# Jarní soustředění KSP 2023 – Seznam přednášek

Tento spisek jest nabídkou přednášek, které byste na soustředění mohli slyšet, čili jakási obdoba matfyzácké Karolínky (ta je ale, pravda, ještě stále o něco tlustší). Přednášek je daleko víc, než kolik se dá za pár dní stihnout, a tak je na vás, abyste si vybrali, o které máte opravdu zájem. Pokud byste rádi slyšeli ještě o něčem dalším, klidně si o to napište (např. na Discord), třeba se najde někdo, kdo by vám o tom rád pověděl. Berte a vychutnávejte!

Přednášky jsou členěné do třech kategorií. Na ty, o kterých si myslíme, že jsou základní a esenciální pro každého programátora, na přednášky pokročilejší, dotýkající se zajímavých oblastí informatiky, a nakonec na přednášky půlnoční, které často rozšiřují obzory až za rámec informatiky. Berte a vychutnávejte!

Údaje o jedné přednášce vypadají asi takto:

**Stručný úvod do základů teorie vlkodlaků** *("Za dne ukryt v hloubi lesa, děs temný zvečera se plazí...")* **LYK**
*RNDr. Á. Cula*
Úvod do moderní teorie vlkodlaků, čili též praktická dæmonologie a naiadologie.
*Předpoklady: Měsíc v úplňku.*

Dozvíte se (čteno v obvyklém pořadí): jméno přednášky, v uvozovkách motto přednášky, kód (pro snadnější odkazování na konkrétní předměty), jméno přednášejícího (nebo nabídku možných přednášejích, pokud je zde více jmen) a nakonec stručný obsah přednášky. Pokud u přednášky není uveden žádný přednášející, umí ji přednést většina přednášejících a jen vás nechceme unavovat přehlídkou našich jmen :-). Hvězdičky znamenají obtížnost.

# Základní přednášky

Mezi těmito přednáškami jsou věci, které by měl každý začínající programátor umět. Bez pochopení většiny věcí přednášených na těchto přednáškách se budete na pokročilých přednáškách, které na ně navazují, jen obtížně chytat. Doporučujeme proto nejdříve zvládnout tyto přednášky a osvěžit si nějaký základní programovací jazyk, než se pustíte do pokročilejších věcí.

**Základy programování** *("Má $x = x + 1$ řešení?")* **ZAKL**
Úvodní trojdílná přednáška pro ty, kteří mají s programováním jen malé, nebo dokonce žádné zkušenosti. Vysvětlíme si od základů problematiku programování, jako je zápis cyklů, podmínek a funkcí, ukážeme si základní datové typy (n-tice, seznamy, slovníky), datové struktury (fronta, zásobník) a zkusíme si prakticky naprogramovat několik základních algoritmů. Vše se bude ukazovat hlavně na jazyku Python, který je jednoduchý na naučení a přesto zároveň velmi mocný. Jednotlivé přednášky se budou prolínat s přednáškami ZALG.

**Základy algoritmizace, složitosti a datových struktur** *("Co by měl každý programátor znát.")* **ZALG**
*David Klement*
Základní vícedílný kurz algoritmů a datových struktur, který se bude prolínat se ZAKL. Jak poznat, který algoritmus je efektivnější? Přehled základních algoritmů. Co je to datová struktura a několik jejích ukázek. Vše si procvičíme na příkladech.

**Grafy & algoritmy** *("Stromy, listy, lesy, pařezy, cesty, kružnice...")* **GA**
*Martin Koreček, Kiki Prokopová, Ondra Sladký, Honza Černý, Jirka Kalvoda*
Spousta algoritmických problémů se dá popsat pomocí teorie grafů. Ukážeme si její základy: co je to graf, jak se dá v programu reprezentovat a k čemu se dá použít. Naučíme se hledat nejkratší cestu v bludišti nebo na mapě.

**Dynamické programování** *("Kampak jsem si to jenom schoval?")* **DYNP**
*Kiki Prokopová, Ondra Sladký, Petr Budai*
Dynamické programování je programátorská technika využívající velice prostinkého nápadu: Proč něco počítat několikrát, když to mohu spočítat jednou a výsledek si uložit? Na této přednášce si ukážeme, že tento jednoduchý nápad může pomoci efektivně vyřešit i poměrně obtížné úlohy.

## Základní programovací jazyky a techniky

**Principy počítačů** *("A opravdu uvnitř počítače běhají malí trpaslíci?")* **HW**
*Lucka Vomelová, David Klement, Ondra Machota, Jirka Kalvoda, Jirka Setnička, Petr Budai*
Ukážeme si, proč programy fungují tak, jak jsme zvyklí. Co umí procesor, co dělá paměť a jak se to dá k něčemu použít. Ukážeme si nějaký program v Céčku a v Asembleru a koukneme se, kolik toho řeší Python za nás. Co dělá operační systém, jak je třeba možné, že na jednom procesoru běží najednou několik procesů. Ukážeme si, že počítače jsou překvapivě hloupá stvoření, co umí jenom základní počty, ale na programování nám to stačí.

**Python** **PYTH**

*David Klement, Jirka Kalvoda, Petr Budai*

Úvod do Pythonu pro ty, kteří již umí programovat v jiném jazyce. V čem se liší od ostatních jazyků a proč se v něm píše tak snadno. Proč se překládá až při spuštění, jaké výhody a jaké nevýhody to s sebou nese. Letmý úvod do balíčků aneb skoro všechno již někdo napsal za nás.

# Pokročilé přednášky

Tyto přednášky by měly jednak dále rozvíjet znalosti ze základních přednášek, ale také nabízet další zajímavé programátorské techniky a technologie, které se mohou každodenně hodit.

## Algoritmizace

**Intervalové stromy *** *("Já bych ty intervaly nejradši... dal do stromu!")* **ITREE**

*Ondra Sladký, Dan Skýpala, Jirka Kalvoda*

Intervalový strom je datová struktura pracující s intervaly, se kterou se můžeme setkat v mnoha úlohách (zejména soutěžních). Řekneme si, co to intervalový strom je, jejich použití si ukážeme na úlohách. Dáme se můžeme zabývat tím, jak je upravovat, aby toho zvládly ještě víc.

**Hledání v textu** *("»Vyšíváme v seníku!« – kde jsem to jen viděl?")* **TEXT**

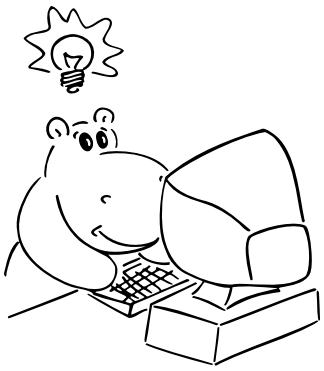
*Kiki Prokopová, Ondra Sladký, Honza Černý, Jirka Kalvoda*

Někdy potřebujeme najít podřetězec ve velkém množství textu. Jak to udělat co nejrychleji? K tomu se nám budou hodit vyhledávací automaty, například Knuthův-Morrisův-Prattův algoritmus. Od toho dojdeme k dalším algoritmům na zpracování textu.

**Datové struktury pro pokročilé *** *("Pojďme na procházku binárním lesem")* **DS**

*Dan Skýpala, Martin „Medvěd" Mareš, Jirka Kalvoda, Jirka Setnička*

Přehled šikovných datových struktur, které se nevešly do ZALG. Vyhledávací stromy a různé způsoby jejich vyvažování a „ozdobení". Hešování aneb hledáme v téměř konstantním čase. Líné datové struktury a amortizovaná složitost.



**Těžké problémy *** **HARD**

*Martin Koreček, Dan Skýpala, Jirka Kalvoda*

V rámci této přednášky se budeme zabývat problémy tak těžkými, že nikdo na světě pro ně neumí vymyslet efektivní (rozuměj polynomiální) algoritmus. Spousta lidí dokonce věří, že to vůbec možné není. Abychom mezi tyto problémy pronikli, seznámíme se s pojmy NP-úplnosti a NP-těžkosti. Především si však konkrétní těžké úlohy ukážeme a naučíme se i některé těžké úlohy rozpoznat. Závěrem si řekneme, jak se s těžkými úlohami vypořádat v praxi.

**Amortizace** *("Celek bývá daleko menší než součet částí.")* **AMORT**

*Martin Koreček, Ondra Sladký, Dan Skýpala, Jirka Kalvoda*

Spousta algoritmů je mnohem rychlejší, než jak na první pohled vypadají. Šikovný způsob, jak takové chování zkoumat, je amortizovaná časová složitost. Předvedeme několik trochu překvapivých příkladů amortizace: dvojková a jiná počitadla, datové struktury založené na přebudovávání, vyhledávací stromy bez otravného vyvažování, dynamizace datových struktur, udržování historie.

## Další programovací jazyky a techniky

**SQL databáze** *("SELECT something FROM knowledge LIMIT 90min")* **SQL**

*Maruška Kalousková, Martin „Medvěd" Mareš, Jirka Setnička, Standa Lukeš*

Představíme si SQL, jazyk databází. Ukážeme si základní příkazy i práci o kus složitější. Jak databázi dát data a jak se na ně potom ptát. K čemu se hodí složený dotaz a klíčové slovo JOIN. Jak si data skupinkovat pomocí GROUP. Co jsou to vlastně ty transakce když mluvíme o databázích a proč je používat.

**Programování v jazyce C** **C**

*Martin „Medvěd“ Mareš, Jirka Kalvoda, Jirka Setnička*

Jazyk C patří k nejrozšířenějším jazykům, hodí se pro low-level programování i kusy kódu, které mají zejména být rychlé. Představíme si datové typy a běžné programové konstrukce, vysvětlíme si základy práce s ukazateli a také se seznámíme se standardními knihovnami jazyka C.

**Procesy a vlákna *** *(“Koupil jsem dalších 15 procesorů, proč je to stále stejně pomalé?”)* **THREAD**

*Jirka Setnička, Standa Lukeš*

Jak vypadá víceprocesorový či vícejádrový počítač a co to znamená pro programátora? Procesy, vlákna a úskalí komunikace mezi nimi aneb jak se stejným kusem paměti může pracovat více procesů. Synchronizační primitiva: mutexy, semafory, podmínkové proměnné. Spinlocky, deadlocky a livelocky. Jde to i bez synchronizace: atomické operace, transakční paměť. Které jazyky nám pomáhají a které spíš škodí. Kdy je lepší vlákna použít, a kdy ne.

*Předpoklady: Trochu představy o hardwaru*

## Počítače, sítě, systémy

**Webové stránky** **WWW**

*Jirka Kalvoda, Standa Lukeš*

Co se děje za oponou, když do prohlížeče zadáte adresu svých oblíbených stránek? A jak si takovou stránku taky pořídit? Přelet nad protokolem HTTP, seznámení s HTML a předvedení kaskádových stylů. Jak fungují dynamické stránky a jaký je rozdíl v tom, jestli něco běží na serveru nebo v prohlížeči.

**Sítě a Internet** *(“Od jednoho drátu k živoucímu ekosystému”)* **NET**

*Lucka Vomelová, Martin „Medvěd“ Mareš, Jirka Setnička, Petr Budai*

Jak funguje Internet a počítačové sítě vůbec: od elektronů v drátech (fotonů v optických kabelech nebo elektromagnetických vln) přes komunikaci na jedné malé síti až ke komunikaci v celém Internetu. Vysvětlíme si rámce, pakety, MAC a IP adresy, routování v malých i ve velkých sítích. Jak to reálně funguje s IPv4 a NATem, co to jsou porty a jak se od sebe liší TCP a UDP. A na závěr radosti a strasti IPv6 (až ho konečně zavedeme).

**Sítě II – protokoly a síťové útoky** *(“Jak si přečíst maily… sousedovy maily.”)* **NET2**

*Lucka Vomelová, Martin „Medvěd“ Mareš, Jirka Setnička*

Volné navázání na NET aneb máme fungující síť a chceme nad ní provozovat složitější komunikaci. ICMP aneb servisní protokol Internetu, DNS a překlad doménových jmen, jednoduché textové protokoly jako je FTP, SMTP, IMAP nebo nejpoužívanější webové HTTP, kterého se zastavíme trochu déle – hlavičky, návratové kódy, cookie, více domén na stejné IP adrese. A pokud zbude čas, využijeme zranitelnosti některých protokolů a provedeme síťový útok.

*Předpoklady: Základní povědomí o počítačových sítích v rozsahu NET*

**Linux pro uživatele** *(“Linux gives you enough rope to hang yourself.”)* **LINUX**

*Martin „Medvěd“ Mareš, Jirka Kalvoda*

Operační systém Linux si původně vyrobili programátoři pro sebe, a dodnes to na něm je vidět. Nenabízí tolik uživatelského pozlátka, ale dá se s ním pracovat mnohem efektivněji. Pojďme ho trochu prozkoumat. Zjistíme, že je to stavebnice složená ze spousty malých kousků, které dělají jednoduché věci a dají se kombinovat. Ovládají se pomocí příkazů, což bývá často rychlejší než klikátka. Také můžete mít svůj systém pod kontrolou a přesně vědět, co se děje uvnitř.

**Linux pro správce serveru** *(“Printer is on fire???”)* **LSERV**

*Martin „Medvěd“ Mareš, Jirka Setnička*

Jak vytvořit jednoduchý Linuxový server, který poskytuje služby vaší domácnosti, nebo třeba nějaké větší síti. Co se tam hodí provozovat? Povíme o SSH, klíčích, šifrování, systemd, Apache a Nginxu, nastavení mailového serveru i DNS. Jak server zabezpečit před útočníky, jak před ztrátou dat a jak před uklízečkou. Vše si vyzkoušíme prakticky, třeba na virtuálním počítači.

*Předpoklady: Základní znalost Linuxu.*

**Mikrokontroléry** *(“Nejlepší debugger je LEDka.”)* **MCU**

*Martin „Medvěd“ Mareš*

Srdcem mnoha dnešních technických hraček je mikrokontrolér. To je čip, na kterém je integrovaný nejen procesor, ale i paměť a spousta zajímavých periferií. Ukážeme si, jak se mikrokontroléry programují, jaké periferie typicky obsahují a jak je používat ke komunikaci s okolním světem. Něco si vyzkoušíme i prakticky na STM32.

*Předpoklady: Hodí se základní znalost jazyka C.*

## Vývoj software

**Systém pro správu verzí Git** *(“U svatýho tučňáka, kdo sem napsal tohle? Ono to tvrdí, že JÁ?!”)* **GIT**

*Maruška Kalousková, David Klement, Jirka Kalvoda, Jirka Setnička*

Když se něco vyvíjí delší dobu, přijde vhod nějaký nástroj, který by uměl zjistit kdo co přidal a proč, uměl by se vrátit k předchozí verzi nebo třeba vrátit jenom jednu změnu, co udělal kamarád před rokem. Na jeden takový, Git, se podíváme. Povíme si, jak Git ukládá změny, co jsou commity, větve, tagy a jak vypadá merge mezi větvemi. Nakonec možná předvedeme i nějaké užitečné triky: třeba hledání bugů půlením historie.

**Vývoj software** **DEV**
*Jirka Setnička, Standa Lukeš*
Výroba software není zdaleka jenom o programování. Pokud chceme vyvíjet větší kus softwaru ve více lidech, pojí se s tím spoustu věcí – verzování, merge requesty, code review, testování a mnoho dalšího. Ukážeme si, jak může vypadat vývoj něčeho většího (ať už je to zadání KSPčka nebo třeba software pro tisíce serverů) a jaké zvyky je dobré si vypěstovat (a jaké ne). Povíme si o různých způsobech testování, o tom, jak udržet v kódu pořádek, a o dalších nástrojích, které pomáhají vyvíjet kvalitní software.

**UX Design aneb Tvorba uživatelského rozhraní** *("Ale mě názor nějakého uživatele přece vůbec nezajímá.")* **UXD**
*Maruška Kalousková*
Návrh uživatelského rozhraní není jen o vzhledu, ale i o tom, jak navrhnout aplikaci, která bude užitečná, efektivní a použitelná. A nejenom aplikaci, ale třeba i tlačítka na přivolání výtahu. A že to přece není nic složitého? Ukážeme si pár případů špatného návrhu uživatelského rozhraní, pár příkladů těch dobrých, a pak se sami vrhneme na návrh - stránky na převod měn, mechanické injekční stříkačky pro nemocnice… - a prodiskutujeme si spolu, co je na nich dobře a co špatně. A možná trochu zabrousíme i do Figmy - takového užitečného a docela jednoduchého nástroje pro návrh rozhraní.

**Textový editor Vim** *("Víš, jaký je nejlepší textový editor? Vim.")* **VIM**
*Martin „Medvěd" Mareš, Jirka Kalvoda*
Odložme na chvíli své myše a pojďme si vyzkoušet textový editor, který umí poslouchat na slovo. Pravda, budeme se ta slova muset chvíli učit, ale výsledek bude proklatě efektivní. Základní příkazy, práce s regulárními výrazy, makra, kouzla. Vimovité ovládání jiných programů, třeba webového prohlížeče.

## Aplikace informatiky a matematiky

**TEX** *("No pages of output. Ask a TEXnician.")* **TEX**
*Martin „Medvěd" Mareš, Jirka Kalvoda*
Donald E. Knuth napsal TEX před desítkami let proto, že mu nikdo nebyl schopen vysázet matematický text podle jeho požadavků. Od té doby se hojně používá pro sazbu nejrůznějších publikací. V této spíše praktické přednášce si ukážeme použití TEXu od hladké sazby knihy až po zběsilosti hraničící s programováním. Pozornost věnujeme i zdrojům informací a rozdílům mezi různými dialekty TEXu.

**LATEX** *("TEX pro smrtelníky")* **LATEX**
*David Klement*
Píšete seminárku či řešení KSP a chcete, aby výsledek vypadal k světu? TEX je pro vás příliš složitý? Pak sáhněte po LATEXu. Ukážeme si, jak málo stačí k tomu, abychom zvládli vysázet hezký dokument.

**Kryptografie** *("Gbgb arav zbp gnwan mcenin.")* **CRYPT**
*Maruška Kalousková, Martin „Medvěd" Mareš*
Kryptografie se zabývá šiframi, jejich konstrukcí a zejména jejich luštěním. Začneme se symetrickými a asymetrickými šiframi a jednosměrnými funkcemi. Z nich pak vybudujeme složitější kryptografické protokoly na bezpečný přenos, autentikaci a digitální podpisy. Vymyslíme dokonce, jak si hodit korunou po telefonu, a také předvedeme nerozluštitelnou šifru a dokonce to o ní dokážeme.

**Čárové kódy** *("Jak naučit počítače číst láhve od Coly")* **BAR**
*Martin „Medvěd" Mareš*
Čárové kódy dnes potkáváme na každém kroku, ale jak doopravdy fungují? Prozkoumáme klasické jednorozměrné kódy (UPC, EAN, Code39, Code128), jakož i novější dvojrozměrné (QR, Aztec, DataMatrix). Kódovací a dekódovací algoritmy plus trocha matematiky okolo zabezpečení proti chybám. Další počítačem čitelné značky: RFID, bíle křížky na asfaltu, …

**Strojové učení** *("Ať si to ten algoritmus radši vymyslí samo.")* **ML**
*Standa Lukeš*
Prakticky si ukážeme jak v Pythonu použít algoritmy, které z příkladů samy nějak usoudí co mají dělat. Na co si dát při jejich používání pozor, co si od nich můžete slibovat a na co se tento přístup (ne)hodí. Pro představu se podíváme jak některé tyto algoritmy fungují.

**Neuronové sítě** **NEURO**
*Ondra Sladký, Jirka Setnička*
Základy neuronových sítí: od biologického neuronu a jeho modelu – perceptronu, přes algoritmus back propagation, až po hluboké a konvoluční sítě a jejich použití na rozpoznávání obrázků.

**Kontrola pravopisu** **SPELL**
*David Klement*
Jak se vyhnout překlepům? Jak zkontrolovat dlouhý dokument za zlomek sekundy? A jak se vypořádat s češtinou, která skloňuje a časuje? Představíme si program Hunspell, který umožňuje kontrolu pravopisu v libovolném jazyce a stačí mu k tomu překvapivě málo. Ukážeme si, jak do něj přidat svá vlastní slova, a následně se podíváme, jak funguje pod pokličkou.

**Zpracování dat** *("Bez práce nejsou koláč... ové grafy.")* **DATA**
*Martin „Medvěd" Mareš*
O světě jde sehnat spousta zajímavých dat ve strojově zpracovatelné podobě: obce a domy v nich, linky hromadné dopravy, katalogy hvězd, slova v češtině, katalog pokémonů, ... Pojďme se podívat, jak s daty zacházet. Naučíme se číst různé formáty dat od CSV až po XML, data zkoumat, filtrovat a kreslit podle nich pěkné grafy. Vyzkoušíme si prakticky v Pythonu. Předvedu své oblíbené nástroje, pojďte ostatním předvést ty své.

**Herní algoritmy** *("Když nemáte na to, abyste vyhráli šachový turnaj...")* **AIGAME**
*Jirka Setnička*
Povídání o tom, jak programovat počítačové soupeře do šachů a her jim podobným. Základní minimaxový algoritmus a jeho vylepšení neboli $\alpha$-$\beta$ ořezávání. Stále pomalé? Několik nápadů na efektivnější ořezávání. Ne u všech her však funguje hrubá síla (minimax) dobře, ukážeme si tedy, jak hru zanalyzovat.

## Matematické přednášky

**Grafy bez algoritmů** **GRAFY**
*Jirka Kalvoda*
Teorie grafů trochu teoretičtěji. Různé druhy grafů a jejich vlastnosti. Stromy a lesy. Kreslení grafů jedním tahem. Princip sudosti a skóre grafu. Jaké speciální vlastnosti mají rovinné grafy a jak je lze obarvit šesti nebo možná i pěti barvami. Jak poznat, že dva grafy (ne)jsou isomorfní. Mosty, artikulace a ušaté lemma. Párování, střídavé cesty a Hallova věta.

**Kombinatorika** *("Nemám rád faktoriály. Faktoriály nemám rád. Rád nemám faktoriály...")* **KOMB**
*David Klement, Martin „Medvěd" Mareš, Jirka Kalvoda*
Pokusíme se vybudovat kombinatoriku intuitivně. Co nejvíce se vyhneme počítání se vzorci, vystačíme si s elegantními úvahami. Kromě základních technik si ukážeme, jak nám mohou pomoci rekurence a jak se úlohy dají převádět mezi sebou. Procvičíme na spoustě příkladů.

**Matematická analýza *** *("Matfyzák se snaží integrovat do společnosti.")* **MA**
*David Klement*
V rychlosti si zavedeme derivace a integrály a následně si ukážeme, k čemu všemu se hodí. Hledání minima funkce, kleslení hladkých křivek, aproximace. Výpočet délky spirály nebo objemu roztodivných těles. Využití ve fyzice: pohyb s odporem vzduchu, zamrzání ledu na rybníce a mnoho dalšího.

**Průlet lineární algebrou** **LIN1**
*Honza Černý, Ondra Machota*
Podíváme se, čím se to ta lineární algebra vlastně zabývá. Řekneme si, co jsou matice, jak se s nimi počítá a k čemu jsou dobré. Seznámíme se s pojmy jako těleso, vektor a vektorový prostor, představíme si jejich zajímavé vlastnosti a uvedeme je do různých souvislostí.

**Logické programování** *("Mohu být svým vlastním dědečkem?")* **LOGP**
*Honza Černý, Standa Lukeš*
Což takhle projednou neříkat počítači, jak má věci počítat, ale jenom mu zadat podmínky, které má výsledek splňovat? Neprocedurální programování vychází přesně z této myšlenky. Podíváme se na programovací jazyk Prolog, který vychází z formální logiky. Zjistíme, které problémy se v něm neobyčejně zjednoduší a které naopak programování promění v noční můru. Pokud jsi milovník rekurze, budeš u této přednášky nejspíš skoro celou dobu spokojeně vrnět.

**Objektově orientované programování** *("Object-oriented system. If we change it, users object.")* **OOP**
*Honza Černý, Standa Lukeš*
Objektově orientované programování přináší jiný náhled na návrh řešení problémů. Vysvětlíme, jak se liší objektové a procedurální programování. Co je to objekt a co třída. Základní vlastnosti objektů (dědičnost, zabalení, polymorfismus). Co je to metoda, překrývání metod, virtuální metody (pozdní vazba) a čistě virtuální (abstraktní) metody. Jak se liší OOP ve statických (C++, C#, Java) a dynamických (Python) jazycích. Jak programovat objektově i bez podpory jazyka, třeba v Céčku.
*Předpoklady: Znalosti procedurálního programování, například v Pascalu, v Pythonu nebo v C.*

**Jazyky, gramatiky a automaty** **AUTO**
*Honza Černý, Jirka Kalvoda*
O jazycích přirozených, počítačových a matematických, jejich popisu a rozpoznávání. Začneme těmi nejjednoduššími: regulární jazyky a výrazy, konečné deterministické a nedeterministické automaty. Pak budeme stoupat po příčkách Chomského hierarchie, kam až to půjde. Jak výpočetně silný je třeba takový automat na kafe?

**Teorie informace *** *("Proč zazipovaný zip není menší než původní?")* **TEOINF**
*Ondra Chwiedziuk*
Zadefinujeme si, co je to informace, jak jí můžeme měřit a jaké to má důsledky. Z počátku se prokoušeme pravděpodobností, řekneme si, co je to entropie a jak souvisí s bitem informace. Následně si povíme, co je to kód a jak nejlépe můžeme zakódovat informaci. Vysvětlíme si Huffmannovo kódování a, pokud zbude čas, i aritmetický kód a Ziv-Lempelův rekurenční kód.

**Úvod do teorie čísel a RSA** *("Pošli mi zprávu, tady máš klíč.")* **RSA**
*Ondra Chwiedziuk*
Řekneme si, čím se zabývá teorie čísel a vysvětlíme si její nejznámější aplikaci, RSA. Vybudujeme teorii kolem dělitelnosti, řekneme si základní větu aritmetiky, vysvětlíme si Eukleidův algoritmus, Bézoutovu rovnost, Eulerovu funkci a důležité vlastnosti prvočísel. Když budeme znát všechny tyto pojmy, můžeme se pustit do vysvětlení, proč vlastně RSA funguje. Na závěr si řekneme slabiny RSA a proč už se dneska přestává používat.

# Půlnoční přednášky

Aneb přednášky přednášené (nejen) o půlnoci na různá zajímavá témata nejen o informatice. Pokud nějaká z nich nebude oficiálně vypsaná, je možné si konkrétního organizátora ve volné chvíli chytit a přesvědčit ho k přednášení.

**Lingvištika** *("Přísudek je v této větě podmět.")* **LING**
*Martin „Medvěd" Mareš*
Převážně nevážné a mírně nepřed-vídatelné po-vídání o jazyku i jazyce. Základní jazykové rodiny a jejich podobnosti i odlišnosti. Co má společného čínština s angličtinou a co nikoliv. Jak se jazyky vyvíjejí a jak se navzájem ovlivňují. Kde jsme přišli k pravidlům a jaký je jejich smysl. Existují synonyma? Proč je jazyk nejednoznačný a proč je to dobře. Jak se na jazyk dívá matematik a jak se na matematiku dívají lingvisté. Jak vzniklo písmo? A jak otazník? Jak zapsat zachrochtání a jak třeba mlasknutí &c.

**GPS** *("Čekám na signál. . . ")* **GPS**
*David Klement*
GPS je vstutku magická technologie, a ačkoli se to nezdá, využívá jedny z nejdůležitějších výsledků moderní vědy. Vysvětlíme si, na jakém pricipu GPS funguje, a proč by to jinak nešlo.

**Teorie množin** *("Jablka a hrušky se dají nejen sčítat, ale třeba i násobit.")* **TEMNO**
*Martin „Medvěd" Mareš*
Základoškolský přístup „množina je kupříkladu miska jablíček" nabízí spoustu otázek: Když jablíčka přesuneme do sáčku, bude to stále tatáž množina? A co když kousek jablíčka ukousneme? V rámci této přednášky se pokusíme o vybudování teorie množin od základů (rozuměj axiomů) a to v duchu Zermelo-Fraenkelovském. Pak uvidíme, jak na teorii množin vystavět zbytek matematiky.

**Typografie** *("What You See Is all What You've Got!?")* **TYPO**
*Martin „Medvěd" Mareš*
Jak na počítači text nejen napsat, ale také vysázet tak, aby pěkně vypadal a aby (což je důležitější) se i příjemně četl. Jak se sází pohádka, jak báseň a jak vzorové řešení KSP plné komplikovaných vzorců. Jak jde dohromady staleté umění typografické a moderní technika. Přineste knihy i letáky, zkritizujeme sazeče, co se do nich vejde.

**Čaj** *("Jak vypadá odvar z nezralých pražců?")* **TEA**
*Martin „Medvěd" Mareš*
Pojďme usednout k šálku lahodného čaje a povídat si o tom, co se v něm skrývá. Kde se čaj vzal, kde se pěstuje, jak se zpracovává a jak ho připravovat. Trocha čajového zeměpisu, dějepisu i čajové chemie a čajové kultury. Též o všelijakých substancích čaji podobných.

# Abecední seznam přednášek



## Základní přednášky



## Pokročilé přednášky



## Půlnoční přednášky