

## 0 Pár poznámek před začátkem

Tento kurs se liší od všech kursů, které z internetu znám ve třech ohledech:

- jen velmi málo věcí se zde vysvětluje. Většinou používám odkazy na jiné kursy v internetu, protože si myslím, že jednou z věcí, které si musí každý programátor osvojit, je schopnost studovat z běžných zdrojů. Odkazované kursy k nim mají rozhodně blíže než moje učebnice. Navíc si myslím, že je zbytečné znovu psát něco, co již někdo v použitelném stylu napsal.
- velkou část textu tvoří příklady. Stejně jako u matematiky a fyziky, i u programování jsem za léta praxe vyzoroval, že studenti většinou pouze pasivně vnímají (nebo předstírají vnímání), ale jakmile má dojít k jakékoliv samostatné práci je to daleko horší. Fakt, že naprostou většinu hodiny nemusím nic vysvětlovat, mi samozřejmě pomáhá v tom, abych hlídal studenty, co vlastně dělají a v případě potřeby jim pomáhal.
- používá dost netradiční dělení, které je kromě logiky postupu přizpůsobeno tomu, co je možné reálně udělat za 45 minut vyučovací hodiny na střední škole.

Kurs byl používán na dvou skupinách studentů:

- studenti, kteří dobrovolně chodí na kroužek programování. U této skupiny fungoval kurs bez problémů a myslím, že svůj účel plnil.
- studenti, kteří se dobrovolně přihlásili na seminář z programování (často však pouze proto, že nechtěli na seminář z dějepisu). Aplikoval jsem ho na tyto skupiny dvakrát a v obou případech do dopadlo podobně. Na začátku byl postup docela rychlý a bezproblémový s příchodem polí se pak projevil těžká krize. Myslím, že nejvíce selhala schopnost studentů udržet si potřebný přehled v paměti. Problém není v tom, že by si přesně nepamatovali syntaxi (tu je vždy možno někde rychle nalézt), ale v tom, že si ani přibližně nepamatojí, jak se jinak zcela zásadní věci dělají (například prostě zapomenou, že existuje něco jako příkaz switch). Tento smutný fakt vnímám jako potvrzení všeho, co je na jiných místech napsáno o „průtokovém učení“. Pokud by chtěl někdo pohnout problematiku výuky programování dál, měl by se s uvedeným problémem nějak vypořádat.

Ke každé hodině jsou maximálně tři odkazy:

- pracovní zadání. Jde o text, který promítám studentům projektorem a studenti si ho sami otevírají na svých počítačích. Obsahuje odkazy na internet a zadání příkladů.
- ukázky. Zdrojové kódy programků, které mají studenti prostudovat nebo upravit.
- řešené příklady. Mnou řešené zadané úkoly. Na školní síti je umísťuji se zpožděním, aby se studentům tak snadno neopisovaly-

Pro výuku používáme [Boodshed DevC++ 4.9.2.2](#).

Kurs neobsahuje přímé komentáře ve formě pedagogických poznámek, jako učebnice matematiky nebo fyziky. Z mého hlediska jde o vedlejší záležitost, kterou mi vedení školy v podstatě vnutilo, neměl jsem na to dostatek času a v budoucnosti již nebudu tento kurs ve své práci potřebovat. Přesto několik obecných postřehů:

- studenti mají tendenci to nějak udělat. Většinou to znamená spravit program tak, že postupně něco mění až to začne fungovat. Je zcela samozřejmé, že tímto způsobem není možné naprogramovat cokoli složitějšího, ale u hodně jednoduchých příkladů se dá uspět. Proto je třeba proti tomu bojovat.

- ve školách není podporován logický postup řešení problému. Jedním z důsledků je skutečnost, že studenti nemají celkový přehled o programovaném úkolu, orientaci v tom, co která část programu dělá. Soustředí se pouze na malý úsek kódu. Velká část příkladů spočívá v přidání nové vlastnosti do předchozího programu. V takovém případě je třeba, aby studenti před tím než začnou zdroják upravovat věděli, co která část kódu dělá a kam mají začít psát. Teprve když budou vědět, kam mají novou funkčnost přidat má smysl, aby začali přemýšlet, jak bude vypadat konkrétní podmínka ...
- notorickým problémem je pojmenování proměnných a komentáře. Obojí patří mezi věci, kterými se studenti odmítají zatěžovat a u počátečních příkladů jim přijdou zbytečné. Už u příkladů s poli se z toho stává neřešitelný problém. Pomáhá také, když jim dáte přeprogramovat kód, který sami napsali před delší dobou (stačí 14 dní).
- U studentů, kteří ke mně chodili na kroužek mám velice dobrý pocit ze série šifrátoru. Programování kódu, který už nebyl velmi krátký a vyžadoval dobrý počátečný návrh, podle mě hodně tlačilo na správný styl. Pokud chcete, aby byli studenti připravení na něco jiného než psaní hodinových trivialit, použijte ho nebo vymyslete něco podobného.

Poslední rada se týká hodnocení (tato poznámka se netýká zkušeností z kroužku). Je smutnou skutečností, že hlavním motivačním faktorem jsou známky. Minimálně polovina studentů není schopná se nějak připravovat, pokud jí nehrozí špatné známky. Jako ideální bych viděl jednu písemkovou hodinu do měsíce (několik příkladů, které musí za hodinu udělat) a pravidelnou kontrolu dokončení příkladů, které nestihli v hodině. Sám jsem to nikdy nedokázal praktikovat (obě mé zkušenosti padají do let psaní učebnic), ale kdybych se k tomu ještě někdy v životě dostal, hodně bych se o to snažil.

Hodně úspěchů přeje Martin