

Informatické myšlení a programování (Computing) - ECDL / ICDL Sylabus 1.1



Upozornění:

Oficiální verze ECDL / ICDL Sylabu verze 1.0 je publikovaná na webových stránkách ICDL Foundation - www.icdl.org a lokalizovaná verze na webových stránkách pracovní skupiny ECDL-CZ - www.ecdl.cz. Sylabus verze 1.1 je schválen ICDL Foundation pro použití na území ČR v rámci pilotního ověřování.

Přes veškerou péči, kterou ICDL Foundation (vlastník práv konceptu ECDL / ICDL) a ČSKI (národní licenciát) věnovaly přípravě a lokalizaci této publikace, ICDL Foundation ani ČSKI neručí za kompletnost informací v ní obsažených a také nezodpovídají za jakékoli chyby, vynechaný text, nepřesnosti, ztrátu nebo poškození informací, instrukcí či pokynů v této publikaci obsažených. Tato publikace nesmí být reprodukována jako celek ani po částech bez předchozího souhlasu vlastníků práv. ICDL Foundation může na základě vlastní úvahy a kdykoli bez ohlášení provádět jakékoli změny.

Copyright 2017-2024 ICDL Foundation Ltd., lokalizace 2024 CertiCon a.s.

Sylabus modulu M16 - *Informatické myšlení a programování* (Computing) definuje základní rozsah teoretických znalostí a praktických dovedností spojených se schopností informaticky a algoritmičticky myslet a vytvářet jednoduché počítačové programy.

Cíle modulu

Modul 16 **Informatické myšlení a programování** - vyžaduje po uchazeči chápat podstatu programování a algoritmické logiky, znát související pojmy a umět používat běžné editory programového kódu a nástroje využívající technologie umělé inteligence pro vytváření jednoduchých programů.

Úspěšný absolvent zkoušky z tohoto modulu by měl být schopen:

- Chápat základní pojmy týkající se informatického myšlení a znát typické činnosti spojené s algoritmizací a programováním.
- Znat a umět používat základní techniky informatického myšlení a umět navrhnout jednoduché algoritmy.
- Umět zapisovat, upravovat a testovat algoritmy pomocí vývojových diagramů a pseudokódu.
- Znat hlavní principy a pojmy spojené se zapisováním programového kódu a uvědomovat si důležitost dobře
- Umět používat základní programové konstrukce.
- Umět zvyšovat efektivitu a funkčnost programu.
- Umět ladit a testovat programový kód a umět posoudit, zda program splňuje požadavky na jeho vydání (release).
- Umět používat nástroje využívající technologie umělé inteligence pro vytváření a úpravy programového kódu.

KATEGORIE	OBLAST ZNALOSTÍ	ODKAZ	ROZSAH ZNALOSTI
16.1 Základní pojmy	16.1.1 <i>Koncepce</i>	16.1.1.1	Rozumět pojmu computing.
		16.1.1.2	Rozumět pojmu informatické myšlení.
		16.1.1.3	Rozumět pojmu počítačový program.
		16.1.1.4	Rozumět pojmu programový kód a chápat rozdíl mezi pojmy zdrojový kód a strojový kód.
		16.1.1.5	Rozumět pojům popis účelu programu a specifikace funkcí programu.
		16.1.1.6	Znat typické fáze procesu programování, jako jsou analýza, návrh algoritmu, kódování, testování a údržba (optimalizace, vylepšování).
		16.1.1.7	Chápat rozdíl mezi formálním a přirozeným jazykem.
16.2 Metody informatického myšlení	16.2.1 <i>Analýza problému</i>	16.2.1.1	Znat typické metody informatického myšlení, např. dekompozice, rozpoznávání vzorů, abstrakce, algoritmizace.
		16.2.1.2	Umět rozkládat (dekomponovat) data, procesy a komplexní problémy na dílčí části.
		16.2.1.3	Umět rozpoznat společné znaky dílčích částí.
		16.2.1.4	Umět abstrahovat (odhlédnout) od nepodstatných detailů při analýze problému.
		16.2.1.5	Rozumět tomu, jak se v informatickém myšlení používají algoritmy.
	16.2.2 <i>Algoritmy</i>	16.2.2.1	Rozumět pojmu posloupnost příkazů při psaní programu a chápat smysl posloupnosti příkazů při návrhu algoritmů.
		16.2.2.2	Vědět, jakým způsobem se dají vyjádřit algoritmy, např. pomocí vývojových diagramů a pseudokódu.
		16.2.2.3	Znat základní symboly používané ve vývojových diagramech, jako jsou mezní značky začátku a konce, zpracování, rozhodování, vstup a výstup dat, spojnice
		16.2.2.4	Umět popsat posloupnost operací zapsanou pomocí vývojového diagramu nebo pseudokódu.
		16.2.2.5	Umět zapsat konkrétní algoritmus na základě vývojového diagramu nebo pseudokódu.

KATEGORIE	OBLAST ZNALOSTÍ	ODKAZ	ROZSAH ZNALOSTI						
16.3	Základy programování	16.3.1	Zápis kódu	16.2.2.6	Umět opravit chyby v algoritmu, jako jsou chybějící příkaz v programu, nesprávné pořadí operací, chyba v logice rozhodování.				
				16.3.1.1	Znát základní pravidla pro psaní kódu, např. odsazování, komentování, používání vhodných a popisných názvů.				
				16.3.1.2	Umět používat operátory pro základní aritmetické operace, tj. pro sčítání, odčítání, násobení a dělení.				
				16.3.1.3	Znát priority jednotlivých aritmetických operátorů a pořadí jejich vyhodnocování. Umět používat závorky pro strukturování složitějších výrazů.				
				16.3.1.4	Rozumět pojmu parametr (funkce nebo procedury) a vědět, proč se v programovém kódu používá.				
				16.3.1.5	Rozumět pojmu komentář a vědět, proč se v kódu používá.				
				16.3.1.6	Umět používat komentáře.				
			16.3.2	Proměnné a datové typy	16.3.2.1	Rozumět pojmu proměnná a vědět, proč se používá.			
					16.3.2.2	Umět deklarovat a inicializovat proměnnou.			
					16.3.2.3	Umět přiřadit proměnné hodnotu.			
					16.3.2.4	Umět vhodně pojmenovat proměnné.			
					16.3.2.5	Umět použít základní datové typy, např. textový řetězec, znak, celé číslo, reálné číslo, logická hodnota.			
					16.3.2.6	Umět používat složené datové typy, jakou jsou pole, seznamy, n-tice.			
					16.3.2.7	Umět přijmout vstupní data zadaná uživatelem.			
					16.3.2.8	Umět používat výstup na obrazovku.			
		16.4			Vytváření kódu	16.4.1	Logické výrazy	16.4.1.1	Rozumět pojmu logický výraz a vědět, proč se při programování používá.
								16.4.1.2	Znát základní relační a logické operátory, jako jsou rovno, nerovno, větší než, menší než, logický součin (AND), logický součet (OR), negace (NOT).
16.4.1.3	Umět sestavit a použít logické výrazy.								
	16.4.2		Iterace	16.4.2.1		Rozumět pojmu cyklus (smýčka) a znát výhody používání cyklů.			
16.4.2.2				Znát základní druhy cyklů, tj. daný počet opakování (for), opakování po dobu splnění podmínky (while), opakování až do splnění podmínky (until).					
16.4.2.3				Umět použít základní druhy cyklů.					
16.4.2.4				Rozumět pojmu nekonečný cyklus (smýčka).					
			16.4.2.5	Rozumět pojmu rekurze.					
	16.4.3		Podmíněné příkazy	16.4.3.1		Rozumět pojmu podmíněný příkaz a vědět, proč se používá.			
16.4.3.2				Umět použít podmíněné příkazy typu IF...THEN...ELSE.					
	16.4.4		Podprogramy (procedury a funkce)	16.4.4.1		Rozumět pojmu procedura a vědět, proč se používá.			
16.4.4.2				Umět vytvořit a pojmenovat proceduru.					
16.4.4.3				Rozumět pojmu funkce a vědět, proč se používá.					
16.4.4.4		Umět vytvořit a pojmenovat funkci.							
	16.4.5	Události a příkazy	16.4.5.1	Rozumět pojmu událost a vědět, proč se používá.					
16.4.5.2			Umět zpracovat základní události, jako jsou kliknutí tlačítkem myši, stisknutí klávesy, kliknutí na tlačítko, časová událost.						

KATEGORIE	OBLAST ZNALOSTÍ	ODKAZ	ROZSAH ZNALOSTI		
		16.4.5.3	Umět používat běžné funkce z knihovny funkcí, např. matematické funkce, generování náhodných čísel, funkce pro datum a čas.		
16.5	Ladění, testování a vydání (release) programu	16.5.1	<i>Ladění a testování programu</i>	16.5.1.1	Chápat, že smyslem ladění a testování programu je odstraňování chyb.
				16.5.1.2	Znát základní druhy chyb, jako jsou logické chyby a syntaktické chyby.
				16.5.1.3	Umět spustit program.
		16.5.1.4	Umět rozpoznat a opravit syntaktické chyby, např. překlepy, chybějící oddělovače.		
		16.5.1.5	Umět rozpoznat a opravit logické chyby, např. chybné logické výrazy, nesprávný datový typ.		
		16.5.2	<i>Vydání (release) programu</i>	16.5.2.1	Umět vyhodnotit, zda byly splněny požadavky na funkce programu.
		16.5.2.2	Umět popsat funkce programu, účel a význam programu.		
		16.5.2.3	Umět rozpoznat možná vylepšení programu.		
16.6	Vytváření kódu pomocí nástrojů využívajících technologie umělé inteligence (AI)	16.6.1	<i>Nástroje využívající AI</i>	16.6.1.1	Znát nejběžnější nástroje využívající AI pro generování a úpravy programového kódu.
				16.6.1.2	Rozumět technologickým omezením nástrojů využívajících AI v oblasti práce s programovým kódem a uvědomovat si potřebu kritického posuzování výsledků.
		16.6.2	<i>Použití nástrojů využívajících AI</i>	16.6.2.1	Používat nástroje využívající AI pro psaní a dokumentování programového kódu v různých programovacích jazycích na základě zadaných požadavků.
				16.6.2.2	Používat nástroje využívající AI pro analýzu, úpravy a optimalizaci programového kódu.
				16.6.2.3	Používat nástroje využívající AI pro konverzi programového kódu do jiného programovacího jazyka.