Studentský matematicko-fyzikální časopis

M&M

Ročník XVII / Číslo 5

Milé řešitelky, milí řešitelé,

jaro se pomalu hlásí ke slovu a my doufáme, že vám ke zlepšení nálady mimo pěkného počasí pomůže i další číslo našeho časopisu. Připravili jsme pro vás novou várku tentokrát opravdu pěkných úloh. A nechybí ani pokračování seriálu o Pythonu. V tomto čísle se naučíme, jak využít Python pro práci s webem.

Zároveň bychom chtěli připomenout soutěž o nejlepší příspěvek k tématkům. Autor nejlepšího článku se může těšit na dort osobně předaný organizátory. Neváhejte proto posílat své příspěvky. Soutěž trvá do konce května.

Ti nejlepší z vás v nejbližší době dostanou pozvánku na jarní soustředění, které se bude konat 2. – 10. dubna v Jizerských horách. Pokud ji ve svých poštovních schránkách nenajdete, tak ale nezoufejte a pilně řešte. Na podzimní soustředění se zve i podle výsledků z předchozího ročníku.

Pěkné chvíle strávené nejen s naším časopisem přejí

Organizátoři M&M

Zadání úloh

Termín odeslání páté série: 25. 4. 2011

Kde to jsem? Podivil se Ferda Mravenec a mžoural zmateně okolo. Když v tom se ozvalo jen „PLESK“ a ležel na zemi. A za první kapkou padala další a potom ještě další. „Asi bych se měl někam schovat,“ pomyslel si. Před sebou zahlédl nějaký dům. Zkusil zabouchat na dveře. Na štítku byla napsáno jméno domácího pána. Bydlel tam pan Hlemýžď. Nikdo ale neotvíral. Ferda zabouchal ještě jednou a silněji. V tom se domek nadzvedl a hlemýžď vystrčil hlavu: „Ty uličníku! To mi nedáš chvíli pokoj? Jak se mám soustředit?“ A rychle zmizel zpátky.

Není divu, že byl hlemýžď tak rozzuřený. Vždyť zrovna uvažoval nad zajímavou úlohou.

Úloha 5.1 – Hlemýžďův problém (4b)

Nalezněte dvě neprázdné disjunktní množiny přirozených čísel A a B takové, že součet k-tých mocnin prvků množiny A se rovná součtu k-tých mocnin prvků množiny B pro každé přirozené $1 \leq k \leq 2011$.

Ferdovi se podařilo najít kámen a pod ním skulinu. Tam se před dešťem schoval. A usnul.

„Takhle to dál nejde, postavím si domeček,“ rozhodl se Ferda. A jen se rozednilo, dal se do stavění. A že mu práce šla od ruky. Za chvíli byl domeček hotový. A i nábytek si vyrobil. Bodejť by ne, když má takové vybavení…

Úloha 5.2 – Provázek (3+2b)

Ferda nemá jen tak nějaký provázek. Ten jeho je dokonale tenký, dokonale ohebný a dokonale pevný.[1] S takovým se to panečku pracuje. Označme jeho hmotnost M a délku L. Ferda si provázek odložil na stůl tak, že přesahoval okraj stolu délkou D, pohodlně se usadil a zálibně si prohlížel svůj domeček. Když v tom mu provázek spadl dolů na zem. Jak dlouho trvalo než provázek spadl? (3b) Třecí a odporové síly při řešení této otázky zanedbejte.

A mohlo by se stát, že by provázek na zem nespadl vůbec? Pokud ano, za jakých podmínek? (2b)

Jednou za Ferdou přišla paní ploštice, jestli by jí neudělal pro její děti nějaké hračky. Ferda ochotně vyskočil a hned se dal do díla.

Úloha 5.3 – Děti paní ploštice (4b)

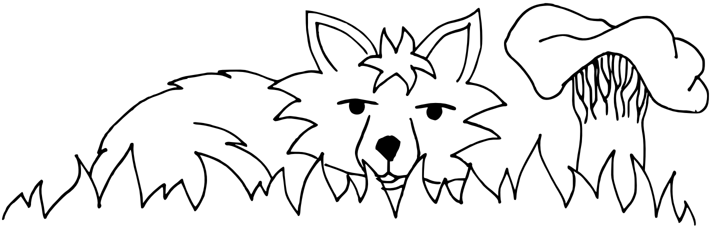
Paní ploštice měla 1111 dětí. Aby je snadno mohla rozlišit, tak si je očíslovala. Každému přidělila jedno přirozené číslo $1, 2, \ldots, 1111$. Jednou si děti hrály a zkoušely, jestli se dokáží postavit všechny do řady tak, aby mezi dvěma dětmi s čísly A a B nikdy nestálo takové, jehož číslo C je rovné aritmetickému průměru A a B.[2] Může se jim to podařit?

Když byly už malé ploštičky vydováděné, vymyslel jim Ferda následující hlavolam.

Úloha 5.4 – Hlavolam (3b)

Namaloval do písku pravidelný 144úhelník. A zajímalo ho, kolik existuje různých trojúhelníků,[3] jejichž vrcholy jsou vrcholy tohoto 144úhelníku. Uměli byste to spočítat?

Protože víc hlav víc ví, ploštičky rychle spočítaly správnou odpověď. Poděkovaly Ferdovi a utíkaly zase na houpačky a na kolotoče.



[1] Nedá se roztáhnout, ale lze libovolně ohnout.

[2] Děti A a B nemusí být bezprostředními sousedy C. Chceme, aby žádné z dětí mezi A a B (těch může být spousta) nemělo číslo rovné $(A+B)/2$

[3] Tojúhelníky lišící se pouze otočením nebo překlopením za různé nepovažuje.

Řešení témat

Téma 1 – Rekurentní posloupnosti

Nejdříve bych se chtěl omluvit za chybu, která se vyskytla ve článku k prvnímu tématu ve 3. čísle našeho časopisu. Rekurentní posloupnost se všemi členy mezi 0 a 1, kterou zaslala Mgr.MM Eva Gocníková, má být správně definována vztahem $A_{n+2} = \dfrac{A_{n+1} + An}{2}$ s počátečními podmínkami $A_0 = 0$ a $A_1 = 1$.

Nyní se ale pojďme zabývat nově došlými články.

Součty posloupností

Hned tři řešitelé se snažili nějak sečíst posloupnosti uveřejněné v třetím čísle. Konkrétně se jednalo o poslopnosti, které měly mít každý třetí člen záporný a konečný součet. Bc.MM Vladimír Sedláček nabídl posloupnost

$$A_1 = 1, A_2 = 0, A_3 = -1, A_{n+3} = A_n \cdot A_{n+1} \cdot A_{n+2} + A_n$$

a Mgr.MM Eva Gocníková zaslala posloupnost

$$A_0 = 0, A_1 = 1, A_{n+2} = -A_n - A_{n+1}.$$

Ačkoli jsou rekurentní předpisy rozdílné, v obou posloupnostech se periodicky opakuje 0, 1 a -1.

Abychom mohli počítat součet, musíme si nejdříve říct, co to vlastně je. Součet posloupnosti (či přesněji řady) se definuje tak, že posloupnost začneme sčítat od začátku a díváme se, k čemu se nám to blíží. Pokud se neustále přibližujeme víc a víc k nějakému jednomu číslu, označíme toto číslo za součet. Formálně pro posloupnost $\{a_k\}_{k=1}^{\infty}$ definujeme její součet jako

$$\lim_{n\to\infty} \sum_{k=1}^{n} a_k \ .$$

Pokud tomuto vzorečku nerozumíš, tak to vůbec nevadí. Říká přesně to, co je popsáno výše.

Pojďme se podívat, jak to vypadá se součtem našich posloupností. Jak správně napsal Mgr.MM Jan Bok, ani jedna z těchto posloupností nemá součet definovaný. Pokud sečteme vždy prvních n členů pro $n = 1,\ 2,\ 3\ldots$, dostaneme postupně $1,\ 1,\ 0,\ 1,\ 1,\ 0,\ 1,\ 1\ldots$ pro posloupnost Bc.MM Vladimíra Sedláčka a $0,\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1\ldots$ pro posloupnost Mgr.MM Evy Gocníkové. Tedy se nám budou vždy pravidelně střídat jedničky a nuly. To ale znamená, že není žádné číslo, ke kterému bychom se neustále víc a víc přibližovali.

Vidíme tedy, že zaslané posloupnosti ze třetího čísla nemají součet a zadaným podmínkám nevyhovují.

Mgr.$^{\text{MM}}$ Jakub Kubečka a Petr Vincena přistupovali k úloze trochu intuitivněji a oba došli k závěru, že by posloupnosti nějaký součet mít mohly. I na tom je něco pravdy.

Pokud chceme mít možnost sčítat více posloupností, můžeme si definici součtu posloupnosti (přesněji řady) rozšířit na tzv. cesarovský součet. Ten je definován jako

$$\lim_{n\to\infty} \frac{1}{n}\left(s_1 + \ldots + s_{n-1}\right), \qquad \text{kde} \qquad s_n = \sum_{k=1}^{n} a_k.$$

Při této definici už opravdu umíme obě posloupnosti sečíst. Součet první posloupnosti bude $2/3$ a druhé $1/3$. Zkuste si rozmyslet, že pokud má posloupnost klasický součet, tak má i cesarovský a tyto součty se rovnají.

Součet 42

Jedním z úkolů zadaných minule bylo najít posloupnost, která bude mít součet roven 42. S tímto problém si všichni, kteří se o to pokoušeli, hravě poradili. Například Mgr.$^{\text{MM}}$ Jan Bok nabízí posloupnost $A_1 = 42/2$ a $A_{n+1} = A_n/2$. Jedná se o geometrickou posloupnost, kterou snadno sečteme.

O aritmetickém průměru

Posledním příspěvkem, který zde zmíním, byl článek Mgr.$^{\text{MM}}$ Evy Gocníkové. Ta zkoumala posloupnost s rekurentním předpisem

$$A_{n+2} = \frac{A_{n+1} + An}{2}.$$

Tedy každý člen této posloupnosti je aritmetickým průměrem dvou předchozích členů. Počáteční podmínky ale volila jako parametry $A_0 = m, A_1 = n$, kde m, $n \in R$. Ocitujeme zde jen závěr podrobného rozboru.

Číslo m	**Číslo n**	**Jak budou vypadat členy posloupnosti**
$m \in \mathbb{R}$	$n = m$	Dostaneme konstantní posloupnost tvořenou čísly m.
$m \in \mathbb{R}^+{}_0$	$n \in \mathbb{R}^+$	Členy budou ležet mezi m a n a budou všechny kladné.
$m \in \mathbb{R}^-{}_0$	$n \in \mathbb{R}^+$	Členy budou ležet mezi m a n a budou všechny záporné.
$m \in \mathbb{R}^-$	$n \in \mathbb{R}^+$	
a) $\lvert m\rvert < n$		První člen bude záporný a ostatní členy budou kladné.
b) $\lvert m\rvert > 3n$		První člen bude záporný, druhý kladný, od třetího budou všechny záporné.
c) $n < \lvert m\rvert < 3n$		V tomto případě bude první člen záporný, druhý kladný, třetí záporný, čtvrtý kladný. O dalších členech už nevíme nic.

Další inspirace

Pokud se vám zdá téma rekurentních posloupností již vyčerpané, můžete si najít cokoli, co s ním alespoň trochu souvisí. A pokud o tom napíšete pěkný článek, rádi ho otiskneme. Obecně zkuste psát své příspěvky jako otisknutelné články. Návod, jak by měl takový článek vypadat nejdete třeba v prvním čísle loňského ročníku. Myslím, že stále máme množství nevyřešených problémů. Například stále nikdo nepřišel s tím, jak spočítat n-tý člen u posloupnosti B_n zadané v prvním čísle. A můžete si pohrát s různými definicemi součtu řady. Najdete třeba řadu s racionálními členy, jejíž součet (nebo cesarovský součet) bude iracionální? Můžete si také zkusit zadefinovat vlastní součet. A zamyslet se, co by vlastně takový součet řady měl splňovat.

Kuba

Téma 2 – Mapování

Druhé mapování, při kterém se měřilo množství srážek, probíhalo na šesti místech naší republiky. Všem řešitelům, kteří se zapojili, děkujeme a odměňujeme je jedním bodem. Výsledky měření jsou uvedeny v tabulce t2.3.1 a na Internetových stránkách tohoto tématka [4]. Jak je vidět, tak v oblasti jižního Valašska a Slovácka 15. prosince slabě sněžilo. Oproti tomu v Praze a středních Čechách sněžilo vydatně. V podhůří Orlických hor byly srážky nulové.

Autor	**Místo měření**	**GPS souřadnice místa měření**	**Srážky [mm]**
Alena Bušáková	Praha-Holešovice	50°7,0 N, 14°26,5 E	6
Kateřina Jiráková	Slavičín	49°5,3 N, 17°52,4 E	1,3
Petr Pecha	Valašské Klobouky	49°8,0 N, 18°0,0 E	1
Lukáš Jančařík	Hovězí	49°16,5 N, 18°2,7 E	0
Jakub Kubečka	Nymburk		7,5
Kristýna Kohoutová	Žamberk	55°5,3 N, 16°27,8 E	0,7

Tabulka t2.3.1: Data naměřená při měření srážek 15. prosince 2010.

Měření probíhalo na velice málo místech, proto nebude snadné data zpracovat, přesto bychom byli velice rádi, kdyby se našel někdo, kdo by se o to pokusil. Může k tomu použít návodu z třetího čísla tohoto ročníku. Zároveň bychom přivítali, kdybychom dostali od některého ze čtenářů návrh na další mapování.

Kromě zpracování dat by bylo zajímavé zamyslet se nad faktory, které mapování srážek mohlo ovlivnit. Doc.ᴹᴹ Alena Bušáková v poznámce ke svému

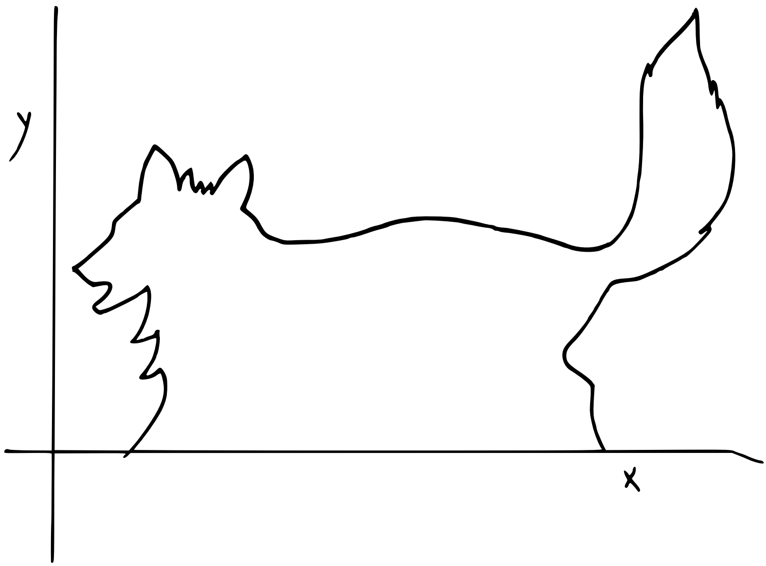
[4] `http://mam.mff.cuni.cz/mapovani`

měření uvádí, že srážky spadly ve formě sněhu. Před měřením objemu počkala, než sníh roztál a ohřál se na pokojovou teplotu. Uvádí, že hustota vody závisí na teplotě, tedy na teplotě závisí i objem srážek, byť „nejspíš nepatrně“. Také Kristýna Kohoutová uvádí, že padaly sněhové srážky, což můžeme, vzhledem k průměrných denním teplotám, předpokládat také u dalších měření. Je tedy zřejmé, že objem srážek byl měřen při různých teplotách. Jak moc mohla teplota vody při měření ovlivnit objem? Pokuste se to určit trochu lépe než jen konstatováním „nejspíš nepatrně“.

Mgr.[MM] Jakub Kubečka v poznámce ke svému měření uvádí, že měření záměrně neprováděl ve městě, ale hned vedle něj. A to z toho důvodu, že si myslí, že budovy by mohly zakrývat nádobu před větrem, a tak by do nádoby napadalo méně sněhu. Zkuste se zamyslet nad tím, jak může mít okolí vliv na množství srážek. Jak závisí výsledky na povětrnostních podmínkách?

Námětů k přemýšlení je dost. Dat na zpracování také. A proto budeme velice rádi, pokud nám od vás přijde nějaký článek.

(R)adim



Řešení úloh

Úloha 3.1 – Tajuplný ostrov (4b)

Zadání:

Vlna, která se mi přelila přes obličej, mě probudila. V poledním slunci jsem se pomalu rozkoukával. Zjistil jsem, že jsem na pláži, kde kromě mě nic není. „Naše loď musela ztroskotat,“ blesklo mi hlavou. „Asi bych nejdřív měl prohledat pobřeží, abych zjistil, zda moře nevyplavilo i něco ze zásob, které byly na lodi,“ napadlo mě při vzpomínce na knihu o Robinsonovi.

A opravdu, po necelé půlhodině hledání jsem zahlédl nějaký předmět, který se leskl v záři slunce. Ukázalo se, že se jedná o mou sbírku úloh z matematiky, která byla velice promočená. Knihu jsem opatrně otevřel. Na první pohled mě zaujal následující příklad.

Číslo 2^{2010^2} *má* 1216192 *cifer a jeho první cifra je* 1*. Uměli byste určit, kolik z čísel* 2^0, 2^1, …, 2^{2010^2-1} *začíná cifrou* 1*? A co kolik jich začíná cifrou* 4*? Všechna čísla uvažujeme v desítkové soustavě.* [5]

„Jídlo počká,“ řekl jsem si. Usadil jsem se na nedaleký kámen a zamyslel se nad tímto příkladem…

Řešení:

Snadno si všimneme, že první mocnina dvojky s daným počtem cifer začíná vždy jedničkou. V opačném případě bychom tuto mocninu dvojky vydělili dvěma a dostali menší číslo se stejným počtem cifer, což není možné. Zřejmý je i fakt, že mezi čísly s daným počtem cifer existuje nejvýše jedna mocnina dvou začínající na jedničku. To znamená, že čísel začínajících jedničkou je tolik, kolik cifer má nejvyšší číslo ze zadané množiny, což je 1216191.

Druhá část je podstatně zajímavější. Ukážeme dva přístupy, jak se dala úloha řešit.

1. možnost: Všimneme si, že pokud první cifra 2^k je 1, tak potom první cifra 2^{k+1} musí být buď 2 nebo 3 a první cifra 2^{k+2} je 4, 5, 6 nebo 7. To znamená, že v zadané množině je 1216191 čísel začínajících 2 nebo 3 a 1216191 čísel začínajících 4, 5, 6 nebo 7. Naopak jiným způsobem žádné z čísel začínajících na 1 až 7 získat nemůžeme. Tedy čísel začínajících na 8 nebo 9 je $2010^2 - 3 \cdot 1216191 = 391527$. Nyní si už stačí uvědomit, že mocnina dvou začíná na 8 nebo 9 právě tehdy, když její polovina začíná na 4. Tím je úloha vyřešena.

2. možnost: Zadanou množinu rozdělíme na bloky

$$\{2^0, 2^1, 2^2, 2^3; \quad 2^4, 2^5, 2^6; \quad 2^7, \ldots 2^{2010^2-1}\}$$

takovým způsobem, aby každý blok začínal číslem s první cifrou 1 a dále v bloku už žádné číslo jedničkou nezačínalo. Protože 2^{2010^2} začíná jedničkou, budou všechny bloky kompletní. Na jedničku začíná 1216191 zadaných mocnin dvojky, proto bude tolik i bloků. Všimneme si, že každý blok má buďto tři nebo čtyři prvky. Pokud má blok tři prvky, tak jejich první cifry jsou 1, 2 nebo 3 a 5 nebo 6 nebo 7. Pokud má čtyři prvky, pak jejich první cifry musí být 1, 2, 4 a 8 nebo

[5] Exponenty bývá zvykem závorkovat odshora, tedy 2^{2010^2} znamená $2^{\left(2010^2\right)}$.

9. Takže nám stačí zjistit počet bloků se čtyřmi prvky. Označme x počet bloků se třemi prvky a y počet bloků se čtyřmi prvky. Potom platí

$$3x + 4y = 2010^2 \quad \text{a} \quad x + y = 1216191.$$

První rovnice počítá celkový počet zadaných mocnin dvojky, druhá počet bloků. Nyní již snadno dopočítáme, že $y = 391527$.

Tedy na cifru 4 začíná 391527 ze zadaných čísel.

Pozn. red.: První snazší část úlohy vyřešilo mnoho řešitelů, správný výsledek druhé části poslalo pouze pět lidí. Z toho tři se nezalekli velkých čísel a využili pro řešení počítač. Za pěkné matematické řešení si zaslouží pochvalu Mgr.[M&M] Alena Harlenderová a Mgr.[M&M] Le Anh Dung.

Kuba

Úloha 3.2 – List (4b)

Zadání:

Po příjemně strávené hodince nad příkladem jsem usoudil, že by bylo vhodné se konečně pustit do prohledávání okolí. Vešel jsem do nedalekého lesa, kde jsem narazil na starou studnu. „Jak může být hluboká?“ říkal jsem si. „To by se určitě dalo lehce změřit, kdybych měl po ruce nějaký kamínek.“ Ten jsem však po ruce neměl. Ovšem kolem bylo plno listí. „To by mohlo pomoct,“ řekl jsem si. Jen bych potřeboval znát, jakou rychlostí padá list…

Změřte dobu, za jakou spadne list z různých výšek na zem. Ne hned od místa „upuštění“, uvažujte jen ustálený pohyb. Ono se to ustálí rychle, uvidíte…

List aproximujte kruhovým lístkem papíru a zkuste tento čas spočítat. Zamyslete se nad tím, proč se experimentální výsledek od teoretického liší. Návod, jak zpracovávat fyzikální měření, naleznete v M&M ve 2. čísle XVI. ročníku.

Řešení:

Než se pustím do samotného řešení, chtěla bych se vám omluvit za dvě věci. Jednak za to, že vlivem běžných zpoždění došlo k tomu, že v době, kdy bylo číslo se zadáním této úlohy vydáno, už mnoho listí nebylo. Zadávala jsem ji ve chvíli, kdy teprve začínalo padat, a nedošlo mi, že se k řešení dostanete podstatně později. Nicméně poradili jste si. Druhá věc je to, že jsem asi špatně formulovala zadání, protože jste ho vesměs nepochopili. Jaksi jsem automaticky očekávala, že když jde o úlohu evidentně o odporu vzduchu, tak vás napadne ho zahrnout i do teoretického výpočtu, když je ten vzoreček tak jednoduchý… a pak uvidíte, jestli funguje pro list dostatečně přesně. Jediný, koho to napadlo, byla Alča Bušáková, jejíž naprosto luxusní řešení předkládám jako vzorové. Lépe už totiž úloha vyřešit ani nejde.

Výsledky velmi závisí na vhodné volbě pokusného materiálu. Nejvhodnějším kandidátem se mi zdál být zelený list skoro kulatého tvaru, jen u stopky (kterou jsem též utrhla) se znatelným zářezem.

Co se dělo poté: házela jsem list z výšky a měřila čas. Zjistila jsem, že někdy list padá rovně, snáší se a plachtí, ale neotáčí se, jindy rotuje jako šílený, proto jsem se rozhodla si to zaznamenávat a rozebrat obě varianty zvlášť.

Čas jsem měřila na stopkách s přesností na jednotky setin sekundy, nicméně u takto nízkých časů (okolo jedné sekundy) se výrazně projeví opoždění mé reakce, které jsem měřila tak, že jsem zkoušela stopky co nejrychleji vypnout a zapnout. Při asi 10ti pokusech byl průměrný čas 14 setin sekundy, nejlepší dosažený 13 setin (a nejhorší dosažený oněch 14). Nicméně toto zpoždění má stejnou hodnotu při spouštění stopek (jelikož se tak dělo v reakci na výšku a ustálenost listu) jako při jejich zastavení (to je zase reakce na dopadnutí na podlahu), takže jsem ho původně nechtěla uvažovat, ale je klidně možné, že se člověk s reakcí „předběhne“, třeba při dopadu na podlahu stopne dřív, proto celkovou chybu určím jako součet, tedy 26,5 setin sekundy (0,5 je chyba měření).

Ohledně měření výšky jsem přišla na to, že když se pokouším zmáčknout stopky v nějaké výšce, většinou je zmáčknu, až když je list asi o 10 cm níž, tudíž ač původní záměr byl metr a metr a půl, skončila jsem na 90 a 140 centimetrech, odchylka měření je 0,5 mm, odchylku metody odhaduji na 5 cm, tedy celkem to činí 0,0505 metru.

Je na čase spočítat statistickou chybu měření. Nutno dodat, že list nemá absolutně žádný důvod padat pořád stejně rychle, neboť někdy rotuje, jindy nerotuje vůbec, a když rotuje, tak velmi různě chaoticky.

U padání z výšky 90 cm se mi ze 13 pokusů podařilo pustit list tak, aby nerotoval, pouze dvakrát, přičemž tato hodnota byla náhodně rozprostřena mezi ty, kdy rotoval, tudíž to nemělo žádný vliv.

Při padání z výšky 140 cm se mi podařilo z 11 pokusů, aby list čtyřikrát nerotoval, přičemž doba padání pak byla výrazně jiná a stálejší, než když rotoval, tudíž jsem tyto dvě situace počítala zvlášť.

Nyní k vlastnímu výpočtu. Vzoreček nám praví

$$v_m = \sqrt{\frac{2mg}{C\varrho S}},$$

kde v_m, je mezní rychlost (kterou se při pádu těleso pohybuje aniž by už zrychlovalo), m je hmotnost tělesa, g je tíhové zrychlení, C je součinitel odporu, ϱ je hustota vzduchu a S je plocha padajícího předmětu. Jelikož měřím jen když padá ustáleně, tedy rovnoměrně, platí

$$t = \frac{s}{v_m} = \frac{s}{\sqrt{\frac{2mg}{C\varrho S}}},$$

přičemž víme, že

- $s = (90 \pm 5)\,\mathrm{cm}$ – tak, jak jsme to měřili, viz výše
- $m = 0{,}128\,\mathrm{g}$ – hmotnost listu; jelikož předměty o takto malé hmotnosti nemám čím vážit, vyrobila jsem si rovnoramennou váhu (primitivní, ale značně citlivou), na jednu stranu jsem položila list, a druhou kousek papíru, a zjistila jsem, že musí být velký $4\times4\,\mathrm{cm}^2$, aby byla váha v rovnováze. Chyba by zde mohla být způsobena špatným odhadnutím a umístěním těžiště listu, nicméně to bude chyba vskutku zanedbatelná. Hmotnost jsem spočítala tak, že m^2 standardního kopírovacího papíru váží 80 g.

- $g = 9{,}81\,\mathrm{m/s^2}$
- $C = 1{,}28$ – údaj pro „flat plate perpendicular to flow 3D“ dle anglické wikipedie
- $\varrho = 1,2759\,\mathrm{kg/m^3}$ – při normálním tlaku a teplotě 0° C, (dle běžných školních tabulek)
- $S = 12{,}8\,\mathrm{cm^2}$ – přičemž jsem to spočítala jako obsah elipsy, jejímuž tvaru tvar listu velmi dobře odpovídá – $1\,\mathrm{cm^2}$ je hrubý odhad vykousnutí. Chyba systémová se sestává z odchylky pravítka 0,5 mm, chyba metody bude nejspíš tkvět v odhadu obsahu vykouslé části, kteroužto chybu bych stanovila odhadem na nejhůře $0{,}25\,\mathrm{cm^2}$. Statistickou chybu nemám, protože měření považuji za snadný úkol, takže by byla beztak mnohem menší než systémová, a už mě to měření moc nebaví, takže bych si připadala opravdu divně, ještě něco 10× měřit pravítkem… Nevím ale, jak ty odchylky ,„složit“ i s tím milimetrem, takže nechám jen $(12{,}8 \pm 0{,}25)\,\mathrm{cm^2}$.

Pozn. red.: Alča se v této situaci ohledně chyb zachovala velice správně. Ve škole se buzeruje s několikanásobným měřením pravítkem jenom proto, že je potřeba „něco“ rychle změřit, aby zbylo dost času na výuku počítání chyb, a měření délky je dostatečně jednoduché a rychlé. V praxi však počítat statistickou chybu měření délky rozumným měřidlem pro danou situaci nemá příliš smysl, chyba měřidla je zpravidla větší a statistická je vůči ní zanedbatelná. Jinak pokud by se měly kombinovat statistická a systematická chyba, prostě by se sečetly.

Celkový čas pádu potom vychází z 90 cm:

$$(0{,}90 \pm 0{,}05)\,\mathrm{m}\sqrt{\frac{2 \cdot 0{,}000128\,\mathrm{kg} \cdot 9{,}81\,\mathrm{m/s^2}}{1{,}28 \cdot 1{,}2759\,\mathrm{kg/m^3} \cdot (0{,}00128 \pm 0{,}000025)\,\mathrm{m^2}}}$$

Neumím dle návodu v čísle započítat chybu pod odmocninou, která je jinou chybou dělena a dole se násobí…, bez ní to vyjde přibližně 0,8211194 s.

Pozn. red.: Spojení slova přibližně a čísla uvedeného na více než dvě platné cifry je tak trochu oxymóron.

A ze 140 cm:

$$\frac{(1{,}40 \pm 0{,}05)\,\mathrm{m}}{\sqrt{\frac{2\cdot 0{,}000128\,\mathrm{kg}\cdot 9{,}81\,\mathrm{m/s^2}}{1{,}28\cdot 1{,}2759\,\mathrm{kg/m^3}\cdot(0{,}00128\pm 0{,}000025)\,\mathrm{m^2}}}} = 1,277297\,\mathrm{s}$$

Na první pohled by to měl být výsledek rozumný… Ačkoli i ve výpočtu jsou některé veličiny dosazovány po experimentálním zjištění, takže zde je určitě nezanedbatelná chyba.
A co jsme naměřili?

Pro výšku $(0{,}9 \pm 0{,}0505)\,\mathrm{m}$ vychází průměrná doba 0,7900 s, směrodatná odchylka 0,0328165 s, systémová chyba je 0,265 s, tedy výsledný čas je $(0{,}8 \pm 0{,}2)\,\mathrm{s}$.

Pro výšku $(1{,}4 \pm 0{,}0505)\,\mathrm{m}$, když se list netočí, vychází průměrná doba 1,1575 s, směrodatná odchylka 0,0125 s, systémová chyba je 0,265 s, tedy výsledný čas je $(1{,}16 \pm 0{,}03)\,\mathrm{s}$.

Pro výšku $(1{,}4 \pm 0{,}0505)$ m, když se list točí vychází průměrná doba 1,00 s, směrodatná odchylka 0,0624 s, systémová chyba je 0,265 s, tedy výsledný čas je $(1,0 \pm 0,3)$ s.

Nutno říct, že pro točící se list jsem to přepočítala i bez prvního měření, které bylo velmi „ulítlé“ nad 3 sigma, ale i poté se opět vyskytlo dokonce více hodnot mimo hranici 3 sigma, takže jsem seznala, že chci-li, aby mi tam nějaké hodnoty zbyly, musím toto pravidlo odignorovat, což je i relativně rozumné vzhledem k tomu, že list opravdu letěl pokaždé úplně jinak.

Zajímavé je, že rovně padající list má doby dopadu daleko méně rozkývané, a potěšilo mě, že lépe odpovídá spočtenému výsledku. Experimentální výsledek má za prvé chybu ještě větší, neb není jista výška, z níž list padal, a za druhé neodpovídá teoretickému výsledku především kvůli rotacím listu při pádu a dále kvůli tvaru, který má na okraji drážky, kudy vzduch dobře klouže, takže se mu padá lépe než dokonale kulaté věci.

Pozn. red.: Konkrétní naměřené hodnoty neuvádíme.

Zuzka

Úloha 3.3 – Palisáda (4b)

Zadání:

Ostrov každoročně sužují nájezdy pirátů. Proto se místní domorodci rozhodli postavit na pobřeží palisádu, aby piráti nemohli přistát. Před stavbou však potřebují zjistit, jak je dlouhé pobřeží ostrova, aby si mohli připravit dostatek dřeva.

Mapování na ostrově nedosáhlo moc dobré úrovně, tak je k dispozici pouze čtverečkovaná mapa, kde 0 znamená vodu a 1 označuje suchou zem. Na okraji celé mapy je proužek široký minimálně jeden čtvereček, kde je voda. Úkol je spočítat počet hran, kde moře sousedí s ostrovem. Na ostrově se nachází několik jezer, z nichž samozřejmě piráti neútočí, a proto se do délky palisády nepočítají.

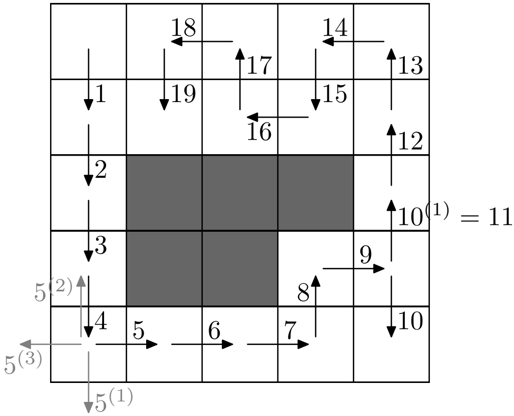
Vzhledem k velikosti ostrova napište program, který spočte délky palisády pro libovolnou mapu, kterou načte ze vstupního souboru `mapa.in` *a uloží ji do výstupního souboru* `mapa.out`*. Toto načítání a ukládání je důležité pro testování správnosti vašeho řešení, neodbývejte ho!*

Řešení:

Na začiatok je dôležité pozorovanie, ktoré mnohí z vás považovali za samozrejmé – len pole vody, na ktoré je možné sa dostať z okraja (mora) cestou cez vzájomne susediace polia vody, je morom a je nutné ho opalisádovať. Kde prepojenie s okrajom neexistuje, tam je voda jazerom. Zároveň je vidieť, že dĺžka palisády je rovná počtu hrán štvorcovej siete, kde more susedí s pevninou. Takže nám stačí pre každé pole mora spočítať, s koľkými pevninami susedí. Tým sa problém redukuje na otázku, ako „navštíviť“ každý vrchol súvislého grafu (každé pole mora).

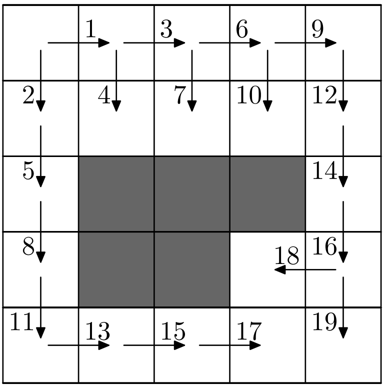
Základná ponuka metód obsahuje prehľadávanie do šírky a prehľadávanie do hĺbky. Tieto metódy sa pri tom líšia v podstate iba poradím, v ktorom polia prechádzame. Predveďme si názorne oba spôsoby.

Spoločný postup vyzerá asi takto: Začneme v poli, o ktorom s istotou vieme, že je morom (túto podmienku spĺňajú podľa zadania všetky body okraja), označíme pole ako navštívené, za každú susediacu pevninu zvýšime dĺžku palisády o jedna a napokon vykonáme to isté pre všetky susediace polia. Rozdiel medzi oboma algoritmami je vlastne len v poslednom bode.



Obr. u3.3.1

Pri prehľadávaní do hĺbky ukladáme ďalšie polia do štruktúry LIFO[6]. Lepší, než zložitý slovný popis je obrázok. Na obrázku u3.3.1 tak vidíme postup prehľadávania do hĺbky pre poradie navštevovaných polí dole, hore, vľavo, vpravo. Tmavou plochou je vyznačená pevnina. Pre piaty krok sú naznačené aj kroky, ktoré program odmietne kvôli tomu, že padajú mimo mapu ($5^{(1)}$ a $5^{(3)}$), alebo preto, že na cieľovom poli už bol ($5^{(2)}$). Podobne krok 11 je vlastne krok $10^{(1)}$, pretože z poľa po kroku 10 nie je možné pokračovať žiadnym smerom a tak program pokračuje na poli po kroku 9. Namiesto zásobníku môžeme použiť rekurziu (funkcia spustená na jedno pole zavolá samú seba na všetkých nenavštívených susedných). Musíme si byť akurát istí, že nám náš prekladač povolí takú hĺbku rekurzie.



Obr. u3.3.2

[6] Last In First Out – „posledný dnu, prvý von“, čiže zásobník – vyberáme z neho prvok, ktorý sme tam vložili ako posledný.

Pri prehľadávaní do šírky ďalšie polia na prehľadanie ukladáme do štruktúry FIFO[7], čiže do radu. Schéme pre poradie krokov vľavo, hore, vpravo, dole je na druhom obrázku.

Napokon navrhnime schématické riešenie úlohy bez konkrétneho programovacieho jazyka pre prehľadávanie do hĺbky:

```
Načítaj vstup.
Vlož do zásobníka pole (0, 0). (alebo iné pole okraja)
Dĺžka palisády = 0.

Kým nie je zásobník prázdny:
  Vyber zo zásobníka ďalšie súradnice a použi ich ako aktuálne.
  Ak je pole vľavo od aktuálneho na mape:
    Ak je vľavo voda a zároveň nenavštívené:
      pridaj pole vľavo do zásobníka a označ ako navštívené.
    Ak je vľavo pevnina:
      predĺž palisádu o jedna.
  Ak je hore na mape:
    Ak je hore voda a zároveň nenavštívené:
      pridaj pole hore do zásobníka a označ ako navštívené.
    Ak je hore pevnina:
      predĺž palisádu o jedna.
  Ak je vpravo na mape:
    Ak je vpravo voda a zároveň nenavštívené:
      pridaj pole vpravo do zásobníka a označ ako navštívené.
    Ak je vpravo pevnina:
      predĺž palisádu o jedna.
  Ak je dole na mape:
    Ak je dole voda a zároveň nenavštívené:
      pridaj pole dole do zásobníka a označ ako navštívené.
    Ak je dole pevnina:
      predĺž palisádu o jedna.
koniec cyklu "Kým".

Vypíš dĺžku palisády.
```

Pár zdrojových kódov z vašich zaujímavých a v podstate iba správnych riešení nájdete na našich stránkach v sekcii „Z řešení“. Aby sme sa ešte zavďačili informatikom, doplňme, že obe metódy majú lineárnu časovú aj pamäťovú náročnosť. Mýliť nás môže, že množstvo údajov rastie na mape s druhou mocninou (s plochou mapy). Náročnosť algoritmu ale závisí na množstve údajov lineárne a to je podstatné.

Jeffer

[7] First in First Out – „prvý dnu, prvý von“, čiže rad (fronta) – vyberáme prvok, ktorý je na rade a ďalšie prvky pridávame na koniec radu.

Úloha 3.4 – Nápis (2b)

Zadání:

Na stěně bylo napsáno toto:

$$(10 \bullet \bullet \bullet 10 \bullet \bullet)_2 \mathbf{\ mod\ } (18)_{10} = 0\,.$$

Poznáte, co bylo původně napsáno tam, kde je nyní napsáno $\bullet$*? Nalezněte všechna řešení.*

Řešení:

Naším úkolem je doplnit číslice A_2, B_2, C_2, D_2 a E_2 u čísla $(10ABC10DE)_2$ tak, aby bylo číslo dělitelné 18. To znamená, že číslo musí být dělitelné dvěma a zároveň devíti[8].

Číslo zapsané v dvojkové soustavě si můžeme přepsat následujícím způsobem

$$(10ABC10DE)_2 = 256 + 64 \cdot A_2 + 32 \cdot B_2 + 16 \cdot C_2 + 8 + 2 \cdot D_2 + E_2\,.$$

O dělitelnosti dvěma rozhoduje pouze číslice E_2, neboť všechny ostatní číslice jsou násobky dvou. Proto se E_2 musí rovnat 0.

Dělitelnost devíti značí, že tímto číslem musí být dělitelný ciferný součet čísla. Uvědomíme si, že dělení a tedy i modulo je distributivní vůči sčítání. Nejdříve sečteme čísla 256 a 8 a určíme jejich zbytek po dělení devíti

$$(256 + 8) \mathbf{\ mod\ } 9 = 3\,.$$

Hledáme tedy kombinaci číslic A_2, B_2, C_2 a D_2 tak, aby

$$(64 \cdot A_2 + 32 \cdot B_2 + 16 \cdot C_2 + 2 \cdot D_2) \mathbf{\ mod\ } 9 = 6\,.$$

$$64 \mathbf{\ mod\ } 9 = 1 \qquad 32 \mathbf{\ mod\ } 9 = 5$$
$$16 \mathbf{\ mod\ } 9 = 7 \qquad 2 \mathbf{\ mod\ } 9 = 2$$

Nyní musíme rozebrat jednotlivé případy. Je zřejmé, že minimálně dvě z číslic musí být jednička, neboť žádný zbytek se nerovná šesti. Pokud by se A_2 rovnalo jedné, tak by spolu s B_2 daly zbytek šest. Dostáváme první řešení

$$(101101000)_2 \mathbf{\ mod\ } (18)_{10} = 0\,.$$

Zároveň je dobré si všimnout, že součet zbytků u C_2 a D_2 dává devět, takže by se nezměnila dělitelnost devíti. Máme tak druhé řešení

$$(101111010)_2 \mathbf{\ mod\ } (18)_{10} = 0\,.$$

Žádná z dalších dvojic či trojic nedá součet, jehož zbytek je šest. Máme tak pouze dvě řešení.

Pozn. red.: Vzhledem k jednoduchosti úlohy byly body za řešení rozdávány stylem „méně je někdy více“. Pokud jste pouze mechanicky vyzkoušeli všechna možná řešení, tedy i liché, tak jste se dočkali jednoho bodu. Ti, kdo se pokusili o nějakou úvahu, se můžou radovat ze dvou bodů.

(R)adim

[8] Čísla v dvojkové soustavě budeme zapisovat $(\cdot)_2$, čísla v desítkové soustavě budeme zapisovat normálně. Číslice A_2, $B_2 \ldots E_2$ jsou v dvojkové soustavě a můžou tak nabývat pouze hodnot 0 nebo 1.

Úloha 3.5 – Seriál o Pythonu (III. díl) (5b)

Zadání:

I (1b): *Napište regulární výraz, který bude odpovídat zápisu teploty (tj. číslo a jednotka). Nezapomeňte, že pro měření teploty máme různé jednotky.*

II (4b): *Napište program, který prolézá webové stránky a sbírá z nich emailové adresy. Využijte* `urllib` *k přístupu na internet a regulární výrazy pro hledání emailových adres a odkazů na další stránky k prozkoumání. Nezapomeňte ošetřit případné chyby a vadné odkazy, prohledávat jen do určité hloubky či počtu stránek a omezit opakované návštěvy téže stránky. Taky nám k programu pošlete stručný návod k použití.*

Pokud budete zkoušet svůj program na obsah cizích stránek, zařaďte mezi přístupy nějaké zpoždění pomocí `time.sleep`*. Pokud nehlídané prolézací skripty „zdivočí“, může to někomu znepříjemnit život. Mechanické zpracování dat je velmi užitečné, ale nezapomínejme na ohleduplnost (na piráty si zatím jen hrajeme).*

Za další luxusní vlastnosti programu nabízíme samozřejmě body navíc.

Řešení:

I (1b): Zápis teploty se skládá z čísla a jednotky. Pomineme-li zápisy v různých číselných soustavách, musíme povolit čísla kladná i záporná, celá i desetinná a navíc i čísla v semilogaritmickém zápisu. Mezi číslo a jednotku patří mezera. Jednotek pro měření teploty máme celou řadu. Kromě stupňů celsia a fahrenheita, které se píšou se stupněm máme taky kelviny, které se píšou bez stupně. Existují však i další jednotky. Příklad regulárního výrazu, který splňuje výše popsané podmínky, je

```
^(-?(\d+([.,]\d+)?(([eE]|x10\^)[+-]?\d+)?)\s(°(F|C|R[aéeø]?|De?|N)|K))$
```

II (4b): Řešení této úlohy přišla řada. Některé nedokázaly sebrat adresu `mam@atrey.karlin.mff.cuni.cz`, jiné spadly, protože neodchytávaly výjimky v případě nefunkčního odkazu. Některé uměly rozšifrovat i relativní odkazy, většina se však chytala pouze absolutních. Téměř žádný však neobsahoval požadovaný návod k použití, neboli dokumentaci. Většina programů se užívala vcelku intuitivně, ale přesto by pár slov o použití nebylo od věci – dokumentace není samoúčelná, ale má svůj hlubší smysl. :-)

Jako vzorové řešení jsme se rozhodli uveřejnit program Dr.$^{\mathrm{MM}}$ Tomáše Pokorného, který do svého velice dobře fungujícího programu vložil i nápovědu. Najdete ho na adrese `http://mam.mff.cuni.cz/python/3.5.II.py`.

Pár slov závěrem: formáty url a e-mailových adres jsou popsány v příslušných normách, např. `RFC 3696`. Jejich prostudováním můžeme získat přesnou představu o tom, jak můžou adresy vypadat. Zkuste se zamyslet nad tím, jak e-mailové adresy chránit před sběrači. Ani zakódování obrázkem dnes není příliš účinné, a tak jediné, co člověku zbývá, je pořádný spamfiltr.

Honza

Seriál o Pythonu (V. díl)

V dnešním dílu seriálu se podíváme na pythoní webový framework Django. Jedná se o skutečnou aplikaci, používanou ve skutečném světě a jak sami uvidíte, není to žádné ořezávátko.

Co budeme potřebovat

Předně budeme muset nainstalovat Django. Pokud používáte linux, pravděpodobně ve vaší distribuci existuje balíček jménem `python-django`. Pokud z nějakého důvodu nechcete použít balíčkovací systém, zavítejte na stránku `http://www.djangoproject.com/download/`, kde si můžete stáhnout nejnovější verzi.

Kromě toho budeme pro serióznější práci s webem potřebovat také databázi[9]. Pro účely tohoto článku jsem zvolil Sqlite. Máte-li python alespoň ve verzi 2.5, je v něm již Sqlite zahrnut a nemusíte se o nic starat. Máte-li starší, nainstalujte balíček `sqlite3`, případně `sqlite` (název se může mírně lišit podle distribuce).

Používáte-li Windows, stáhněte si instalační balíček ze stránek `http://www.instantdjango.com/`. Obsahuje Python, Django i Sqlite, takže se o nic dalšího nemusíte starat.

Založení nového projektu

Teď, když máme vše nainstalováno, můžeme vytvořit projekt. Uděláme to příkazem

```
django-admin startproject mujprojekt
```

kde „ujprojekt“ je název našeho projektu. Vytvoří se nám tím stejnojmenný adresář. Než se pustíme do programování, otestujeme, jestli nám Django funguje. Vlezeme proto do adresáře `mujprojekt` a spustíme příkaz

```
python manage.py runserver
```

Tím spustíme zabudovaný webový server Djanga a když zadáme do prohlížeče adresu `127.0.0.1:8000`, měla by na nás vykouknout úvodní stránka.

V adresáři `mujprojekt` se nám vytvořilo několik užitečných souborů[10]. Nejdůležitějším je už použitý `manage.py`. Pomocí tohoto skriptu spravujeme svůj projekt – startujeme server, vytváříme aplikace, obnovujeme databázi, ...

Druhý, neméně důležitý skript se jmenuje `settings.py`. Když si ho otevřete ve svém oblíbeném textovém editoru, zjistíte, že je složený z nastavení různých proměnných, např. `ADMINS`, `TIME_ZONE`, `LANGUAGE_CODE` aj. Nastavíme v něm tyto proměnné:

---

[9] využijeme ji ale až v příštím dílu, zatím její přítomnost není nutná

[10] kdykoliv budu odteď psát jména souborů, budou se nacházet v tomto adresáři

```
TEMPLATE_DIRS = ( "/cesta/k/adresari/se/sablonami" )
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'kalendar',
)
```

Cesta musí být zadána absolutně[11] a oddělena normálními lomítky (i ve Windows). Proměnnou `TIME_ZONE` můžete nastavit na `'Europe/Prague'` a proměnnou `LANGUAGE_CODE` na `'cs'`, resp. `'sk'`, ale není to nutné.

Aplikace

Projekt v Djangu se skládá z dílčích programů, tzv. aplikací. Tyto jsou na sobě značně nezávislé a pokud si dáte trochu práce s jejich návrhem, můžete tyto aplikace používat opakovaně v různých projektech s neobyčejnou lehkostí.

V našem projektu „mujprojekt“ vytvoříme aplikaci „kalendar“:

```
python manage.py startapp kalendar
```

Tím nám v aktuálním adresáři přibyl adresář `kalendar`. Proto jsme do proměnné `INSTALLED_APPS` složili řetězec `'kalendar'` – říkáme tím djangu, že má s touto aplikací v současném projektu počítat.

Vytvoření pohledu

Nyní bude naším úkolem naprogramovat „pohled“, který tvoří jakousi mezivrstvu mezi daty a vzhledem.

Vytvoříme soubor `kalendar/views.py` s tímto obsahem:

```
from django.http import HttpResponse
def index(request):
    return HttpResponse("<h1>Hello World!</h1>");
```

A upravíme soubor urls.py do této podoby:

```
from django.conf.urls.defaults import *

urlpatterns = patterns('',
    (r'^kalendar/$', 'kalendar.views.index'),
)
```

[11] tj. vzhledem ke kořeni vašeho disku, nikoliv k aktuálnímu adresáři

V souboru urls.py jsou uložena pravidla pro překládání webových adres. Prvním parametrem je regulární výraz – pokud tento sedí na zadanou adresu, je provedena druhá část dvojice. Pokud tedy zadáte do prohlížeče adresu `http://127.0.0.1:8000/kalendar/`, zavolá se funkce `index` ze souboru `views.py` v aplikaci `kalendar`.

Nastartujte teď server pomocí příkazu `python manage.py runserver` a podívejte se na adresu `http://127.0.0.1:8000/kalendar/`. Pokud jste všechno provedli správně, měli byste vidět nápis „Hello world!“.

Architektura webových frameworků

Asi si teď říkáte, proč se zatěžovat takovými hroznými složitostmi, když by přece stačilo napsat webovou stránku opakováním příkazu `print`, který by vypisoval jednotlivé tagy. Django je totiž postaven na dnes asi nejpoužívanější architektuře Model—View—Template, neboli Model—Pohled—Šablona. Toto oceníte hlavně ve chvíli, kdy se budete chtít k napsané aplikaci opakovaně vracet a znovu a znovu ji upravovat. Tato architektura důsledně odděluje objektový návrh od obsahu a vzhledu, což ji činí, obzvláště co se týče velkých projektů, neobyčejně přehlednou.

Šablony

Šablony tvoří fasádu aplikace – určují, jak bude vypadat zvenku. Přitom se vůbec nezajímají o to, jakým způsobem se k nim data dostávají a jak se uvnitř aplikace zpracovávají – to mají na starosti pohledy. Než vytvoříme šablonu, upravíme soubor `kalendar/views.py` do následující podoby:

```
from datetime import datetime
from django.shortcuts import render_to_response

def index(request):
        pozdrav = "Buďte zdrávi!"
        return render_to_response('index.html', {'aktualni_cas':
          datetime.now(), 'pozdrav': pozdrav})
```

Funkce render_to_response otevře soubor `index.html`, který se nachází v adresáři se šablonami (tento adresář jsme nastavovali na začátku pomocí proměnné `TEMPLATE_DIRS`) a předá do něj proměnné `aktualni_cas` a `pozdrav`.

Vytvoříme teď v adresáři se šablonami soubor `index.html` a předané proměnné v něm použijeme:

```
<html>
<head>
        <meta http-equiv="content-type" content="text/html;
          charset=UTF-8">
        <title>Ahoj, světe!</title>
```

```
</head>
<body>
        <h1>{{ pozdrav }}</h1>
        <p>Aktuální čas je {{ aktualni_cas|time:'H:i:s' }}</p>
</body>
</html>
```

Když se podíváte na adresu `http://127.0.0.1:8000/kalendar/`, dostanete stránku s pozdravem a aktuálním časem. Pokud tento nesedí, pravděpodobně jste v `settings.py` nenastavili správně časové pásmo.

Řetěžec `time:'H:i:s'` je tzv. filtr – způsob jak upravovat předané proměnné přímo uvnitř šablony. Užitečný je např. filtr default ({{ `x|defaut="42"` }}), který, pokud je proměnná `x` prázdná (nebo není vůbec zadána), vyplní na její místo `42`. Výchozí chování je nevkládat na místo neexistujících proměnných nic. Dalším filtrem, který přijde často vhod je `linenumbers`, který vloží do proměnné obsahující větší počet řádků jejich očíslování. Hromadu dalších filtrů naleznete v dokumentaci a dokonce si můžete napsat i svoje vlastní – k tomu se možná dostaneme v příštím dílu.

Mezi dvojité složené závorky se zapisují proměnné, jejihž obsah chceme do stránky vložit. Všechny ostatní konstrukce (podmínky, cykly, ...) se zapisují mezi {% a %}. Těmto konstrukcím se říká tagy.

Django template language

Tag `for` projde všechny položky pole:

```
{% for osoba in lidi %}
    <li>{{ osoba.jmeno }}</li>
{% endfor %}
```

Tag `if ... else` je klasická podnímka, která testuje, zda je zadaný výraz `True`. V podmínkách jde použít i porovnávání, ale obecně nelze volat funkce Pythonu.

```
{% if lidi %}
    Počet přítomných lidí: {{ lidi|length }}
{% else %}
    Nikdo tu není.
{% endif %}
```

Asi nejsilnější věcí na šablonách v Djangu je jejich dědičnost.
Zjednodušeně řečeno můžeme nadefinovat šablonu, ve které se vyskytují bloky kódu, které můžou být dále pozměňovány podšablonami. Uveďmě si příklad:

```
<html>
<head>
    <title>{% block title %}Moje stranka{% endblock %}</title>
</head>
<body>
    <div id="menu">
        {% block menu %}
        <a href="/">Homepage</a>
        <a href="veverky">O inteligentních veverkách</a>
        {% endblock %}
    </div>
    <div id="content">
        {% block content %}{% endblock %}
    </div>
</body>
</html>
```

Teď si představme, že chceme změnit menu a obsah stránky. Použijeme k tomu tag `extends`, který určuje soubor, ze kterého se bude dědit:

```
{% extends "base.html" %}
{% block menu %}
    <a href="/">Homepage</a>
    <a href="krokodyli">O inteligentních krokodýlech</a>
{% endblock %}
{% block content %}
    {% for prispevek in forum.prispevky %}
        <h2>{{prispevek.title}}</h2>
        <p>{{prispevek.text}}</p>
    {% endfor %}
{% endblock %}
```

A to je pro dnešek vše. Příště se podíváme na modely, práci s databází a automatické generování administrace.

Honza

Úloha 5.5 – Seriál o Pythonu (V. díl) (2b)

Vytvořte jednoduchou knihu návštěv (stačí když bude příspěvky zobrazovat, vkládání si ukážeme až příště). Nějaké příspěvky (testovací, aby bylo vidět, že váš program funguje) načítejte ze souboru. Jeho formát je zcela na vás.

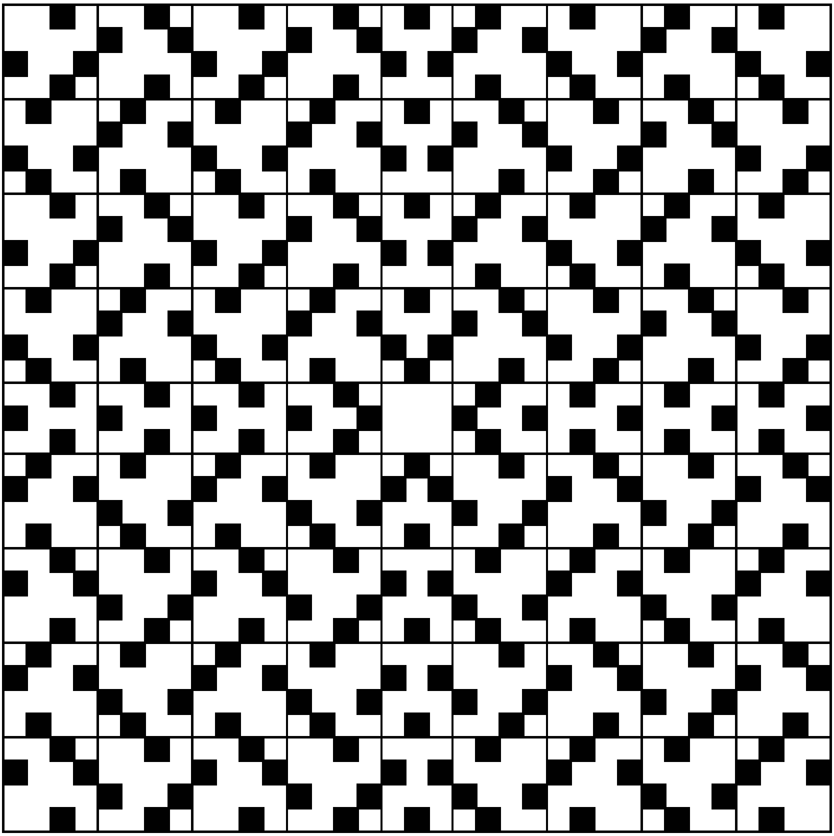
Výsledková listina

Poř.	Jméno	R.	$\sum_{-1}$	Úlohy r1	r2	r3	r4	t1	t2	s3	+	$\sum_0$	$\sum_1$
1.	Prof.$^{\mathrm{MM}}$ P. Pecha	4.	242	4	2	3	2		1	4	0	16	49
2.	Doc.$^{\mathrm{MM}}$ A. Bušáková	4.	148	2	5	4	2		1	3	1	18	48
3–5.	Dr.$^{\mathrm{MM}}$ T. Pokorný	4.	91	1		4	2			8	0	15	39
	Mgr.$^{\mathrm{MM}}$ J. Sopoušek	4.	39	4	3		2			3	0	12	39
	Prof.$^{\mathrm{MM}}$ Š. Šimsa	2.	247	1		4	2			3	1	11	39
6.	Mgr.$^{\mathrm{MM}}$ J. Kubečka	3.	36	2	1		1	3	1		0	8	36
7.	Doc.$^{\mathrm{MM}}$ F. Štědronský	4.	111	2		4	2				0	8	33
8.	Mgr.$^{\mathrm{MM}}$ J. Setnička	4.	30			4	2			5	0	11	30
9.	Mgr.$^{\mathrm{MM}}$ E. Gocníková	3.	23	2	2		2	6			0	12	23
10–13.	Mgr.$^{\mathrm{MM}}$ L. Dung	1.	20	4							1	5	20
	Mgr.$^{\mathrm{MM}}$ A. Harlenderová	3.	46	4			2				0	6	20
	Dr.$^{\mathrm{MM}}$ F. Hlásek	4.	71	4		4	2				0	10	20
	Dr.$^{\mathrm{MM}}$ M. Kocián	4.	51	2		4	1			1	0	8	20
14–15.	Mgr.$^{\mathrm{MM}}$ J. Bok	4.	23	0		4	2	3		0	0	9	19
	Bc.$^{\mathrm{MM}}$ V. Sedláček	3.	19								0	0	19
16.	Bc.$^{\mathrm{MM}}$ M. Töpfer	3.	16								0	0	16
17.	Mgr.$^{\mathrm{MM}}$ K. Zemková	3.	29	2			2				0	4	15
18.	Dr.$^{\mathrm{MM}}$ T. Kubelka	3.	91								0	0	13
19–22.	Bc.$^{\mathrm{MM}}$ L. Grund	1.	12								0	0	12
	Dr.$^{\mathrm{MM}}$ M. Kochmanová	4.	81				2				0	2	12
	Bc.$^{\mathrm{MM}}$ R. Kubíček	2.	12								0	0	12
	Bc.$^{\mathrm{MM}}$ J. Novotná	2.	12								0	0	12
23–25.	Bc.$^{\mathrm{MM}}$ B. Mólová	3.	11								0	0	11
	Bc.$^{\mathrm{MM}}$ J. Svoboda	2.	11								0	0	11
	Dr.$^{\mathrm{MM}}$ L. Zavřel	4.	59								0	0	11
26–27.	Bc.$^{\mathrm{MM}}$ M. Bílý	4.	10								0	0	10
	Bc.$^{\mathrm{MM}}$ F. Lux	4.	10								0	0	10
28–30.	K. Kohoutová	2.	9				2		1		0	3	9
	Bc.$^{\mathrm{MM}}$ P. Kratochvíl	3.	17	2			2			3	0	7	9
	Mgr.$^{\mathrm{MM}}$ J. Škoda	4.	37								0	0	9
31–34.	Dr.$^{\mathrm{MM}}$ M. Bekrová	4.	61								0	0	8
	Mgr.$^{\mathrm{MM}}$ O. Cífka	2.	21								0	0	8
	L. Langerová	1.	8		1						0	1	8
	D. Tělupil	3.	8								0	0	8
35–39.	Mgr.$^{\mathrm{MM}}$ B. Böhmová	3.	37								0	0	7
	S. Havadej	4.	7								0	0	7
	Mgr.$^{\mathrm{MM}}$ G. Kubíčková	4.	21								0	0	7
	P. Kubincová	4.	7				2				0	2	7
	B. Said	3.	7								0	0	7

Poř.	Jméno	R.	$\sum_{-1}$	r1	r2	r3	r4	t1	t2	s3	+	$\sum_0$	$\sum_1$
40–41.	O. Fiedler	4.	7								0	0	6
	M. Kopf	3.	6								0	0	6
42–45.	V. Kletečka	3.	7								0	0	5
	M. Landa	1.	5								0	0	5
	M. Poppr	1.	5								0	0	5
	P. Vincena	1.	5					1			0	1	5
46–49.	J. Erhart	1.	4	1							0	1	4
	L. Jančařík	4.	4				1		1		0	2	4
	Bc.$^{\mathrm{MM}}$ O. Mička	2.	10								0	0	4
	D. Vít	1.	4								0	0	4
50–52.	E. Bušáková	2.	3								0	0	3
	J. Kadlec	1.	3								0	0	3
	S. Ondrčková	2.	3								0	0	3
53.	O. Krčmář	2.	2								0	0	2
54–56.	O. Darmovzal	1.	1								0	0	1
	K. Jiráková	4.	3						1		0	1	1
	J. Křivonožka	3.	1								0	0	1
57.	V. Václavík	1.	0								0	0	0

Sloupeček $\sum_{-1}$ je součet všech bodů získaných v našem semináři, $\sum_0$ je součet bodů v aktuální sérii a $\sum_1$ součet všech bodů v tomto ročníku. Sloupeček „+“ značí bonusové body udělované podle ročníku a součtu bodů za úlohy, sloupeček k jsou body za konferenční příspěvky a s jsou body za řešení úloh k seriálu. Tituly uvedené v předchozím textu slouží pouze pro účely M&M.

Optický klam na poslední stránce je převzat z `http://hps.mallat.cz/`.



Adresa redakce:

M&M, OVVP, UK MFF
Ke Karlovu 3
121 16 Praha 2

Telefon: +420 221 911 235
E-mail: `MaM@atrey.karlin.mff.cuni.cz`
WWW: `http://mam.mff.cuni.cz`

Časopis M&M je zastřešen Oddělením pro vnější vztahy a propagaci Univerzity Karlovy, Matematicko-fyzikální fakulty a vydáván za podpory středočeské pobočky Jednoty českých matematiků a fyziků.