

JARNÍ SOUSTŘEDĚNÍ KSP 2018 – 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 zkuste říct, třeba se najde někdo, kdo by vám o tom rád pověděl.

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 přednášejících, 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 přehledkou 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**
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 I (“*Pokud jste ještě neslyšeli o informatických grafech, tak tato přednáška je právě pro vás.*”) **GA1**
Úvod z teorie grafů a použití grafů při řešení algoritmických problémů. Naučíme se hledat nejkratší cestu v bludišti, pochopíme základní princip, jak funguje GPS navigace a mnohé další. Vše si procvičíme na konkrétních příkladech.

Dynamické programování (“*Kampak jsem si to jenom schoval?*”) **DYNP**
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

Programování v jazyce C **C**
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 knihovnamy jazyka C.

Python **PYTH**
Ríša Hladík, Filip Štědronský
Jak programovat v Pythonu a jak v něm „nepsat Cčko“. Syntaxe, datové typy, funkce, třídy, . . . Na co si dát pozor, v čem se Python liší od ostatních jazyků a proč je mezi nimi tak oblíbený.

Filip Štědranský

Slýcháte v programátorských kruzích o objektech, třídách, dědičnosti a metodách, ale nemáte pořádně jasno v tom, co to přesně znamená? Představíme si objektově orientované programování (OOP) – techniku, která často pomáhá psát větší programy tak, aby se o nich snáze přemýšlelo. Ukážeme si základní principy a příklady použití. Čím se liší OOP v různých programovacích jazycích? Kdy je lepší se objektům vyhnout?

Předpoklady: Základy programování

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

Parsing čili analýza textu * (*“ $1+2*4 = 12$ ”*)**PARSE**

Martin Mareš

Často potřebujeme načíst nějaký složitý textový vstup: matematický výraz, webovou stránku v HTML, zdroják programu, . . . Ukážeme si, jak texty analyzovat (neboli parsovat), aniž bychom v nich zabloudili: rozdělení na lexikální a syntaktickou vrstvu, železničářský algoritmus na parsování výrazů, popis syntaxe pomocí regulárních výrazů a gramatik.

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

Někdy potřebujeme najít podřetězec ve velkém množství textu. Stromček trochu připomínající ten biologický aneb trie. Proč se ve vstupu vracet neboli Knuthův-Morrisův-Prattův algoritmus. Hledání více řetězců najednou podle Aha a Corasickové. Okénkové hešování Rabina a Karpa.

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

Riša Hladík, Filip Štědranský

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, jaké všechny druhy intervalových stromů existují a jejich použití si ukážeme na úlohách. Na závěr si představíme jednu „magickou“ datovou strukturu jménem Fenwickův strom.

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

Jirka Setnička, Martin Mareš, Filip Štědranský

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.

Geometrie a počítače (*“Nerušte mé kruhy! (ani jiné kvadriky)”*)**GEOM**

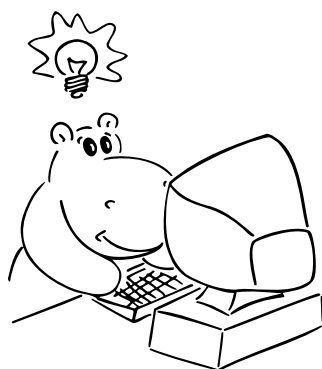
Jirka Setnička

Základní algoritmy pro řešení geometrických úloh – konvexní obal, dva nejbližší body v rovině, výpočet obsahu nekonvexního mnohoúhelníka, lokalizace bodu, scanline algoritmus a jeho použití, Voroného diagramy a souvislost s persistentními datovými strukturami.

Toky v sítích * (*“Když je v grafu povodeň, těsní?”*)**TOKY**

Jirka Sejkora, Pěta Pelikánová, Filip Štědranský

K čemu je dobré, když grafem teče voda. Předvedeme si klasický problém toků v sítích a jeho všelijaké, mnohdy dosti překvapivé aplikace. Jak rozestavět n věží na šachovnici a jak ji místo toho pokrýt dominovými kostkami? Další souvislosti, jako třeba násobná souvislost grafů.

**Těžké problémy *****HARD**

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.

Další programovací jazyky a techniky

Programování v jazyce C# (*“Co se stane, když strčíme Céčko za mřížku?”*)

CIS

Jirka Sejkora, Standa Lukeš

C# je moderní objektově orientovaný jazyk, který za patnáct let svého bouřlivého vývoje dostal do vínku některé funkcionální rysy. Mimo popisu základních konstrukcí si projdeme také nejzákladnější vestavěné třídy.

Perl (*“Jak Pejsek a Kočička vymýšleli programovací jazyk?”*)

PERL

Martin Mareš

Jednoho dne se Larry Wall rozhodl, že nasype do jednoho velkého kotle spousty programovacích jazyků a unixových utilit, za stálého míchání povaří, posléze přecedí, přikoření a implementuje. Tak vznikl Perl, jazyk původně určený hlavně na zpracování textu, ovšem jak se ukázalo, též šikovný na spoustu dalších věcí. Asociativní pole, libovolně složité datové struktury za pomoci referencí, balíčky a objekty zdarma a hlavně regulární výrazy zde a všude. Zkrátka jazyk, který lze jedinečně milovat nebo nenávidět, nic mezi tím.

Jazyk Go

GOLANG

Jirka Setnička

Go je moderní kompilovaný jazyk (vyvinutý původně v Google), který se pokouší být takovým Cčkem na steroidech. Umí být podobně rychlý, zaručuje větší typovou bezpečnost, ale má i prvky dynamicky typovaných jazyků. Jeho velkou silou je velmi snadné provázání na existující kód v C/C++, systém balíčků distribuovaných převážně přes Github, funkce vracující libovolný počet hodnot nebo třeba vestavěná podpora Unicode a vestavěná hashovací tabulka. Také se silně dbá na coding-style pro zajištění snadné čitelnosti programů.

Prolog * (*“Co s jazykem bez přiřazovacího příkazu?”*)

LOGP

Péťa Pelikánová

Proč psát dlouhé a složité programy, když stačí dostatečně přesně popsat situaci a pak se prostě zeptat? Toť princip logického programování, který si ukážeme na Prologu.

Haskell * (*“Pro ty, kdo uncínofobií trpí”*)

HASK

Péťa Pelikánová

Základní kurz Haskellu ■ moderního funkcionálního jazyka. Na skladě máme skoro všechno, co měl Lisp, o zbytku ukážeme, že mít to by byla chyba; a samozřejmě spoustu věcí navíc. Základní konstrukce, typový systém, třídy a jak se obejít bez výjimek a speciálních případů, vstup a výstup.

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

SQL

Standa Lukeš

Představíme si SQL, jazyk databází. Ukážeme si základní příkazy i práci o kus složitější. Jak ušetřit skriptu práci a sobě čas, aneb jak se zeptat rovnou na to, co chcí vědět. K čemu se hodí složený dotaz a klíčové slovo JOIN. Protože dnes už databáze typicky umí i mnoho (bohužel nestandardních) rozšíření, tak si také ukážeme jak se dá využít nějakých Postgresovin, třeba jak jej použít jako skoro-dokumentovou databázi.

JavaScript

JS

Standa Lukeš

Každá chytrá webová aplikace musí dneska mít aspoň trochu (nebo co nejvíc?) JavaScriptu. Ukážeme si, jak v něm psát, rozhybeme pomocí něj webovou stránku, předvedeme si, jak fungují funkce, objekty, moduly a tak. A taky jak to debugovat, jak si pořídit typovou kontrolu nebo novější featury než prohlížeče umí.

Předpoklady: WWW

Webové UI funkcionálně

REACT

Standa Lukeš

Vyrábět složitější aplikaci čistě v Javascriptu a HTML může být trochu peklo, tak si ukážeme jak se v tom neutopit. Vyrobit si totiž magickou abstrakci a z každé části stránky uděláme jenom funkci která dostane stav svojí části a vrátí HTML, které nám prohlížeč zobrazí. Pomocí toho se dá v aplikaci udělat hezky pořádek, tak si ukážeme jak se v tomto modelu dá naprogramovat všechno potřebné.

Předpoklady: JS

Počítače, sítě, systémy

Webové stránky

WWW

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 od formulářů až po JavaScript běžící v prohlížeči.

Sítě a Internet (*“Sítě nejen na ryby.”*)

NET

Jak funguje Internet a počítačové sítě vůbec. Lokální sítě s dráty i bez nich a různé způsoby, jak je mezi sebou propojovat. Protokoly rodiny TCP/IP a nad nimi postavené aplikační protokoly: DNS, SMTP, HTTP a celý zvěřinec dalších. Bezpečnost sítí a všelijaké útoky na ni. Pár taktů hudby budoucnosti: IPv6, multicasting, přenos v reálném čase atd.

Principy počítačů (*“A opravdu uvnitř počítače běhají malí trpaslíci?”*)

HW

Jirka Setnička, Standa Lukeš, Filip Štědronský

Vydáme se do země skřítků, kteří pohánějí počítače. Počítačové architektury od hodiněk po superpočítač od Craye, jejich křivolaká historie i současnost. Co je to procesor, jak se programuje a jak se chová. Různé druhy pamětí a jejich cacheování. Jak procesory komunikují s okolím – sběrnice, čipové sady, vstupní a výstupní zařízení. A co když je procesorů několik, nebo třeba pár tisíc? Přednáška bude praktická: pár počítačů při ní rozebereme a možná i nějaký postavíme.

Od zdrojáku k programu (*“Před spuštěním program přeložte. Stačí třikrát podélně? Po spuštění se přeloží ještě jednou sám.”*) **KOMI**

Standa Lukeš, Martin Mareš, Filip Štědronský

Mezi programem ve vašem oblíbeném příčetném programovací jazyce, který jste právě dopsali, a tranzistory uvnitř vašeho procesoru leží obrovské území obývané překladači, JITy, knihovničky, operačními systémy, loadery a jinými bájnými bytostmi. Pojdme zjistit, co jsou zač a co všechno s programem provádějí. Co udělá kompilátor za nás a co musíme naopak udělat my za něj.

UNIX (*“UNIX gives you enough rope to hang yourself.”*)

UNIX

Unixové operační systémy (zejména Linux) dobývají svět. Jak fungují uvnitř a jaké nabízejí výhody? Unixová filosofie a historie. Proč je systém složený ze spousty malých a jednoduchých kousků stabilnější a bezpečnější? Proč je ovládání prostřednictvím textových příkazů často efektivnější než klikátka? Jaké to je mít svůj systém pod kontrolou a „vidět mu pod ruce“? V čem spočívá moc textových souborů?

Skriptování v shellu

SHELL

Filip Štědronský

Praktičtější zaměřená přednáška než UNIX, zabývající se hlavně tím, jak efektivně používat příkazovou řádku. Ukážeme si na spoustě příkladů, jak nám může automatizace všedních činností ulehčit život a jak silné nástroje pro ni UNIXový shell (který navzdory svému názvu existuje i pro Windows) svou jednoduchostí a flexibilitou poskytuje. Budeme spojovat spoustu jednoduchých příkazů do mocných celků a s nimi plnit i na první pohled komplexní úlohy, jako třeba automatické stahování a parsování věcí z webu. Některé činnosti vyžadují lidskou nápaditost a vhled. Ty ostatní bychom měli přenechat strojům.

Vývoj software

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

GIT

Jirka Setnička, Standa Lukeš, Martin Mareš

Když se něco vyvíjí delší dobu, přijde vhod nějaký sofistikovaný nástroj. 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 kouzelnické triky: hledání bugů púlením historie, prepisování dějin.

Testování a kvalita softwaru

QA

Standa Lukeš

Výroba software není zdaleka jenom o programování. Ukážeme si, jak psát kód tak, aby nejenom fungoval, ale aby vyzářoval krásu a pokoj všem programátorům dobré vůle. Povíme si o různých způsobech testování, o tom jak udržet v kódu pořádek a dalších nástrojích, které pomáhají vyvíjet kvalitní software.

Předpoklady: Znáť aspoň jeden příčetný programovací jazyk

Aplikace informatiky a matematiky

T_EX (*“No pages of output. Ask a T_EXnician.”*)

TEX

Jirka Setnička, Ríša Hladík, Martin Mareš

Donald E. Knuth napsal T_EX před desítkami let proto, že mu nikdo nebyl schopn vysázet matematický text podle jeho požadavků. Od té doby se hojně používá pro sazbu nejrůznějších publikací. V této spíše praktické přednášce si ukážeme použití T_EXu od hladké sazby knihy až po zběsilosti hraničící s programováním. Pozornost věnujeme i zdrojům informací a rozdílům mezi různými dialekty T_EXu.

Programujeme obrázky (*“Teď ten obrázek takhle zkroutím a pak ho přeložím.”*)

MF

Jirka Setnička, Filip Štědronský

Ač se to tak na první pohled nezdá, občas nejjednodušší způsob, jak nakreslit nějaký obrázek, je popsat ho pomocí příkazů ve specializovaném programovacím jazyce. Týká se to zejména obrázků, kde něco chceme zobrazit přesně, případně obsahují nějakou pravidelnost: geometrické náčrtky, grafy, diagramy a podobné. Tímto způsobem vzniká například většina obrázků, které potkáváte v zadání a řešení KSP (kromě hrochů ;-)). Ukážeme si jazyk Asymptote, který se k tomuto účelu dá dobře použít (případně jeho blízkého příbuzného Metapost) a možná další související nástroje.

Kryptografie (*“Gbgg arav zbp gnwan mcenin.”*)

CRYPT

Standa Lukeš, Martin Mareš, Filip Štědronský

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.

Kompresie dat (“*Jnm idln kpln j nstlčtln.*”)

ZIP

Jirka Setnička, Filip Štědranský

Přehled základních kompresních algoritmů: triviální algoritmy (RLE), statistické metody (Huffmanovo a aritmetické kódování), slovníková komprese (LZ77, LZ78, LZW), Burrowsova-Wheelerova transformace (BZIP). Pokud zbude čas, tak i něco o ztrátové kompresi obrázků a zvuku (prediktory, wavelets, JPEG, MPEG, fraktály).

Čárové kódy (“*Jak naučit počítače číst láhve od Coly*”)

BAR

Martin 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, bílé křížky na asfaltu, ...

Matematické přednášky

Pravděpodobnost (“*S poloviční pravděpodobností bude tato přednáška, tak si jí s poloviční pravděpodobností připravím.*”)**PST**

Standa Lukeš, Pěta Pelikánová

Jak matematici zacházejí s náhodou? Co je to pravděpodobnost, náhodná veličina, střední hodnota, k čemu je ten kloboukový graf a lemma o džbánu, kterému se ucho utrhne? Podíváme se na odvětví matematiky zkoumané původně kvůli karbaníkům a gamblerům a také na to, jak se dá pravděpodobnost využít v programování: výpočet čísla π a algoritmy Monte Carlo.

Diskrétní matematika (“*O Dlouhém, Širokém a šatnářce*”)

DM1

Pěta Pelikánová

Úvodní minikurz diskrétní matematiky (to je opak matematiky spojité, čili mimo jiné kombinatorika). Seznámení s relacemi a jejich vlastnostmi. Dozvíte se také něco o uspořádaných, nezávislých a jiných množinách. S pomocí kombinatoriky možná vyřešíme problém zmatené šatnářky. Hallova věta nám pomůže určit, jestli má cenu snít o perfektním párování.

Funkce a jejich chování

MA

Pěta Pelikánová

Profrčíme letem světem základní tvary funkcí. Vrhne se na derivování a integrování. Podíváme se na aplikace při popisu chování funkcí a počítání rozměrů ohraničených ploch. Pak to všechno dáme do pohybu a budeme schopni určit objem rotačních těles.

Jak vypadá zrcadlo v číslech

LIN1

Pěta Pelikánová

Na přednášce se podíváme na to, jaká matematika stojí za grafickými transformacemi obrázků. Ukážeme si, jak můžeme otáčet předmět pomocí jeho zobrazení osovou souměrností, a budeme si hrát s dalšími transformacemi. Dozvíte se, jak vypadají matice a jak je násobit s vektory. Povíme si, co všechno stojí na lineární algebře a jaké problémy dokáže vyřešit.

Lineární algebra *

LIN2

Pěta Pelikánová

Naučíme se hledat řešení soustav rovnic na papíře tak, abychom toho museli napsat co nejméně (Gaussova eliminace). Zabředneme hlouběji do vektorových prostorů a budeme hledat jejich hezké ortogonální a ortonormální báze.

Předpoklady: Je dobré vědět, co je to matice a jak se násobí (přednáška LIN1).

Teorie nemožného * (“*Neexistence důkazu není důkazem neexistence. Dokažte.*”)

NONEX

Martin Mareš

Existenci slona v Africe snadno dokážete tím, že ho přivedete. Jak ale ukázat, že tam žádný slon není, případně že sice je, jenže ho nejde najít pomocí pravítka, kružítka a jeepu? Přímou se to dělá těžko, ale existuje spousta krásných triků, jak neřešitelnost problémů dokazovat. Nesložitelné hlavolamy, nerozvázatelné uzly, nepopsatelná čísla, neroztřetitelné úhly, nealgoritmické problémy a jiné slasti nekonstruktivní matematiky. Jak naopak ukázat, že něco existuje, aniž bychom věděli, jak to vypadá?

Intervalové počítání (“*Prima, sekunda, tercié, ... nekonečno.*”)

INT

Pěta Pelikánová

Co s čísly, která si počítač nezapamatuje přesně? Vezmeme si na pomoc intervaly a zabalíme je do nich. Naučíme se s intervaly počítat a nakrmíme nimi i funkce. Řekneme si, kde se dají využít v praxi, ale i v teoretické informatice.

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.

Základy první pomoci (“*Jak někomu zachránit život a jak málo k tomu stačí*”)

ZDRAV

Jirka Setnička

Pobavíme se o základech první pomoci. Jak správně vyhodnotit situaci a kdy je potřeba volat pomoc? Jak se postarat o člověka v bezvědomí, jak kontrolovat životní funkce a jak člověka stabilizovat do příjezdu pomoci? Ukážeme si, jak málo stačí k záchraně života a naučíme se nebát se první pomoci. A také, že naše bezpečí je v každé situaci na prvním místě.

Létání a natáčení s dronem (*“Ne ne, to není řízená raketa, tam jen visí velký foťák.”*)

DRON

Jirka Setnička

Na co je dobré mít drona, kdy se dá reálně použít a kdy je to jen drahá hračka. Přehled o technologiích, jejich možnosti a omezení. Jak se liší armádní drony od dronů pro kameramany a nebo od dronů, které pořídíte za pár šupů v hračkářství. Také lehký výlet do legislativy – s čím smíme létat, kde smíme létat a co s tím můžeme dělat. Pokud počasí dá, tak i praktické hrátky s dronem.

Lockpicking (*“Jak si odemknout, když si náhodou my (nebo soused) zapomeneme klíč :-)”*)

PICK

Jirka Setnička

Jak fungují dnešní zámky, co jsou to stavítka a jak vlastně fungují klíče. A jak se pomocí jednoduchých nástrojů dají využít výrobní nedokonalosti zámků k jejich odemčení. Použití planžet, napínáků, praktické ukázky odemykání, nastínění technik bumpingu a dalších postupů, jak se dostat přes zamčené dveře.

Soukromí na webu (*“Naši zákazníci si samozřejmě můžou navolit, jaké údaje o Vás budeme skladovat. Kupte si u nás taky reklamu a Standa Lukeš*

Co všechno o vás webové služby umí zjistit? V čem to vlastně vadí? Jak to dělají? A proč nás sledují? Dobře, dá se nějak efektivně bránit? Facebook je jedinný zlý, že? Z trochu jiného soudku - co se o vás dá na veřejném internetu jednoduše najít, a jestli je to vlastně špatně. No ale GDPR nás spasí, že? Navzdory divné anotaci bude přednáška spíše praktická a technická než filosofická, a rozhodně nebude právníká.

Teorie množin (*“Jablka a hrušky se dají nejen sčítat, ale třeba i násobit.”*)

TEMNO

Martin 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.

Lingvištika (*“Přísudek je v této větě podmět.”*)

LING

Martin Mareš

Převážně nevážné a mírně nepřed-vídatelné po-vídaní 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.

Typografie (*“What You See Is all What You’ve Got!?”*)

TYPO

Martin 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.

Orientace

ORI

Martin Mareš

Jak ze neztratit v terénu a jak se neztratit na moři. Vývoj umění navigace. K čemu je důležité slunce a hvězdy, ale proč mořeplavcům nestačí, alespoň dokud neobjevíme hodinky. Použití mapy, busoly a GPSky. Orientace bez pomůcek a použití Ariadniny nitě. Bleskový úvod do sférické astronomie a časoměry čili jak (ne)postavit sluneční a třeba i měsíční hodiny. Jak reprezentovat mapu v počítači a jak raději ne. Jak zapisovat polohu místa na Zemi (přestože Země má tvar podivně nakousnuté hrušky) a kolika způsoby to jde. Různé druhy map a jejich (z)kreslení. Jak se neztratit v kartografii. Praktické cvičení v terénu.

Čaj (*“Jak vypadá odvar z nezralých pražců?”*)

TEA

Martin 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.

Vektorizace obrázků (*“Jak se rodí hroši?”*)

SVG

Péťa Pelikánová

Každý hezký text se neobejde bez obrázků. Dozvíte se jak se z hlavy někoho kreativního dostane obrázek hrocha až na leták KSP. Jak a proč z náčrtku tužkou udělat vektorový obrázek a v čem se dá tvořit vektorová grafika.

Akrobacie aneb létání (*“Vy jste všichni blázni jen já jsem letadýlko.”*)

ACRO

Péťa Pelikánová

Kde je hranice sportem k u zdraví či k trvalé invaliditě? Různé pohledy na cirkus. „Wow, to je nemožné!“ vs. „Jé, jak to dělají?“ Co je to nový cirkus, akrojóga, handstandy a headstandy ... Ať už si chceš o tom něco poslechnout nebo si chceš něco z toho vyzkoušet, ulov si Péťu.

Péťa Pelikánová

Nahlédneme co vše se skrývá pod pojmem Internet of Things. Jak internet propojuje spoustu různých zařízení a jak se z nich shromažďují informace. Představíme si model klaudových služeb. Jak může vypadat IoT chytrá domácnost.

Racionalita (*“Racionalita neslouží k vyhrávání debat, nýbrž k tomu, abychom se rozhodli, na kterou stranu se postavit.”*)**RAT**
Filip Štědranský

Naše přesvědčení o světě jsou jako mapa nějakého území. Měli bychom se snažit, aby mapa co nejlépe odrážela skutečnost, neustále ji upravovat podle nově objevených faktů. Přesto se lidi často spíš snaží přesvědčit ostatní i sebe, že jejich názor je správný, než aby věnovali veškeré úsilí nalezení názoru nejlépe kopírujícího realitu. Odhlížíme od nepříjemných pravd. Proč? Jak se naučit své názory absolutně podříditi realitě, přijímat svět, jaký je? Jaké systematické chyby v uvažování lidský mozek dělá a jak nám jejich znalost (ne)může pomoci? K čemu se hodí znát teorii pravděpodobnosti či úvahy z oblasti umělé inteligence?

Počítač bez myši (*“Může počítač trpět fobií z hlodavců?”*)

KEYB

Filip Štědranský

Ruce programátora patří na klávesnici, přesto se spousta věcí běžně dělá pomocí myši. Podíváme se třeba na to, jak bez myši editovat text (editor Vim), spravovat okna či ovládat webový prohlížeč. Posuďte sami, kolik cenných vteřin to může pokaždé ušetřit.

Biologická evoluce (*“Ale vždyť je to jen teorie!”*)

EVOL

Filip Štědranský

Evoluční teorie je asi jedním z největších převratů v lidském náhledu na svět. Jak se na takovou věc vůbec přišlo, proč to tak dlouho trvalo a proč si myslíme, že je to pravda? Jak si to vše alespoň trochu představit? Mnoho nečekaných překvapení, informatické a fyzikální souvislosti. Proč jsou stromy vysoké, jak souvisí pohlavní rozmnožování s počítačovou bezpečností, co má evoluce společného s Windows a co s halting problémem?

Předpoklady: Základní přednáška nevyžadující předchozí biologické znalosti.

Abecední seznam přednášek

LYK Stručný úvod do základů teorie vlkodlaků.. 1

Základní přednášky

DYNP	Dynamické programování	1	ZALG	Základy algoritmizace, složitosti a datových struktur	1
GA1	Grafy & algoritmy I	1	ZAKL	Základy programování	1
OOP	Objektově orientované programování	2			
C	Programování v jazyce C	1			
PYTH	Python	1			

Pokročilé přednášky

DS	Datové struktury pro pokročilé	2	HW	Principy počítačů	4
DM1	Diskrétní matematika	5	CIS	Programování v jazyce C#	3
MA	Funkce a jejich chování	5	MF	Programujeme obrázky	4
GEOM	Geometrie a počítače	2	LOGP	Prolog	3
HASK	Haskell	3	SQL	SQL databáze	3
TEXT	Hledání v textu	2	SHELL	Skriptování v shellu	4
INT	Intervalové počítání	5	GIT	Systém pro správu verzí Git	4
ITREE	Intervalové stromy	2	NET	Sítě a Internet	3
LIN1	Jak vypadá zrcadlo v číslech	5	NONEX	Teorie nemožného	5
JS	JavaScript	3	QA	Testování a kvalita softwaru	4
GOLANG	Jazyk Go	3	TOKY	Toky v sítích	2
ZIP	Kompresce dat	5	HARD	Těžké problémy	2
CRYPT	Kryptografie	4	UNIX	UNIX	4
LIN2	Lineární algebra	5	REACT	Webové UI funkcionálně	3
KOMP	Od zdrojáku k programu	4	WWW	Webové stránky	3
PARSE	Parsing čili analýza textu	2	TEX	T _E X	4
PERL	Perl	3	BAR	Čárové kódy	5
PST	Pravděpodobnost	5			

Půlnoční přednášky

ACRO	Akrobacie aneb létání	6	RAT	Racionalita	7
EVOL	Biologická evoluce	7	PRIVACY	Soukromí na webu	6
IOT	Internet of Things	7	TEMNO	Teorie množin	6
LING	Lingvištika	6	TYPO	Typografie	6
PICK	Lockpicking	6	SVG	Vektorizace obrázků	6
DRON	Létání a natáčení s dronem	6	ZDRAV	Základy první pomoci	5
ORI	Orientace	6	TEA	Čaj	6
KEYB	Počítač bez myši	7			