



Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozási nyelvek és
Fordítóprogramok tanszék

DIPLOMAMUNKA-TÉMA BEJELENTŐ

Név: Bócsi Rajmund

Tagozat: nappali

Értek. programtervező matematikus

Témavezető neve: Nyékyné Dr. Gaizler Judit

A GENERATÍV PROGRAMOZÁS MÓDSZERTANÁNAK VIZSGÁLATA

Bócsi Rajmund

Nappali tagozat

Programtervező matematikus szak

Témavezető: Nyékyné Dr. Gaizler Judit

TARTALOMJEGYZÉK

TARTALOMJEGYZÉK	4
BEVEZETÉS	6
1. fejezet A generatív programozásról	7
1.1 A cél	7
1.2 A generatív programozás definíciója	7
1.3 A generatív domain modell	7
1.4 Két fejlesztési ciklus	8
1.5 Előnyök és hátrulütök	8
2. fejezet Az objektumorientált elemzés és tervezés és az újrafelhasználás	9
3. fejezet A domain alapú fejlesztés módszere	11
3.1 Definíciók és fázisok	11
3.2 Domain elemzés	12
3.3 Domain tervezés és implementáció	13
3.4 Alkalmazásfejlesztés	14
3.5 Feature-modell: definíciók és feature-diagram	15
3.6 Feature-modell: a közös és változó tulajdonságok kifejezése	17
3.7 Feature-modell: további információk a diagramhoz	18
3.8 Feature-modell: alkotási folyamat	19
4. fejezet Az objektumorientáltság és a domain alapú tervezés integrálása	21
5. fejezet A generatív programozás folyamata	23
5.1 A generatív domain modell	23
5.2 A fő fejlesztési lépések a generatív programozásban	24
5.3 A domain alapú fejlesztés módosítása a generatív programozáshoz	25
5.4 Domain-specifikus nyelvek	25
5.5 Generátorok	28
6. fejezet Generikus programozás	33
6.1 A generikus programozás különböző definíciói	33
6.2 Polimorfizmus	34
6.3 Generikusság az Ada95-ben	36
6.4 Generikusság az Eiffel-ben	40
6.5 Generikusság a Java-ban	44
6.6 Generikusság a C++-ban	50
6.7 A látens típusosság hiányának kiküszöbölése a Java-ban	58
6.8 Generikusság a D-ben	60
7. fejezet Aspektusorientált programozás	66
7.1 A vonatkozások szétválasztása (Separation of Concerns)	66
7.2 Aspektus-orientált dekompozíciós megközelítések	67

7.3 Aspektusok	69
7.4 Aspektus-orientált programok	69
7.5 Kompozíciós mechanizmusok	71
7.6 Példa: egy szinkronizált verem	72
7.7 AspectJ	78
7.8 Példa: egy szinkronizált verem	84
7.9 Az aspektusorientált programozás néhány eredménye	86
8. fejezet Statikus metaprogramozás	88
8.1 A metaprogramozással kapcsolatos fogalmak	88
8.2 Statikus metaprogramozás	89
8.3 Template metaprogramozás	90
8.4 Metafüggvények	92
8.5 Metainformáció reprezentációja	93
8.6 Fordítási idejű vezérlési szerkezetek	95
8.7 Kódgenerálás	97
8.8 Részleges kiértékelés C++-ban	100
8.9 A részleges specializáció kiküszöbölése	102
8.10 A C++ template metaprogramozás problémái	102
8.11 Template metaprogramozás a D-ben	103
9. fejezet Szándékorientált programozás	105
9.1 A fix programozási nyelvek problémái	105
9.2 Mi a szándékorientált programozás?	106
9.3 Az IP technológiája	108
9.4 Új szándékok és kérdések	111
9.5 Összegzés	112
9.6 Nyelvorientált programozás	112
ÖSSZEFOGLALÁS	114
A függelék – Az AspectJ primitív vágási pont kijelölői	115
B függelék – Fordítási idejű vezérlési szerkezetek a C++-ban	117
B.1 IF	117
B.2 IF (részleges specializáció nélküli változat)	117
B.3 SWITCH	118
B.4 SWITCH (részleges specializáció nélküli változat)	118
B.5 WHILE	119
B.6 DO	119
B.7 FOR	120
IRODALOMJEGYZÉK	121