

JARNÍ SOUSTŘEDĚNÍ KSP 2013 – 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 to k přednáškám připište, třeba se najde někdo, kdo by vám o tom rád pověděl. 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 a nakonec stručný obsah přednášky.

Povinně dobrovolné přednášky

Základy programování (“*Má $x = x + 1$ řešení?*”) **ZAKL**

Michal Pokorný, Jirka Setnička, Honza Kubálek

Úvodní několikadílná přednáška pro ty, kteří mají s programováním malé nebo dokonce žádné zkušenosti. Od základů si vysvětlíme problematiku programování a představíme programovací jazyk Python, který je jednoduchý na naučení, ale přitom není jen dětskou hračkou a je používán ve spoustě důležitých projektů. Ukážeme si základní datové typy (n-tice, seznamy, slovníky) a datové struktury (fronta, zásobník) a základní algoritmy a programátorské techniky, bez kterých se žádný programátor neobejde.

Základní algoritmy **ZALG**

Základní výbavou každého informatika jsou různé standardní algoritmy, zde si ukážeme ty nejdůležitější z nich: Třídící algoritmy včetně vnějšího třídění. Vyhledávání v polích, hledání mediánu, resp. k -tého největšího prvku v lineárním čase. Vyhodnocování výrazů, předzpracování vstupních dat.

Grafy & algoritmy I (“*Pojďme si hrát s obrázky*”) **GA1**

Co to jsou grafy, jak je v programech reprezentovat a hlavně k čemu se dají použít. Prohledávání grafu do šířky i do hloubky. Hledání nejkratších cest: Dijkstrův a Floydův algoritmus. Topologické třídění grafů a kreslení grafů jedním tahem.

Datové struktury pro začátečníky **DS1**

Jak si ukládat data natolik šikovně, abychom je nejen neztratili, ale také našli dříve, než si pro nás přijde Smrt. Klasické struktury jako pole, seznamy, trie, vyhledávací stromy (vyvážené, AVL, $a-b$), haldy (binární a obecně regulární) a v neposlední řadě hešování.

Algoritmy a jejich složitost (“*Čím menší je časová složitost algoritmu, tím větší je složitost kódu.*”) **SLOZ**

Problém, algoritmus a program. Časová a paměťová složitost problémů i algoritmů. Složitost rekurzivních algoritmů, složitost v průměrném případě.

Dokazování **DOKAZ**

Pavel Veselý

Ověřit správnost programu můžeme buď spuštěním algoritmu na opravdu velkém množství různorodých vstupů, nebo rozumovou úvahou, která mimo vši pochybnost existenci vstupu, na který by program odpověděl špatně, vyloučí.

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

Karel Tesař, Jirka Setnička

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.

Jak vypadá řešení (“*Jen jeden bod, když jsem napsal 18 stránek?*”) **SOL**

Jak má správně vypadat řešení KSP? Na co si dát pozor, co je úplně špatně a za co organizátoři strhávají body a sobě vlasy. Přednáška, která by mohla pomoci i mnohým déle aktivním řešitelům.

Algoritmizace

Složitější složitost *

SLOZ2

Martin Mareš, Pavel Veselý

Trochu hlouběji o složitosti: amortizovaná časová složitost, dolní odhady, nedeterministické výpočty a třída NP, NP-úplné problémy a příklady redukcí.

Předpoklady: SLOZ

Datové struktury pro pokročilé *

DS2

Karel Tesař, Martin Mareš, Pavel Veselý

Důmyslnější datové struktury: splay stromy, BB- α stromy, skip-listy, treapy a geometrické struktury pro lokalizaci bodů v rovině.

Grafy & algoritmy II

GA2

Karel Tesař, Pavel Veselý

Pokročilejší grafové algoritmy: union-find problem, hledání minimální kostry, mostů a maximálního párování, testování silné souvislosti.

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

ITREE

Karel Tesař, Jirka Setnička, Lukáš Folwarczný, Karolína Burešová

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.

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

TOKY

Karel Tesař, Pavel Veselý, Jirka Setnička, Honza Škoda

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.

Předpoklady: Umět plavat (zejména v matematice)

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

REGEX

Karel Tesař, Jirka Setnička, Karolína Burešová

Vyhledávání čehokoliv ve velkém množství textu: Prostá vylepšení hledání hrubou silou: Karp-Rabin, Boyer-Moore. A algoritmy chytrější: Morris-Pratt, Knuth-Morris-Pratt, Aho-McCorasicková. Konečné automaty prakticky, regulární a „regulární“ výrazy.

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

GEOM

Martin Mareš, 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.

Herní algoritmy (*“Když nemáte na to, abyste vyhráli šachový turnaj. . .”*)

AIGAME

Pavel Veselý, Karolína Burešová

Povídání o tom, jak programovat počítačové soupeře do šachů a her jim podobným. Základní minimaxový algoritmus a jeho vylepšení neboli α - β ořezávání. Stále pomalé? Několik nápadů na efektivnější ořezávání. Ne u všech her však funguje hrubá síla (minimax) dobře, ukážeme si tedy, jak hru zanalyzovat.

Logické úlohy a informatika

LOGINF

Karel Tesař

Povíme si, jak logické úlohy souvisí se základy informatiky. Ačkoliv to tak na první pohled nevypadá, tak ve většině logických úloh jsou skryty informatické postupy jako je například kódování informací, hledání správné cesty nebo splňování podmínek.

Programování s omezujícími podmínkami * (*“Celé prázdniny budu plánovat a řešit sudoku.”*)

CSTR

Michal Pokorný

Trochu jiný přístup k obtížným úlohám. Některé úlohy sice vypadají, jako by se za dobu existence vesmíru nedaly vyřešit, nicméně pro rozumně velké vstupy to přesto potřebujeme. Jak backtrackovat rychleji a radostněji – backjumping, backmarking, limited discrepancy search, a jak neprobírat úplné nesmysly – hranová konzistence, konzistence po cestě, bodová konzistence. Programování „Řekni mi, co chceš, a já ti to seženu“.

Programovací jazyky a techniky

Programování v jazyce C

C

Datové typy jazyka C, programové konstrukce, základy práce s ukazateli. Seznámení se standardními knihovny jazyka C.

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

CIS

Pavel Veselý, Honza Škoda

C# je moderní objektově orientovaný jazyk, který za víc než deset let svého bouřlivého vývoje dostal do vínku některé funkcionální rysy. Mimo popisu základních konstrukcí si projdeme zajímavé části dotnetí Base Class Library.

Objektově orientované programování v C++ a Object Pascalu (*“I život je objektový, tak proč ne programování...?”*)

OBJ

Honza Škoda, Honza Kubálek

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. Syntaxe deklarací tříd v C++ a Delphách, viditelnost a přístupová práva ke členům tříd. A také, že to není jediný přístup k objektům.

Předpoklady: Znalosti procedurálního programování v Pascalu a/nebo v C.

Základy webu (*“Proč se cpát do Flashe, když to jde hezky a standardně jinak?”*)

WWW

Honza Škoda, Jirka Setnička, Karolína Burešová

Úvod do servírování informací skrze World Wide Web. Co znamená `http://`, jaký je rozdíl mezi HTML a XHTML a na co se hodí znát kaskádové styly. Kdy a proč (ne)použít kterou technologii.

Jazyk SQL (*“HAVING COUNT(*) = -1 Urghh?”*)

SQL

Honza Kubálek

Jak správně položit dotaz do relační databáze obsahující tisíce záznamů? Jaké operace s daty můžeme přenechat serveru a jaké si raději zpracovat v lokální aplikaci? Co je to JOIN, jaký je rozdíl mezi WHERE a ON. Zbyde-li nám čas, nebo v případě zájmu, si ukážeme, jak na relační algebru a jak pomocí ní zefektivnit SQL dotazy.

PHP

PHP

Honza Škoda, Jirka Setnička, Michal Pokorný, Karolína Burešová

Úvod do problematiky vytváření dynamicky generovaných webových stránek. Syntaxe jazyka PHP, netypovost, (ne)pole. Jak psát stránky, za které se nemusím stydět, objektové programování, pár slov o bezpečnosti. Zamícháno s XHTML, JavaScriptem, CSS a možná i (My)SQL.

Předpoklady: WWW

Haskell * (*“fac n = product [1..n]”*)

HASK

Karolína Burešová

Funkce v obvyklých (imperativních) programovacích jazycích nejsou funkce v matematickém slova smyslu: mají „vedlejší efekty“, projevují se na běhu programu i jinak, než svou návratovou hodnotou. Ve funkcionálních jazycích, mezi něž Haskell patří, mají takové chování zakázáno. K čemu to?

Programování v UNIXu (*“Segmentation fault. Core dumped.”*)

PUNIX

Martin Mareš

Jak se programuje v operačním systému, který si programátoři navrhli pro sebe? Vyměníme klikací vývojová prostředí za stavebnici se spoustou kostiček. Textový editor Vim, překladač GCC a otrokář zvaný make. Jak zjistit, co program doopravdy dělá, i když běží hodinu – core dumpy, GDB, valgrind, perf. Jak programovat ve více lidech a nezbláznit se z toho – diff, patch, Git.

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

LOGP

Jirka Setnička, Honza Škoda, Karolína Burešová

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.

Python (*“import antigravity”*)

PYTH

Honza Škoda, Karolína Burešová

Základy rychlého programování v Pythonu. Syntaxe, datové typy, funkce, třídy, ... Na co si dát pozor, v čem se liší od ostatních jazyků a proč je tak oblíbený.

Ruby (*“Jak při psaní webů záplatovat opice a mlátit kachny a přitom být stokrát rozumnější než PHP.”*)

RUBY

Michal Pokorný

Ruby je super jazyk: je ohebný a snadno pochopitelný a navíc má okolo sebe skvělý ekosystém. Uvidíte jeho základy a snad i poznáte, proč si ho tolik programátorů zamilovalo.

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

GIT

Michal Pokorný, Jirka Setnička, Honza Kubálek, Martin Mareš

Jak vyvíjet program delší dobu a nezbláznit se u toho. Různé systémy pro správu verzí od diff/patch přes CVS a SVN až ke Gitu. Jak Git funguje: stromy, commity, větve, tagy. Merge mezi větvemi nebo mezi různými počítači. Kouzelnické triky: hledáme bugy pūlením historie, přepisujeme dějiny. Jak se liší správa zdrojáků v projektech o jednom, deseti a tisíci programátorech. Udržujeme patche k cizímu programu aneb quilt a StGit.

Matematické přednášky

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

DM1

Pavel Veselý, Karel Tesař, Lukáš Folwarczný

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.

Derivace a integrály pro zelenáče

DIFF

Honza Kubálek, Michal Pokorný, Jenda Hadrava

Rychlokurz derivování a integrování. Rychle prolétneme limity, nadefinujeme si derivace a procvičíme jejich výpočty. Dále si řekneme, co je to integrál, jak se definuje a počítá. Na závěr si ukážeme, k čemu je to všechno dobré v reálném či středoškolském světě – „rychlé“ odvozování fyzikálních vzorců, grafy funkcí, všemožné optimalizace.

Komplexní čísla (*“ $1 = \sqrt{(-1)^2} = (\sqrt{-1})^2 = i^2 = -1$. WTF???”*)

COMPL

Honza Kubálek, Jenda Hadrava

Ukážeme si, co se stane, když začneme počítat s čísly mimo reálnou osu: co je to imaginární jednotka a jak se s ní počítá, komplexní čísla a počítání s nimi v algebraickém, goniometrickém a exponenciálním tvaru, dále převod mezi těmito tvary. Zobrazení a výpočty v Gaussově rovině. Také si řekneme, k čemu se komplexní čísla dají použít v praxi.

Lineární algebra (*“Vektorový prostor je místo, kde žijí vektory.”*)

LA1

Pavel Veselý, Honza Kubálek

Lineární algebra vznikla jako formalizace geometrie a tuto souvislost si ukážeme. Popíšeme vektorové prostory, které se skládají z vektorů. Jaké operace s nimi umíme provádět a co všechno musí splňovat? Kdy jsou vektory závislé a kdy nezávislé? Co je to lineární kombinace, obal a generátor? Co je to dimenze vektorového prostoru a jaké má souvislosti s předchozími pojmy?

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

CRYPT

Martin Mareš, Jirka Setnička

Kryptologie čili tajuplná nauka o šifrách, jejich konstrukci a hlavně o jejich luštění. Přísně tajné. Šifrovací systémy jako lego: základními kostičkami nám budou symetrické a asymetrické šifry a jednosměrné funkce, stavět z nich budeme kryptografické protokoly na bezpečný přenos, autentikaci, digitální podpisy a třeba i jak si hodit korunou po telefonu. Předvedeme nerozluštitelnou šifru a dokonce to o ní i dokážeme.

Teorie kombinatorických her (*“Život je jen hra. . . Jakou má vyhrávající strategii?”*)

GAME

Pavel Veselý

Rozličné kombinatorické hry se zápalkami, kamínky, barvičkami či grafy a jejich abstrakce. U některých si ukážeme výherní či obranné strategie, u některých dokážeme, že sice příslušná strategie existuje, ale nikdo ji ve skutečnosti nezná. Teorie pana Conwaye: jak pozicím přiřazovat čísla, popř. jiné symboly. Piškvorky a jejich zobecnění na hypergrafy.

Teorie grafů

GRAF

Martin Mareš, Pavel Veselý

Grafy, které vás na střední škole neučili! Žádné osy a funkce, jsou to města a silnice. Jednoduché, představitelné objekty. Často je vidáme v algoritmech, ale i z čistě matematického úhlu pohledu jsou zajímavé. Ukážeme si nečekaná použití grafů v geometrii, teorii čísel i jinde. Na grafech si také představíme dualitu – vlastnost, kterou mají snad všechny matematické pojmy společnou. Vše jednoduše a srozumitelně i pro ty, kdo o grafech slyší prvně.

Rovinné grafy

ROG

Pavel Veselý, Karel Tesař, Jirka Setnička

Povídání o grafech, které jde nakreslit na papír bez křížení hran. O tom, co všechno pro takové grafy platí a jak je poznáme, aniž bychom je museli kreslit. Existuje pouze 5 pravidelných mnohostěnů a my se o tom pomocí teorie grafů přesvědčíme. Barvení rovinného grafu šesti a možná i méně barvami. Když zbude čas, zkusíme grafy kreslit i na jiné plochy: kupříkladu Möbiovu pásku, pneumatiku nebo ušatou kouli.

Matematika náhody (*“Světlem vládne náhoda a současně i s ní zákon, který vzniká z obrovského souboru náhod.”*)

PST

Lukáš Folwarczný

V této přednášce se podíváme na to, jakým způsobem matematika zachází s pojmem náhody. Podíváme se na motivaci a způsob zavedení teorie pravděpodobnosti, odvodíme pár užitečných vztahů a především budeme řešit zajímavé příklady. Budete-li během přednášky vyčerpaní, přečteme si část Dopisů o pravděpodobnosti.

Deskriptivní geometrie

DG

Lukáš Folwarczný

Letmé představení této disciplíny. Existence deskriptivy je sice silně spjata s aplikací v technické praxi, ovšem i bez ní je disciplínou velmi pěknou. Podíváme se společně na některé základní projekce a s jejich pomocí se pokusíme vyřešit směsici úloh. Pokud jste měli deskriptivu na střední škole, přednáška vám pravděpodobně mnoho nepřinese.

Hardware, operační systémy a další technické

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

UNIX

Honza Škoda, Martin Mareš, Honza Kubálek, Jenda Hadrava

Kamarád u černobílého textového okna září blahem. Chcete poznat, proč? Jak UNIX vznikl, k čemu je dobrý a k čemu třeba není. UNIXová filosofie. Kouzlo scriptů. Kouzlo speciálních souborů. Kouzlo propojování programů. Kouzlo nechtěného. UNIX byl napsán v C a C vzniklo pod UNIXem.

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

HW

Martin Mareš, Jirka Setnička, Jenda Hadrava

Historie a vývoj počítačových technologií a současných architektur. Co je to procesor, jak se programuje (instrukční sady) a jak se chová. Přehled tříd procesorů (skalární a vektorové, RISC). Jak procesor komunikuje s okolím (čipová sada). Nezbytné periferie: vnitřní a vnější paměti, vstupní a výstupní zařízení.

TEX

TEX

Martin Mareš, Karolína Burešová, (Jirka Setnička)

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. Probereme základy TEXu, jeho využití při psaní řešení KSP, ale třeba také povídek, ročenky, slajdů pro prezentaci, zpěvníku nebo rovnou not.

Hradla (*“Vše co jste chtěli vědět o hradlech a báli jste se zeptat.”*)

HRAD

Jirka Setnička, Jenda Hadrava

Shrnutí hradlového seriálu KSP. Tedy krátký úvod do historie, nástin vnitřností, schémata, schematické značky, konvence. Trocha teorie, Booleova algebra a De Morganovy zákony. V neposlední řadě si ukážeme, jak rychle a efektivně umíme problémy hradly řešit.

Počítačová grafika (*“Namaluj mi beránka. . .”*)

GFX

Pavel Veselý, Martin Mareš

Kreslení a zpracování obrazu na počítači. Souřadnice (rovinné, prostorové i barevné) a jejich transformace. Základní grafická primitiva: body, úsečky, kružnice, elipsy, Bézierovy křivky a jejich rasterizace. Vyplňování n -úhelníků a křivkou ohraničených oblastí, flood fill. Pár triků navíc: maticové filtry, anti-aliasing a dithering. Grafické formáty a komprese obrázků. Základy trojrozměrného promítání a vykreslování scény.

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

NET

Martin Mareš, Honza Škoda

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.

Od zdrojáku k programu (*“Před spuštěním program přeložte. Stačí třikrát podělně?”*)

KOMP

Martin Mareš

Mezi programem v Céčku, který jste právě dopsali, a tranzistory uvnitř vašeho procesoru leží obrovské území obývané překladači, linkery, knihovníky, operačními systémy, loadery a jinými bájujícími 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.

Ostatní

Úvod do programátorských soutěží (*“Za pár let chci třímát v rukou pohár!”*)

SOUT

Karel Tesař, Lukáš Folwarczyný

Tipy a triky, jak uspět v programátorských soutěžích a olympiádě. Čeho si všimnout při vymýšlení algoritmů a na co si dávat pozor při samotné implementaci. Jakých soutěží se na střední škole můžete účastnit, kde se dají získat zkušenosti a kde se naopak dají vyhrát velké ceny.

Půlnoční přednášky

MFF UK aneb co obnáší matfyzákem být (*“Maminko, ptá se tatínka, kdy už budu matfyzákem?”*) **MFF**

Nezávazné povídání o Matfyzu a základním matfyzáckém folklóru. Určitě si přečteme matfyzáky sepsané Úvod do matfyzáka a zazpíváme pár matfyzáckých písní. Zbytek už bude záležet na tom, co budete chtít slyšet.

Mapování mysli a jiné techniky (*“Někdo mapuje terén, my mapujeme mysl”*) **MIND**

Pavel Veselý

Mapy mysli (neboli myšlenkové mapy) jsou speciální diagramy, které mají v člověku povzbudit kreativitu. Kromě vymýšlení všemožných věcí od přednášky po výpravu na Mars je lze využít pro organizaci myšlenek a při učení. Na příkladě si tedy nějakou mapu mysli vytvoříme a představíme si program na jejich tvorbu. V případě času rovněž: brainstorming, duševní rozcvičky, nasazování klobouků a další užitečné kreativní techniky.

Ďábelské herní algoritmy * (*“Pořádně se zamyslet, nebo pořádit výkonější počítač?”*) **AIGAME2**

Pavel Veselý

Minimax s α - β ořezáváním nestačí? Pokročilejší techniky na efektivnější ořezávání. Jiné přístupy než minimaxovaný algoritmus aneb Proof number search a metody Monte Carlo.

Předpoklady: AIGAME

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

Jirka Setnička, Karolína Burešová

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.

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

Lukáš Folwarczný, 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. Tempo přednášky bude spíše rychlejší.

Teorie čísel trochu jinak (*“Horst Fuchs uvádí $11 = 0$ ”*) **TORC**

Honza Kubálek

Co je to malá Fermatova věta a jak ji dokázat, aniž bychom ji ubili k smrti formalismy? Jak souvisí generál Wilson s malým Fermatem? Nejsou kongruence jenom pro vtipálky? Toto a ještě mnohem víc za pouhých 999 mod 1000 v klidu a teple našich chatrčí.

Základy psaní pro Android **ANDROID**

Michal Pokorný

Android se v našich mobilech, tabletech, netboocích a ledničkách zabydlel, a asi se v nejbližší době neodstěhuje. Na nějakém jednoduchém zadání si ukážeme, jak se pro něj programuje. Vlastní notebook s nainstalovaným Android SDK se bude hodit.

Předpoklady: Aspoň pasivní znalost Javy.

Ruby on Rails * (*“Webíme bez španělských bot”*) **RAILS**

Michal Pokorný

V přednášce navazující na Ruby si ukážeme, jak se programují webové aplikace v jednom z nejpoužívanějších frameworků: jednoduché věci budou jednoduché, složité věci budou často jednoduché taky, a nad věcmi, které by měly být automatické, nebudeme muset ani přemýšlet. Hodí se mít ponětí o MVC.

Předpoklady: Ruby, nějaké zkušenosti s weby

Auto z pohledu technika (*“Co mi to vrčí pod kapotou a proč bliká ta kontrolka?”*) **CAR**

Jirka Setnička

Nahlédneme do tajů starších i novějších aut. Podle zájmu se můžeme pobavit o tom, jaký je rozdíl mezi benzínovým a naftovým motorem. Na praktické ukázce probereme auto z pohledu běžné údržby a jednoduchých oprav. Na co si mohu troufnout sám a co bych měl raději svěřit servisu. Praktické typy, jak se chovat ke svému autu a mnoho dalšího podle zájmu. Určeno pro každého, koho čeká autoškola, nebo ho jen zajímá mechanika.

Literárně-filosofický seminář **LFS**

Lukáš Folwarczný

Pojďme diskutovat. Četli jste v poslední době zajímavou knihu, o níž byste se chtěli podělit? Co říkáte na státní podobu maturity z literatury, jaký seznam knih zvolila vaše škola? Je Franz Kafka enormně nadhodnocený? Je tragédií soudobé společnosti skutečnost toho, že nečtete poezii? Dá se vůbec ve světě, kde lze zažít tolik zajímavých věcí, najít čas na beletrii? Je filosofie na matfyzu sprostým slovem? Jaký je společenský přínos filosofů? Máte svého oblíbeného filosofa? Chcete vést spor o universálie? Jsoucno, nebo bytí? Jaká je aplikace Platónova trojúhelníku v geometrii a jak souvisí Kant s Eulerem? A co na to Jan Tleskač?

Martin 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 mlasknutí &c.