



Prekladač nového modulárního jazyka

Semestrální projekt

Daniel Čejchan | xcejch00

1 | Obsah

1	Obsah	1
2	Úvod	2
2.1	Motivace	2
3	Vlastnosti jazyka	3

2 | Úvod

Tento dokument popisuje a zdůvodňuje část z mnoha rozhodnutí, která byla vykonána při procesu návrhu programovacího jazyka NATI a vzorového překladače pro něj. Rámcově prozkoumává syntaktické i sémantické prvky moderních programovacích jazyků a popisuje i nové koncepty a prvky.

Programovací jazyk se hodně inspiruje jazyky C++ a D¹. Tyto jazyky budou používány pro srovnávání syntaxe a efektivity psaní kódu.

2.1 Motivace

Mou hlavní motivací je nespokojenost se stávajícími programovacími jazyky. Programování mám jako koníčka už od nějakých dvanácti let, a tak jsem ještě před nastoupením na FIT získal nějaké zkušenosti z psaní aplikací v Object Pascalu (Delphi), C++ (Qt, SDL + OpenGL) a PHP (webové aplikace). FIT pak můj repertoár (i když jen rámcově) rozšířil na valnou většinu dnes používaných jazyků. Bohužel jsem ale nenarazil na žádný, který by splňoval mé požadavky.

Jazyk D Mému srdci nejbližší jazyk, na který jsem ve svém pátrání narazil, byl jazyk D. Jedná se o kompilovaný jazyk vycházející z C++ (binárky jsou do jisté míry kompatibilní) s velice podobnou syntaxí. Největší rozdíly jsou modulový systém² (zdrojový kód je rozdělen do modulů, které se vzájemně, i rekurzivně, importují; odpadá nutnost psát hlavičkové soubory), značně rozšířená funkčnost metaprogramování s šablonami a rozšířená schopnost vykonávat funkce za doby kompilace.

D mi byl velikou inspirací při navrhování mého jazyka. Bohužel i v D jsem narazil na strop možnosti (ačkoli byl značně výš než třeba v C++), kdy některé věci nešly napsat tak jednoduše, jak bych chtěl. Tento jazyk však dokazuje, že tato bariéra může být mnohem dál. Já ji chci ve svém jazyku ještě více posunout.

¹ <http://dlang.org/>

² <http://dlang.org/spec/module.html>

3 | Vlastnosti jazyka

Nejdříve musíme určit základní rámce, ve kterých budeme operovat.

Použití jazyka Naším cílem je navrhnout tzv. *general purpose language*, tedy jazyk nezaměřený na konkrétní případy užití.

Kompilovaný vs. interpretovaný jazyk Ačkoli interpretované jazyky mají jisté výhody, platí se za ně pomalejším kódem a nutností zavádět interpret. V rámci této práce budeme navrhovat kompilovaný jazyk. Pro zjednodušení práce při psaní překladače (předmětem projektu je jazyk, ne překladač) nebudeme ale překládat přímo do strojového kódu, ale do jazyka C.

Syntaxe jazyka Abychom maximálně usnadnili přechod případných programátorů k našemu jazyku a zkrátili učební křivku, je vhodné se co nejvíce inspirovat již existujícími jazyky. Vzhledem k tomu, že se jedná o kompilovaný jazyk, je nejrozhodnější vycházet se syntaxe rodiny jazyků C.

3.1 Prvky moderních programovacích jazyků