# 2. Framework, Programovací jazyk

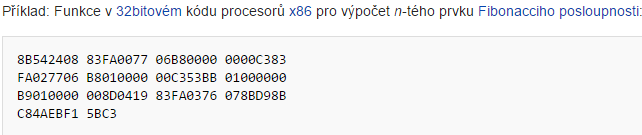
**Programovací jazyk** je prostředek pro zápis algoritmů, jež mohou být provedeny na počítači. Zápis algoritmu ve zvoleném programovacím jazyce se nazývá program.

Programovací jazyk je komunikačním nástrojem mezi [programátorem](https://cs.wikipedia.org/wiki/Program%C3%A1tor), který v programovacím jazyce formuluje postup řešení daného problému, a počítačem, který program interpretuje technickými prostředky. Programovací jazyk je vlastně soubor pravidel pro zápis algoritmu, odborně řečeno se jedná o [formální jazyk](https://cs.wikipedia.org/wiki/Form%C3%A1ln%C3%AD_jazyk).

# Základní rozdělení Programovacích jazyků:

* **Nižší programovací jazyky**

Nižší programovací jazyk může být snadno převeden do strojového kódu procesoru. Jsou typicky považovány za jednoduché, avšak za obtížněji osvojitelné, protože je pro jejich efektivní používání nutné seznámit se s technickými detaily fungování [hardware](https://cs.wikipedia.org/wiki/Hardware). Programy napsané v nižším programovacím jazyce mohou být rychlé a mít jen malé nároky na paměť. Ekvivalentní programy napsané ve [vyšším programovacím jazyce](https://cs.wikipedia.org/wiki/Vy%C5%A1%C5%A1%C3%AD_programovac%C3%AD_jazyk) jsou typicky pomalejší a mají větší paměťové nároky, avšak oddělují sémantiku [počítačové platformy](https://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%A1_platforma) od zápisu programu, což by mělo zjednodušit jeho vývoj.



Označení „nižší“ odkazuje na velmi malý nebo žádný rozdíl mezi daným programovacím jazykem a [strojovými instrukcemi](https://cs.wikipedia.org/wiki/Strojov%C3%A1_instrukce) procesoru (resp. jeho [instrukční sadou](https://cs.wikipedia.org/wiki/Instruk%C4%8Dn%C3%AD_sada)). Proto bývají nízkoúrovňové programovací jazyky označovány jako „těsně spjaté s [hardwarem](https://cs.wikipedia.org/wiki/Hardware)“.

Jejich instrukce (víceméně přesně) odpovídají příkazům procesoru. To znamená, že procesor bude vykonávat ty instrukce, které programátor napíše. Jsou závislé na svém procesoru a nepřenositelné na jiný procesor.

* **Vyšší programovací jazyky**

Vyšší programovací jazyky by měly být člověku lépe srozumitelné, než nižší programovací jazyky, čímž by měl být jednodušší vlastní vývoj programů. Programy zapsané ve vyšších jazycích jsou obvykle kratší a lépe čitelné, než zápis v nižších programovacích jazycích. Tím by měly šetřit čas programátora a zmenšit pravděpodobnost výskytu [programátorských chyb](https://cs.wikipedia.org/wiki/Program%C3%A1torsk%C3%A1_chyba). Ve vyšších programovacích jazycích je možné používat prvky přirozeného jazyka.

Struktura zdrojového kódu je u vyšších programovacích jazyků logická. Další výhodou vyšších programovacích jazyků je jejich přenositelnost. Programy po malých (někdy i žádných) úpravách mohou běžet na různých [počítačových platformách](https://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%A1_platforma).

Nevýhodou vyšších programovacích jazyků je fakt, že počítače umí přímo zpracovávat kód zapsaný v nejnižších programovacích jazycích (tzv. [Jazyk symbolických adres](https://cs.wikipedia.org/wiki/Jazyk_symbolick%C3%BDch_adres)). Proto musejí být programy zapsané ve vyšších programovacích jazycích [překládány](https://cs.wikipedia.org/wiki/P%C5%99eklada%C4%8D) překladačem (kompilátorem) do nižších jazyků.

## Rozdělení vyšších programovacích jazyků

[**Procedurální**](https://cs.wikipedia.org/wiki/Procedur%C3%A1ln%C3%AD_programov%C3%A1n%C3%AD)– popisuje výpočet pomocí posloupností příkazů a určuje přesný postup ([algoritmus](https://cs.wikipedia.org/wiki/Algoritmus)), jak danou úlohu řešit

* [Strukturované](https://cs.wikipedia.org/wiki/Strukturovan%C3%A9_programov%C3%A1n%C3%AD) – [algoritmus](https://cs.wikipedia.org/wiki/Algoritmus) se rozděluje na dílčí úlohy, které se spojují v jeden celek

**Objektově orientované** - je [programovací paradigma](https://cs.wikipedia.org/wiki/Programovac%C3%AD_paradigma), založené na myšlenkách Objektů, Abstrakce, Zapouzdření, Dědičnosti a Polymorfismu.

# Druhy kódů

## Zdrojový kód

Zdrojový kód je v informatice označení zápisu textu počítačového programu v některém programovacím jazyce, který je uložen v jednom nebo více textových souborech (tzv. projekty). Zdrojový kód je následně buď přímo prováděn (interpretován, viz interpret) nebo je z něj nejprve vytvořen samostatný spustitelný soubor (přeložen, viz kompilátor) složený ze strojových instrukcí a teprve ten je pak přímo spuštěn (prováděn procesorem počítače).

## Strojový kód

Strojový kód je v informatice posloupnost strojových instrukcí prováděných procesorem počítače, která je zapsána pomocí posloupnosti číselných kódů těchto strojových instrukcí. Hnutí GNU pojem strojový kód chápe obecnějším způsobem.

# Programovací jazyky

## C

Jazyk relativně nízké úrovně, původně zaměřený na programování operačních systémů. Má velmi úsporné vyjadřování, a proto je poněkud nepřehledný. Existuje velké množství rozšíření jazyka, nejznámější je objektově orientované C++. Syntaxi "céčka" převzala spousta dalších jazyků (PHP, Java, Perl,…).

## C#

(vyslovované anglicky jako C Sharp, /siː ʃɑɹp/, doslova to označuje notu cis) je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework. Microsoft založil C# na jazycích C++ a Java (a je tedy nepřímým potomkem jazyka [C](https://cs.wikipedia.org/wiki/C_(programovac%C3%AD_jazyk)), ze kterého čerpá syntaxi).

## PHP

PHP (Hypertext Preprocessor) je skriptovací programovací jazyk. Je určený především pro programování dynamických internetových stránek a webových aplikací například ve formátu HTML, XHTML či WML.

Internetový skriptovací jazyk (serverový) používaný k tvorbě webových stránek. Instrukce se vkládájí do HTML kódu, syntaxe je podobná jazyku C. Práce programu probíhá v serveru, na uživatelský počítač se posílají jen výsledky. Má ohromné možnosti: dokáže pracovat s databázemi, s velkým množstvím internetových protokolů (IMAP, POP3, SMTP - zpracování emailů; NNTP - news protokol; ...)

## ASP

ASP (Active Server Pages) je skriptovací platforma společnosti [Microsoft](https://cs.wikipedia.org/wiki/Microsoft), primárně určená pro dynamické zpracování [webových stránek](https://cs.wikipedia.org/wiki/Webov%C3%A1_str%C3%A1nka) na straně serveru. Její nástupce, [ASP.NET](https://cs.wikipedia.org/wiki/ASP.NET), lze chápat jako širší a komplexnější technologii, která se od ASP v mnoha ohledech fundamentálně liší.

Na správně nakonfigurovaném [serveru](https://cs.wikipedia.org/wiki/Webov%C3%BD_server) (většinou je to [IIS](https://cs.wikipedia.org/wiki/IIS)) se webová stránka s příponou .asp před odesláním klientovi, který si ji vyžádal, zpracuje (tj. pošle se výsledek toho, co vygeneroval [interpret](https://cs.wikipedia.org/wiki/Interpret_(software)) ASP kódu, zavolaný s vyžádanou stránkou jako parametrem).

## SQL

SQL je zkratka (anglicky Structured Query Language) pro standardizovaný strukturovaný dotazovací jazyk, který je používán pro práci s daty v relačních databázích. SQL je nástupcem jazyka SEQUEL (anglicky Structured English Query Language).

## Programovací jazyky Nižších prg. jazyků

Jazyk symbolických adres (Assembler) je programovací jazyk nejnižší úrovně a je závislý na [strojovém kódu](https://cs.wikipedia.org/wiki/Strojov%C3%BD_k%C3%B3d) [procesoru](https://cs.wikipedia.org/wiki/Mikroprocesor). Každá rodina procesorů má svůj vlastní odlišný JSA, protože ve strojových instrukcích různých rodin procesorů a možnosti rozdělování a adresování paměti bývají zásadní rozdíly. Každá firma vyrábějící procesory si definuje vlastní pravidla pro JSA svých procesorů, z kterých mohou (ale také nemusejí) vycházet nezávislí autoři a firmy.

V JSA programujeme za pomocí instrukcí. Tento jazyk ulehčil programátorovi práci, protože nemusel psát ve strojovém kódu, ale psal za pomocí mnemotechnickch symbolů, které se lépe zapamatovaly. Výhodou je používání symbolů oproti obyčejným číslům, používají se symboly třeba pro adresy, které se při překladu nahradí skutečnou adresou.

# Framework

Framework se snaží jít novou cestou, poskytnou programátorovi pomoc, nový efektnější způsob řešení a také ušetřit mnoho kódu, který by musel stále znova a znova dokolečka psát.

Framework (aplikační rámec) je softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů.

Může obsahovat podpůrné programy, knihovny API, podporu pro návrhové vzory nebo doporučené postupy při vývoji.

Cílem frameworku je převzetí typických problémů dané oblasti, čímž se usnadní vývoj tak, aby se návrháři a vývojáři mohli soustředit pouze na své zadání. Například, tým, který používá Apache Struts k vývoji webových stránek pro banku, se může zaměřit na to, jak se budou provádět bankovní operace a ne jak zajistit bezchybnou navigaci mezi jednotlivými stránkami, protože Framework obsahuje knihovny pro tyto funkce (Předprogramované funkce).

Další velmi důležitým faktorem při výběru je to, k čemu framework potřebuji a k čemu je určen. Obecně lze frameworky rozdělit na dvě skupiny:

1. Sady scriptů – knihoven – pokrývající všemožné potřeby
2. Scripty vytvářející jednu konkrétní webovou aplikaci

Představitelem první skupiny je například [Zend Framewok](http://framework.zend.com/) (patří také do 2. skupiny), [PRADO](http://www.pradosoft.com/) framework. Do druhé skupiny lze zařadit [CakePHP](http://cakephp.org/), [CodeIgniter](http://codeigniter.com/). Zřejmě by stálo za to blíže vysvětlit princip fungování 2. skupiny. Tato skupina většinou aplikuje určitý návrhový vzor, nejčastěji MVC. Všechny požadavky míří na jeden index.php, jenž následně řídí celou aplikaci, načítá potřebné soubory apod.

## MVC

MVC aneb **model-view-controller** je návrhový vzor, tedy objektová struktura, jež odděluje data a jejich zpracování od jejich zobrazení. Model představuje datové úložiště, obstarává získání dat a práci s nimi a vrací je controlleru, který si o ně zažádal, controller je následně předá view, který je jakýmsi způsobem zobrazí.

### Celý průběh skrze MVP

Úkolem **Routeru** je podle URL adresy poznat co uživatel chce a zavolat mu příslušný **Presenter**, který jeho dotaz obslouží. Presenter je komponenta, se kterou komunikuje uživatel. Předá jí parametry a ona mu vrátí HTML stránku. Presenter typicky parametry předá **Modelům**, od kterých získá data. Tato data předá **Templatům** (šablonám), které data začlení do nějakého HTML kódu. Tento HTML kód pošle presenter uživateli do prohlížeče.

## Nette Framework

* Nette Framework je open source framework pro tvorbu webových aplikací v PHP 5.
* Zaměřuje se na eliminaci bezpečnostních rizik
* Nette Framework je svobodný software, nabízený pod licencemi GNU GPL a licencí Nette
* Vlastnosti: Ladící nástroje, Aktivní komunita uživatelů v ČR

### Co potřebujeme k tomu, abychom mohli pracovat s nette

Pro zjednodušení uvedu, že Nette je tedy vlastně balík knihoven pro PHP. Když programujete v Nette, programujete stále v PHP a navíc používáte rozhraní těchto knihoven, které je objektové a staví na MVC architektuře. Proto pro použití Nette musíte mít s PHP pokročilé zkušenosti, znát dobře [objektově orientované programování](http://www.itnetwork.cz/php-objektove-orientovane-programovani-navody-a-tutorialy) a také [alespoň základy MVC architektury](http://www.itnetwork.cz/objektovy-mvc-redakcni-system-v-php). Učit se Nette bez těchto znalostí je zbytečný výdej energie, protože neporozumíte jeho principům a jen ztratíte svůj čas.

#### Sandbox

Abychom mohli Nette použít, potřebujeme tzv. **sandbox**. To je kostra webové aplikace s Nette frameworkem, do které budeme dále přidávat stránky a tak vytvoříme svůj web na Nette.