

## Témata bakalářské státní závěrečné zkoušky

Cílem tohoto textu je rozvést stručné znění témat bakalářské státní závěrečné zkoušky tak, aby bylo zřejmé (1) jaké konkrétní znalosti a dovednosti patří do obecně uvedených témat, (2) v jaké hloubce budou tyto znalosti a dovednosti vyžadovány, a (3) v jakých předmětech jsou témata obvykle rozebírána.

### Formulace požadavků

Text obsahuje společné požadavky z matematiky a informatiky a požadavky pro jednotlivé specializace. Požadavky pro jednotlivé specializace vždy zahrnují společné požadavky, které mohou prohlubovat. Všechny požadavky také implicitně zahrnují příslušné vstupní předpoklady, jako například zvládnutí kalkulu nebo zvládnutí konstrukcí příslušného programovacího jazyka.

Požadavky mohou být formulovány jako konkrétní ověřitelné znalosti, dovednosti a schopnosti, nebo jako výčet pojmů:

- Požadavek “pro konkrétní adresy a obsah jednoúrovňové stránkovací tabulky řešit úlohy překladu adres” formuluje ověřitelnou schopnost. Zkouška v takovém případě může testovat právě tuto schopnost nebo schopnosti přiměřeně související, v tomto případě například také schopnost popsat formát obsahu stránkovací tabulky, ze kterého schopnost použít tento obsah při překladu adres vyplývá.
- Požadavek “třídy složitosti P a NP, převoditelnost problémů, NP-těžkost a NP-úplnost” uvádí výčet pojmů. Zkouška v takovém případě může testovat znalost a použití uvedených pojmů v rozsahu, ve kterém se aktuálně probírají v uvedených předmětech.

V případě, že hloubka požadovaných znalostí není jasná, kontaktujte garanta specializace pomocí odkazu níže.

### Závaznost požadavků

Ačkoliv text se odkazuje na předměty, požadavky platí bez ohledu na volbu povinně volitelných předmětů, historické odchylky od aktuálního obsahu absolvovaných předmětů a další individuální okolnosti studia. Pokud okolnosti vedly k tomu, že část požadavků nebyla studiem pokryta, komise očekává, že si studenti chybějící znalosti doplní.

Pokud rozepsané znění požadavků výslovně definuje varianty pro různé individuální volby povinně volitelných předmětů označené symbolem ©, zkouška bude nabízet obdobné varianty pro všechny volby povinně volitelných předmětů podporované studijním plánem.

Text může být mezi zkušebními termíny aktualizován podle potřeb studijního programu a zpětné vazby zkušební komise a studentů. Rozepsané znění požadavků může přitom stručné znění požadavků ze studijních plánů jak rozšiřovat, tak zužovat. Změny jsou však omezeny tak, aby v období jednoho měsíce před zkušebním termínem nedocházelo k žádnému rozšiřování požadavků, a aby v období jednoho roku před zkušebním termínem nedocházelo k rozšiřování požadavků, které by vyžadovalo absolvování celého nového předmětu.

Toto je verze ze 28. července 2025, připravená pro bakalářské státní závěrečné zkoušky od září 2025.

V případě problémů prosím otevřete issue na <https://gitlab.mff.cuni.cz/mff/statnice/informatika-public-forum>.

# Společná matematika

## 1. Základy diferenciálního a integrálního počtu

- Posloupnosti reálných čísel a jejich limity
  - definice, aritmetika limit
  - věta o dvou policajtech, limity a uspořádání
- Řady
  - definice částečného součtu a součtu
  - geometrická řada, harmonická řada
- Reálné funkce jedné reálné proměnné
  - limita funkce v bodě
    - definice, aritmetika limit
    - vztah s uspořádáním
    - limita složené funkce
  - funkce spojité na intervalu
    - nabývání mezihodnot
    - nabývání maxima
- Derivace a její aplikace
  - definice a základní pravidla pro výpočet
  - l'Hospitalovo pravidlo
  - vyšetření průběhu funkcí: extrémy, monotonie a konvexita/konkavita
  - Taylorův polynom (limitní forma)
- Integrály a jejich aplikace
  - primitivní funkce: definice a metody výpočtu (substituce, per-partes)
  - Riemannův integrál: definice, souvislost s primitivní funkcí (Newtonovým integrálem)
  - aplikace
    - odhady součtu řad (konečných i nekonečných)
    - obsahy rovinných útvarů
    - objemy a povrchy rotačních útvarů v prostoru
    - délka křivky

### Předměty

- NMAI054 Matematická analýza 1 (5 kr)

## 2. Algebra a lineární algebra

- Algebraické struktury:
  - grupy a podgrupy, permutace
  - tělesa a speciálně konečná tělesa
- Soustavy lineárních rovnic:
  - maticový zápis, elementární řádkové úpravy, odstupňovaný tvar matice
  - Gaussova a Gaussova-Jordanova eliminace, popis množiny řešení
- Matice:
  - operace s maticemi a základní typy matic, hodnota matice
  - regulární a inverzní matice
- Vektorové prostory:
  - vektorový prostor, lineární kombinace, lineární závislost a nezávislost, lineární obal, systém generátorů
  - Steinitzova věta o výměně, báze, dimenze, souřadnice
  - vektorové podprostory, zejména maticové (řádkový, sloupcový, jádro) a jejich dimenze
- Lineární zobrazení:
  - definice, maticová reprezentace lineárního zobrazení, matice složeného zobrazení
  - obraz a jádro lineárních zobrazení
  - isomorfismus prostorů
- Skalární součin:
  - skalární součin, norma indukovaná skalárním součinem

- Pythagorova věta, Cauchyho-Schwarzova nerovnost, trojúhelníková nerovnost
- ortonormální systémy vektorů, Fourierovy koeficienty, Gramova-Schmidtova ortogonalizace
- ortogonální doplněk, ortogonální projekce, projekce jako lineární zobrazení
- ortogonální matice a jejich vlastnosti
- Determinanty:
  - definice a základní vlastnosti determinantu (multiplikativnost, determinant transponované matice, vztah s regularitou a vlastními čísly)
  - Laplaceův rozvoj determinantu
  - geometrická interpretace determinantu
- Vlastní čísla a vlastní vektory:
  - definice, geometrický význam a základní vlastnosti vlastních čísel, charakteristický polynom, násobnost vlastních čísel
  - podobnost a diagonalizovatelnost matic, spektrální rozklad
  - symetrické matice, jejich vlastní čísla a spektrální rozklad
- Positivně semidefinitní a pozitivně definitní matice:
  - charakterizace a vlastnosti, vztah se skalárním součinem, vlastními čísly
  - Choleského rozklad (znění věty a praktické použití)

#### Předměty

- NMAI057 Lineární algebra 1 (5 kr)
- NMAI058 Lineární algebra 2 (5 kr)

### 3. Diskrétní matematika

- Relace
  - vlastnosti binárních relací (reflexivita, symetrie, antisymetrie, tranzitivita)
- Ekvivalence a rozkladové třídy
- Částečná uspořádání
  - základní pojmy (minimální a maximální prvky, nejmenší a největší prvky, řetězec, antiřetězec)
  - výška a šířka částečně uspořádané množiny a věta o jejich vztahu (o dlouhém a širokém)
- Funkce
  - typy funkcí (prostá, na, bijekce)
  - počty různých typů funkcí mezi dvěma konečnými množinami
- Permutace a jejich základní vlastnosti (počet a pevný bod)
- Kombinační čísla a vztahy mezi nimi, binomická věta a její aplikace
- Princip inkluze a exkluze
  - obecná formulace (a důkaz)
  - použití (problém šatnářky, Eulerova funkce pro počet dělitelů, počet surjekcí)
- Hallova věta o systému různých reprezentantů a její vztah k párování v bipartitním grafu
  - princip důkazu a algoritmické aspekty (polynomiální algoritmus pro nalezení SRR)

#### Předměty

- NDMI002 Diskrétní matematika (5 kr)
- NDMI011 Kombinatorika a grafy 1 (5 kr)

### 4. Teorie grafů

- Základní pojmy teorie grafů
  - graf, vrcholy a hrany, izomorfismus grafů, podgraf, okolí vrcholu a stupeň vrcholu, doplněk grafu, bipartitní graf
- Základní příklady grafů
  - úplný graf a úplný bipartitní graf, cesty a kružnice
- Souvislost grafů, komponenty souvislosti, vzdálenost v grafu
- Stromy
  - definice a základní vlastnosti (existence listů, počet hran stromu)
  - ekvivalentní charakteristiky stromů
- Rovinné grafy
  - definice a základní pojmy (rovinný graf a rovinné nakreslení grafu, stěny)

- Eulerova formule a maximální počet hran rovinného grafu (důkaz a použití)
- Barevnost grafů
  - definice dobrého obarvení
  - vztah barevnosti a klikovosti grafu
- Hranová a vrcholová souvislost grafů
  - hranová a vrcholová verze Mengerovy věty
- Orientované grafy, silná a slabá souvislost
- Toky v sítích
  - definice sítě a toku v ní
  - existence maximálního toku (bez důkazu)
  - princip hledání maximálního toku v síti s celočíselnými kapacitami (například pomocí Ford-Fulkersonova algoritmu)

#### Předměty

- NDMI002 Diskrétní matematika (5 kr)
- NDMI011 Kombinatorika a grafy 1 (5 kr)

### 5. Pravděpodobnost a statistika

- Pravděpodobnostní prostor, náhodné jevy, pravděpodobnost
  - definice těchto pojmů, příklady
  - základní pravidla pro počítání s pravděpodobnostmi
  - nezávislost náhodných jevů, podmíněná pravděpodobnost
  - Bayesův vzorec
- Náhodné veličiny a jejich rozdělení
  - diskrétní i spojitý případ
  - popis pomocí distribuční funkce a pomocí pravděpodobnostní funkce/hustoty
  - střední hodnota
    - linearita střední hodnoty
    - střední hodnota součinu nezávislých veličin
    - Markovova nerovnost
  - rozptyl
    - definice
    - vzorec pro rozptyl součtu (závislých či nezávislých veličin)
  - práce s konkrétními rozděleními: geometrické, binomické, Poissonovo, normální, exponenciální
- Limitní věty
  - zákon velkých čísel
  - centrální limitní věta
- Bodové odhady
  - alespoň jedna metoda pro jejich tvorbu
  - vlastnosti
- Intervalové odhady: metoda založená na aproximaci normálním rozdělením
- Testování hypotéz
  - základní přístup
  - chyby 1. a 2. druhu
  - hladina významnosti

#### Předměty

- NDMI002 Diskrétní matematika (5 kr)
- NMAI059 Pravděpodobnost a statistika 1 (5 kr)

### 6. Logika

- Syntaxe
  - znalost a práce se základními pojmy syntaxe výrokové a predikátové logiky (jazyk, otevřená a uzavřená formule, instance formule, apod.)
  - normální tvary výrokových formulí
    - prenexní tvary formulí predikátové logiky

- znalost základních normálních tvarů (CNF, DNF, PNF)
- převody na normální tvary
- použití pro algoritmy (SAT, rezoluce)
- Sémantika
  - pojem modelu teorie
  - pravdivost, lživost, nezávislost formule vzhledem k teorii
  - splnitelnost, tautologie, důsledek
  - analýza výrokových teorií nad konečně mnoha prvovýroky
- Extenze teorií
  - schopnost porovnat sílu teorií
  - konzervativnost, skolemizace
- Dokazatelnost:
  - pojem formálního důkazu, zamítnutí
  - schopnost práce v některém z formálních dokazovacích systémů (např. tablo metoda, rezoluce, Hilbertovský kalkul)
- Věty o kompaktnosti a úplnosti výrokové a predikátové logiky
  - znění a porozumění významu
  - použití na příkladech, důsledky
- Rozhodnutelnost
  - pojem kompletnosti a její kritéria, význam pro rozhodnutelnost
  - příklady rozhodnutelných a nerozhodnutelných teorií

### **Předměty**

- NAIL062 Výroková a predikátová logika (5 kr)

# Společná informatika

## 1. Automaty a jazyky

- Regulární jazyky
  - regulární gramatiky
  - deterministický a nedeterministický konečný automat
  - regulární výrazy
- Bezkontextové jazyky
  - bezkontextové gramatiky, jazyk generovaný gramatikou
  - zásobníkový automat, třída jazyků přijímaných zásobníkovými automaty
- Rekurzivně spočetné jazyky
  - gramatika typu 0
  - Turingův stroj
  - algoritmicky nerozhodnutelné problémy
- Chomského hierarchie
  - schopnost zařazení konkrétního jazyka do Chomského hierarchie (zpravidla sestrojení odpovídajícího automatu či gramatiky)

### Předměty

- NTIN071 Automaty a gramatiky (2/5 z 5 kr)

## 2. Algoritmy a datové struktury

- Časová složitost algoritmů
  - časová a prostorová složitost algoritmu
  - měření velikosti dat
  - složitost v nejlepším, nejhorším a průměrném případě
  - asymptotická notace
- Třídy složitosti
  - třídy P a NP
  - převoditelnost problémů, NP-těžkost a NP-úplnost
  - příklady NP-úplných problémů a převodů mezi nimi
- Metoda rozděl a panuj
  - princip rekurzivního dělení problému na podproblémy
  - výpočet složitosti pomocí rekurentních rovnic
  - Master theorem (kuchařková věta) (bez důkazu)
  - aplikace
    - Mergesort
    - násobení dlouhých čísel
- Binární vyhledávací stromy
  - definice vyhledávacího stromu
  - operace s nevyvažovanými stromy
  - AVL stromy (definice)
- Třídění
  - primitivní třídící algoritmy (Bubblesort, Insertsort)
  - Quicksort
  - dolní odhad složitosti porovnávacích třídících algoritmů
- Grafové algoritmy
  - prohledávání do šířky a do hloubky
  - topologické třídění orientovaných grafů
  - nejkratší cesty v ohodnocených grafech (Dijkstrův a Bellmanův-Fordův algoritmus)
  - minimální kostra grafu (Jarníkův a Borůvkův algoritmus)
  - toky v sítích (Ford-Fulkerson algoritmus)

## Předměty

- NTIN060 Algoritmy a datové struktury 1 (3/5 z 5 kr)
- NTIN061 Algoritmy a datové struktury 2 (1/5 z 5 kr)

### 3. Programovací jazyky

Některé následující body definují varianty požadavků pro různé individuální volby povinně volitelných předmětů. Vyžaduje se zvládnutí všech bodů bez označení ☹ a zvládnutí všech bodů s označením ☹ pro jeden z jazyků C#, C++ nebo Java.

- Koncepty pro abstrakci, zapouzdření a polymorfismus.
  - související konstrukty programovacích jazyků
    - třídy, rozhraní, metody, datové položky, dědičnost, viditelnost
    - ☹ zapouzdření poskytované moduly v Javě
  - (dynamický) polymorfismus, statické a dynamické typování
  - jednoduchá dědičnost
    - ☹ virtuální a nevirtuální metody v C++ a C#
  - vícenásobná dědičnost a její problémy
    - ☹ vícenásobná a virtuální dědičnost v C++
    - ☹ interfaces a defaultní metody v Javě
    - ☹ interfaces v C#
  - implementace rozhraní (interface)
  - vhodné použití uvedených konceptů
- Primitivní a objektové typy a jejich reprezentace.
  - číselné a výčtové typy
  - ☹ ukazatele a reference v C++
  - ☹ hodnotové a referenční typy v C#
  - ☹ reference, imutabilní typy a boxing v C# a Javě
- Generické typy a funkcionální prvky (procedurálních programovacích jazyků).
  - ☹ šablony (templates) a statický polymorfismus v C++
  - ☹ generické typy v Javě a C# (bez omezení typových parametrů)
  - ☹ typy reprezentující funkce v C++, C#, nebo Javě
  - lambda funkce a funkcionální rozhraní
- Manipulace se zdroji a mechanismy pro ošetření chyb.
  - správa životního cyklu zdrojů v případě výskytu chyb
    - ☹ RAII v C++
    - ☹ `using` v C#
    - ☹ `try-with-resources` v Javě
  - konstrukce pro obsluhu a propagaci výjimek
- Životní cyklus objektů a správa paměti.
  - alokace (alokace statická, na zásobníku, na haldě)
  - inicializace (konstruktory, volání zděděných konstruktorů)
  - destrukce (destruktory, finalizátory)
  - explicitní uvolňování objektů, reference counting, garbage collector
- Vlákna a podpora synchronizace.
  - reprezentace vláken v programovacích jazycích
  - specifikace funkce vykonávané vláknem a základní operace na vlákny
  - časově závislé chyby a mechanismy pro synchronizaci vláken
- Implementace základních prvků objektových jazyků.
  - základní objektové koncepty v konkrétním jazyce
  - implementace a interní reprezentace primitivních typů
  - implementace a interní reprezentace složených typů a objektů
  - implementace dynamického polymorfismu (tabulka virtuálních metod)
- Nativní a interpretovaný běh, řízení překladač a sestavení programu.
  - reprezentace programu, bytecode, interpret jazyka
  - just-in-time (JIT) a ahead-of-time (AOT) překlad
  - proces sestavení programu, oddělený překlad, linkování
  - staticky a dynamicky linkované knihovny
  - běhové prostředí procesu a vazba na operační systém

## Předměty

Podle volby programovacího jazyka

- NPRG013 Programování v jazyce Java (5 kr)
- NPRG035 Programování v jazyce C# (5 kr)
- NPRG041 Programování v jazyce C++ (5 kr)

## 4. Architektura počítačů a operačních systémů

- Základní architektura počítače, reprezentace čísel, dat a programů.
  - reprezentace a přístup k datům v paměti, adresa, adresový prostor
  - ukládání jednoduchých a složených datových typů
  - základní aritmetické a logické operace
- Instrukční sada, vazba na prvky vyšších programovacích jazyků.
  - Implementovat běžné programové konstrukce vyšších jazyků (přiřazení, podmínka, cyklus, volání funkce) pomocí instrukcí procesoru
  - Zapsat běžnou konstrukci vyššího jazyka (přiřazení, podmínka, cyklus, volání funkce), která odpovídá zadané sekvenci (vysvětlených) instrukcí procesoru
- Podpora pro běh operačního systému.
  - privilegovaný a neprivilegovaný režim procesoru
  - jádro operačního systému
- Rozhraní periferních zařízení a jejich obsluha.
  - Popsat roli řadiče zařízení při programem řízené obsluze zařízení (PIO), pro zadané adresy a funkce vstupních a výstupních portů implementovat programem řízenou obsluhu zadaného zařízení (myš, disk)
  - Popsat roli přerušování při programem řízené obsluze zařízení (PIO), na úrovni vykonávání instrukcí popsat reakci procesoru (hardware) a operačního systému (software) na žádost o přerušování
- Základní abstrakce, rozhraní a mechanismy OS pro běh programů, sdílení prostředků a vstup/výstup.
  - neprivilegované (uživatelské) procesy
  - sdílení procesoru
    - procesy, vlákna, kontext procesu a vlákna
    - přepínání kontextu, kooperativní a preemptivní multitasking
    - plánování běhu procesů a vláken, stavy vlákna
  - sdílení paměti
    - Vysvětlit rozdíl mezi virtuální a fyzickou adresou a identifikovat, zda se v zadaném kontextu či fragmentu kódu používá virtuální nebo fyzická adresa
    - Na zadaném příkladu identifikovat a vysvětlit význam komponent virtuální a fyzické adresy (číslo stránky, číslo rámce, offset)
    - Pro konkrétní adresy a obsah jednoúrovňové stránkovací tabulky řešit úlohy překladu adres
    - Vysvětlit roli virtuálních adresových prostorů v ochraně paměti procesů a vláken
  - sdílení úložného prostoru
    - soubory, analogie s adresovým prostorem
    - abstrakce a rozhraní pro práci se soubory
- Paralelismus, vlákna a rozhraní pro jejich správu, synchronizace vláken.
  - časově závislé chyby (race conditions)
  - kritická sekce, vzájemné vyloučení
  - základní synchronizační primitiva, jejich rozhraní a použití
    - zámky
    - aktivní a pasivní čekání

## Předměty

- NSWI120 Principy počítačů (3 kr)
- NSWI170 Počítačové systémy (5 kr)

Podle volby programovacího jazyka

- NPRG013 Programování v jazyce Java (1/5 z 5 kr)
- NPRG035 Programování v jazyce C# (1/5 z 5 kr)
- NPRG041 Programování v jazyce C++ (1/5 z 5 kr)

# Webové a datově orientované programování

Studenti této specializace mají ke společným tématům ještě následující témata specializace.

## 1. Databáze

- Architektury databázových systémů
  - Vysvětlit pojmy konceptuální, logická a fyzická úroveň při návrhu dat.
  - Vysvětlit pojem normální formy při návrhu relací a důvody pro normalizaci relací. Uvést příklad relace, která danou normální formu porušuje a navrhnout opravu daného návrhu.
  - Vytvořit konceptuální model (UML, ER) pro zadaný problém. Převést konceptuální model na (relační) logický datový model.
- Transakční zpracování
  - Popsat ACID vlastnosti transakcí.
  - Vysvětlit a rozpoznat vlastnosti rozvrhů (uspořadatelnost, zotavitelnost).
  - Popsat a použít uzamykací protokoly pro dosažení vlastností rozvrhů.
  - Vysvětlit pojem zablokování.
- Přehled SQL
  - Vysvětlit význam jednoduchého SQL dotazu.
  - Vysvětlit účel jednotlivých klauzulí SQL dotazu.
  - Napsat jednoduchý SQL dotaz (včetně spojování tabulek) pro zadaný problém.
  - Použít klauzule pro seskupování a agregaci dat.
  - Použít vnořené dotazy a testy na NULL hodnotu.
- Moderní databázové systémy
  - Vyjmenovat základní třídy moderních databázových systémů a krátce popsat jejich specifické vlastnosti. Pomocí jednoduchého příkladu srovnat jejich vlastnosti s tradičními (SQL) databázemi.
  - Vysvětlit pojem Big Data, alespoň čtyři základní vlastnosti a jejich význam. Specifikovat alespoň dva příklady zdrojů velkých dat. Vysvětlit nové výzvy a problémy pro tradiční databázové systémy.
  - Vysvětlit princip MapReduce a zapsat v pseudokódu jednoduchý příklad jeho aplikace na konkrétní úlohu. Popsat výhody a nedostatky tohoto principu, zmínit alternativní přístupy.
  - Specifikovat typy, vlastnosti, výhody a nevýhody NoSQL databází. Ukázat rozdíly příslušných modelů na jednoduchém příkladu.
  - Specifikovat datový model grafové databáze, vytvořit jednoduchý příklad. Specifikovat třídy grafových dotazů, uvést jednoduché příklady použití.
  - Specifikovat rozdíly, výhody a nevýhody multi-model databáze a polystore. Vytvořit jednoduchý příklad multi-model dotazu a popsat možné problémy při jeho vyhodnocení.

### Předměty

- NDBI025 Databázové systémy (5 kr)
- NDBI026 Vývoj databázových aplikací (4 kr)
- NDBI040 Moderní databázové systémy (5 kr)

## 2. Datový management

- Datové formáty.
  - Popsat základní typy strukturovaných dat, uvést jejich reprezentanty a použití. Popsat, ve kterých situacích se který formát hodí, ukázat příklady.
  - Vysvětlit rozdíl mezi pojmy datový model, datový formát a datové schéma. Popsat základní vlastnosti textových formátů, uvést příklady standardizačních organizací, popsat jejich fungování, uvést příklady standardů definujících datové formáty.
  - Uvést, popsat, porovnat a použít modely a formáty pro grafová data - RDF a jeho serializace, Labeled Property Graf. Uvést, popsat a použít slovník pro definici slovníků použitelných v RDF - RDF Schema a uvést příklady použití. Uvést, popsat a použít jazyky pro dotazování a transformaci grafových dat - SPARQL, Cypher.
  - Uvést, popsat, porovnat a použít formáty pro stromová (hierarchická) data (XML, JSON), uvést, popsat a použít jazyky pro schémata stromových (hierarchických) dat XML Schema a JSON Schema, včetně příkladů.

- Vysvětlit, jakým způsobem lze zajistit, že na data v JSON lze nahlížet také jako na data v RDF vhodná pro výměnu na Webu (JSON-LD). Popsat a použít jazyk pro transformaci XML dat (XSLT).
- Uvést, použít a popsat formát pro tabulková data CSV a jeho specifikace. Popsat použití standardu *CSV on the Web* pro tvorbu schémat a zajištění, že na data v CSV lze nahlížet také jako na data v RDF vhodná pro výměnu na Webu.
- Vysvětlit a ukázat na příkladech, co je to souřadnicový referenční systém v kontextu prostorových dat, a ukázat, jak lze prostorová data reprezentovat v různých formátech, zejména WKT, GML, GeoJSON a GeoSPARQL.
- Vysvětlit, jak lze sémanticky popsat data pomocí RDF Schema a slovníků jako je Dublin Core či SKOS. Popsat datový model Wikidata a vysvětlit, jak se lze ve Wikidata dotazovat.
- Procesy zpracování dat
  - Popsat a na příkladu vysvětlit datové operace data selection, data projection, data summarization, data reduction, data lifting a data lowering.
  - Vysvětlit pojem datové kvality a její důležitost pro uživatele dat.
  - Uvést příklady dimenzí datové kvality a jejich měření.
  - Na zvoleném příkladu vysvětlit data provenance a ten popsat pomocí PROV-O ontologie.
- Katalogizace dat, metadata
  - Vyjmenovat druhy metadat a uvést jejich význam.
  - Popsat význam a využití datového katalogu.
  - Vysvětlit, jak je možné realizovat datový katalog pomocí DCAT (Data Catalog Vocabulary).
  - Na příkladu ukázat použití DCAT - zejména třídy `dcat:Dataset`, `dcat:DataService`, `dcat:Distribution` a `dcat:Catalog`.
- Sémantický popis dat, slovníky
  - Popsat vznik a řešení problému datových sil.
  - Popsat výhody využití kontrolovaných slovníků (controlled vocabulary) k popisu dat.
  - Vysvětlit rozdíl mezi typy slovníků (controlled list, taxonomy, thesaurus, classification scheme, ontology).
  - Popsat základní strukturu SKOS (Simple Knowledge Organization System).
  - Definovat ontologii pomocí SKOS a následně jí použít k popisu dat.
- Základy šifrování a komprese dat
  - Vysvětlit význam Shannonovy věty o kódování zdrojů (Shannon's source coding theorem) v kontextu komprese dat.
  - Na příkladech vysvětlit základní myšlenky algoritmů pro kompresi: Run-Length Encoding, Huffman Coding, Lempel-Ziv-Welch a Arithmetic coding.
  - Vysvětlit pojem digitálního certifikátu a jeho použití.
  - Vysvětlit potřebu existence PKI (public key infrastructure) ve vztahu k digitálnímu certifikátu.
  - Vysvětlit použití symetrického a asymetrického šifrování v kontextu digitálního certifikátu.
- Základy indexování
  - Na příkladech vysvětlit práci se základními typy organizace souborů (hromada, sekvenční soubor, indexovaný sekvenční soubor) a jejich (ne)výhody.
  - Vysvětlit a na příkladech demonstrovat pojmy přímé/nepřímé indexování a primární/sekundární index.
  - Vysvětlit principy hashování na vnější paměti, vybraný konkrétní přístup (např. Cormack, Larson & Kalja, Fagin, ...) demonstrovat na příkladu.
  - Vysvětlit, k čemu slouží a jaké jsou výhody hierarchických indexů. Na příkladu demonstrovat datovou strukturu B-strom a související operace. Popsat další modifikace (B+ strom, B\* strom).
- Indexování v prostorových databázích
  - Nakreslit a vysvětlit výhody křivek vyplňujících prostor (Z-křivka, Hilbertova křivka).
  - Pro zadaný příklad nakreslit Quad-tree, vysvětlit princip, výhody a nevýhody.
  - Pro zadaný příklad nakreslit k-d-tree, vysvětlit princip, výhody a nevýhody.
  - Pro zadaný příklad nakreslit R-strom, vysvětlit princip, výhody a nevýhody. Vysvětlit rozdíly oproti R+ stromu, popř. R\* stromu.
  - Vysvětlit pojem prostorové spojení - princip, problémy. Na příkladu demonstrovat konkrétní přístup podrobněji.

## Předměty

- NDBI046 Úvod do datového inženýrství (5 kr)
- NDBI007 Principy organizace dat (4 kr)
- NPRG036 Datové formáty (5 kr)

### 3. Web

- Principy www, HTML, XHTML, HTML5 a CSS
  - Vytvořit statickou webovou stránku pomocí HTML, HTML5 v rozsahu osobního blogu, či e-shopu.
  - Na příkladu ukázat výhody HTML5 sémantických tagů.
  - Na příkladu ukázat využití HTML formulářů včetně validace vstupních polí.
  - Vysvětlit principy fungování CSS: syntaxe, specifická selektorů, vložení do stránky.
  - Vytvořit responzivní layout stránky v rozsahu: menu, hlavní obsah se sloupci, patička.
- Architektury, základní principy, návrhové vzory a techniky webových aplikací
  - Vysvětlit použití návrhových vzorů: Front Controller, MVC/MVP, MVVC.
- Programování na straně klienta, JavaScript, standardní API v prohlížeči
  - Napsat klientský kód (JavaScript), který v reakci na událost provede dotaz na server, zpracuje odpověď a modifikuje DOM.
  - Uvést příklady standardních API dostupných v prohlížeči.
  - Vysvětlit a použít mechanismy pro asynchronní programování v JavaScriptu: callbacks, promises, async/await, event loop.
- API webových aplikací a webové služby
  - Vysvětlit základní principy REST API.
  - Popsat úroveň REST API.
  - Popsat REST API pomocí OpenAPI.
  - Položit dotaz v GraphQL, popsat výhody a nevýhody GraphQL.
- Single-page aplikace, udržování stavu a uživatelské relace
  - Vysvětlit princip fungování single-page aplikací.
  - Popsat možnosti udržování stavu pro webové aplikace v kontextu single-page aplikací.
- Programování na straně serveru, CGI a CGI-like aplikace
  - Vysvětlit fungování CGI a CGI-like aplikací.
  - Popsat možnosti udržování stavu pro webové aplikace a využití uživatelských relací.
  - Na příkladu demonstrovat PHP interleaving.
  - Vytvořit jednoduchou stránku v PHP, s využitím HTTP wrapperu a připojením k SQL databázi.
- Základy bezpečnosti webových aplikací
  - Vysvětlit vztah HTTPS a HTTP, popsat výhody.
  - Na příkladu (JWT) vysvětlit použití autentizačních tokenů.
  - Identifikovat a popsat základní bezpečnostní rizika webových aplikací.
- Doporučovací systémy
  - Vysvětlit typické workflow doporučovacího systému a popsat typické vstupy a výstupy. Popsat problémy, které způsobuje dynamičnost doporučovacího procesu (např. cold start, new item problem, online model updates).
  - Vysvětlit princip fungování, výhody a nevýhody kolaborativního filtrování. Vysvětlit funkci jednoduchých algoritmů (user/item-based KNN, varianty faktorizace matic).
  - Vysvětlit princip fungování, výhody a nevýhody content-based a knowledge-based doporučování.
  - Vysvětlit cíle, rozdíly a omezení v offline/online/user-studies hodnocení doporučovacích systémů a uvést typické hodnotící metriky.
- Vyhledávání na webu a v multimediálních databázích
  - Popsat booleovské a vektorové modely, word2vec.
  - Popsat vyhledávání v hypertextu, ranking, PageRank.
  - Vysvětlit optimalizaci webových stránek pro vyhledávače.
  - Popsat metrické indexování podobnosti (filtrování pomocí pivotů, maticové, stromové, hašované a hybridní indexy)
  - Uvést základní formáty spojené s vizuálními daty (konkrétně BMP, JPEG, MP4) a vysvětlit základní principy komprese videa (vysvětlit, proč P a B snímky zlepšují kompresi, vysvětlit kódovací strom u HEVC), popsat algoritmy detekce střihů ve videu pomocí podobnosti snímků a konvolučních sítí (stačí inference).
  - Formalizovat a vysvětlit základní podobnostní model (deskriptor, funkce podobnosti, Kosinova a Euklidovská vzdálenost), vysvětlit principy kombinace více modelů (early/late fusion).
  - Vysvětlit způsob vyhledávání a klasifikace v obrázkové databázi na základě textu s využitím neuronové sítě CLIP, vysvětlit základní stavební prvky architektury CLIP (stačí inference).
  - Popsat techniky vizualizace výsledků hledání v gridu pomocí různých technik zobrazení rankované množiny, popsat a vysvětlit možnosti zobrazení obrázkových dat pomocí SOM (self-organizing map) a algoritmus řazení obrázkových dat ve 2D gridu (self-sorting map)
  - Popsat techniky vyhodnocování efektivity vyhledávacího modelu (zejména pojmy přesnost, úplnost, mAP, F1-score), popsat možnosti vyhodnocování efektivity interaktivních systémů.

## **Předměty**

- NDBI038 Vyhledávání na webu (4 kr)
- NDBI045 Vyhledávání ve videu (5 kr)
- NSWI142 Programování webových aplikací (5 kr)
- NSWI153 Pokročilé programování webových aplikací (5 kr)
- NSWI166 Úvod do doporučovacích systémů a uživatelských preferencí (4 kr)

# Obecná informatika

Studenti této specializace si při přihlášení ke státní zkoušce vyberou ke společným tématům a následujícím tématům 1 a 2 ještě dvě z následujících témat 3 až 6.

## 1. Kombinatorika

- Vytvořující funkce
  - použití vytvořujících funkcí k řešení lineárních rekurencí
  - zobecněná binomická věta (formulace)
  - Catalanova čísla (příklad kombinatorické interpretace, odvození vzorce v uzavřeném tvaru)
- Odhady faktoriálu a kombinačních čísel
  - formulace základních odhadů
    - $(n/e)^n \leq n! \leq en(n/e)^n$
    - $(n/k)^k \leq \binom{n}{k} \leq (en/k)^k$
    - $2^{2m}/(2\sqrt{m}) \leq \binom{2m}{m} \leq 2^{2m}/\sqrt{2m}$
- Ramseyovy věty
  - Ramseyova věta (formulace konečné a nekonečné verze pro p-tice, důkaz verze p=2 pro 2 barvy)
  - Ramseyova čísla (definice, pro 2 barvy horní odhad z důkazu Ramseyovy věty a dolní odhad pravděpodobnostní konstrukcí)
- Extremální kombinatorika
  - obecné povědomí co extremální kombinatorika studuje
  - Turánova věta (formulace, Turánovy grafy)
  - Erdős-Ko-Radoova věta (formulace)
- Samoopravné kódy
  - přehled o používané terminologii
  - vzdálenost kódu a její vztah k počtu opravitelných a detekovatelných chyb
  - Hammingův odhad (formulace a důkaz)
  - perfektní kódy (definice a příklady, Hammingův kód bez přesné konstrukce)

### Předměty

- NDMI011 Kombinatorika a grafy 1 (5 kr)
- NDMI012 Kombinatorika a grafy 2 (5 kr)

## 2. Diferenciální a integrální počet ve více rozměrech

- Riemannův integrál jedno- i vícerozměrný
- Funkce více proměnných
  - parciální derivace: definice a výpočet
  - výpočet extrémů pomocí parciálních derivací
  - existence extrémů pro funkce několika reálných proměnných
  - vázané extrémy: výpočet pomocí Lagrangeových multiplikátorů
- Metrický prostor
  - definice a základní příklady
  - otevřené a uzavřené množiny: definice, příklady
  - spojitost funkce na metrickém prostoru
  - kompaktnost: definice a důsledky pro extrémy funkcí více proměnných
  - stejnoměrná spojitost

### Předměty

- NMAI055 Matematická analýza 2 (5 kr)

### 3. Optimalizace

- Základy lineárního a celočíselného programování
  - dualita lineárního programování, Farkasovo lemma
  - simplexová metoda, pivotovací pravidla
- Kombinatorická geometrie
  - konvexní obal objektů
  - mnohostěny
  - Minkowski-Weylova věta
- Základy matematického programování
  - unimodularita, Königovo lemma, toky v sítích, souvislost s dualitou LP
  - vážené maximální párování v bipartitních grafech a primárně-duální algoritmus na jeho nalezení
- Celočíselné programování
  - metoda řezů
- Matroidy
  - řádová funkce, existence a submodularita
  - hladový algoritmus
- Aproximační algoritmy
  - definice: aproximační poměr, aproximační schéma
  - hladové algoritmy
    - obchodní cestující - Christofidesův 3/2-aproximační algoritmus
    - rozvrhování
    - množinové pokrytí
  - použití lineárního programování
    - algoritmy pro splnitelnost (MAXSAT, pravděpodobnostní zaokrouhlování)
    - vrcholové a množinové pokrytí (deterministické zaokrouhlování, primárně-duální algoritmus)
- Využití pravděpodobnosti při návrhu algoritmů
  - minimální globální řez v grafu
  - hashování a jeho využití pro slovník s konstantním časem vyhledávání
  - pravděpodobnostní testování maticových a polynomiálních identit
  - paralelní algoritmus pro hledání maximální nezávislé množiny
  - paralelní algoritmy pro hledání párování (bipartitní grafy)

#### Předměty

- NDMI084 Úvod do aproximačních a pravděpodobnostních algoritmů (5 kr)
- NOPT046 Diskrétní a spojitá optimalizace (5 kr)
- NOPT048 Lineární programování a kombinatorická optimalizace (5 kr)

### 4. Pokročilé algoritmy a datové struktury

- Dynamické programování
  - princip dynamického programování (řešení podproblémů od nejmenších k největším)
  - aplikace: nejdelší rostoucí podposloupnost, editační vzdálenost
- Grafové algoritmy
  - komponenty silné souvislosti orientovaných grafů
  - toky v sítích (Dinicův a Goldbergův algoritmus)
  - toky v celočíselně ohodnocených grafech, aplikace na párování v bipartitních grafech
- Algoritmy vyhledávání v textu
  - algoritmy Knuth-Morris-Pratt a Aho-Corasicková
- Algebraické algoritmy
  - diskrétní Fourierova transformace a její aplikace
  - výpočet Fourierovy transformace algoritmem FFT
- RSA
  - šifrování, dešifrování a generování klíčů
- Aproximační algoritmy
  - aproximační poměr a relativní chyba
  - aproximační schémata
  - příklady: obchodní cestující, batoh
- Paralelní třídění pomocí komparátorových sítí

## Předměty

- NDMI010 Grafové algoritmy (3 kr)
- NMAI062 Algebra 1 (5 kr)
- NTIN060 Algoritmy a datové struktury 1 (5 kr)
- NTIN061 Algoritmy a datové struktury 2 (5 kr)

## 5. Geometrie

- Základní věty o konvexních množinách
  - Hellyho věta
  - Radonova věta
  - věta o oddělování
- Minkowského věta o mřížkách
- Konvexní mnohostěny
  - základní vlastnosti
  - V-mnohostěny, H-mnohostěny
  - kombinatorická složitost (maximální počet stěn mnohostěnu, cyklický mnohostěn a jeho vlastnosti)
- Geometrická dualita
  - duální množina
  - duální mnohostěn
  - dualita mezi body a nadrovinami
- Voroného diagramy
  - definice
  - souvislost s mnohostěny
  - horní odhad složitosti
- Arrangementy (komplexy) nadrovin
  - definice
  - počet buněk jednoduchého arrangementu  $n$  nadrovin v  $d$ -rozměrném prostoru
  - věta o zóně
- Incidence bodů a přímek
  - dolní odhad počtu incidencí
  - Szemerédi-Trotterova věta
- Základní algoritmy výpočetní geometrie
  - konstrukce arrangementu přímek v rovině
  - konstrukce konvexního obalu v rovině

## Předměty

- NDMI009 Základy kombinatorické a výpočetní geometrie (5 kr)

## 6. Pokročilá diskrétní matematika

- Barvení grafů
  - definice a základní vlastnosti
  - hranová barevnost (definice, formulace Vizingovy věty, souvislost s párováními v grafech)
  - Brooksova věta (formulace)
  - základní metody z důkazů Vizingovy a Brooksovy věty (Kempeho řetězce, hladový algoritmus)
  - silná a slabá věta o perfektních grafech (formulace), chordální grafy a další příklady tříd perfektních grafů
- Párování v grafech
  - definice párování a perfektního párování
  - párování v obecných grafech (formulace Tutteovy věty včetně důkazu jednodušší implikace, Petersenova věta a její důkaz použitím Tutteovy věty)
  - Edmondsův algoritmus (pouze vědět o jeho existenci)
- Kreslení grafů na plochách
  - základní topologické pojmy (homeomorfismus, křivka, plocha)
  - konstrukce ploch pomocí přidávání uší a křížtek (formulace), orientovatelné a neorientovatelné plochy, Eulerova charakteristika
  - pojem buňkového (2-cell) nakreslení

- zobecněná Eulerova formule, její použití pro horní odhad počtu hran a minimálního stupně v grafu nakresleném na dané ploše
- Grafové minory
  - definice a základní vlastnosti
  - zachovávání nakreslení při minorových operacích
- Množiny a zobrazení
  - přehled o používané terminologii (třídy a vlastní třídy, kartézský součin, relace, zobrazení, suma, potenční množina, ...)
- Subvalence a ekvivalence množin
  - definice
  - Cantorova-Bernsteinova věta (bez důkazu)
  - spočetné množiny
    - definice
    - zachovávání spočetnosti při množinových operacích
  - mohutnost množin racionálních a reálných čísel, důkaz neekvivalence diagonální metodou
- Dobré uspořádání
  - definice
  - ordinální a kardinální čísla (definice)

### **Předměty**

- NAIL063 Teorie množin (3 kr)
- NDMI012 Kombinatorika a grafy 2 (5 kr)

# Počítačová grafika, vidění a vývoj her

Studenti této specializace si při přihlášení ke státní zkoušce vyberou ke společným tématům ještě zaměření Počítačová grafika, Počítačové vidění nebo Vývoj počítačových her. Temata 1 až 3 jsou společná všem zaměřením, témata 4 až 6 jsou určena pro zaměření Počítačová grafika, témata 7 až 9 pro zaměření Počítačové vidění, témata 6 a 10 až 11 pro zaměření Vývoj počítačových her.

## 1. Matematická analýza

- Metrické prostory
  - Prostory  $E_n$ , podprostory, konvergence
  - Otevřené a uzavřené množiny
  - Kompaktní prostory, maxima a minima funkcí na kompaktním prostoru
  - Stejněměrná spojitost
- Reálné prostory více proměnných
- Parciální derivace
  - Definice
  - Totální diferenciál, geometrická interpretace
  - Spojitost parciálních derivací
  - Počítání parciálních derivací
- Riemannův integrál
  - Riemannův integrál v jedné proměnné
  - Riemannův integrál ve více proměnných

### Předměty

- NMAI055 Matematická analýza 2 (5 kr)

## 2. Základy 2D počítačové grafiky

- Základy lidského vidění a vnímání barev (jen přehledově)
  - Anatomie oka a sítnice, čočka, tyčinky a čípky (tři typy S, M, L)
  - Viditelné spektrum a metamerismus
  - Významné vlastnosti (nedokonalosti) systému vidění (barevná aberace, setrvačnost, schopnost integrace)
- Barevné systémy v digitálním světě
  - RGB, HSV, CMY[K], jejich převody (jen princip)
  - Aditivní a subtraktivní skládání barev
- High Dynamic Range grafika (HDR)
  - Princip, pořizování a aplikace HDR
- Průhlednost (alfa-kánál) a její aplikace v kompozici
  - Principy a algebra alfa kanálu
  - Kompozitní operace (over, atop, ...)
- Rastrová a vektorová grafika
  - Principy a rozdíly mezi rastrovým a vektorovým přístupem
  - Příklady rastrových a vektorových grafických formátů
  - Aplikace (který formát se hodí pro který účel)
- Základy rasterizace (převod z vektorové do rastrové reprezentace)
  - Kreslení čar, midpoint algoritmy (jen přehled)
  - Scanline vyplňování mnohoúhelníka
  - Záplavové vyplňování (jen přehled)
  - Reprezentace a vykreslování písma (jen přehled)
- Anti-aliasing v rastrové grafice (obecně)
  - Principy a metody v implementaci anti-aliasingu (analytický přístup, super-sampling)
- Kódování rastrového obrazu a grafické formáty
  - Bezeztrátová vs. ztrátová komprese
  - Populární přístupy v kompresi obrazu (RLE, GIF, PNG, JPEG komprese - jen přehled)
  - Kódování 2D tvarů (Quad-tree, Span-list)

## Předměty

- NPGR003 Základy počítačové grafiky (5 kr)

### 3. Základy 3D počítačové grafiky

- Souřadnicové systémy a transformace v 3D grafice
  - Kartézské a homogenní souřadnice
  - Lineární transformace maticemi 3x3 a 4x4
  - Kompozice složitějších transformací ve 3D (skládání z jednodušších kroků, příklady)
- Projekce do průmětny pro 3D zobrazování
  - Taxonomie projekčních metod (jen přehled), středová projekce pro perspektivní zobrazení
  - Repräsentace projekce maticí 4x4
- Posloupnost souřadných systémů v typické 3D aplikaci
  - Objektový systém, světový systém, systém kamery, normalizovaný systém po projekci, průmět na obrazovku
  - Logika práce s těmito systémy v realtime aplikaci
- Základy HW podporovaného programování 3D grafiky v OpenGL
  - Základní architektura GPU
  - Formáty dat - vektorová data 3D scény - posílaných do GPU (data pro jeden vrchol, buffery, index-buffery)
  - Programování GPU - shaders (jen přehled)
  - Textury a texturové souřadnice, transformační matice
  - Repräsentace výsledků, double-buffering (Present/SwapBuffers)
- Architektura typické 3D realtime aplikace
  - Jednotlivé fáze inicializace, hlavní smyčka, interaktivita
- Repräsentace 3D scén v počítači
  - Povrchové a objemové repräsentace, topologicky bohatší přístupy (Winged-edge, přehled)
  - CSG repräsentace (elementární tělesa a množinové operace)
  - Hierarchické repräsentace a jejich výhody (příklady)
  - Propojení se zobrazováním na GPU (hierarchie, maticové transformace ve scéně, vertex buffery...)
- Algoritmy na výpočet viditelnosti (jen principy a přehled)
  - Třídění podle hloubky (malíř, aplikace)
  - Hloubkový buffer (depth-buffer)
- Základy stínování
  - Model odrazu světla na povrchu tělesa
  - Phongův model odrazu, materiály
- Rekurzivní sledování paprsku (Ray-tracing) - jen přehled
  - Základní princip Ray-tracingu, jednotlivé komponenty RT systému - jen přehled
  - Nedostatky a principy jejich potlačení (anti-aliasing) - jen přehled

## Předměty

- NPGR003 Základy počítačové grafiky (5 kr)

### 4. Fotorealistická grafika

- Základy stínování
  - Modely odrazu světla na povrchu materiálu, definice
  - Spojité stínování
  - Jednoduché modely odrazu světla
  - Fyzikálně věrnější modely odrazu světla, mikroplošky
- Rekurzivní sledování paprsku (RT)
  - Základní princip RT, nevýhody a nedostatky naivního přístupu
  - Výpočet průsečíku paprsku se scénou
  - Urychlovací techniky vedoucí ke složitosti  $O(\log N)$
- Kvalitativní vylepšení naivního RT
  - Anti-aliasing a vzorkování, stochastické vzorkování, nestrannost
  - Texturování, 2D a 3D Textury
  - Simulace přírodních fenoménů, šumové funkce ve 2D a 3D
  - Distribuované sledování paprsku jako základní krok směrem k Monte-Carlo renderingu
- Monte-Carlo výpočet osvětlení

- Orientace a přehled
- Zobrazovací rovnice
- Princip Path-tracingu, nestranný (tj. korektní) odhad šíření světla ve scéně

#### Předměty

- NPGR004 Fotorealistická grafika (5 kr)

### 5. Základy vědy o barvách

- Fundamental causes of colour
  - Properties of visible light, and where it comes from (emission and reflection, key characteristics)
  - The human eye and visual system
  - Colour blindness
  - Colour mixing
- Colour spaces
  - Colour matching experiments and the CIE RGB and XYZ colour spaces
  - The CIE Luv and Lab colour spaces
  - Gamuts and gamut mapping
  - Colour atlases, such as Munsell or NCS
  - Colour collections, such as Pantone or RAL
- Colour measurement
  - Colour measurement devices vs. spectral measurement devices
- Colour reproduction technology
  - Additive devices: monitors, projectors
  - Subtractive devices: 2D printers (laser, inkjet, offset), 3D colour printers
- Colour Management Systems
  - Purpose of ICC files
  - Device calibration procedures
  - Rendering intents

#### Předměty

- NPGR025 Introduction to Colour Science (3 kr)

### 6. Geometrie pro počítačovou grafiku

- Eukleidovské shodnosti v rovině a prostoru
  - Klasifikace a reprezentace shodností
  - Aplikace na geometrické úlohy
  - Animace spojitého pohybu v rovině a prostoru
- Diferenciální geometrie křivek
  - Parametrizace a reparametrizace
  - Křivost a torze
  - Frenetovy vzorce
- Kvaterniony
  - Definice a vlastnosti
  - Popis rotací pomocí kvaternionů
  - Použití při animaci v prostoru, LERP a SLERP
- Projektivní prostor a projektivní zobrazení
  - Definice a základní vlastnosti
  - Vnoření Eukleidovského prostoru do projektivního
  - Aplikace na panoramatické lepení fotografií a rekonstrukci scény
  - Dvojpoměr a jeho využití při odečítání velikostí ze snímků

#### Předměty

- NPGR020 Geometrie pro počítačovou grafiku (3 kr)

## 7. Digitální zpracování obrazu

- Vzorkování a kvantizace obrazu
  - Digitalizace obrazu, vzorkování a kvantování spojitéch funkcí
  - Shannonova věta
  - Moiré, Anti-aliasing
- Základní operace nad obrázky
  - Histogram
  - Změny kontrastu
  - Redukce šumu
  - Zvyšování ostrosti
- Lineární filtrace obrazu
  - Filtrace v obrazovém prostoru
  - Konvoluce a Fourierova transformace
  - Filtrace v spektrálním prostoru
- Detekce hran a rohů
  - Detekce hran s využitím 1. derivace
  - Detekce hran s využitím 2. derivace
  - Detekce rohů
- Matematické modelování degradace obrazu
  - Pořízení obrazu, zdroje degradace, PSF
  - Potlačování základních zkreslení obrazu
  - Inverzní a Wienerův filtr

### Předměty

- NPGR002 Digitální zpracování obrazu (4 kr)

## 8. Počítačové vidění

- Pořizování obrazu, vlastnosti digitálního obrazu
  - Geometrie dírkové komory, rovnice tenké čočky
  - Hloubka ostrosti, aberace
  - Vzorkování, kvantování
- Matematická morfologie
  - Základní operace (dilatace, eroze, otevření, uzavření)
  - Pokročilé operace, šedotónová morfologie
  - Morfologické algoritmy, příklady použití
- Segmentace obrazu
  - Definice segmentace, Prahovací metody
  - Metody pracující s hranicemi
  - Metody pracující s oblastmi
  - K-means, Mean Shift
- Detekce, popis a párování lokálních příznaků
  - Detektory rohů
  - Deskriptory příznaků
  - Párování, RANSAC
- Významné oblasti v obraze
  - Sledování pohybu očí, vizualizace
  - Výpočtové modely vizuální pozornosti
- Detekce a sledování objektů
  - Gausovská a Laplaceovská pyramida a jejich použití
  - Optický tok
  - Kalmanův filtr

### Předměty

- NPGR036 Počítačové vidění (5 kr)

## 9. Strojové učení

- Výběr a předzpracování příznaků
  - Filter, wrapper a funkce vhodnosti příznaků
  - PCA, SVD
  - LDA
- Bayesovská teorie rozhodování
  - Bayesovo pravidlo
  - Kritérium minimální chyby
- Rozhodovací stromy
- Lineární klasifikátor
- Support Vector Machines
- Neuronové sítě
  - Dopřední sítě, zpětné šíření chyby
  - Konvoluční sítě
- Kombinace klasifikátorů
  - Bagging
  - Boosting
- Shluková analýza
  - Nehierarchické metody
  - Hierarchické metody
- Hodnocení kvality klasifikace
  - Matice záměn a z ní odvozené metriky
  - ROC a PR křivky

### Předměty

- NPGR035 Strojové učení v počítačovém vidění (5 kr)

## 10. Vývoj počítačových her

- 2D hry: sprite-based animace, 2D kostra, parallax scrolling, dlaždicové systémy, pixel art.
- 3D hry: 3D scény, modely, kosterní animace, rigging.
- 3D rendering: shadery, stíny, částicové systémy, billboards, screenspace efekty.
- Zvuk: zvukové efekty, 3D zvuk, sound engine, kompozice zvuku.
- Návrh architektury herního kódu, návrhové vzory pro počítačové hry.
- Herní design: definice, historie, taxonomie hráčů.
- Úvod do architektury herních engine, engine Unity.
- Řízení vývoje počítačových her a životní cyklus herního projektu.

### Předměty

- NPGR038 Základy vývoje počítačových her (5 kr)

## 11. GPU grafika

- Architektura moderního GPU
  - Programovatelná pipeline, pevné a programovatelné komponenty
  - Souřadná systémy používané při kreslení hierarchické 3D scény
  - Různé typy shaderů používané v moderních GPU
  - Předávání dat z aplikace do GPU, variabilní i konstantní data
  - Buffery, atributy vrcholů, textury
  - Render Targets
- Programování shaderů
  - Základy jazyka HLSL nebo GLSL
  - Kompilace a linkování shaderů
  - Constant-buffers
- Moderní a pokročilejší partie
  - Instancing
  - Programovatelný clipping/culling
  - Geometry/tesselation/mesh shadery

- Víceprůchodové algoritmy, Multiple Render Targets
- Vulkan nebo DirectX 12 - hlavní změny
- GPGPU
  - Masivně paralelní počítání na GPU
  - Základy CUDA nebo OpenCL

### **Předměty**

- NPGR019 Realtime grafika na GPU (5 kr)

# Programování a vývoj software

Studenti této specializace mají ke společným tématům ještě následující témata specializace.

## 1. Architektura počítačů, operačních systémů a sítí

- Organizace paměti za běhu programů
  - Paměť procesu
  - Dynamická alokace paměti
  - Aktivační záznam funkce
  - Volací konvence, předávání parametrů
  - Uložení proměnných, heap
- Paměťová reprezentace datových struktur
  - Reprezentace polí, struktur a tříd
  - Zarovnání paměťových struktur
- Cache a její vliv na výkonnost
  - Asociativní paměť
- Provádění instrukcí procesorem
  - Třídy instrukčních sad
  - Pipeline, superskalarita, out-of-order execution
- Operační systémy
  - Kooperativní a preemptivní přepínání kontextu
  - Souborové systémy
  - Pevný disk, plánovací algoritmy pro pohyb hlavy pevného disku
  - Segmentace, sdílená paměť, paměťově mapované soubory
- Síťová, linková a transportní vrstva, směrování, VLAN, NAT, adresování v TCP/IP
  - Vysvětlit základní vlastnosti a úkoly vrstvy síťového rozhraní
  - Vyjmenovat či identifikovat nejdůležitější položky ethernetového rámce, vysvětlit pojem MAC adresy a její význam
  - Propojovací zařízení na úrovni linkové vrstvy, přepínače (switch)
    - Popsat princip činnosti linkového rozhraní (přijímání, zpracování a odesílání rámců)
  - Vysvětlit základní vlastnosti a úkoly síťové vrstvy a její návaznost na vrstvu síťového rozhraní (hop-to-hop komunikace, přímé vs. nepřímé doručování)
  - Vymezení problému směrování (routing) a předávání (forwarding), směrovací tabulky
    - Na zadaném příkladu sítě (topologie, adresy, obsahy tabulek) popsat postup zpracování (doručení) konkrétního IP datagramu
  - IP adresy (IPv4 i IPv6)
    - Vysvětlit pojmy "IP adresa" a "síťová maska" a jejich význam pro síťovou vrstvu (doručování datagramů)
    - Popsat způsob přidělování IP adres a speciální prostory adres (privátní adresy, multicast)
    - Popsat na konkrétním problému možnou aplikaci NA(P)T překladu a vysvětlit jeho přínosy a nevýhody
    - Na příkladu dané sítě s použitím NA(P)T vysvětlit průchod datového paketu z vnitřní sítě do vnější a naopak
  - Popsat protokol DHCP (jak probíhá přidělování adres)
  - Transportní vrstva, adresace v TCP a UDP, spolehlivost, řízení toku
    - Vysvětlit základní vlastnosti a úkoly transportní vrstvy a nejdůležitější protokoly a koncepty (TCP, UDP, porty)
    - Vysvětlit způsoby alokace portů, uvést příklady dobře známých portů
    - Popsat algoritmy navazování a ukončení spojení v TCP (three way handshake)

### Předměty

- NSWI090 Počítačové sítě (3 kr)
- NSWI120 Principy počítačů (3 kr)
- NSWI141 Úvod do počítačových sítí (3 kr)
- NSWI143 Architektura počítačů (3 kr)
- NSWI170 Počítačové systémy (5 kr)

## 2. Programovací jazyky

Některé následující body definují varianty požadavků pro různé individuální volby povinně volitelných předmětů. Vyžaduje se zvládnutí všech bodů bez označení ☹ a zvládnutí všech bodů s označením ☹ pro jeden z jazyků C#, C++ nebo Java. Dále tato specializace vyžaduje zvládnutí společných požadavků z informatiky v sekci Programovací jazyky ve variantě pro jazyk C++.

- Programování v C++
  - Chápat sémantiku syntaktických konstrukcí a konceptů jazyka (r-value reference, move sémantika, lambdy) a umět je použít
  - Být schopen použít běžné standardní knihovní typy, funkce a třídy, znát jejich efektivitu
- Programování v dynamických jazycích
  - Rozumět typovým mechanismům dynamických jazyků (duck typing, prototypy)
  - Umět použít imutabilní typy, boxing/unboxing
- Statický vs. dynamický polymorfismus, generické programování, typová dedukce
  - Chápat výkonové důsledky různých implementací polymorfismu
  - Rutinně ovládat použití šablon v C++ (typově parametrizovaný kontejner, datový typ s operátory)
  - Umět prakticky využít parametrizaci generického kódu a politiky
  - Znat principy typové dedukce a umět je použít v běžném kódu
- Paralelní programování a synchronizace
  - Detailně chápat koncept vlákna a jeho explicitního a implicitního vzniku a zániku
  - Znat a umět prakticky použít prostředky pro implicitní (synchronized, atomic) a explicitní (mutex, semaphore, condition variable) synchronizaci
  - Znat další abstrakce paralelismu (task, future/promise)
- Komponenty, dependency injection
  - Být schopen vysvětlit vlastnosti a účel komponent, znát prostředky pro jejich popis
  - Rozumět reprezentaci komponent v objektovém programovacím jazyce a jejich běhové podpoře
- Reflexe a introspekce
  - ☹ Chápat princip a umět použít statickou introspekci v C++ (type\_traits) a run-time introspekci a reflexi v C# a Javě
  - ☹ Umět použít atributy v C++ a C#, anotace v Javě
- Návrhové vzory
  - Znat konkrétní návrhové vzory dle GoF, jejich aplikaci a vzájemné vztahy
- Principy WWW, HTTP, URL, statické webové stránky
  - Umět vysvětlit vztah URI, IRI a URL, umět popsat strukturu URL
  - Umět popsat strukturu HTTP požadavku
  - Umět vysvětlit rozdíl mezi HTTP GET, POST, PUT a DELETE požadavky
  - Umět vysvětlit, jak mohou hlavičky v HTTP požadavku ovlivnit zpracování požadavku serverem a podobu odpovědi od serveru
  - Umět vysvětlit a na příkladu demonstrovat princip využití stavových kódů
  - Mít přehled o základních stavových kódech (20x, 30x, 40x, 50x) a jejich použití
  - Umět vytvořit statickou webovou stránku s pomocí HTML a CSS
  - Umět uvést příklady sémantických HTML tagů (HTML5) a význam jejich použití
  - Umět vysvětlit princip zpracování webové stránky (HTML, CSS, JavaScript) prohlížečem
  - Umět vysvětlit pojem DOM a jeho možnosti použití a interakce
  - Umět vytvořit HTML formulář
  - Mít povědomí o existujících vstupních prvcích pro HTML formuláře a jejich použití
  - Umět vysvětlit a na příkladu demonstrovat možnosti validace HTML formuláře s pomocí HTML
  - Umět vysvětlit princip zpracování HTML formuláře na straně serveru
  - Umět použít CSS ke stylování webové stránky
  - Umět použít základní CSS selektory (id, class, element, attribute, \*, >, ...)
  - Umět vysvětlit a využít specifitu CSS selektorů
- Principy webových aplikací
  - Chápat principy single-page a multiple-page aplikací a správy stavu webové aplikace
  - Umět základní konstrukce a koncepty JavaScriptu
  - Umět vysvětlit princip event modelu v JavaScriptu
  - Umět vysvětlit princip asynchronního programování v JavaScriptu pomocí promise, callback, nebo async/await
- Programování na straně klienta
  - Umět modifikovat DOM pomocí JavaScriptu
  - Umět popsat různé způsoby registrace obsluhy událostí pomocí JavaScriptu
  - Umět vysvětlit využití Fetch API

- Umět vysvětlit vliv CORS na využití Fetch API

### Předměty

- NPRG024 Návrhové vzory (3 kr)
  - NPRG041 Programování v C++ (3 kr)
  - NPRG043 Doporučené postupy v programování (5 kr)
  - NSWI142 Programování webových aplikací (5 kr)
- Podle volby programovacího jazyka
- NPRG021 Pokročilé programování v jazyce Java (5 kr), nebo
  - NPRG038 Pokročilé programování v jazyce C# (5 kr), nebo
  - NPRG051 Pokročilé programování v jazyce C++ (5 kr)

## 3. Softwarové inženýrství

- Procesy vývoje software
  - Vysvětlit, co je to proces vývoje software
  - Znat základní klasifikaci procesů vývoje software (plan-driven : vodopád, iterativní vývoj + agile) a popsat jejich výhody/nevýhody.
  - Umět vyjmenovat a definovat základní aktivity v procesu vývoje (analýza, návrh, implementace, testování) a vysvětlit rozdíly mezi nimi.
- Analýza požadavků
  - Umět vysvětlit, že do analýzy spadá minimálně identifikace stakeholderů a porozumění jim, formulace vize projektu/produktu a potom identifikace, strukturování a formalizování požadavků jako základního driveru vývoje aplikačního software.
    - Umět vysvětlit, co je to stakeholder, proč je důležité identifikovat všechny a porozumět jim, nezapomenout na někoho.
    - Umět říci a vysvětlit alespoň jednu techniku, jak se stakeholdery komunikovat (event-storming, interviews)
  - Umět klasifikovat požadavky na funkční a kvalitativní (=ne-funkční)
  - Umět identifikovat požadavky ve vágním textovém zadání stakeholderů a umět odlišit v tomto zadání funkční a kvalitativní (ne-funkční).
  - Funkční požadavky
    - Znat způsoby specifikace funkčních požadavků (user stories, use cases, wireframes, UML - use case diagram, activity diagram, state diagram)
      - Znat strukturu user story a umět v ní user stories vyjádřit (věta: WHO, WHAT, WHY)
      - Znat strukturu use case a umět v ní use case vyjádřit (preconditions, basic flow, alternative flows, postconditions, actors)
      - Znat následující detaily UML diagramů pro specifikaci požadavků (include + extend vazby v use case diagramech, executable+control+object nodes v activity diagramech, a stavy ve state-machine diagramech)
    - Umět vyjádřit funkční požadavky v libovolném z těchto způsobů na základě vágního textového vyjádření požadavků stakeholderem.
    - Umět vysvětlit, kdy/pro co se hodí použít jaký způsob.
  - Doménový model (lze použít pojem business vocabulary, konceptuální model)
    - Umět vysvětlit, k čemu je doménový model dobrý při tvorbě software
    - Vědět, jak jej vyjádřit (UML diagram tříd) a umět jej vytvořit na základě vágního textového popisu domény.
    - Znat následující detaily UML diagramů tříd (třída, atribut, asociace, multiplicity atributů a tříd, asociační třídy, generalizace/specializace tříd)
    - Vědět, že existují pravidla na vytváření názvů tříd, atributů a asociací (ne ad-hoc pojmenovávání, musí to být ukotveno v byznys slovníku non-technical lidí v organizaci/doméně).
    - Vědět, že třídy (volitelně i atributy a asociace ale to nezkoušíme) musí mít dobrou definici, která vysvětluje sémantiku dané třídy a že máme nějaká pravidla na to, jak mají definice vypadat.
- Návrh/Design
  - Umět vysvětlit, že do designu spadá minimálně návrh architektury, datových struktur, interfaces/API, návrh UX/UI a ke každému něco říct.
  - Návrh architektury
    - Umět vysvětlit, co je architektura.
    - Umět na malém zadání funkčních požadavků (v podobě user stories nebo usecases) a kvalitativních

- požadavků navrhnout architekturu:
  - identifikovat responsibilities (malé features, které systém musí umět pro naplnění požadavku na prezentační, aplikační/business, datové vrstvě)
  - dekomponovat do základních modulů kódu, tyto v případě potřeby dále dekomponovat
  - navrhnout závislosti mezi moduly
  - navrhnout runtime architekturu v jednom z těchto stylů (layered monolith, modularized monolith, service-based architecture)
- Umět vysvětlit základní principy návrhu architektury (high cohesion, low coupling, separation of concerns, information hiding, layering, domain boundaries) a aplikovat je při návrhu architektury v bodu výše.
- Návrh datových struktur
  - Umět odlišit datové struktury navrhované pro optimalizaci práce s daty pro potřeby algoritmů vs. doménové datové struktury (tabulky, JSONy, XMLka, ...) pro reprezentaci dat o reálném světě v software.
  - Doménové datové struktury - umět vyjmenovat, že potřebujeme typicky navrhnout datové struktury pro databáze, rozhraní/API
  - Vědět, že je dobré a proč je dobré navrhovat doménové datové struktury ve vazbě na doménový model z analýzy.
  - Umět navrhnout doménovou datovou strukturu z textového zadání domény a požadavků nebo z doménového modelu (vyjádřeného jako UML diagram tříd) - netřeba znát detailně věci jako JSON Schema nebo XML Schema, student by ale měl být schopen tabulkovou nebo hierarchickou (ala JSON/XML) strukturu navrhnout v nějakém pseudo schema jazyku (ala SQL, JSON Schema, XML Schema, ...)
- Návrh rozhraní/API
  - Umět vysvětlit, proč jsou rozhraní a API důležitá z pohledu modularizace software a dobré architektury.
  - Umět ukázat na pseudo-kódu definici nějakého rozhraní/API na základě textového popisu nějakého architektonického modulu (tj. v našem sw bude modul/komponenta, která umí to a to - navrhnete strukturu a operace API/rozhraní)
- Testování
  - Umět popsat model procesu testování software a fáze testování (development testing, release testing, user testing testing)
  - Umět popsat druhy development testování (unit/jednotkové testy, integrační testy, systémové testy)
    - Popsat a na ukázkách kódu identifikovat typickou strukturu testu (set up, tear down, execute and validate)
    - Umět napsat unit testy pro daný malý modul zadaný konkrétním kódem
    - Umět vysvětlit mockování a uplatnit v unit testech, alespoň principiálně.
    - Popsat hlavní koncepty black box, white box
  - Vysvětlit omezení běžných testovacích postupů a základní možnosti řešení
  - Umět definovat, co je testovací scénář pro release/user testing
    - Chápat pojem a umět použít use cases resp. user stories připravené v analýze.
    - Vědět, že identifikujeme test conditions (a vysvětlit co to je) a pak sestavujeme test cases na základě scénářů z use case, příp. na základě user stories.
    - Vědět, že test cases se dají automatizovat.
    - Umět prakticky použít
      - k danému user story/use case umět napsat pár konkrétních test conditions, sestavit test case, navázat na pokryté test conditions.
      - znát rozdíl mezi prováděním test cases ručně a automatizovaně za pomoci nástrojů, umět v nějakém pseudo-kódu napsat testovací skript na základě test case.
  - Popsat způsoby automatizace testování
- Správa verzí
  - Popsat účel a použití systémů pro správu verzí při vývoji rozsáhlého software a práci v týmech (typicky nabízené funkce těchto systémů a jejich použití pro řešení běžných situací)
  - Popsat běžné způsoby integrace verzovacích systémů s nástroji pro správu projektů
  - Vysvětlit hlavní koncepty (lokální a vzdálené repozitáře, pracovní kopie (working copy), commit, větve)
  - Popsat a použít operace clone, pull, push, add, diff, commit, branch, merge, log, blame nástroje git
  - Popsat koncept feature branch v kontextu procesu vývoje a udržování software
  - Vysvětlit konflikty mezi verzemi, důvody vzniku a způsoby řešení
- Systémy pro sestavování software
  - Popsat hlavní účel a typicky nabízené funkce
  - Vysvětlit hlavní koncepty (cíl, závislosti, akce)

- Měření výkonnosti
  - Vysvětlit postup zvaný “profilování” (základní kroky, běžné metriky, interpretace výsledků měření, hlavní praktická omezení)
- Návrh API, tříd a metod, objektový návrh
  - dekompozice, zapouzdření, návrh struktury a rozhraní tříd
  - návrh efektivních rozhraní tříd, modulů a knihoven
  - Umět popsat základní principy návrhu REST API.
  - Umět vysvětlit rozdíl mezi REST API a HTTP API využívající JSON.
  - Umět navrhnout a REST API.
- Základy bezpečnosti webových aplikací
  - Umět vysvětlit následující druhy útoků:
    - SQL injection,
    - cross-side scripting.
  - Umět navrhnout možnosti ochrany aplikace před útoky:
    - SQL injection,
    - cross-side scripting.
  - Umět vysvětlit princip protokolu HTTPS jako nadstavby nad protokolem HTTP.
  - Umět vysvětlit využití digitálního certifikátu v kontextu HTTPS.
  - Umět vysvětlit princip autentizace a autorizace.

### Předměty

- NSWI041 Úvod do softwarového inženýrství (5 kr)
- NPRG043 Doporučené postupy v programování (5 kr)
- NSWI130 Architektury softwarových systémů (5 kr)
- NSWI142 Programování webových aplikací (5 kr)
- NSWI154 Nástroje pro vývoj software (2 kr)

## 4. Databáze

- Architektury databázových systémů.
  - Popsat rozdíly mezi jednotlivými úrovněmi (konceptuální, logická, fyzická) pohledů na data a jejich úlohu při návrhu databázové vrstvy aplikace.
  - Převody mezi jednotlivými úrovněmi pohledů (konceptuální na logický relační a naopak, převod relace na fyzický model v jazyce SQL).
  - Popsat normální formy schémat relací a jejich úlohu při návrhu logického relačního modelu.
  - Identifikovat normální formu relace na základě známých funkčních závislostí.
  - Použít dekompozici relace pro dosažení požadované normální formy.
  - Vysvětlit pojmy bezetrátovost a zachování závislostí při dělení univerzální relace.
  - Popsat a použít prostředky pro dosažení referenční integrity při návrhu relací.
- Transakční zpracování.
  - Popsat a vysvětlit ACID vlastnosti transakcí.
  - Popsat pojem transakční rozvrh a jeho vlastností (typy uspořadatelnosti, zotavitelnost, dodržení (striktního) dvoufázového protokolu)
  - Umět identifikovat splnění vlastností transakčního rozvrhu.
  - Popsat příčiny uvážnutí transakčního rozvrhu a jeho řešení.
- Přehled SQL.
  - Umět napsat jednoduchý SQL dotaz - SELECT, FROM, WHERE, ORDER BY
  - Umět použít seskupování dat a agregaci.
  - Umět použít spojování tabulek v dotazech.
  - Vnořené dotazy, operátory, testy na NULL hodnotu.
  - Umět slovně popsat význam dotazu napsaného v jazyce SQL a identifikovat základní syntaktické chyby v něm
- Moderní databázové systémy
  - Vyjmenovat základní třídy moderních databázových systémů a krátce popsat jejich specifické vlastnosti. Pomocí jednoduchého příkladu srovnat jejich vlastnosti s tradičními (SQL) databázemi.
  - Vysvětlit pojem Big Data, alespoň čtyři základní vlastnosti a jejich význam. Specifikovat alespoň dva příklady zdrojů velkých dat. Vysvětlit nové výzvy a problémy pro tradiční databázové systémy.
  - Vysvětlit princip MapReduce a zapsat v pseudokódu jednoduchý příklad jeho aplikace na konkrétní úlohu. Popsat výhody a nedostatky tohoto principu, zmínit alternativní přístupy.

- Specifikovat typy, vlastnosti, výhody a nevýhody NoSQL databází. Ukázat rozdíly příslušných modelů na jednoduchém příkladu.
- Specifikovat datový model grafové databáze, vytvořit jednoduchý příklad. Specifikovat třídy grafových dotazů, uvést jednoduché příklady použití.
- Specifikovat rozdíly, výhody a nevýhody multi-model databáze a polystore. Vytvořit jednoduchý příklad multi-model dotazu a popsat možné problémy při jeho vyhodnocení.

### **Předměty**

- NDBI025 Databázové systémy (5 kr)
- NDBI040 Moderní databázové koncepty (5 kr)

# Systemové programování

Studenti této specializace mají ke společným tématům ještě následující témata specializace.

## 1. Architektura počítačů

- Výkonnost počítače a procesoru, metriky a omezení
  - Vyjádřit a na příkladech demonstrovat vztah mezi dobou běhu programu a metrikami na úrovni architektury (CPI, IPC)
  - Popsat vliv instrukčního mixu na hodnoty metrik CPI a IPC, identifikovat typické hodnoty CPI a IPC
  - Formulovat Amdahlův zákon a použít ho pro odhad limitů zrychlení díky paralelismu
- Zpracování instrukcí procesorem, paralelismus, predikce a spekulace
  - Na diagramu datové cesty procesoru vysvětlit postup zpracování základních instrukcí (add, load, store, branch)
  - Popsat kroky vykonané procesorem pro obsluhu přerušení a obsluhu výjimek
  - Na příkladu kódu popsat zřetěžené zpracování (pipelining), identifikovat datové závislosti a popsat jejich řešení, odhadnout zrychlení
  - Popsat zřetěžené zpracování (pipelining) za přítomnosti (také podmíněných a nepřímých) skoků, popsat a na příkladu identifikovat rozhodnutí základních prediktorů (saturating counter), popsat spekulativní vykonávání kódu a jeho vliv na výkon
  - Popsat rozdíl mezi hardware cores a hardware threads a jeho vliv na výkon
- Architektura paměťového subsystému, architektura cache
  - Použít metriky hit či miss ratio a access latency pro odhad rychlosti přístupu do paměti
  - Pro zadanou architekturu (přímo mapovaná, množinově či plně asociativní) a relevantní parametry (velikost, stupeň asociativity, victim replacement policy) cache a zadanou sekvenci přístupů do paměti určit chování cache (hit/miss, 3C model, obsazení konkrétních cache lines)
  - Diskutovat roli vrstev ve víceúrovňové architektuře cache a identifikovat obvyklé parametry vrstev
- Multi-core a multi-socket systémy, koherence cache
  - Na diagramu identifikovat UMA a NUMA architektury
  - Popsat záruky poskytované mechanismy cache coherence při přístupu do paměti
  - Popsat a na příkladech ilustrovat fungování cache coherence protokolů (IV, MSI, MESI)
  - Popsat a na příkladech kódu ilustrovat false sharing

### Předměty

- NSWI143 Architektura počítačů (3 kr)

## 2. Operační systémy

- Spouštění procesů, dynamicky linkované knihovny, volací konvence
  - Použít systémové API (fork, exec, join ...) pro spuštění a (počkání na) ukončení procesu
  - Použít pthread API (pthread\_create, pthread\_join ...) pro spuštění a (počkání na) ukončení vlákn
  - Identifikovat statické a dynamické linkování, použít obvyklé nástroje pro vytvoření staticky a dynamicky linkovaného programu
  - Použít obvyklé nástroje pro identifikaci linkovaných symbolů, identifikovat externí a exportované symboly
  - Porovnat použití a režii statického a dynamického linkování
  - Identifikovat na ukázkách kódu pozičně závislý a pozičně nezávislý program, použít překladač pro vytvoření takového programu
  - Identifikovat a vysvětlit použití absolutních adres pro adresaci proměnných na ukázkách kódu
  - Identifikovat a vysvětlit použití relativních adres pro adresaci proměnných na ukázkách kódu
  - Porovnat použití a režii absolutní a relativní adresace
  - Popsat obvyklé umístění argumentů, proměnných a dalších informací na zásobníku programů
  - Popsat obvyklé použití registrů procesoru uvnitř funkcí a při volání funkcí
- Paralelismus a synchronizace na multiprocsorech
  - Popsat a sestavit příklad race condition nad sdíleným čítačem s použitím interleaving sémantiky běhu programu

- Popsat a sestavit příklad race condition nad sdíleným seznamem s použitím interleaving sémantiky běhu programu
- Popsat sémantiku prostředků pro implementaci synchronizace na multiprocsorech (IPI, T&S, CAS, LL/SC)
- Implementovat spin lock s použitím zadaného prostředku (T&S, CAS, LL/SC)
- Implementovat blokující zámek s použitím rozhraní futex
- Rozhraní pro synchronizaci
  - Popsat sémantiku atomických typů, bariér, zámků, semaforů, condition variables jako nástrojů pro synchronizaci
  - Použít zadaný nástroj (atomické typy, bariéry, zámky, semaforey, condition variables) pro odstranění race condition v ukázkách kódu
  - Použít zadaný nástroj (atomické typy, bariéry, zámky, semaforey, condition variables) pro implementaci problému producent-konzument
- Správa paměti na multiprocsorech, alokátory, garbage collection
  - Na zadaném příkladu datových struktur heap alokátoru popsát jeho funkci a odhadnout prostorovou a časovou režii
  - Na zadaném příkladu obsahu heapu demonstrovat algoritmy mark and sweep a copying garbage collection
  - Ovládat související terminologii (heap, mutator, collector, reachability, garbage)
- Rozhraní pro práci se soubory, paměťově mapované soubory
  - Použít systémové API (open, seek, read, write, close, readv, writev, mmap) pro práci se soubory
  - Popsat interakci systému souborů a správy paměti při přístupu k souborům pomocí mapování (mmap)
- Interní struktura základních systémů souborů
  - Vyjmenovat obvyklá metadata asociovaná se souborem
  - Na zadaném příkladu datových struktur systému souborů (FAT, inode, extent, dentry, bitmap) popsát realizaci operací se souborem (open, read, write)
- Princip komunikace mezi perifériemi a operačním systémem
  - Popsat a na příkladech kódu identifikovat jednotlivé kroky obsluhy žádosti o přerušení od zařízení
  - Popsat omezení nástrojů pro synchronizaci uvnitř kódu obsluhujícího žádosti o přerušení
  - Popsat jednotlivé kroky přenosu dat mezi zařízením a pamětí při použití DMA

### Předměty

- NSWI200 Operační systémy (5 kr)

## 3. Počítačové sítě

- Linková vrstva, adresace v Ethernetu
  - Vysvětlit základní vlastnosti a úkoly vrstvy síťového rozhraní
  - Vyjmenovat/identifikovat nejdůležitější položky ethernetové rámce, vysvětlit pojem MAC adresy a její význam
  - Propojovací zařízení na úrovni linkové vrstvy, přepínače (switch)
    - Popsat princip činnosti linkového rozhraní (přijímání, zpracování a odesílání rámců)
    - Popsat principy základní optimalizace síťového toku (filtrování, cílený forwarding, metoda zpětného učení)
- Síťová vrstva, adresace v IPv4 a IPv6, statické směrování, NAT
  - Vysvětlit základní vlastnosti a úkoly síťové vrstvy a její návaznost na vrstvy síťového rozhraní (hop-to-hop komunikace, přímé vs. nepřímé doručování)
  - Vymezení problému směrování (routing) a předávání (forwarding), směrovací tabulky
    - Vysvětlit na příkladu strukturu záznamů ve směrovacích tabulkách a význam jednotlivých položek
    - Na zadaném příkladu sítě (topologie, adresy, obsahy tabulek) popsát postup zpracování (doručení) konkrétního IP datagramu
  - IP adresy (v4 i v6)
    - Vysvětlit pojem IP adresa a síťová maska a jejich význam pro síťovou vrstvu (doručování datagramů)
    - Popsat způsob přidělování IP adres a speciální prostory adres (privátní adresy, multicast)
    - Popsat na konkrétním problému možnou aplikaci NA(P)T překladu a vysvětlit jeho užitečnost vs. nevýhody
    - Na příkladu dané sítě s použitím NA(P)T vysvětlit průchod datového paketu z vnitřní sítě do vnější a naopak
  - Protokol IPv4, základní vlastnosti, struktura IPv4 datagramu
    - Na příkladu identifikovat hlavní položky hlavičky (verze, time-to-live, protokol, kontrolní součet, adresy)

- Na daném příkladu hlaviček datagramu (a konfigurace routeru) vysvětlit, jak bude datagram zpracován
  - Včetně atypických situací jako je např. zahození datagramu
- Vysvětlit základní princip fungování nástroje `traceroute`
- Popsat princip IPv4 fragmentace (s návazností na hlavičky IP datagramu) a navrhnout alespoň jeden postup, jak se fragmentaci vyhnout
- Protokol ICMPv4, jeho účel a použití, základní typy zpráv, fungování nástroje Ping
- Popsat protokol ARP a jeho význam pro interakci mezi síťovou a linkovou vrstvou
- Popsat protokol DHCP (jak probíhá přidělování adres)
- Transportní vrstva, adresace v TCP a UDP, spolehlivost, řízení toku
  - Vysvětlit základní vlastnosti a úkoly transportní vrstvy a nejdůležitější protokoly a koncepty (TCP, UDP, porty)
  - Vysvětlit způsoby alokace portů, uvést příklady dobře známých portů
  - Na příkladu identifikovat a vysvětlit nejdůležitější položky UDP packetu a jejich souvislost s IP protokolem
  - Na příkladu identifikovat a vysvětlit nejdůležitější položky TCP packetu, vysvětlit souvislost tohoto packetu s přenášeným proudem dat
  - Popsat algoritmy navazování a ukončení spojení v TCP (three-way handshake)
  - Popsat princip řízení toku (flow control) a předcházení zahlcení (congestion control) v TCP, uvést konkrétní příklad, jak může dojít k zahlcení a jak se s tím TCP vypořádá
  - Vysvětlit, jak ovlivňuje dynamický překlad adres (NAPT) obsah hlaviček TCP a UDP paketů
- Aplikační rozhraní a abstrakce pro síťovou komunikaci
  - Napsat pseudo-kód části síťové aplikace (klient nebo server) řešící zadaný problém s použitím socket API (např. statický web server)
  - Na příkladu vysvětlit jednotlivé části URI/URL a jejich význam
  - Popsat princip fungování systému DNS a DNS protokolu (základní typy a význam záznamů, zóny, DNS servery, vyřizování DNS dotazů, bezpečnostní rizika)
  - Popsat jednotlivé (síťové) kroky (např. překlad doménového jména na IP adresu, navázání spojení na konkrétních portech, odeslání a příjem zpráv s konkrétním obsahem, ...), které musí dobře znát síťová aplikace (prohlížeč, mailový klient, ...) provést, když vykonává svou činnost (zobrazuje stránku, odesílá mail, ...)
- Zabezpečení komunikace, autentizace, šifrování
  - Popsat jednoduché metody detekce poškozených bloků (kontrola parity, kontrolní součty) s uvedením konkrétních příkladů na různých vrstvách
  - Popsat principy a typy firewallů (co řeší a jak se konfiguruje)
  - Na konkrétním příkladu problému (a konfigurace sítě) doporučit vhodné zabezpečení (nastavení firewallů, demilitarizovaná zóna, aplikační brány)
  - Popsat základní principy symetrického a asymetrického šifrování, význam použití certifikátů, zabezpečení síťové komunikace SSL/TLS

#### Předměty

- NSWI090 Počítačové sítě (1/3 ze 3 kr)
- NSWI141 Úvod do počítačových sítí (1/3 ze 3 kr)

#### 4. Překladače a programovací jazyky

- Lexikální analýza
  - Použít regulární výraz pro definici lexikálního elementu
  - Použít stavy lexikálního analyzátoru (např. flex) pro definici složitějších lexikálních elementů (např. řetězec s escape-sekvencemi)
- Syntaktická analýza
  - Transformovat opakování a alternativy (např. v syntaktických diagramech) na bezkontextovou gramatiku
  - Zapsat gramatikou typické konstrukce programovacích jazyků
  - Odstranit nejednoznačnosti v gramatice (např. if-then-else)
  - Odstranit LL(1) konflikty v gramatice (např. levá rekurze)
- Sémantická analýza
  - Definovat pomocí atributů propojení lexikální a syntaktické analýzy
  - Použít syntetizované atributy (např. bison) pro předání mezivýsledků (např. při vyhodnocování výrazu)
  - Použít atributovou gramatiku (včetně dědičných atributů) pro řešení složitějších konstrukcí
- Mezikód
  - Zkonstruovat control-flow graf pro danou proceduru (v C++ nebo v C#/Java)

- Přepsat úsek zdrojového kódu (v C++ nebo v C#/Java) do zvoleného mezikódu

## Předměty

- NSWI098 Principy překladačů (1/2 ze 6 kr)

Některé následující body definují varianty požadavků pro různé individuální volby povinně volitelných předmětů. Vyžaduje se zvládnutí všech bodů bez označení ☹ a zvládnutí všech bodů s označením ☹ pro jeden z jazyků C#, C++ nebo Java. Dále tato specializace vyžaduje zvládnutí společných požadavků z informatiky v sekci Programovací jazyky ve stejné variantě.

- Pokročilé využití generických typů a metod
  - ☹ Vysvětlit fungování generických typů a metod s typovým omezením (constraints) v C# nebo Javě
  - ☹ Umět vhodně implementovat generické typy a metody s typovým omezením v C# nebo Javě
  - ☹ Umět implementovat šablony (templates) typů a funkcí, a vhodně využít konceptu duck-typingu v C++
  - ☹ Využít parciální specializaci na vhodných příkladech v C++
  - ☹ Aplikovat perfect-forwarding a variadické šablony v C++
- Funkcionální prvky
  - Lambda výrazy, closure, a variable capture
    - ☹ Vysvětlit fungování type inference pro parametry a návratovou hodnotu lambda výrazů v C#, C++ nebo Javě
    - ☹ Vysvětlit princip fungování a rozdíly mezi capture-by-copy, capture-by-reference, a no-capture sémantikami v C++
      - Na příkladu umět zvolit vhodný způsob variable capture
    - ☹ Vysvětlit princip fungování, výhody a nevýhody capture-by-“move-to-closure” sémantiky v C#
      - Umět ukázat nutnost ruční emulace capture-by-value
    - ☹ Vysvětlit princip fungování variable capture a potřebu final nebo “effectively final” zachycených proměnných v Javě
  - Paralelní a asynchronní programování
    - Vysvětlit výhody a nevýhody thread a task-based concurrency
      - Vysvětlit princip thread poolu
      - ☹ Vhodně využít typů Task, Task<T> a ThreadPool místo vláken v C#
      - ☹ Vhodně využít tasků pomocí std : : async místo vláken v C++
      - ☹ Vhodně využít exekutory a fork/join pool místo vláken v Javě
    - ☹ Vysvětlit koncepty future a promise a jejich využití v asynchronním programování v C#, C++ nebo Javě
      - ☹ Umět implementovat a využívat asynchronní metody vracející svůj budoucí výsledek ve formě future v C#, C++ nebo Javě
      - ☹ Umět implementovat asynchronní kód, který si drží informaci o slíbeném výsledku ve formě promise, a umí tento promise naplnit v C#, C++ nebo Javě
    - Vysvětlit koncept coroutine a možnosti jeho využití
      - Vysvětlit v kontextu coroutine princip a možnosti využití kooperativního multitaskingu a vysvětlit jeho výhody a nevýhody oproti preemptivnímu multitaskingu
      - ☹ Umět implementovat coroutine a použít v zadaném příkladu připravenou coroutine v C++ nebo Javě
      - ☹ Využít iterátorových metod pro implementaci coroutines a využít takové coroutines na příkladu v C#
      - ☹ Využít asynchronní metody s await pro implementaci coroutines a využít takové coroutines na příkladu v C#
    - Synchronizační primitiva
      - Definovat sémantiku a vysvětlit princip použití podmínkové proměnné, resp. monitoru
      - ☹ Využít podmínkové proměnné, resp. monitoru, v C#, C++ nebo Javě pro řešení základních synchronizačních problémů (např. producent-konzument)
      - Definovat sémantiku a vysvětlit princip použití semaforu
      - ☹ Využít semaforů v C#, C++ nebo Javě pro řešení základních synchronizačních problémů (např. producent-konzument)
  - Významné prvky standardních knihoven a jejich aplikace
    - Základní síťování pomocí socketů
      - ☹ Umět implementovat klientskou aplikaci pomocí TCP socketů v C#, C++ nebo Javě
      - ☹ Umět implementovat serverovou aplikaci pomocí TCP socketů v C#, C++ nebo Javě
      - ☹ Využít na jednoduchém příkladu UDP sockety pro komunikaci mezi aplikacemi v C#, C++ nebo Javě
    - ☹ Využít koncept asynchronního programování v tvorbě síťových aplikací v C#, C++ nebo Javě

## Předměty

Podle volby povinně volitelných předmětů

- NPRG051 Pokročilé programování v jazyce C++ (1/2 z 5 kr), nebo
- NPRG038 Pokročilé programování v jazyce C# (1/2 z 5 kr), nebo
- NPRG021 Pokročilé programování v jazyce Java (1/2 z 5 kr)

## 5. Návrh a tvorba software

- Principy objektového návrhu
  - Abstrakce, zapouzdření, SOLID principy a jejich aplikace, dekompozice
  - Předpokládá se schopnost identifikovat porušení principů v návrhu a navrhnout změny vedoucí k nápravě
- Návrh API, tříd a metod
  - Rozhraní aplikací (API) a poskytovatelů služeb (SPI), evoluce rozhraní
  - Návrh tříd, dědičnost a kompozice, immutability
  - Návrh metod, zohlednění účelu/zodpovědnosti, funkční dekompozice
  - Testovatelnost tříd a metod, návrh pro testovatelnost
- Návrhové vzory
  - Identifikace principů objektového návrhu v návrhových vzorech
  - Struktura, záměr (intent) a aplikace základních návrhových vzorů

## Předměty

- NPRG043 Doporučené postupy v programování (1/2 z 5 kr)
- Paralelní programování, paměťový model.
- Atomické operace a neblokující datové struktury.

Pro detailnější informace o tomto tématu kontaktujte garanta specializace.

## Předměty

- NPRG042 Programování v paralelním prostředí (1/3 ze 6 kr)
- Správa verzí
  - Popsat účel a použití systémů pro správu verzí při vývoji rozsáhlého software a práci v týmech (typicky nabízené funkce těchto systémů a jejich použití pro řešení běžných situací)
  - Popsat běžné způsoby integrace verzovacích systémů s nástroji pro správu projektů
  - Vysvětlit hlavní koncepty (lokální a vzdálené repozitáře, pracovní kopie (working copy), commit, větve)
  - Popsat a použít operace `clone`, `pull`, `push`, `add`, `diff`, `commit`, `branch`, `merge`, `log`, `blame`
  - Nástroje `git`
  - Popsat koncept feature branch v kontextu procesu vývoje a udržování software
  - Vysvětlit konflikty mezi verzemi, důvody vzniku a způsoby řešení
- Systémy pro sestavování software
  - Popsat hlavní účel a typicky nabízené funkce
  - Vysvětlit hlavní koncepty (cíl, závislosti, akce)
- Nástroje pro testování software
  - Popsat hlavní koncepty (test, pokrytí, jednotkové testování, integrační testování, systémové testování, black box, white box)
  - Popsat a na ukázkách kódu identifikovat typickou strukturu testu (set up, tear down, execute and validate)
  - Vysvětlit omezení běžných testovacích postupů a základní možnosti řešení
  - Popsat způsoby automatizace testování

## Předměty

- NSWI154 Nástroje pro vývoj software (1/4 ze 2 kr), nebo
- NSWI177 Úvod do Linuxu (1/5 ze 3 kr)

Potřebné znalosti a dovednosti jsou rozvíjeny také v rámci předmětů věnovaných programovacím jazykům a dalším předmětů, u kterých je tvorba software důležitým aspektem hodnocení.

# Umělá inteligence

Studenti této specializace si při přihlášení ke státní zkoušce vyberou ke společným tématům ještě zaměření Robotika, Strojové učení nebo Zpracování přirozeného jazyka. Téma 1 je společné všem zaměřením, téma 2 je určené pro zaměření Robotika, téma 3 pro zaměření Strojové učení, téma 4 pro zaměření Zpracování přirozeného jazyka.

## 1. Základy umělé inteligence

- Řešení úloh prohledáváním
  - formulace úlohy
  - stromové vs. grafové prohledávání
  - neinformované prohledávání (DFS, BFS, uniform-cost search)
  - informované prohledávání (algoritmus A\*, přípustné a konzistentní heuristiky)
- Splňování omezujících podmínek
  - problém splňování podmínek
  - hranová konzistence (algoritmus AC-3)
  - algoritmus hledání řešení (MAC)
- Logické uvažování
  - základy výrokové logiky (konjunktivní a disjunktivní normální forma)
  - algoritmus DPLL
  - dopředné a zpětné řetězení (Hornovské klauzule)
  - rezoluce
- Pravděpodobnostní uvažování
  - základy teorie pravděpodobnosti (úplná sdružená distribuce, nezávislost, Bayesovo pravidlo)
  - pravděpodobnostní uvažování (vysčítání, normalizace)
  - Bayesovské sítě
    - konstrukce a vztah k úplné sdružené distribuci
    - exaktní odvozování (enumerace, eliminace proměnných)
    - aproximační odvozování (Monte Carlo, zamítání, vážení věrohodností)
- Rerezentace znalostí
  - situační kalkulus, problém rámce
  - Markovské modely
    - filtrace, predikce, vyhlazování, nejpravděpodobnější průchod
    - skryté Markovské modely (HMM) vs. dynamické Bayesovské sítě
- Automatické plánování
  - formulace plánovacího problému (definice operátoru)
  - dopředné a zpětné plánování
- Markovské rozhodovací procesy (MDP)
  - formulace problému (výpočet užítku, strategie)
  - Bellmanova rovnice
  - iterace hodnot, iterace strategií
  - POMDP (základní definice)
- Hry a teorie her
  - algoritmy Minimax a alfa-beta prořezávání
  - základy teorie her (věžňovo dilema, Nashovo ekvilibrium)
  - mechanism design (typy aukcí)
- Strojové učení
  - základní druhy učení (s učitelem, bez učitele, zpětnovazební)
  - rozhodovací stromy (definice, konstrukce)
  - regrese, SVM (základní principy)
  - Bayesovské učení, EM algoritmus
  - zpětnovazební učení
    - pasivní učení (definice, metody ADP a TD)
    - aktivní učení (definice, explorace vs. exploitace, Q-učení, SARSA)

## Předměty

- NAIL120 Úvod do umělé inteligence (5 kr)

## 2. Robotika

- Kinematika
  - pohyb a transformace
    - 2D, 3D translace, rotace a obecné transformace
    - Kartézské a polární souřadnice
    - Homogenní transformace
  - řešení základních úloh
    - jednoduché transformace
    - skládání transformací
    - Denavit-Hartenberg systém
- Řídicí systémy
  - architektury a jejich hierarchie
    - reaktivní, deliberative, subsumption, behaviorální architektury, třívrstevný systém
  - implementace
    - monolytický systém
    - stavové automaty
    - middleware
  - specifická běhová prostředí
    - nízko- a vysokoúrovňové systémy
    - programování pro prostředí s omezenými prostředky
- Pohyb, senzorka
  - způsob pohybu
    - pozemní kolové a pásové systémy a jejich řízení (diferenciální, Ackermannovo, varianty všesměrových systémů)
    - létající systémy
  - základní typy aktuátorů a senzorů
    - elektrické motory
    - kontaktní a bezkontaktní senzory
  - zpětnovazební řízení, PID
  - zpracování vstupních dat, filtrace.
- Lokalizace a mapování
  - způsoby určování polohy
    - absolutní a relativní
    - pasivní a aktivní
    - odometrie, triangulace a trilaterace, satelitní systémy
    - pravděpodobnostní přístup
  - typy map
    - metrické, geometrické, topologické, mřížka obsazenosti
    - úrovně abstrakce
  - volba použití v modelových situacích
  - simultánní lokalizace a mapování
- Zpracování obrazu a počítačové vidění
  - segmentace obrazu
  - vyhledávání a sledování objektů

## Předměty

- NAIL028 Úvod do robotiky (5 kr)
- NPGR036 Počítačové vidění (5 kr)
- NPRG037 Programování mikrokontrolerů (5 kr)

## 3. Strojové učení

- Učení s učitelem
  - klasifikace, regrese (základní typy úloh)

- standardní evaluační metriky
- ohodnocení modelu (testovací data, křížová validace, maximální věrohodnost)
- přeučení a regularizace (generalizační chyba, včasné zastavení trénování, L2 a L1 regularizace)
- prokletí dimenzionality
- Učení založené na příkladech
  - algoritmus k-nejbližších sousedů
- Lineární regrese
  - analytické řešení metodou nejmenších čtverců
  - trénování pomocí stochastic gradient descent
- Logistická regrese
  - binární klasifikace (sigmoid, metody trénování)
  - klasifikace do více tříd (softmax, metody trénování)
- Rozhodovací stromy
  - algoritmus učení a kritéria větvení
  - prořezávání
- Metoda podpůrných vektorů
  - klasifikátor pro lineárně separabilní třídy
  - klasifikátor pro lineárně neseperabilní třídy
  - jádrové funkce
  - klasifikace do více tříd
- Kombinace více modelů
  - bagging a boosting
  - metoda náhodných lesů
- Statistické testy
  - studentův t-test (jednovýběrový a dvouvýběrový)
  - chí-kvadrát test (test dobré shody)
- Učení bez učitele
  - shlukování (algoritmus k-means)
  - hierarchické shlukování
  - redukce dimenzionality (analýza hlavních komponent)

#### Předměty

- NAIL121 Seminář dobývání znalostí (4 kr)
- NPFL054 Úvod do strojového učení v systému R (5 kr)
- NPFL129 Úvod do strojového učení v Pythonu (5 kr)
- NPGR035 Strojové učení v počítačovém vidění (5 kr)

#### 4. Zpracování přirozeného jazyka

- Roviny popisu jazyka, morfologická a syntaktická analýza
  - soustava rovin popisu jazyka, vztahy mezi rovinami
  - morfologická analýza, lemmatizace, slovní druhy a další morfologické kategorie, sady morfologických značek, příklady algoritmů pro statistický tagging
  - formální popis syntaxe přirozených jazyků, složkový a závislostní přístup
  - syntaktická analýza, příklady algoritmů pro statistický parsing
- Základy teorie pravděpodobnosti a teorie informace
  - definice pravděpodobnosti, základní vlastnosti
  - statistická nezávislost
  - Bayesova věta
  - entropie
  - vzájemná informace
  - Kullbackova–Leiblerova divergence
- Statistické metody zpracování přirozeného jazyka, jazykové modely
  - n-gramový jazykový model
  - odhad jazykového modelu metodou maximální věrohodnosti
  - vyhlazování jazykového modelu, využití, příklady metod
- Strojové učení, klasifikace, regrese
  - řízené a neřízené strojové učení
  - konstrukce ztrátových funkcí pomocí metody maximální věrohodnosti (NLL, MSE)

- úloha klasifikace, klasifikační algoritmy (rozhodovací stromy, logistická regrese, metoda podpůrných vektorů - SVM, perceptron, vícevrstevný perceptron - MLP)
- příklady využití klasifikace ve zpracování přirozeného jazyka
- úloha regrese, regresní algoritmy (rozhodovací stromy, lineární regrese)
- kombinace více modelů (voting, ensembling, boosting)
- shluková analýza (k-means)
- Odhad generalizační chyby, přetrénování, regularizace
  - vyhodnocení úspěšnosti v klasifikačních a regresních úlohách (RMSE, accuracy, confusion matrix, precision, recall, F1 skóre, Fbeta skóre, precision-recall křivka, specificita, sensitivita, ROC křivka)
  - problém přetrénování, možné příčiny, možná řešení
  - příklady regularizace v lineární a logistické regresi, regularizační členy L1 a L2
- Vektorové reprezentace slov, základy hlubokého strojového učení
  - základní techniky pro trénování word embeddings (skipgram, CBOW)
  - perceptron, vícevrstevný perceptron (aktivační funkce skrytých a výstupních vrstev), stochastické gradientní klesání
  - Universal approximation theorem
  - konvoluční sítě a základní architektury pro jejich využití ve zpracování přirozeného jazyka
  - rekurentní sítě a základní architektury pro jejich využití ve zpracování přirozeného jazyka
- Aplikace zpracování přirozeného jazyka, příklady evaluačních měř
  - získávání informací, invertovaný index, model bag-of-words, TF-IDF,
  - statistický strojový překlad (phrase-based)
  - neuronový strojový překlad
  - využití precision, recall a f-score ve zpracování přirozeného jazyka
  - příklady aplikačně specifických evaluačních měř (BLEU, UAS apod.)
  - křížová evaluace

### **Předměty**

- NPFL012 Úvod do počítačové lingvistiky (3 kr)
- NPFL054 Úvod do strojového učení v systému R (5 kr)
- NPFL124 Zpracování přirozeného jazyka (4 kr)
- NPFL129 Úvod do strojového učení v Pythonu (5 kr)

## Historie změn požadavků

Všechny významné změny požadavků jsou dokumentovány v této sekci.

### 2025-07-28 (používané od září 2025)

- doplnění požadavků na počítačové sítě v Programování a vývoji software
- zpřesnění formulací v Programování a vývoji software u Programovacích jazyků, Softwarového inženýrství a Databází

### 2024-12-18

- opravy překlepů v Obecné informatice
- zpřesnění formulací u vektorových prostorů v Základech matematiky
- zrušeny požadavky týkající se Hadwigerovy hypotézy a horního odhadu počtu hran grafových minorů v Obecné informatice
- doplnění celočíselných toků a jejich aplikací a zrušení červeno-černých stromů a transitivních uzávěrů v Obecné informace

### 2024-01-05 (používané od února 2024)

#### Rozšíření

- doplnění požadavků na povinnou znalost programovacího jazyka C++ v Programování a vývoji software

#### Ostatní

- reorganizace požadavků u programovacích jazyků v Systémovém programování
- zpřesnění formulací k variantám týkajícím se volby programovacích jazyků v Základech informatiky, Systémovém programování a Programování a vývoji software

### 2023-08-12 (používané od září 2023)

#### Ostatní

- zpřesnění formulací u konceptů týkajících se dědičnosti v Základech informatiky
- zpřesnění formulací u vyhodnocení klasifikace v Umělé inteligenci

### 2022-12-08 (používané od února 2023)

#### Rušení

- zrušeny požadavky týkající se počítačových sítí v Obecné informatice

#### Ostatní

- zpřesnění formulací k variantám týkajícím se volby programovacích jazyků