



poziom: **Core**

wersja: **3.0**

moduł: **B : Wytwarzanie**

SYLABUS

Niniejszy dokument jest sylabusem obowiązującym dla certyfikatu EUCIP ver. 3.0.
Prezentuje obszary wiedzy, których znajomość jest niezbędna do zaliczenia modułu „**Wytwarzanie**”.

Zastrzeżenie:

Dokument ten został opracowany ze szczególną starannością na podstawie materiałów źródłowych pochodzących z Fundacji ECDL. Polskie Towarzystwo Informatyczne i Fundacja ECDL zastrzegają sobie prawo do zmian treści dokumentu oraz wyłączenia od odpowiedzialności za jakiegokolwiek straty i szkody powstałe na skutek wykorzystania niniejszego dokumentu i treści w nim zawartych.

Cel

Obszar wiedzy: **Wytwarzanie** **Moduł B: Wytwarzanie** – obejmuje aspekty tradycyjnych technik projektowania, specyfikacji, implementacji, testowania i integracji systemów informatycznych (SI). Wymaga on od kandydata zrozumienia cyklu życia systemu, znajomości procesów budowy SI oraz trendów rozwoju SI. Kandydat powinien również potrafić zastosować różne metodyki i narzędzia budowy SI. Od kandydata wymaga się również znajomości zasad projektowania interfejsu użytkownika, stron internetowych, hipermediów jak również zasad związanych z wykorzystaniem relacyjnych baz danych i hurtowni danych. Kandydat powinien posiadać podstawową wiedzę z zakresu języków zapytań oraz podstawowych zagadnień administrowania bazami danych. Wymagana jest również znajomość umiejętności wykorzystania podstawowych struktur danych i algorytmów, rozpoznawania różnych języków programowania oraz konstrukcji programistycznych, szacowania utrzymywania systemu oraz dokumentowania i testowania systemów informatycznych.

Kategoria	Obszar wiedzy	Nr Ref.	Zakres
B.1 Proces i metody tworzenia systemów	B.1.1 Oprogramowanie w systemach przetwarzania danych	B.1.1.1	Opisz system przetwarzania danych, jako kombinację sprzętu oraz oprogramowania wbudowanego, systemu operacyjnego, aplikacji, systemowych danych konfiguracyjnych i danych zdefiniowanych przez użytkownika.
		B.1.1.2	Rozpoznaj i wymień przykłady oprogramowania systemowego.
		B.1.1.3	Rozpoznaj i wymień przykłady aplikacji.
	B.1.2 Cykle życia tworzenia systemów	B.1.2.1	Opisz typowe etapy tworzenia systemów.
		B.1.2.2	Dokonaj porównania różnych klasycznych modeli cyklu życia tworzenia systemów, takich jak kaskadowy, spiralny, przyrostowy lub prototypowanie.
		B.1.2.3	Opisz cykl życia systemu w kategoriach analizy, tworzenia i wdrażania, wykorzystania, konserwacji oraz wycofania.
		B.1.2.4	Opisz specyfikacje wymagań i projektów, takie jak specyfikacje organizacyjne oraz techniczne.
	B.1.3 Narzędzia związane z tworzeniem oprogramowania	B.1.3.1	Opisz zastosowanie narzędzi na różnych etapach tworzenia systemów, takich jak narzędzia CASE wyższego lub niższego poziomu oraz zintegrowane narzędzia CASE.
		B.1.3.2	Opisz silne i słabe strony różnych środowisk tworzenia oprogramowania.
		B.1.3.3	Opisz zastosowanie prostych narzędzi programistycznych w dziedzinie edycji, kompilacji, testowania i debugowania.
	B.1.4 Testowanie i wdrażanie systemów	B.1.4.1	Opisz różne rodzaje testowania i udowodnij, że mogą być zastosowane w czasie cyklu życia tworzenia systemów.
		B.1.4.2	Opisz najważniejsze zagadnienia związane z etapem implementacji i wdrażania systemów, takie jak udostępnienie systemów użytkownikom, migracja danych, szkolenie użytkowników i wsparcie początkowe.
		B.1.4.3	Opisz silne i słabe strony różnych podejść implementacyjnych, takich jak "big bang", model krokowy i rdzeniowy, lub metoda typu "rollout".
		B.1.4.4	Wymień typowe komponenty podręczników użytkownika systemów oraz dokumentacji technicznej.
	B.1.5 Kontrola i bezpieczeństwo systemów	B.1.5.1	Dokonaj rozróżnienia pomiędzy środowiskami tworzenia, testowania i produkcji oraz uzasadnij znaczenie podejścia strukturalnego do udostępniania systemu użytkownikom, takiego jak systemy kontroli wersji lub procedury dystrybucji oprogramowania.

		B.1.5.2	Rozpoznaj ryzyka związane z awariami systemów i opisz środki ochrony poufnych danych przedsiębiorstwa na różnych poziomach, takich jak fizyczny lub proceduralny.
		B.1.5.3	Opisz codzienne procedury bezpieczeństwa w systemach rozproszonych, takie jak tworzenie kopii bezpieczeństwa i kontrola dostępu.
	B.1.6 Trendy w tworzeniu systemów	B.1.6.1	Opisz standardowe i innowacyjne podejścia do tworzenia systemów, takie jak ISO12207, SEI/CMMI, metodologie typu "agile".
		B.1.6.2	Uzasadnij wpływ bieżących dokonań w dziedzinie technicznej architektury, takich jak dwu- lub trzywarstwowe warianty klient-serwer, "n" warstwowe systemy oparte na sieci, architektury zorientowane na usługi, czy rozszerzenia i integracja systemów typu "mainframe" na tworzenie systemów.
		B.1.6.3	Opisz złożoność współczesnego "systemu systemów" i podejścia do zarządzania tą złożonością, takie jak systemy autonomiczne.
B.2 Zarządzanie danymi i bazy danych	B.2.1 Dane i transakcje	B.2.1.1	Opisz znaczenie trwałego nagrywania danych dla przetwarzania transakcji i systemów raportowania.
		B.2.1.2	Opisz w jaki sposób atomowość, spójność, izolacja i trwałość pomagają w zapewnianiu bezpieczeństwa transakcji bazodanowych.
		B.2.1.3	Opisz kwestie związane z projektowaniem i utrzymaniem systemów przeznaczonych dla wielu użytkowników, takie jak redundancja i niespójność danych, problemy z integralnością, elastyczność danych, dostęp współbieżny i bezpieczeństwo.
	B.2.2 Struktura baz danych	B.2.2.1	Dokonaj rozróżnienia pomiędzy systemem zarządzania plikami, a systemem zarządzania bazami danych (DBMS).
		B.2.2.2	Opisz komponenty systemu bazodanowego, takie jak pliki danych, słownik danych, indeksy, dane statystyczne.
		B.2.2.3	Opisz obszary biznesu, w których stosowane są systemy DBMS i wskaż oferowane przez nie korzyści.
		B.2.2.4	Opisz komponenty systemu DBMS, takie jak język zapytań, generator raportów, narzędzia administracyjne, narzędzia kontroli współbieżności, zarządzanie transakcjami, narzędzia kopii bezpieczeństwa i przywracania danych.
		B.2.2.5	Opisz różne role administratora bazy danych, projektanta i programisty bazy danych oraz jej użytkownika.
	B.2.3 Modelowanie danych	B.2.3.1	Zdefiniuj abstrakcję danych i opisz różnice pomiędzy poziomami: fizycznym, konceptualnym (logicznym) oraz widoku (użytkownika).

	B.2.3.2	Dokonaj rozróżnienia pomiędzy różnymi grupami modeli danych, takimi jak logiczny model oparty na obiektach, logiczny model oparty na rekordach, bądź fizyczny model danych.
	B.2.3.3	Opisz reguły logicznych modeli opartych na rekordach, takich jak model hierarchiczny lub sieciowy.
	B.2.3.4	Opisz reguły logicznych modeli opartych na obiektach, takich jak model związków encji lub model obiektowy.
B.2.4 Model relacyjny	B.2.4.1	Opisz korzyści wynikające z zastosowania modelu relacyjnego, takie jak brak redundancji, elastyczność, skalowalność.
	B.2.4.2	Opisz ważne terminy związane z modelem relacyjnym, takie jak relacja, klucz, klucz główny, klucz wtórny, klucz obcy, integralność odwołań.
	B.2.4.3	Opisz za pomocą prostych przykładów proces normalizacji dla pierwszej, drugiej oraz trzeciej postaci normalnej.
B.2.5 Języki zapytań	B.2.5.1	Wskaż różnice pomiędzy proceduralnymi, a nieproceduralnymi językami zapytań.
	B.2.5.2	Opisz podstawowe operacje algebry relacyjnej, takie jak: wybór, projekt, zmiana nazwy, produkt kartezjański, unia, złączenie, różnica zbiorów.
	B.2.5.3	Opisz komponenty strukturalnego języka zapytań SQL, takie jak DDL (ang. Data Definiton Language), DML (ang. Data Manipulation Language), DCL (ang. Data Control Language).
	B.2.5.4	Wyjaśnij polecenia SQL DDL takie jak: create, drop, alter table.
	B.2.5.5	Wyjaśnij polecenia SQL DCL takie jak: grant, revoke.
B.2.6 Zapytania SQL	B.2.6.1	Wyjaśnij podstawowe polecenia SQL DML takie jak insert, delete, update, select.
	B.2.6.2	Wyjaśnij wyrażenia stosowane w SQL takie jak where, order by, group by.
	B.2.6.3	Opisz zastosowanie widoków i specjalnych poleceń SQL, takich jak commit, rollback.
B.2.7 Administracja i bezpieczeństwo bazy danych	B.2.7.1	Opisz najważniejsze procedury administrowania bazą danych, takie jak definicja schematu, struktura pamięci, metody dostępu, modyfikacja schematu i fizycznej organizacji, autoryzacja dostępu do danych,
	B.2.7.2	Opisz problemy związane z bezpieczeństwem i integralnością zawarte w akronimie CIA (ang. Confidentiality, Integrity, Availability), takie jak więzy integralności, przypadkowa utrata integralności danych, przypadkowa utrata
	B.2.7.3	Opisz przykłady różnych rodzajów polityki bezpieczeństwa, takie jak bezpieczeństwo człowieka, fizyczne bezpieczeństwo, bezpieczeństwo systemu operacyjnego, bezpieczeństwo bazy danych.
	B.2.7.4	opisz schematy przywracania w oparciu o różne rodzaje awarii, takich jak błędy logiczne, błędy systemowe, awarie systemu, awarie dysków.

	B.2.8 Hurtownie danych i eksploracja danych	B.2.8.1	Opisz koncepcję i komponenty systemu hurtowni danych.
		B.2.8.2	Opisz koncepcję eksploracji danych.
		B.2.8.3	Wskaż główne zastosowania systemów hurtowni danych.
B.3 Programowanie	B.3.1 Metody i techniki projektowania oprogramowania	B.3.1.1	Opisz główne cechy różnych metod projektowania programów, takie jak projektowanie zorientowane obiektowo, projektowanie według zasady „top-down”, programowanie strukturalne.
		B.3.1.2	Opisz zastosowanie abstrakcji jako techniki rozwiązywania problemów i projektowania oprogramowania.
		B.3.1.3	Opisz konkretne potrzeby starych systemów w projektowaniu programów, takie jak złożona struktura, słaba dokumentacja, wycofane oprogramowanie bądź sprzęt, kluczowy system biznesowy..
		B.3.1.4	Wskaż różnice pomiędzy oprogramowaniem otwartym (ang. open source), a zamkniętym (prawnie zastrzeżonym).
		B.3.1.5	Opisz różne wymagania licencyjne oprogramowania prawnie zastrzeżonego, otwartego, darmowego oraz na licencji freeware.
	B.3.2 Struktury danych i algorytmy	B.3.2.1	Opisz typu danych o określonej strukturze i pozbawione struktury oraz dokonaj rozróżnienia struktur danych, takich jak rekordy, tablice, listy powiązanej.
		B.3.2.2	Oceń dopasowanie typowych algorytmów wyszukiwania i sortowania do określonych struktur danych.
	B.3.3 Języki programowania	B.3.3.1	Opisz zalety głównych rodzajów języków programowania, takich jak funkcjonalne, proceduralne, obiektowe.
		B.3.3.2	Opisz zastosowanie procedur i funkcji, dokonaj rozróżnienia pomiędzy odwołaniem przez wartość i odwołaniem przez referencję.
		B.3.3.3	Zdefiniuj pojęcie składni i opisz jej znaczenie w językach programowania.
		B.3.3.4	Wskaż różnice pomiędzy kompilacją, a interpretacją języków programowania.
	B.3.4 Programowanie zorientowane obiektowo	B.3.4.1	Opisz główne koncepcje projektowania zorientowanego obiektowo.
		B.3.4.2	Opisz koncepcję programowania zorientowanego obiektowo.
		B.3.4.3	Opisz koncepcję klasy, obiektu, instancji, metody, oraz ich związku z programowaniem zorientowanym obiektowo.
		B.3.4.4	Opisz koncepcję dziedziczenia oraz znaczenie tego mechanizmu dla pracy programisty.
		B.3.4.5	Opisz koncepcje abstrakcji i enkapsulacji (ukrywania informacji).

		B.3.4.6	Opisz, w jaki sposób polimorfizm przyczynia się do wydajnego projektowania oprogramowania poprzez tworzenie komponentów wielokrotnego użytku.
	B.3.5 Konstrukcje elementarne	B.3.5.1	Zinterpretuj i oceń instrukcje wejścia / wyjścia.
		B.3.5.2	Zinterpretuj i oceń instrukcje sterowania.
		B.3.5.3	Zinterpretuj i oceń operacje arytmetyczne i logiczne.
	B.3.6 Testowanie	B.3.6.1	Zdefiniuj podstawowe koncepcje w testowaniu, takie jak błąd, wada, awaria. Opisz różne poziomy sprawdzania, testowania, debugowania.
		B.3.6.2	Opisz różne zastosowania i zakresy testów jednostkowych, testów systemowych, testów akceptacji.
		B.3.6.3	Wskaż różnice pomiędzy statyczną, a dynamiczną metodologią testowania, wskaż przykłady automatycznych narzędzi testujących.
	B.3.7 Dokumentacja i utrzymanie	B.3.7.1	Opisz standardową dokumentację stosowaną dla tworzenia i dostarczania oprogramowania, jak np. drzewa decyzyjne, język UML (ang. Unified Modelling Language), kod, komentarze w kodzie, schematy blokowe.
		B.3.7.2	Uzasadnij wartość kodu posiadającego poprawną strukturę i dobrą dokumentację.
		B.3.7.3	Opisz sposoby dokumentowania zmian w oprogramowaniu i dokumentacji oprogramowania.
		B.3.7.4	Opisz metody osiągnięcia wysokiej jakości w konserwacji oprogramowania, takie jak inspekcje kodu, reguły komentowania kodu, dokumenty referencji technicznej.
	B.3.8 Przykłady programowania	B.3.8.1	Dokonaj interpretacji małych segmentów programów zbudowanych na podstawie danej hipotezy.
		B.3.8.2	Wskaż błędy lub braki w kodzie i dokonaj niezbędnych poprawek aby program spełniał dane wymagania.
B.4 Interfejs użytkownika i projektowanie stron internetowych	B.4.1 Interakcja człowieka z komputerem: wskazówki i standardy	B.4.1.1	Zdefiniuj podstawowe założenia teorii komunikacji, takie jak nadawca, wiadomość, odbiorca.
		B.4.1.2	Wyjaśnij wpływ komunikacji na człowieka i wskaż efektywne sposoby przekazywania informacji.
		B.4.1.3	Zdefiniuj koncepcję interfejsu użytkownika i wymień różne rodzaje interfejsów, takie jak tekstowe, graficzne i dźwiękowe.
		B.4.1.4	Opisz różne technologie komunikacyjne trafiające do ludzkich zmysłów, takie jak dźwięki, wskazówki wizualne, cyfrowe zapachy, impulsy dotykowe.

	B.4.1.5	Wymień modele testowania wydajności interfejsu użytkownika pod względem wymagań projektowych i celów.
B.4.2 Projekt graficzny	B.4.2.1	Opisz koncepcje grafiki i animacji (mapa bitowa i grafika wektorowa), cyfrowych nagrań dźwiękowych i wideo, opisz ich różnice, zastosowania, standardowe formaty.
	B.4.2.2	Opisz korzyści wynikające z zastosowania rysunków, zdjęć, kolorów, animacji oraz wskaż podstawowe reguły w zakresie projektowania grafiki, takie jak balans, harmonia, kontrast, różnorodność.
	B.4.2.3	Opisz podstawowe narzędzia do prostych operacji na obrazie w dziedzinie rozmiaru, kształtu, kolorów, kontrastu, przeźroczystości.
B.4.3 Internet i hipermedia: możliwości i ograniczenia	B.4.3.1	Opisz historię rozwoju Internetu i sieci World Wide Web.
	B.4.3.2	Zdefiniuj hipertekst oraz hipermedia i wyjaśnij ich znaczenie w projektowaniu stron internetowych.
	B.4.3.3	Opisz komponenty stosowane najczęściej w budowie stron internetowych, takie jak belka górna i boczna (ang. top bar, side bar), mapa strony, kontakt, funkcja wyszukiwania, pomoc, informacja o ostatniej aktualizacji, ikony nawigacyjne.
	B.4.3.4	Opisz zastosowanie i wartość wewnętrznych i zewnętrznych stron internetowych w przedsiębiorstwie.
	B.4.3.5	Opisz wyzwania, jakie niesie z sobą utrzymanie i konserwacja biznesowej strony internetowej.
B.4.4 Wymagania i metody w projektowaniu stron internetowych	B.4.4.1	Rozpoznaj potrzeby danej grupy docelowej, do której zaprojektowana została strona internetowa.
	B.4.4.2	Opisz zagrożenia związane z umieszczeniem nadmiernej ilości informacji na jednej stronie.
	B.4.4.3	Opisz problemy związane ze słabym doбором kolorów.
	B.4.4.4	Opisz zasady regulujące tworzenie stron przyjaznych dla użytkownika, takie jak czytelność, odpowiedni dobór treści, łatwa i konsekwentna nawigacja, funkcjonalność „gdzie jestem”.
	B.4.4.5	Opisz ogólne kryteria jakościowe dla tekstu w ramach stron internetowych, takie jak ograniczone możliwości przeglądarki, walidacja HTML, zwięzłość tekstu, poprawność językowa tekstu, niewielkie grafiki.
	B.4.4.6	Wyjaśnij potrzebę łatwej nawigacji na stronie internetowej.
	B.4.4.7	Opisz narzędzia wykorzystywane w tworzeniu strony internetowej.

	B.4.4.8	Opisz zastosowanie i cel stosowania diagramów struktury w projektowaniu stron internetowych.
	B.4.4.9	Opisz najważniejsze metody nawigacyjne.
	B.4.4.10	Opisz różne podejścia projektowe do projektowania stron internetowych i powszechnie stosowane techniki, takie jak scenariusze, szkice.
B.4.5 Projektowanie stron internetowych	B.4.5.1	Opisz koncepcję języków znaczników i opisz główne cechy HTML.
	B.4.5.2	Zastosuj podstawowe znaczniki HTML i zinterpretuj znaczniki związane z układem graficznym, takie jak znaki specjalne, znaki podziału, wyrównanie, nagłówki, znaczniki obrazów, tła, kolory, hiperłącza, listy, tabele, formularze, ramki.
	B.4.5.3	Opisz podstawowe zasady graficzne dla zwykłego tekstu pisanego, takie jak rozmiar czcionki czy białe znaki.
	B.4.5.4	Opisz podstawowe elementy XML, jego zastosowania oraz ewolucję HTML do XHTML.
	B.4.5.5	Opisz koncepcję arkuszy stylów, w tym CSS (ang. Cascading Style Sheets) oraz XSL (ang. Extensible Stylesheet Language). Wyjaśnij ich zastosowanie w projektowaniu.
B.4.6 Programowanie zastosowań internetowych	B.4.6.1	Przedstaw różnice pomiędzy technologiami po stronie klienta oraz po stronie serwera i wskaż różne typy języków programowania przeznaczonych dla zastosowań internetowych.
	B.4.6.2	Wymień najważniejsze wyzwania związane z integracją systemów internetowych z istniejącymi systemami.