Název vyučovaného předmětu: Programování a vývoj aplikací ve Windows
Zkratka předmětu: PVA
Obor vzdělání: 18-20-M/01 Informační technologie
Forma vzdělání: denní
Celkový počet vyučovacích hodin: 332 hodin (10 hodin týdně)

|  |  |
| --- | --- |
| **Ročník studia** | **Časová dotace (hod./ týden)** |
| 1. | 1 |
| 2. | 3 |
| 3. | 2 |
| 4. | 3 |

## **Pojetí vyučovacího předmětu**

### Obecné cíle

Cílem předmětu je seznámit žáka s technikou programování v různých programovacích prostředích a naučit ho vyvíjet vlastní aplikace a představit mu aplikace do běžných problémů. Důraz je kladen především na výuku praktické schopnosti programovat. Žák také bude připraven analyzovat a navrhovat IT řešení jednotlivých problémů. Žák se naučí základy některého z vyšších programovacích jazyků a bude schopen používat algoritmy k řešení problémů. Žák bude schopen vytvořit algoritmus pro klasickou kalkulačku a tento algoritmus převést do kódu v některém z vyšších programovacích jazyků. Žák bude rozumět termínům I/O, datový typ, komentář, podmínka, cyklus, pole a bude je umět naimplementovat v některém z vyšších programovacích jazyků. Žák bude rozumět termínu operační systém a bude v něm umět efektivně pracovat. Žák bude umět pracovat se souborovými složkami v příkazovém řádku. Žák bude rozumět tomu, jak jsou zdrojové soubory pro jednotlivé programovací jazyky uloženy v počítači. Žák se naučí pracovat v objektově orientovaném programovacím jazyku a bude znát výhody tohoto přístupu. Žák dokáže vytvořit desktopovou aplikaci s grafickým rozhraním. Žák bude znát základní algoritmy běžně používané při programování. Žák získá přesah programování do dalších předmětů a dostane návod, jak je možné programovat v Excelu, s připojenou databází a vyvíjet síťové aplikace.

### Charakteristika předmětu

Předmět PVA připravuje žáky k tomu, aby byli schopni navrhovat aplikace podle nejnovějších trendů a pravidel, učí je strukturálně přemýšlet a řešit problémy, v konečném součtu dává ucelený a souvislý přehled základních technik a teorií, které vedou k efektivnímu vývoji desktopových aplikací. Dovednosti a znalosti z tohoto předmětu mohou studenti uplatnit jako analytici, programátoři či vývojáři aplikací. Význam budou mít i v dalším studiu i běžném životě.

### Pojetí výuky, metody a formy práce

Výuka předmětu PVA je pojata popularizačně. Hlavní metodou výuky je výklad složený především ze zobecnění problémů a jejich řešení, aby se žákům učivo snadněji chápalo. Dále je využíván přesný popis postupu, který učitel ukazuje na projektoru v počítačové učebně a žáci ho aktivně sledují a snaží se sami přicházet na řešení praktických úloh a rozvíjet své dovednosti. Úvodním vstupem do předmětu bude motivační téma: konkretizace problémů a návrhy způsobů jejich řešení. Při jejich řešení by měl každý žák individuálně přijít na to, že k vyřešení problému potřebuje nějaký aparát, v tomto případě programovací prostředí a programovací jazyk. Poté by si měl tento aparát budovat na základní a následně pokročilé úrovni. Výuka základního programového vybavení bude tento způsob vhodně doplňovat a bude realizovaná buď teoretickým výkladem, nebo formou prezentací jednotlivých studentů.

### Hodnocení výsledků žáků

Hodnocení práce žáků je složeno minimálně z 50 % hodnocením kvality jejich individuálních prací a projektů. Dále se budou hodnotit teoretické znalosti pomocí písemných testů. Učitel nastaví objektivní kritéria pro hodnocení. Jasně stanovené váhové podmínky a plán odevzdávání prací a testů by měl žákům poskytnout na začátku každého pololetí.

### Přesah do dalších předmětů

* Matematika
využívání číselných datových typů v programování, algoritmy postavené na matematice
* Anglický jazyk

programovací jazyky jsou v angličtině
veškerá oficiální podpora programovacích jazyků a nejlepší zdroje informací vychází z angličtiny

* Aplikace na PC

práce s databází v programovacím prostředí, Visual Basic a jeho využití v Excelu

* Hardware a sítě

tvorba síťových programů, využívané protokoly, programování na nejnižší úrovni

* Webdesign

Základy programování a algoritmizace pro vývoj webových aplikací

* Počítačová grafika

Grafické rozhraní desktopových aplikací, programování počítačové grafiky

## **Realizace odborných kompetencí**

Programování a vývoj aplikací ve Windows – 1. ročník (1 hodina týdně)
Koncipováno jako základní rychlokurz, většinou na každou hodinu 1 téma včetně praktické části.

|  |  |
| --- | --- |
| **Výsledky vzdělávání – Žák:** | **Tématické celky** |
| * rozumí pojmu problém
* umí určit třídy problému, sestavit jejich diagram
* umí navrhnout řešení problému pomocí algoritmu
* umí zanalyzovat jednotlivé aktéry a přiřadit mu činnosti
* rozumí významu architektury MVC pro řešení problémů
* umí sestavit sekvenční diagram
* chápe, co je funkce a je motivován k vytvoření aparátu pro její implementaci

  | 1. Obecný úvod do modelování dat*úvodní motivace k programování*
* existence problému a jeho řešení
* diagram tříd
* vývojový diagram
* případy užití se scénáři
* architektura MVC
* sekvenční diagram
 |
| * zná datové typy a jejich funkci
* umí přidat uživatelský vstup do proměnné, vypsat výstup z proměnné
* umí používat debugger a komentáře
* chápe a umí použít podmínku
* umí sestrojit jednoduchou kalkulačku
* chápe a umí použít různé typy cyklů
* umí sestrojit program vypisující prvočísla
* chápe, co je to pole a k čemu slouží
* ví, kde jsou v počítači uloženy zdrojové soubory
 | 1. Základy programování

*budování základního aparátu pro řešení problému* * datové typy
* I/O, debugger, komentáře
* zdrojové soubory, zápis do textového souboru, čtení z textového souboru
* podmínka
* cyklus
* pole
 |
| * chápe princip objektového kódu
* chápe princip třídy jako datového typu
 | 1. Úvod do objektově orientovaného programování

*kód může být i konkrétní, návrat k úvodu** třída jako datový typ
 |
| * nainstaluje operační systém
* nakonfiguruje operační systém pro použití periferních zařízení
* nastaví účty uživatelů a skupin a jejich oprávnění
* připojí a nakonfiguruje počítač v rámci počítačové sítě
* připojí počítač k internetu
* zálohuje OS a data
* zabezpečí počítače proti zneužití
* ochrání data před zničením
* orientuje se v používaných OS a umí zvolit vhodný OS s ohledem na jeho nasazení
* umí používat základní příkazy v příkazové řádce
 | 1. Základní programové vybavení

*aby se vůbec aplikace mohli tvořit** instalace, konfigurace a správa operačního systému
* operační systém
* práce v příkazové řádce
 |

Programování a vývoj aplikací ve Windows – 2. ročník (3 hodiny týdně)
Nejvíce informací v tomto předmětu studenti získají v tomto ročníku.

|  |  |
| --- | --- |
| **Výsledky vzdělávání – Žák:** | **Tématické celky** |
| * zanalyzuje úlohu a sestaví algoritmus
* zapíše algoritmus vhodným způsobem
* umí správně používat podmínky a cykly
* chápe, co je to pole a k čemu slouží
* zná rozdíl mezi procedurou a funkcí
* umí vytvořit aplikaci ve vyšším programovacím jazyku
 | 1. Základy programování 2*opakování potřebného aparátu a přidání dalších funkcí*
* algoritmizace
* podmínka, cyklus, debugger
* pole
* procedury a funkce
 |
| * umí měnit datový typ proměnné
* umí vybrat jakoukoliv část znaků z řetězce a čísla
* umí pracovat s třídou jako datovým typem
* rozumí systému zdrojových souborů v počítači a ví, co obsahují jednotlivé soubory
* umí používat pole s předem nedefinovanou délkou
* umí navrhnout desktopovou aplikaci s grafickým rozhraním
* umí obsluhovat objekty grafického rozhraní
 | 1. Objektově orientované programování

*přecházíme na vyšší level** práce s datovými typy
* implementace tříd
* třída jako datový typ
* arrayList
* zdrojové soubory
* tvorba procedur a funkcí náležející třídám
* grafické rozhraní desktopových aplikací
* události
 |
| * umí alespoň jeden základní třídící algoritmus
* umí naimplementovat binární strom pomocí pole
* zná princip symetrické a asymetrické šifry
 | 1. Základní algoritmy

*matematické algoritmy pomáhající programátorům** druhy třídění
* vyhledávací algoritmy
* základy kryptografie
 |
| * analyzuje požadavky
* umí vytvořit dokumentaci k projektu
* pro vybrané téma zpracuje analýzu
 | 1. Analýza požadavků a návrh aplikace

*pokud někdo přijde, že chce IT systém, tohle mu dáte jako první výstup, zároveň si tím hodně pomůžete při implementaci** business požadavky
* diagram tříd
* systémové požadavky
* vývojové diagramy
* diagram případů užití se scénáři
* diagram komponent
* sekvenční diagram
 |

Programování a vývoj aplikací ve Windows – 3. ročník (2 hodiny týdně)
Celý třetí ročník je proložen vývojem vlastní aplikace s dokumentací.

|  |  |
| --- | --- |
| **Výsledky vzdělávání – Žák:** | **Tématické celky** |
| * je schopen naplnit spojový seznam
* je schopen naimplementovat bubble sort
 | 1. Programování s využitím algoritmů

*budeme opakovat a využijeme matematické algoritmy z minulého roku** podmínka, cyklus, pole
* bubble sort
* spojový seznam
 |
| * umí zapsat data do textového souboru a data z něj zapsat do proměnné
* umí připojit databázi k programovacímu prostředí a otevřít a zavřít spojení
 | 1. Data

*potřebuji někde něco zachovat při zavření** práce s textovým souborem
* práce s jednoduchou databází
 |
| * umí vytvořit dokumentaci k projektu na odpovídající úrovni
 | 1. Dokumentace

*začínáme si vytvářet něco svého** modelování dat
* vytváření šablon pro dokumentaci
 |
| * umí vytvořit desktopovou aplikaci s grafickým rozhraním na odpovídající úrovni
 | 1. Implementace projektu

*začínáme psát něco svého** zvolení vývojového prostředí
* čistý kód
* vlastní program
 |

Programování a vývoj aplikací ve Windows – 4. ročník (3 hodiny týdně)

|  |  |
| --- | --- |
| **Výsledky vzdělávání – Žák:** | **Tématické celky** |
| * umí bez problému naimplementovat pole a spojový seznam
* rozumí termínům procedura a funkce, zná mezi nimi rozdíl
* chápe princip rekurze
 | 1. Algoritmizace

*opakujeme a přidáváme další věci** pole, spojový seznam
* procedury a funkce
* rekurze
 |
| * rozumí základní koncepci objektově orientovaného programování i v teoretické rovině
* umí popsat pojmy dědičnost, zapouzdření, kompozice, polymorfismus
 | 1. Teoretické principy objektově orientovaného programování

*tak a teď tomu budeme více rozumět** dědičnost
* zapouzdření
* kompozice
* polymorfismus
 |
| * zná funkci a význam jednotlivých síťových služeb
* zaktivuje a nakonfiguruje jednotlivé síťové služby
* je schopen obecně popsat způsob, jakým je možné vytvořit síťovou aplikaci
 | 1. Síťové aplikace

*a nakonec to nahrajeme na web, nebo ještě lépe - uděláme multiplayer** server a jeho funkce
* DHCP, DNS, http, file server, print server, SQL server, SMTP server aj.
* konfigurace síťových rozhraní
* síťové aplikace

  |