOM**
- Zbudowany jako warstwa oparta na proceduralnym DCE RPC.
 - Specyfikuje język definicji interfejsów IDL, przypominający składnię C++. Podobnie jak w wypadku CORBA interfejsy te są kompilowane celem uzyskania kodu tzw. szkieletów i pieńków w jednym z obsługiwanych języków. Interfejsy są również rejestrowane w rejestrze systemowym.
 - Podobnie jak RMI obsługuje rozproszone zbieranie nieużytków (*distributed garbage collection*).
 - Technologia ta wprowadza również własny protokół: ORPC (*Object Remote Procedure Call*).
- 94. Wyjaśnij pojęcie usług Webu (Web Services).**
- W przeciwieństwie do tradycyjnej treści Webu (informacja wprowadzana przez człowieka i prezentowana człowiekowi), usługa Webu jest **dostępną poprzez sieć komponentem aplikacyjnym przeznaczonym do wykorzystania przez inne aplikacje, zwane aplikacjami klienckimi**.
 - Definicja według [Webservices.org](http://www.webservices.org): *Hermetyzowane, luźno skojarzone „kontraktowane” funkcje, oferowane poprzez standardowe protokoły*.
 - „Hermetyzowane”: implementacja nigdy nie jest widoczna z zewnątrz.
 - “Luźno skojarzone”: modyfikowanie danej implementacji nie generuje problemu propagacji zmiany.
 - “Kontraktowane” opisy działania funkcji oraz specyfikacja ich interfejsów jest publicznie dostępna.
- 95. Objasnij architekturę Web Services. Na jakich protokołach się opiera?**
- Podobnie jak w wypadku tradycyjnego oprogramowania pośredniczącego niezbędne jest wyróżnienie **klienta, dostawcy usług** oraz serwisu umożliwiającego klientowi zlokalizowanie dostawcy.
 - Serwis taki, zwany **brokerem usług**, umożliwia klientowi wyszukiwanie usług, zaś dostawcy publikowanie ich opisów.
 - Protokół nie jest binarny: środowisko (Internet) wymaga oparcia komunikacji na rozpowszechnionym protokole. Zakłada się użycie http (ominięcie problemu zapór ogniowych), choć potencjalnie mogą to być też inne protokoły.
- 96. Czym jest SOAP (Simple Object Access Protocol)?**
- SOAP jest w swojej istocie protokołem bezstanowym i jednokierunkowym. Umożliwia jednak budowę bardziej złożonych wzorców komunikacji.
- 97. Omów budowę komunikatu SOAP.**
- Elementem-korzeniem jest tzw. koperta (*envelope*). Może ona zawierać: nagłówek (*Header*) – opcjonalnie oraz ciało komunikatu (*Body*).

-Blok nagłówkowy (*header blocks*), będące podelementami elementu *Header*, zawierają informacje precyzujące sposób traktowania wiadomości (np. kodowanie zawartości czy autentykacja).

-Pozycja nagłówkowa jest opisywana przez atrybuty:

- a. *role*: określa adresata danej pozycji (może to być adresat docelowy komunikatu lub jakiś węzeł pośredniczący). Standard określa m.in. rolę „next”, wskazującą na przetwarzanie tej pozycji nagłówkowej przez najbliższy węzeł SOAP. Brak atrybutu *role* oznacza przeznaczenie danej pozycji nagłówkowej dla ostatecznego odbiorcy.

- b. *mustUnderstand*: określa, czy pozycja musi zostać przetworzona przez jej adresata.

98. Jakie informacje można wyróżnić w opisie usługi oraz potencjalnego klienta usługi? Dlaczego warto przedstawiać taki opis w postaci czytelnej maszynowo?

99. Wyjaśnij, czym jest semantyka interakcji. W jakim zakresie jest ona opisywana w dzisiejszych technologiach usług Webu?

100. Czemu służy specyfikacja WSDL? Jakie informacje zawiera dokument WSDL?

-Ponieważ opisuje usługę, nie zaś wymagania klienta, stąd budowa swoistej giełdy, gdzie zarówno usługi jak i zapotrzebowania byłyby przedmiotem wyszukiwania i ew. kojarzenia przez odrębnego brokera, wymaga dodatkowo określenia sposobów opisu takich zapotrzebowań.

-WSDL opisuje technikę współdziałania z usługą. Zakłada, że semantyka usługi jest opisana na zewnątrz i może być jednoznacznie wskazana poprzez identyfikator. Tym samym „kontrakt” dotyczący znaczenia i celu jest oddzielony od „kontraktu” określającego mechanikę współdziałania.

101. Omów budowę dokumentu WSDL.

-Jest to opis abstrakcyjnego interfejsu, tj. niezależnego od protokołu transportowego oraz od języka programowania.

-Struktura pliku WSDL jest zdefiniowana w specyfikacji w postaci schematów XML Schema.

-Typy danych używanych w opisywanych przez dokument WSDL interfejsach mogą być zdefiniowane w dowolnym systemie typów, choć domyślnie stosuje się właśnie XML Schema.

-Funkcjonalność odpowiada np. IDL ze standardu OMG CORBA.

102. Scharakteryzuj technologię UDDI. Jakie cele ma ona realizować?

-Specyfikacja, tworzona przez konsorcjum OASIS, określa rozproszony katalog, zawierający zarówno informacje o samych firmach jak i o udostępnianych przez nie usługach Webu, czyli swoistą „książkę telefoniczną”. Rozwijając tę metaforę, specyfikacja wyróżnia:

-„zielone strony”: techniczny opis usług wraz z odnośnikami URL (w założeniach nie muszą to być koniecznie usługi Webu);

-„białe strony”: identyfikacja, adresy i inne dane kontaktowe firm;

-„żółte strony”: wykaz firm ułożony według klasyfikacji przemysłowej;

-Rejestr zaprojektowano jako logicznie scentralizowany, zaś fizycznie rozproszony i replikowany. Może być dostępny zarówno tradycyjnie (interfejs WWW), jak i programowo (jak usługi Webu).

-Interfejs programistyczny wyróżnia część służącą formułowaniu zapytań oraz część służącą publikowaniu opisów.

103. Omów budowę rejestru UDDI.

-**Informacja o usłudze**: Informacja ta znajduje się wewnątrz elementu zawierającego informacje biznesowe. Składa się z elementów *businessService* (opis usługi) oraz *bindingTemplate* (informacja niezbędna dla wywołania usługi): informacje techniczne niezbędne do połączenia się z usługą; m.in. czy jest samodzielna, jej URL itp.

-**Specyfikacja usługi**: Element *bindingTemplate* zawiera zbiór odnośników do specyfikacji (zawartych w elementach *tModel*). Każdy element *tModel* stanowi opis pewnej specyfikacji, na której oparta jest usługa. Komplet takich specyfikacji określa wzorzec zgodności z daną usługą.

104. Jaką funkcjonalność oferuje API do rejestru UDDI?

105. Czym jest ebXML? W jakim celu powstał?

106. Jakie dokumenty w technologii ebXML muszą powstać, aby nawiązać współdziałanie partnerów biznesowych?

107. Czym jest i jak został skonstruowany „Message Service” w technologii ebXML?

108. Czym charakteryzuje się Architektura Zorientowana na Usługi (Service Oriented Architecture) SOA?