

Vzdělávací oblast: Informatika a informační a komunikační technologie

Vzdělávací obor: Programování

Předmět: Programování

Charakteristika vyučovacího předmětu

Cílem předmětu je poskytnout žákům ucelený kurz programování. Žáci se naučí všechny klíčové konstrukce a datové struktury poskytované programovacím jazykem. Výuka je postavena především na řešení praktických úloh, jejichž náročnost se stupňuje. Žáci jsou vedeni k psaní dobře strukturovaného a čitelného kódu, který odpovídá užívaným konvencím. Od třetího ročníku jsou zařazena témata moderních technologií programování na základě znalostí a zájmu žáků.

Obsahové vymezení

Žáci jsou seznámeni s následujícími tématy:

- principy programování a programovacích jazyků
- základní a pokročilé programové struktury
- základní a pokročilé datové struktury
- práce s konzolí, soubory a datovými sítěmi
- objektově orientované programování
- základní knihovny
- výjimky
- dobré zdrojové kódy
- implementace algoritmů a datových struktur
- grafické uživatelské rozhraní
- počítačová grafika
- vícevláknové programování
- databáze
- návrhové vzory
- pokročilé technologie*
- správa paměti*
- základní kurz nižšího programovacího jazyka (např. C)*
- programování malých a mobilních zařízení*
- softwarové inženýrství*

* takto označená témata jsou zařazena na základě znalostí a zájmu studentů

Časové a organizační vymezení

Předmět je zařazen do prvního až čtvrtého ročníku čtyřletého studia programátorské třídy s dotací dvě hodiny týdně. Výuka probíhá ve skupinách s maximálním počtem studentů 16.

Pokud je to možné vzhledem k rozvrhu, skupiny jsou rozděleny podle znalostí a dovedností studentů. U žáků s pokročilejšími znalostmi je výklad látky podrobnější, nebo jsou přidána další témata.

Součástí předmětu je pětidenní odborný kurz ve druhém ročníku.

Ve druhém ročníku výuka částečně navazuje na učivo probírané v souběžném předmětu Algoritmy.

Žáci v prvním a druhém ročníku povinně vypracovávají individuální ročníkový projekt, ve třetím ročníku týmový ročníkový projekt. Každý projekt je završen jeho obhajobou před komisí a publikem. Ve čtvrtém ročníku žáci vypracovávají ročníkový projekt, který je závěrečnou maturitní prací pro ty, kteří se rozhodnou maturovat v profilové části z programování.

Po celou dobu výuky se používá jeden vyšší programovací jazyk – po domluvě s vyučujícím mohou žáci úlohy řešit i v jiném programovacím jazyce.

Výchovné a vzdělávací strategie

Kompetence k učení

- učitel poskytuje žákům vhodnou názornou formou požadované informace a ověřuje schopnost žáka jejich použití v praxi
- učitel vede žáky ke studiu odborné (většinou cizojazyčné) literatury a dalších zdrojů informací a ukazuje, jak je při řešení úloh využívat
- učitel podněcuje myšlení žáků jdoucí za povrchní pohled na skutečnosti
- učitel nabádá žáky k zodpovědnému přístupu k předmětu a pravidelnému řešení úloh

Kompetence k řešení problémů

- učitel vede žáky k systematickému a logickému postupu při vyhodnocení zadání a řešení úloh
- učitel vede žáky k tvořivé práci a hledání originálních řešení
- učitel vede žáky k hledání různých postupů a výběru optimálního způsobu řešení
- učitel na příkladech seznamuje žáky s častými problémy a diskutuje s žáky možnosti jejich řešení

Kompetence komunikativní

- učitel vyžaduje od žáků stručné, přesné, jasné a logické vyjádření myšlenek (v písemném i mluveném projevu)
- učitel vyžaduje používání správných odborných termínů
- učitel vede žáky ke srozumitelné a efektivní prezentaci výsledků své práce

Kompetence sociální a personální

- učitel vede žáky ke sdílení poznatků, spolupráci a pomoci spolužákům
- učitel některé vhodné úlohy koncipuje tak, aby si žáci vyzkoušeli práci v menším týmu
- učitel vede žáky k respektování a toleranci názorů ostatních

Kompetence občanské

- učitel ukazuje žákům možnosti využití získaných kompetencí ve společnosti a jejich budoucím životě
- učitel vede žáky k užívání informačních technologií a získaných kompetencí v souladu s etickými, bezpečnostními a legislativními požadavky

Kompetence k podnikavosti

- učitel vede žáky k hledání a realizaci vlastních témat a zkoumání jejich možného uplatnění na trhu,
- žáci jsou pobízeni k účasti v předmětových olympiádách a soutěžích, které umožňují srovnání v konkurenci svých vrstevníků nejen v rámci školy

Zabezpečení výuky žáků se speciálními potřebami, zabezpečení výuky nadaných žáků

- učitel se jednotlivě věnuje jak dobrým, tak i slabším žákům
- učitel připravuje různě náročné úlohy
- učitel při výkladu využívá dílčí znalosti lepších žáků
- učitel volí vhodné tempo výkladu a procvičování
- učitel volí náročnost požadavků tak, aby neodradil slabší a začínající programátory a vzbudil u nich zájem o programování

Náplň předmětu

Výstupy ŠVP	Učivo - téma	Konkretizace	Průřezová témata, souvislosti, metody
1. ročník			
<ul style="list-style-type: none"> rozumí způsobům tvorby počítačového programu, principům jeho vykonávání 	Principy programování a programovacích jazyků	<ul style="list-style-type: none"> funkce interpretu a překladače, řízený kód přehled programovacích jazyků vývojové prostředí, práce s projekty, spuštění programu ladění programů, krokování 	
	Základní programové struktury	<ul style="list-style-type: none"> statická třída, hlavní metoda příkaz, blok příkazů podmíněný příkaz, cyklus statické metody, jejich parametry a návratové hodnoty, přetěžování metod 	
	Základní datové struktury	<ul style="list-style-type: none"> jednoduché datové typy (čísla, znaky, pravdivostní typ) proměnná, identifikátor, deklarace, rozsah platnosti konstanty, jejich zápis a použití převody mezi datovými typy výrazy, matematické, relační a logické operátory, použití metod ve výrazu, řetězce pole výčty 	
	Práce s konzolí a soubory	<ul style="list-style-type: none"> parametry příkazové řádky vstup a výstup do konzole načítání a ukládání textových souborů kódování textu 	
	Základní knihovny	<ul style="list-style-type: none"> náhodná čísla kolekce - seznam, množina, mapa použití generických datových typů třídy pro práci s poli a kolekcemi matematické funkce 	

Výstupy ŠVP	Učivo - téma	Konkretizace	Průřezová témata, souvislosti, metody
	Objektově orientované programování	<ul style="list-style-type: none"> třída a objekt vlastnosti a metody objektu, konstruktor statické vlastnosti a metody třídy práce s referencemi, existence objektů, garbage collector dědičnost 	
	Základní práce s datovými strukturami	<ul style="list-style-type: none"> součet, průměr, hledání minima a maxima nad polem a kolekcí permutace pole a kolekce načítání dat ze souboru 	
	Dobré zdrojové kódy	<ul style="list-style-type: none"> volba názvů konvence spojené s jazykem formátování zdrojového kódu modifikátory přístupu 	
2. ročník			
	Základní knihovny	<ul style="list-style-type: none"> zásobník, fronta, spojový seznam práce s časem a kalendářem další třídy pro práci se soubory 	
	Objektově orientované programování	<ul style="list-style-type: none"> polymorfismus, rozhraní 	
	Výjimky	<ul style="list-style-type: none"> typy výjimek a jejich vlastnosti obsluha výjimky vlastní výjimka, vyvolání výjimky 	
	Implementace algoritmů a datových struktur	<ul style="list-style-type: none"> permutace základní číselné algoritmy (převody soustav, Hornerovo schéma, dělitelnost, Eukleidův algoritmus, prvočíselnost, prvočíselný rozklad, Eratosthenovo síto) lineární spojový seznam, zásobník, fronta, halda, strom, graf prohledávání do šířky a prohledávání do hloubky vyhledávání a řazení rekurzivní algoritmy základní grafové algoritmy vyhodnocení výrazů 	

Výstupy ŠVP	Učivo - téma	Konkretizace	Průřezová témata, souvislosti, metody
	Grafické uživatelské rozhraní	<ul style="list-style-type: none"> práce s oknem formulářové prvky, jejich umístění do okna základy práce s více vlákny událostmi řízené programování, události klávesnice a myši, události prvků okna 	
	Počítačová grafika (odborný kurz)	<ul style="list-style-type: none"> grafický výstup práce s obrázky, skládání scény animace základy 3D grafiky 	
3. ročník			
	Databáze	<ul style="list-style-type: none"> návrh relačních databází tabulky, indexy správa záznamů, import a export dat dotazovací jazyk relace, klíče, integrita dat, transakce využití v programovacím jazyce ORM mapování* objektové databáze* 	
	Vícevláknové programování	<ul style="list-style-type: none"> spuštění a kontrola vlákn synchronizace objektů, monitor správné použití vláken, deadlock 	
	Dobré zdrojové kódy	<ul style="list-style-type: none"> dokumentace a komentáře logování (protokolování) refactoring 	
	Softwarové inženýrství	<ul style="list-style-type: none"> metodiky vývoje – tradiční a agilní analýza a návrh aplikací, UML týmová práce, verzování a sdílení kódu 	

Výstupy ŠVP	Učivo - téma	Konkretizace	Průřezová témata, souvislosti, metody
	Pokročilé technologie*	<ul style="list-style-type: none"> • vytváření tříd s generickými typy • použití XML • programování síťových aplikací (protokol HTTP, socketová komunikace) • návrhové vzory • práce se zvukem a videem • regulární výrazy • anotace a reflexe • neprocedurální a funkcionální jazyky • skriptovací jazyky • serverové technologie 	
	Základní kurz nižšího programovacího jazyka (např. C)*	<ul style="list-style-type: none"> • kompilace do nativního kódu • rozdílné konstrukce • správa paměti (bez garbage collectoru) • pointerová aritmetika 	
	Programování malých a mobilních zařízení*	<ul style="list-style-type: none"> • přehled platforem • nástroje pro vývoj • jednoduchá robotika 	
4. ročník			
	Síťová komunikace*	<ul style="list-style-type: none"> • nízkourovňové programování síťové komunikace • vyšší vrstvy, referenční model • IP adresa, MAC adresa • stream, datagram, paket • topologie sítí, vícesměrové vysílání • privátní sítě, šifrování • proxy • jiné komunikační protokoly 	
	Návrhové vzory*	<ul style="list-style-type: none"> • NV chování • NV struktury • NV vytváření • NV konkurence 	

Výstupy ŠVP	Učivo - téma	Konkretizace	Průřezová témata, souvislosti, metody
	Pokročilé technologie*	<ul style="list-style-type: none"> • anotace, jejich využití, definování vlastních anotací • reflexe, inspekce • profilování, optimalizace programu 	
	Nižší jazyky*	<ul style="list-style-type: none"> • assembler, linker a strojový kód • správa paměti • přehled nižších jazyků 	
	Vyšší jazyky*	<ul style="list-style-type: none"> • přehled vyšších jazyků • rozdíly mezi jazyky • moderní trendy • funkcionální vs. procedurální programování 	
	Skriptovací jazyky*	<ul style="list-style-type: none"> • interpretovaný jazyk vs. kompilovaný • použití skriptovacích jazyků • tvorba skriptovacího jazyka 	

* takto označená témata jsou zařazena na základě znalostí a zájmu studentů