

Vzdělávací oblast: Informatika

Vzdělávací obor: Informatika

Předmět: Programování

Charakteristika vyučovacího předmětu

Cílem předmětu programování je poskytnout žákům tříd se zaměřením programování ucelený kurz programování. Žáci se naučí všechny klíčové konstrukce a datové struktury poskytované programovacím jazykem a algoritmizaci úloh. 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 systémové knihovny
- systémové a vlastní výjimky a jejich ošetření
- dobré zdrojové kódy
- algoritmy a implementace algoritmů a datových struktur
- grafické uživatelské rozhraní
- počítačová grafika
- vícevláknové programování
- databáze v programovacím jazyce
- objektově relační mapování
- 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. U žáků s pokročilejšími znalostmi je výklad látky podrobnější, nebo jsou přidána další témata.

Součástí předmětu je pětidenní odborný kurz ve druhém ročníku.

Žáci v prvním, druhém a čtvrtém ročníku 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 může být po doplnění závěrečnou maturitní prací pro ty, kteří se rozhodnou maturovat v profilové části z programování.

Výuka probíhá na počítačích nebo noteboocích buď v počítačové učebně, nebo v běžné učebně s přenosnými notebooky s připojením k internetu, výjimečně i mimo školní budovu.

U žáků prvního ročníku se nepředpokládají předchozí znalosti z programování nad rámec informatiky základní školy.

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

Kompetence digitální:

- žák je veden k efektivnímu využití digitálních zařízení, aplikací a služeb nejen ve školním prostředí
- žák je veden k tomu, aby rozuměl a vyvaroval se nebezpečí v digitálním prostředí
- žák je veden, aby v digitálním prostředí jednal eticky, s ohleduplností a respektem k druhým

Zabezpečení výuky žáků se speciálními potřebami, zabezpečení výuky nadaných žáků

- učitel se jednotlivě věnuje jak dobrým, tak i slabším žákům
- učitel připravuje různé 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

Výstupy ŠVP, žák	Učivo - téma	Konkretizace	Souvislosti, metody
1. ročník			
DATA, INFORMACE A MODELOVÁNÍ <ul style="list-style-type: none"> • rozlišuje a používá různé datové typy • navrhuje a porovnává různé způsoby kódování z různých hledisek a vysvětlí proces a úskalí digitalizace 	<ul style="list-style-type: none"> • kódování a přenos dat – kódování dat v počítačích obecně, binární soustava, bity a bajty; kódování čísel, vliv množství informace (počtu bitů) na možný rozsah a dostupnou přesnost; kódování textů; přenos dat, kódování a dekódování zprávy, komunikační kanál, kontrolní součty 	<ul style="list-style-type: none"> • bit, byte • LSB/MSB, BE/LE • celá čísla, znaménko, plovoucí čárka, pravdivostní typ, výčet • znak, pole, řetězec, soubor • ASCII, UNICODE, UTF • dvojková, osmičková, šestnáctková soustava • datum a čas, časová zóna, časové razítko • CRC 	<ul style="list-style-type: none"> • získávání, vyhledávání a ukládání dat obecně a v počítači • data a jejich význam, pojem informace
ALGORITMIZACE A PROGRAMOVÁNÍ <ul style="list-style-type: none"> • vysvětlí daný algoritmus, program • vytvoří přehledný program pro vyřešení konkrétního problému s ohledem na jeho možné důsledky a svou odpovědnost za ně • používá opakování, větvení programu se složenými podmínkami, proměnné, seznamy, podprogramy s parametry a návratovými hodnotami 	<ul style="list-style-type: none"> • algoritmizace – zadání úlohy, vstup, výstup, podmínky řešení; pojem algoritmus, vlastnosti algoritmu, přirozené a formální jazyky, různé zápisy algoritmů • programovací koncepty – programovací jazyk; proměnné, datové typy a jejich vlastnosti, vstup a výstup dat; podprogramy s parametry a s 	<ul style="list-style-type: none"> • vývojový diagram • podmíněně vykonání příkazů, okrajová podmínka, cyklus • rekurze • zpracování vstupních a formátování výstupních dat • proměnná, konstanta, identifikátor • matematické funkce 	<ul style="list-style-type: none"> • model jako zjednodušení reality • schéma, diagram • pojmová a myšlenková mapa

	návratovými hodnotami; větvení programu se složenými podmínkami, cykly, seznamy	<ul style="list-style-type: none"> • operátor, operand • náhoda • příkaz, blok příkazů • třída, objekt, struktura, hlavní metoda • reference, ukazatel, null • generický datový typ • vývojové prostředí, překladač a interpret, projekt, spuštění programu • veřejné rozhraní, abstrakce, zapouzdření • imperativní a funkcionální jazyk 	
<ul style="list-style-type: none"> • rozumí způsobům tvorby počítačového programu, principům jeho vykonávání 	<ul style="list-style-type: none"> • základní programové struktury • principy programování a programovacích jazyků • základní programové struktury • základní datové struktury • práce s konzolí a soubory • základní knihovny • objektově orientované programování • základní práce s datovými strukturami • dobré zdrojové kódy 	<ul style="list-style-type: none"> • přehled programovacích jazyků • statické a instanční vlastnosti a metody, jejich parametry a návratové hodnoty, přetěžování metod, konstruktor a destruktor • deklarace, rozsah platnosti • převody mezi datovými typy • výrazy, matematické, relační a logické, použití metod ve výrazu, porovnání objektů • parametry příkazové řádky • vstup a výstup do konzole • načítání a ukládání textových souborů a obecných dat • základní kolekce - seznam, množina, mapa • třídy pro práci s poli a kolekcemi • garbage collector • dědičnost • konvence spojené s jazykem a pojmenováním identifikátorů • formátování zdrojového kódu • modifikátory přístupu 	<ul style="list-style-type: none"> • softwarové inženýrství • data • objektově orientované programování
<ul style="list-style-type: none"> • vytvoří přehledný program pro vyřešení konkrétního problému s ohledem na jeho možné důsledky a svou odpovědnost za ně • vysvětlí proces a úskalí digitalizace 	<ul style="list-style-type: none"> • tvorba uceleného programu na zvolené téma 	<ul style="list-style-type: none"> • individuální ročníkový projekt • technická dokumentace • vědecký poster • obhajoba projektu 	<ul style="list-style-type: none"> • projektová výuka • prezentace díla • práce s odborným textem • tabulková a grafická prezentace dat • rétorika
2. ročník			

<p>DATA, INFORMACE A MODELOVÁNÍ</p> <ul style="list-style-type: none"> rolišuje a používá různé datové typy navrhne a porovná různé způsoby kódování z různých hledisek 	<ul style="list-style-type: none"> modelování – model jako zjednodušení reality, schéma, diagram, pojmová a myšlenková mapa; graf, vrcholy, hrany, orientovaný graf, ohodnocený graf, kritická cesta 	<ul style="list-style-type: none"> schéma grafů implementace pomocí primitivních datových typů a polí implementace pomocí objektového programování matice sousednosti 	<ul style="list-style-type: none"> modelování pomocí grafu schéma, diagram
<p>ALGORITMIZACE A PROGRAMOVÁNÍ</p> <ul style="list-style-type: none"> určí, zda je daný postup algoritmem analyzuje problém, rozdělí problém na menší části, rozhodne, které je vhodné řešit algoritmicky, své rozhodnutí zdůvodní sestaví a zapíše algoritmy pro řešení problému ve vztahu k charakteru a velikosti vstupu hodnotí nároky algoritmů porovná algoritmy podle různých hledisek, vybere pro řešení problému ten nejvhodnější vylepší algoritmus podle zvoleného hlediska zobecní řešení pro širší třídu problémů ve snaze o vyšší efektivitu navrhuje, řídí a hodnotí souběh procesů ověří správnost, najde a opraví případnou chybu v algoritmu, otestuje, odladí a optimalizuje program 	<ul style="list-style-type: none"> algoritmizace – rozdělení problému na části, identifikace návazností dat, opakujících se vzorů a míst pro rozhodování testování, optimalizace – syntaktické, běhové a logické (funkční) chyby, krokování a ladění programu; vliv vstupních dat na spotřebované výpočetní zdroje vývoj programu – návrh přehledného uživatelského rozhraní programu 	<ul style="list-style-type: none"> algoritmus hašovací funkce soubor, proud, buffer, kruhový buffer výjimka událostmi řízené programování, události klávesnice a myši, události grafických prvků GUI vlákno pixel, RGB paralelizace úloh rozděl a panuj lineární spojový seznam, zásobník, fronta, halda, strom, graf paměťová a časová náročnost složitost algoritmů řadící algoritmy základní grafové algoritmy prohledávání do šířky a prohledávání do hloubky vyhodnocení výrazů permutace 	<ul style="list-style-type: none"> programovací koncepty počítačové jazyky počítačová grafika optika, barva vidění matematické abstrakce
<ul style="list-style-type: none"> rozumí způsobům tvorby počítačového programu, principům jeho vykonávání 	<ul style="list-style-type: none"> základní knihovny objektově orientované programování výjimky implementace algoritmů a datových struktur grafické uživatelské rozhraní počítačová grafika (odborný kurz) 	<ul style="list-style-type: none"> další třídy pro práci se soubory polymorfismus, rozhraní typy výjimek a jejich vlastnosti obsluha výjimek vlastní výjimka, vyvolání výjimek základní číselné algoritmy převody číselných soustav vyhledávání a řazení rekurzivní algoritmy práce s oknem, formulářové prvky, 	<ul style="list-style-type: none"> Hornerovo schéma dělitelnost Eukleidův algoritmus prvočíselnost prvočíselný rozklad Eratosthenovo síto

		jejich umístění do okna <ul style="list-style-type: none"> • základy práce s více vlákny • grafický výstup • práce s obrázky, skládání scény, animace • základy 3D grafiky 	
<ul style="list-style-type: none"> • vytvoří přehledný program pro vyřešení konkrétního problému s ohledem na jeho možné důsledky a svou odpovědnost za ně • ověří správnost, najde a opraví případnou chybu v algoritmu, otestuje, odladí a optimalizuje program 	<ul style="list-style-type: none"> • tvorba uceleného programu na zvolené téma 	<ul style="list-style-type: none"> • individuální ročníkový projekt • technická dokumentace • vědecký poster • veřejná obhajoba projektu 	<ul style="list-style-type: none"> • projektová výuka • prezentace díla • práce s odborným textem • tabulková a grafická prezentace dat • rétorika
3. ročník			
ALGORITMIZACE A PROGRAMOVÁNÍ <ul style="list-style-type: none"> • analyzuje problém, rozdělí problém na menší části, rozhodne, které je vhodné řešit algoritmicky, své rozhodnutí zdůvodní • sestaví a zapíše algoritmy pro řešení problému • ve vztahu k charakteru a velikosti vstupu hodnotí nároky algoritmů 	<ul style="list-style-type: none"> • vývoj programu – volba nástroje podle zadání úlohy; nápověda a dokumentace k programu; autorství a licence programu; etika programátora 	<ul style="list-style-type: none"> • jednotkové testy • programování řízené testy • inspekce dat, debugging • logování, profilování • dokumentace a komentáře • šifrování • CPU, jádro, logická jednotka, reálné a virtuální vlákno • automatizace vývoje • data binding • relační databáze • datový model, entity • serializace 	<ul style="list-style-type: none"> • softwarové inženýrství • počítačový hardware • databázové systémy
<ul style="list-style-type: none"> • rozumí způsobům tvorby počítačového programu, principům jeho vykonávání 	<ul style="list-style-type: none"> • vícevláknové programování • dobré zdrojové kódy • softwarové inženýrství • pokročilé technologie 	<ul style="list-style-type: none"> • spuštění a kontrola vlákn • synchronizace objektů, monitor • správné použití vláken, deadlock • refactoring • metodiky vývoje – tradiční a agilní • analýza a návrh aplikací, UML • týmová práce, verzování a sdílení kódu • vytváření tříd s generickými typy • použití XML, JSON • programování síťových aplikací (protokol HTTP, socketová komunikace) 	<ul style="list-style-type: none"> • softwarové inženýrství • diagramy • správa verzí obsahu • webové aplikace

		<ul style="list-style-type: none"> • regulární výrazy • anotace a reflexe, vlastní anotace • správa paměti • neprocedurální a funkcionální jazyky, lambda výrazy • serverové technologie 	
<ul style="list-style-type: none"> • vytvoří přehledný program pro vyřešení konkrétního problému s ohledem na jeho možné důsledky a svou odpovědnost za ně • ve snaze o vyšší efektivitu navrhuje, řídí a hodnotí souběh procesů • vysvětlí proces a úskalí digitalizace 	<ul style="list-style-type: none"> • tvorba uceleného programu na zvolené téma • týmová práce • sdílení zdrojového kódu 	<ul style="list-style-type: none"> • týmový ročníkový projekt • technická kolektivní dokumentace • vědecký poster • společná obhajoba projektu 	<ul style="list-style-type: none"> • projektová výuka • prezentace díla • práce s odborným textem • tabulková a grafická prezentace dat • rétorika • týmová spolupráce
4. ročník			
<p>DIGITÁLNÍ TECHNOLOGIE</p> <ul style="list-style-type: none"> • identifikuje a řeší problémy a výzvy vznikající při práci s digitálními zařízeními a poradí s nimi druhým • chrání digitální zařízení, digitální obsah i osobní údaje před poškozením či zneužitím s vědomím změn v technologiích, které ovlivňují bezpečnost 	<ul style="list-style-type: none"> • počítačové sítě – lokální počítačové sítě a internet; paketový přenos dat, firewall; zabezpečený přenos dat • bezpečnost počítačových zařízení a dat – způsoby útoků na počítačová zařízení 	<ul style="list-style-type: none"> • socket, client-server, TCP/UDP, SSL/TLS, ISO/OSI • MITM, packet sniffer, DoS, DDoS • datagram • IP adresa • IPv4 a IPv6 • port • komunikační protokol • standardní vstup, standardní výstup, chybový výstup • ověřování uživatelů • online služby • aplikační servery • regulární výrazy • online zdroje • XML • data mining • vzdálený přístup k počítači • příkazový řádek • operační systém • trendy v programování 	<ul style="list-style-type: none"> • webové aplikace • operační systémy • internet a komunikace • bezpečnost na internetu a v počítačových sítích • digitální stopa
<ul style="list-style-type: none"> • rozumí způsobům tvorby počítačového programu, principům jeho vykonávání 	<ul style="list-style-type: none"> • síťová komunikace • sledování síťového provozu • návrhové vzory • pokročilé technologie 	<ul style="list-style-type: none"> • nízkourovňové programování síťové komunikace • vyšší vrstvy, referenční model • IP adresa, MAC adresa 	<ul style="list-style-type: none"> • softwarové inženýrství • návrhové vzory • síťová komunikace

		<ul style="list-style-type: none"> • stream, datagram, paket • topologie sítí, vícesměrové vysílání • privátní sítě, šifrování • proxy • komunikační protokoly • NV vytváření • NV struktury • NV chování • NV konkurence • funkcionální vs. procedurální programování 	
<ul style="list-style-type: none"> • vytvoří přehledný program pro vyřešení konkrétního problému s ohledem na jeho možné důsledky a svou odpovědnost za ně • identifikuje a řeší problémy a výzvy vznikající při práci s digitálními zařízeními a poradí s nimi druhým • vysvětlí proces a úskalí digitalizace 	<ul style="list-style-type: none"> • tvorba uceleného programu na zvolené téma • vývoj programu – volba nástroje podle zadání úlohy • náповěda a dokumentace k programu • autorství a licence programu • etika programátora 	<ul style="list-style-type: none"> • individuální ročníkový projekt • individuální maturitní práce (jen maturantů z programování) • technická dokumentace • vědecký poster • obhajoba projektu 	<ul style="list-style-type: none"> • projektová výuka • prezentace díla • práce s odborným textem • tabulková a grafická prezentace dat • rétorika