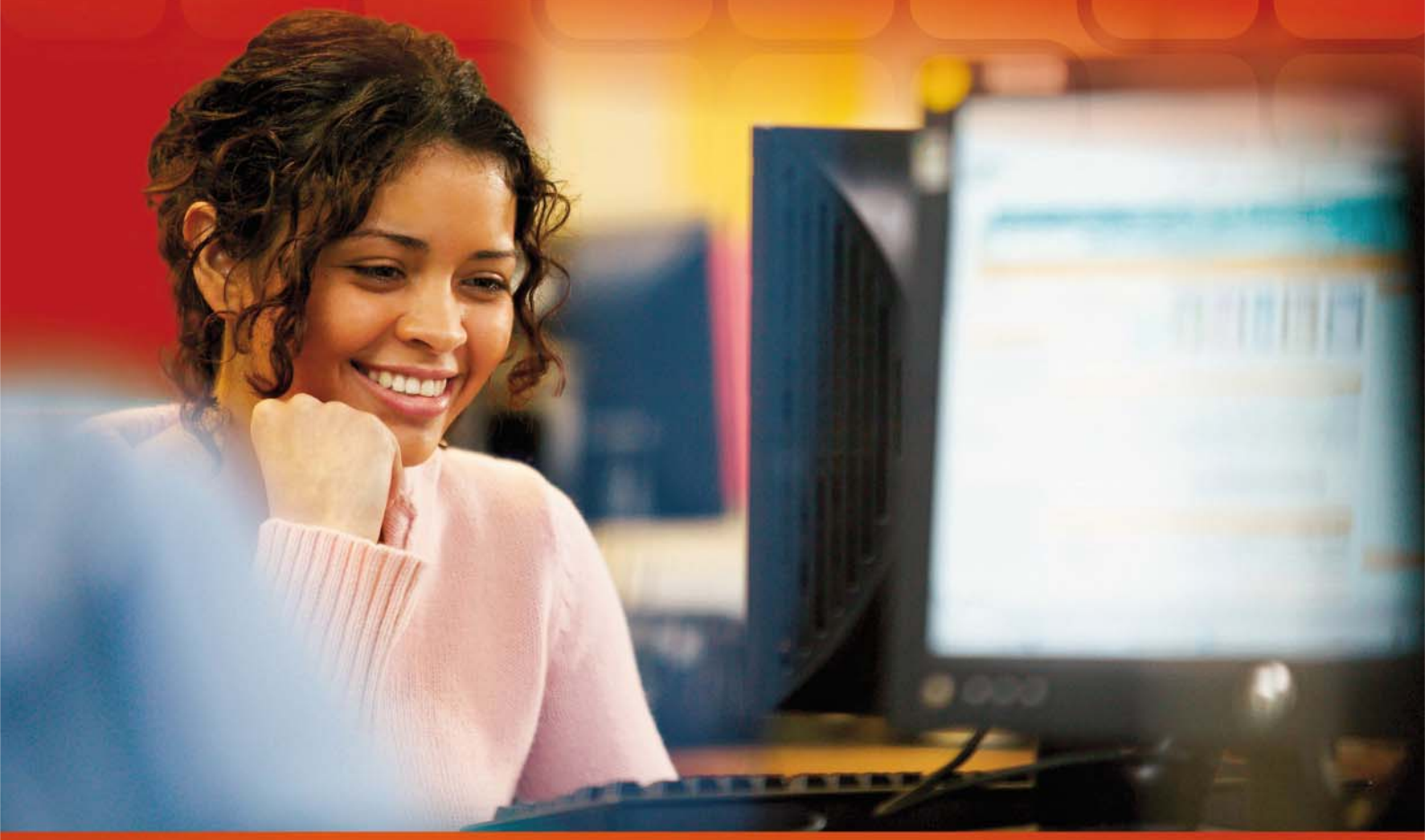
Microsoft



Ján Hanák

# Objektovo orientované programovanie v jazyku C# 3.0

Príručka pre vývojárov, programátorov
a softvérových expertov

Microsoft Visual Studio 2008

Microsoft Visual Studio Express Editions

Ján Hanák

# Objektovo orientované programovanie v jazyku C# 3.0

(Príručka pre vývojárov, programátorov a softvérových expertov)

Microsoft
2008

# Obsah

# Úvod

Počas uplynulého polstoročia sa v informatických vedách uskutočnil taký progres, aký sa v iných oblastiach ľudských činností zaznamenáva spravidla za niekoľko storočí. Z pôvodne sálových počítačov sa stali mikropočítače s ohromným výkonom, ktoré sú kedykoľvek a kdekoľvek pripravené pomáhať ľuďom s riešením ich každodenných problémov. Vedno s prevratným technologickým tempom držala krok aj teoretická informatika, ktorá definovala metodiky, postupy a paradigmy vývoja počítačového softvéru. Je to totiž práve softvér, ktorý dokáže vdýchnuť dušu rozmanitým hardvérovým strojom a zariadeniam. Bez sofistikovaného softvéru by nemohol byť úspešne realizovaný ani zlomok súčasnej technologicko-informačnej revolúcie.

Hoci z dnešného pohľadu je to snáď až nepredstaviteľné, paradigmy vývoja počítačového softvéru neboli vždy také vyspelé ako v súčasnosti. Začiatky sa spájajú s priamym programovaním počítačov pomocou rýdzo technických úkonov, ktoré boli neskôr nahradené tvorbou programov v strojovom kóde. Explicitné programovanie číslicových počítačov na nízkej úrovni sa vyznačovalo nielen veľmi malou mierou pracovnej produktivity, ale aj náročnosťou na abstraktné myslenie a technickú precíznosť. Presun od strojového kódu smerom k jazykom symbolických adries priniesol vo svojej dobe signifikantnú úroveň abstrakcie. Hoci aj dnes sa časti jadier mnohých operačných a informačných systémov programujú v jazyku symbolických adries, v akademickej a komerčnej sfére sa udomácnili paradigmy vývoja počítačového softvéru, ktoré sú typické vyššou mierou abstrakcie od hardvérovej infraštruktúry. Štruktúrované programovanie umožnilo naplno rozvinúť myšlienku problémovej dekompozície, kedy sa problém rozložil na viacero algoritmicky riešiteľných podproblémov, ktorých spracovanie mali na starosti viaceré funkcie jedného programu. Štruktúrované programovanie sa stalo prvou vývojovou vetvou, ktorej sa podarilo zapojiť do programovania počítačov široké cieľové publikum. O perspektívnosti štruktúrovaného programovania svedčí jeho stále veľká praktická obľúbenosť, a to aj napriek tomu, že od uvedenia tohto štýlu vývoja softvéru uplynulo už niekoľko desaťročí.

Veľký posun vpred zaznamenala objektová paradigma, ktorá sa stala základným pilierom objektovo orientovaného programovania. Dáta a manipulačné operácie, ktoré sú s dátami uskutočňované, sú zapuzdrené do logických jednotiek, ktorým vravíme objekty. Virtuálne objekty vznikajú v procese objektového modelovania, pričom reflektujú vlastnosti a schopnosti skutočných objektov tvoriacich reálny svet. Virtuálne objekty sú navyše inteligentné a vedia kooperovať tak, aby spoločne vyriešili mnohé aj z tých najnáročnejších problémov, s akými sa teoretická informatika stretla.

Netrvalo dlho a vývoj softvéru sa z pôvodne akademických a výskumných centier preniesol do rýdzo komerčnej sféry. A v nej, ako je známe, sa cení najmä veľká pracovná

produktivita, ktorej cieľom je kreácia produktov s vysokou pridanou hodnotou. Proces vývoja softvéru sa začal podobať výrobe iných zložitých strojov a zariadení, pričom sa doň implementovali prvky hromadnej výroby. Komponentové programovanie predstavilo komponentovú technológiu vývoja softvéru, podľa ktorej sú aplikácie tvorené kolekciou vhodne nakonfigurovaných počítačových súčiastok, teda komponentov. Komponentové programovanie využíva všetky silné stránky objektovo orientovaného programovania, veď koniec koncov, komponenty ako také sú tvorené súpravou spolupracujúcich objektov. Komponentové programovanie umožnilo naplno rozvinúť ideu rýchleho vývoja počítačových aplikácií.

Výzvou súčasných a určite aj budúcich rokov je paralelné programovanie, ktoré možno realizovať v súčinnosti s objektovo orientovaným a komponentovým programovaním. Paralelné programovanie sa stáva hitom, pričom tento stav je podmienený predovšetkým príchodom počítačov, ktoré sú osadené viacjadrovými procesormi. Práve takéto stroje umožňujú dosiahnuť skutočný paralelizmus, kedy sú toky programových inštrukcií spracúvané paralelne na jednotlivých jadrách procesora. Keďže môžeme s istotou predpovedať, že v budúcnosti sa budú objavovať procesory so stále väčším počtom exekučných jadier, je nutné v záujme optimálneho využitia celkovej výpočtovej kapacity systému venovať zvýšenú pozornosť masívnej paralelizácii.

Cieľom tejto knihy je vysvetliť všetky princípy objektovo orientovaného programovania v jazyku C# 3.0. V publikácii sme sa snažili vytvoriť robustné teoreticko-praktické zázemie, ktoré pomôže vývojárom pripraviť sa na novo prichádzajúcu paradigmu tvorby počítačového softvéru, ktorou je paralelné objektovo orientované programovanie (POOP). Tak budú môcť vývojári, programátori a softvéroví experti čeliť novým apelom, ktoré prinesie paralelná epocha počítačovej evolúcie.

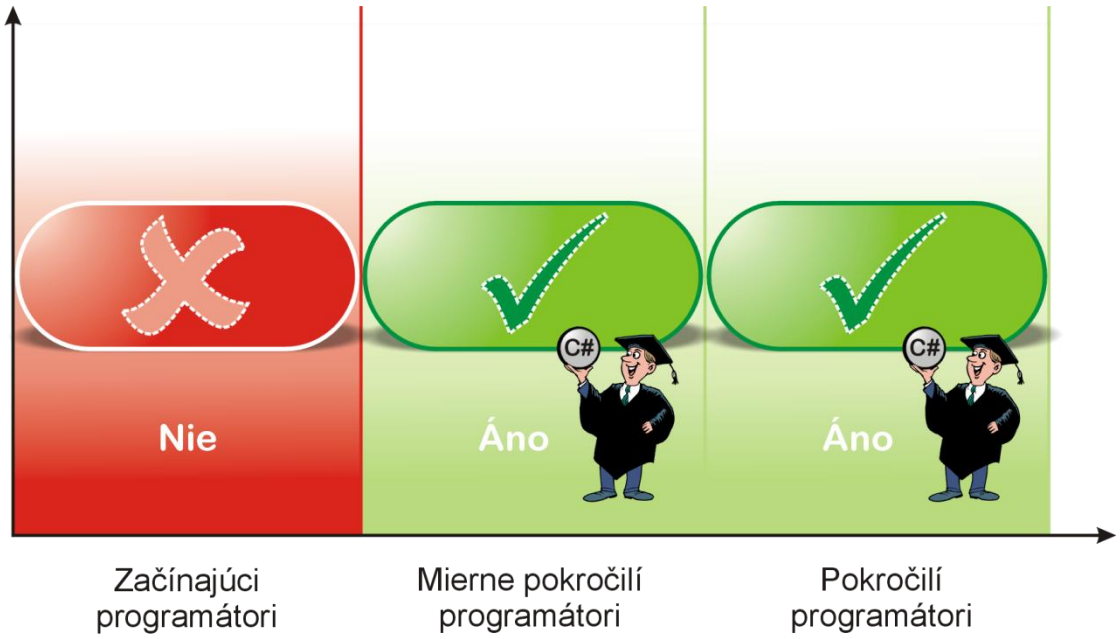
Ján Hanák

Praha, september 2008

# Pre koho je táto kniha určená

Kniha **Objektovo orientované programovanie v jazyku C# 3.0** sa sústreďuje na výklad základných pilierov všeobecnej teórie objektovo orientovaného programovania s ich praktickou aplikáciou v programovacom jazyku C# 3.0, ktorý je implementovaný v produktoch Microsoft Visual C# 2008 a Microsoft Visual C# 2008 Express. Primárnym cieľovým segmentom publikácie sú mierne pokročilí programátori a vývojári v jazyku C#, ktorí ovládajú základy tohto programovacieho jazyka.

Kniha predpokladá, že čitatelia vedia, ako funguje program v jazyku C# 3.0 a rovnako sú oboznámení s elementárnymi programovými konštrukciami, ku ktorým patria premenné, operátory, rozhodovacie príkazy a cykly. Ak ste začiatočníkom v algoritmizácii alebo v programovaní vo všeobecnosti, odporúčame vám, aby ste si pred štúdiom tejto publikácie najskôr prečítali niektorú z množstva kníh, ktoré podávajú základný výučbový kurz programovania v jazyku C# 3.0. Výklad kladie mimoriadny dôraz na vysokú technickú kvalitu, terminologickú exaktnosť a vedecké spracovanie problematiky, čím napĺňa koncepciu primeraného študijného zaťaženia čitateľov.




Obr. A: Vhodnosť knihy pre rôzne segmenty používateľov

Publikácia je vhodná pre poslucháčov vysokých škôl univerzitného typu so zameraním na výučbu informatických a počítačových vied. Svoje uplatnenie nachádza v špecializovaných predmetoch 1., resp. 2. stupňa vysokoškolského štúdia.

# Obsahová štruktúra knihy

Kniha **Objektovo orientované programovanie v jazyku C# 3.0** sa skladá z dvoch hlavných tematických celkov:

1. Teoretické základy objektovo orientovaného programovania.
2. Objektovo orientované programovanie v jazyku C# 3.0.

1. tematický celok oboznamuje čitateľov s tromi hlavnými paradigmami vývoja počítačového softvéru, ktoré sa vyvinuli za posledné desaťročia. Ide o štruktúrované, objektovo orientované a komponentové programovanie. Dôraz je kladený predovšetkým na komplexné ozrejmenie princípov všeobecnej teórie objektovo orientovaného programovania.

2. tematický celok nadväzuje na teoretické základy položené v 1. tematickom celku. Hlavný výkladový prúd tvoria kapitoly, ktoré sa venujú praktickému objektovo orientovanému programovaniu v jazyku C# 3.0. Komplementárnou pridanou hodnotou je výklad syntakticko-sémantických inovácií jazyka C# 3.0, ktoré sa viažu s filozofiou objektovo orientovaného programovania.

# Typografické konvencie

Aby sme vám čítanie tejto knihy spríjemnili v čo možno najväčšej miere, bol prijatý kódex typografických konvencií, pomocou ktorých došlo k štandardizácii a unifikácii použitých textových štýlov a grafických symbolov. Veríme, že prijaté konvencie zvýšia prehľadnosť a používateľskú prívetivosť výkladu. Prehľad použitých typografických konvencií uvádzame v tab. A.

Tab. A: Prehľad použitých typografických konvencií

| **Typografická konvencia** | **Ukážka použitia typografickej konvencie** |
|---|---|
| Štandardný text výkladu, ktorý neoznačuje zdrojový kód, identifikátory, modifikátory a kľúčové slová jazyka C# 3.0, ani názvy iných syntaktických elementov a entít, je formátovaný týmto typom písma. | Vývojovo-exekučná platforma Microsoft .NET Framework 3.5 kreuje spoločne s jazykom C# 3.0 jednotnú technologickú bázu na vytváranie moderných riadených aplikácií pre Windows, web a inteligentné mobilné zariadenia. |

| Tab. A: Prehľad použitých typografických konvencií (pokračovanie) | |
|---|---|
| **Typografická konvencia** | **Ukážka použitia typografickej konvencie** |
| Názvy ponúk, položiek ponúk, ovládacích prvkov, komponentov, dialógových okien, podporných softvérových nástrojov, typov projektov ako aj názvy ďalších súčastí grafického používateľského rozhrania sú formátované **tučným** písmom.<br><br>**Tučným** písmom sú rovnako formátované aj identifikátory programových entít, ktoré sú uvádzané priamo vo výkladovom texte. | V záujme založenia nového projektu štandardnej aplikácie pre systém Windows (**Windows Forms Application**) v prostredí produktu Visual C# 2008 postupujeme takto:<br><br>1. Otvoríme ponuku **File**, ukážeme na položku **New** a klikneme na príkaz **Project**.<br><br>2. V dialógovom okne **New Project** klikneme v stromovej štruktúre **Project Types** na položku **Visual C#**.<br><br>3. Zo súpravy projektových šablón (**Templates**) vyberieme ikonu šablóny **Windows Forms Application**.<br><br>4. Do textového poľa **Name** zapíšeme názov pre novú aplikáciu a stlačíme tlačidlo **OK**. |
| Fragmenty zdrojového kódu jazyka C# 3.0, prípadne akýchkoľvek iných programovacích jazykov, sú formátované neproporcionálnym písmom `Courier New`. | `// Definícia premennej typu string.`<br>`string správa;`<br>`// Inicializácia premennej.`<br>`správa = "Vitajte v jazyku " +`<br>`"C# 3.0!";`<br>`// Zobrazenie okna so správou`<br>`// pomocou metódy Show triedy`<br>`// MessageBox.`<br>`MessageBox.Show(správa);` |

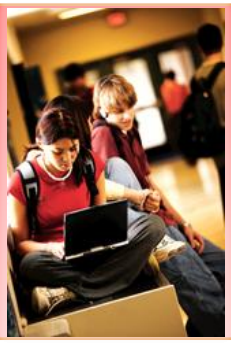
# Poďakovanie

Na tomto mieste by autor knihy rád vyjadril svoje poďakovanie pánovi Jiřímu Burianovi, ktorý zastáva pozíciu produktového manažéra vo vývojárskej divízii českej pobočky spoločnosti Microsoft za výbornú niekoľkoročnú spoluprácu, ktorá priniesla vývojárskej komunite mnoho hodnotných produktov. Rovnako srdečne ďakuje autor za skvelú spoluprácu, ochotu a podporu aj pánovi Miroslavovi Kubovčíkovi, ktorý je kmeňovým členom vývojárskeho tímu slovenskej pobočky spoločnosti Microsoft. Bez ich podpory

by nemohla vzniknúť nielen táto kniha, ale ani mnohé ďalšie diela, ktoré permanentne zlepšujú stav vývojárskeho ekosystému. Takže páni, ešte raz, úprimná vďaka!

1. tematický celok

# Teoretické základy objektovo orientovaného programovania



Microsoft® Visual Studio® 2008

# 1. Teoretické základy objektovo orientovaného programovania

## 1.1 Vývojové vetvy a paradigmy programovania

Počas uplynulých päťdesiatich rokov zaznamenali prístupy k vývoju počítačového softvéru nebývalý rozmach. Pri bližšej analýze môžeme determinovať tri základné vetvy programovania:

1. Štruktúrované programovanie.
2. Objektovo orientované programovanie.
3. Komponentové programovanie.

Každá z uvedených vetiev prichádza s vlastnou paradigmou vývoja počítačového softvéru.

### 1.1.1 Štruktúrované programovanie

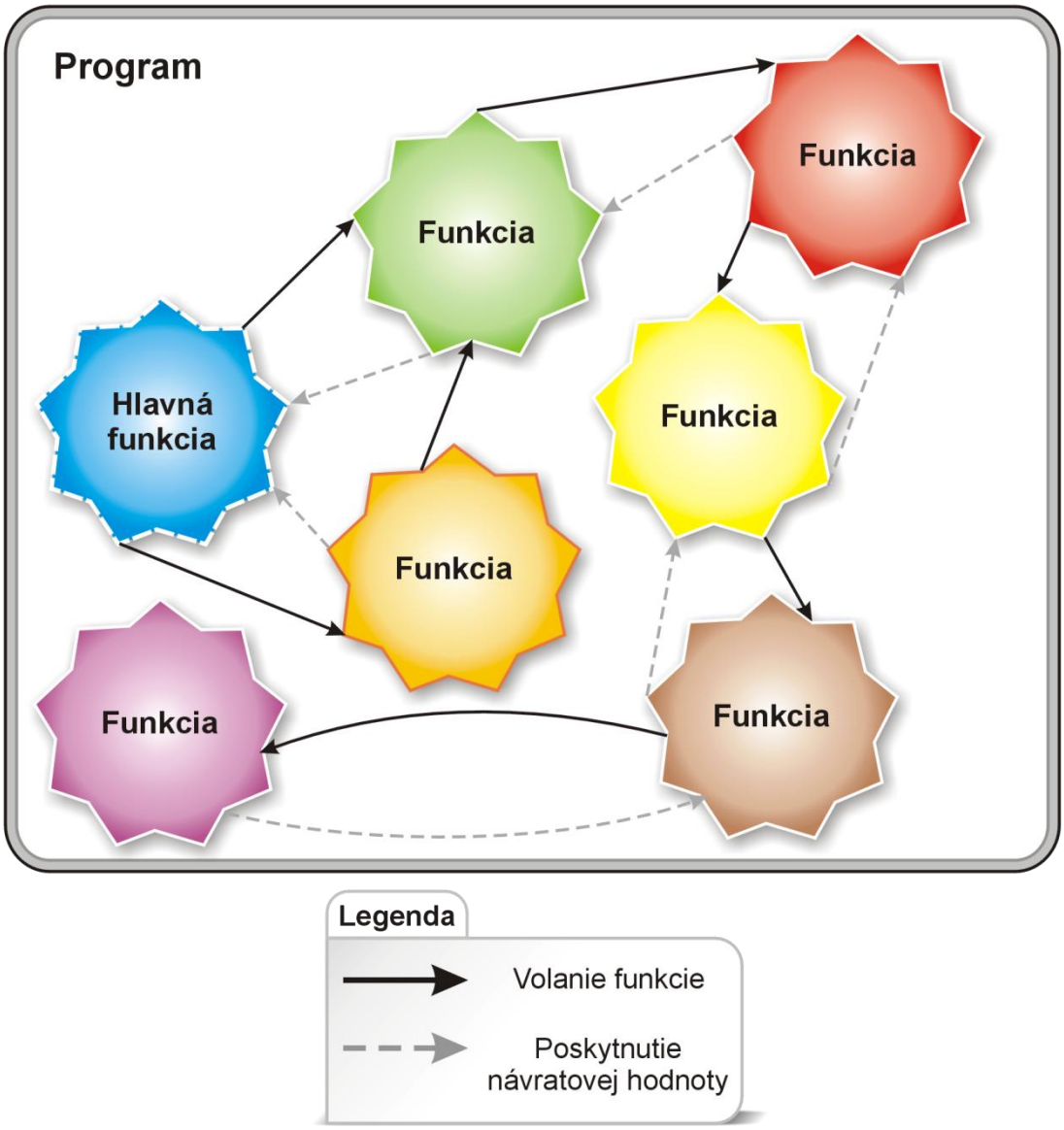
Štruktúrované (procedurálne) programovanie sa sústreďuje na návrh imperatívnych programov, ktoré implementujú algoritmy prostredníctvom troch základných grafov riadenia, resp. riadiacich konštrukcií, ktorými sú: sekvencia, selekcia a iterácia. Návrh štruktúrovaných programov spravidla prebieha postupným zjemňovaním zložitosti, kedy sa pôvodne riešený komplexný problém rozkladá na menšie podproblémy, ktoré disponujú nižšou zložitosťou. Naznačená problémová dekompozícia je spojená s intenzívnym využitím základných riadiacich konštrukcií, ktoré je možné spoľahlivo definovať syntaktickými príkazmi aplikovaného programovacieho jazyka. Štruktúrované programovanie využíva štruktúrované algoritmy. Štruktúrovaný algoritmus môžeme charakterizovať induktívno-rekurzívnym spôsobom:

1. Každý jednotlivý krok je štruktúrovaným algoritmom.
2. Základné riadiace konštrukcie (sekvencia, selekcia a iterácia) štruktúrovaných algoritmov predstavujú taktiež štruktúrovaný algoritmus.
3. Štruktúrovaným algoritmom je všetko to, čo získame opakovaným použitím premís položených v 1. a 2. kroku tohto postupu.

Z praktického hľadiska je štruktúrovaný program tvorený množinou funkcií, ktoré medzi sebou vedú prospešný informačný dialóg. Funkcia je základnou stavebnou jednotkou štruktúrovaného programu a ako taká sa často stotožňuje s podprogramom. Z uvedeného vyplýva, že štruktúrovaný program $P$ je množinou spriaznených funkcií $F$, čo môžeme matematicky formalizovať takto:

$$P = F = \{f_1, f_2, \dots, f_n\}$$

Jedna z funkcií štruktúrovaného programu má výsadné postavenie, pričom sa klasifikuje ako hlavná funkcia programu. Hlavná funkcia predstavuje vstupný bod programu, čo znamená, že exekúcia programu sa začína volaním a spracovaním tejto funkcie. Hlavná funkcia je automaticky aktivovaná operačným systémom vždy, keď je zaregistrovaná požiadavka na spustenie štruktúrovaného programu. V rámci implementácie štruktúrovaných algoritmov je ideálne, ak jedna funkcia programu rieši práve jeden algoritmus. Tak je možné dosiahnuť efektívnu úlohovú dekompozíciu. Úlohová dekompozícia pracuje v súčinnosti s technikou postupného zjemňovania zložitosti riešeného problému. Hlavná funkcia sa preto nekoncentruje na samostatné riešenie problému, jej úlohou je optimálna dekompozícia problému na jednotlivé podproblémy, ktorých spracovanie je delegované rôznym funkciám (podprogramom). Vizuálna kompozícia štruktúrovaného programu je znázornená na obr. 1.




Obr. 1: Štruktúrovaný program