

プログラミング教育におけるHCPチャートの再評価

杉浦 学[†] 松澤 芳昭[†] 大岩 元^{††}
Manabu Sugiura,[†] Yoshiaki Matsuzawa[†] and Hajime Ohiwa^{††}

筆者らは、プログラミング教育でHCPチャートを利用している。この教育では、プログラムの設計をHCPチャートによって表現し、チャートに対するレビューを実施する。しかし、HCPチャートを教育に利用し、分かりやすいHCPチャートについて議論するカリキュラムの意義が、筆者らの教育を受けた者以外には理解されない。この原因は、HCPチャートがプログラムの論理構造を表現する図法だということが、一般に正しく理解されていないからである。本稿では、HCPチャートに対する誤解を解き、「目的/手段の構造を整理する」という考え方の重要性を述べて、プログラミング教育ツールとしてのHCPチャートを再評価する。

1. はじめに

筆者らは、プログラミング教育で、HCPチャート¹⁾²⁾を利用している。この教育では、プログラムの設計をHCPチャートによって表現し、チャートに対するレビューを実施する³⁾⁴⁾⁵⁾⁶⁾。しかし、HCPチャートをプログラミング教育に利用し、分かりやすいHCPチャートについて議論するカリキュラムの意義が、筆者らの教育を受けた者以外に理解されることが悩みである。

本稿では、HCPチャートに対する誤解を解き、「目的/手段の構造を整理する」という考え方の重要性を述べて、プログラミング教育ツールとしてのHCPチャートを再評価する。

2. HCPチャートに対する誤解

2.1 目的/手段の構造

プログラムにおける処理は「目的 (What)」とそれを達成するための「手段 (How)」という視点から整理することができる。整理された「目的/手段の構造」は、階層的にプログラムの論理構造を表現する。これを図解するのがHCPチャートである⁷⁾⁸⁾。

2.2 教科書における誤解の例

HCPチャートで表現されるべきものが、一般に正しく理解されていないことが問題である。この事を、SPDの教科書に掲載されてい

るHCPチャート(図1)を例にして述べる*。図1を「プログラムの論理構造(目的/手段の構造)が表現されているか?」という視点から捉えると、次のような問題点が指摘できる。

- [全体]: (チャートとは別のページで解説されているが) 最上位の目的がない。
- [2行目]: 「データが存在する」は目的ではなく、繰り返し処理の継続条件である。何のために繰り返し処理を行うのか(複数の三角形の面積を求めること)が明確ではない。
- [3行目]: 何のために「底辺・高さが0より大きい」という振り分け処理を行うのが明示されていない。
- [4,7行目]: 「T」や「F」というのは実装の手段が記述されているだけであり、振り分けの条件の意味が表現されていない。
- [全体]: 3番目の指摘と関連するが、本処理とエラー処理が振り分け処理を使って表現されており、処理の目的の違いが表現されていない。振り分け処理を使ってエラー処理を行うということは、実装段階で考えればよい事柄である**
- [全体]: 繰り返し処理が、特定の制御構

* SPDはHCPチャートと同じく、プログラムの論理構造を階層的に記述するための設計図法である。

** 本来は、エラー処理を削除してもチャートが理解できるように記述すべきである。図2のチャートでは、×印で表現されたエラー処理の項目を削除しても、「三角形の面積を求める」という本処理に影響がない。

[†] 慶應義塾大学 政策・メディア研究科
^{††} 慶應義塾大学 環境情報学部

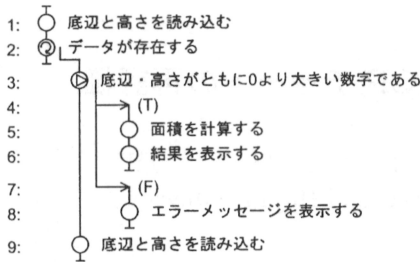


図1 三角形の面積を求めるプログラムの HCP チャート (構造化プログラム設計図法 SPD P.35 より)

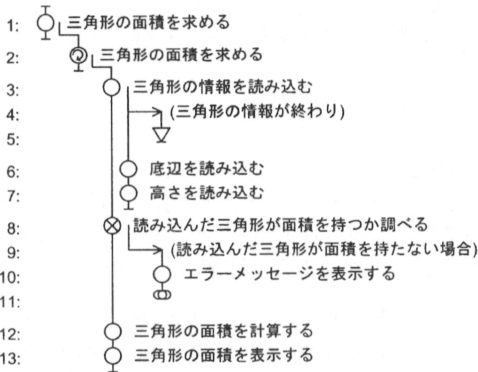


図2 三角形の面積を求めるプログラムの HCP チャート (目的/手段の論理構造を表現した場合)

造 (繰り返したい処理を実行する前に、繰り返しの継続条件を判定すること) を前提に記述されている。それが原因で、「底辺と高さを読み込む」という項目が2個 (1行目と9行目) 必要になっている。

同じ「三角形の面積を求めるプログラム」でも、「目的/手段の構造」を表現した HCP チャートは図2のようになる。

図1に記述されている項目は処理 (手段) だけであり、表現されているのはプログラムの制御構造である。それに対して図2は、処理が目的という観点からまとめられており、プログラムにおける目的と手段の関係が階層構造で表現されている。

2.3 HCP チャート≠フローチャート

図1のような HCP チャートを記述してしまうのは、処理を目的によってまとめて、プログラムの論理構造を考えるという視点が不足していることが原因である。このことを HCP チャートとフローチャートを比較しながら考察する。

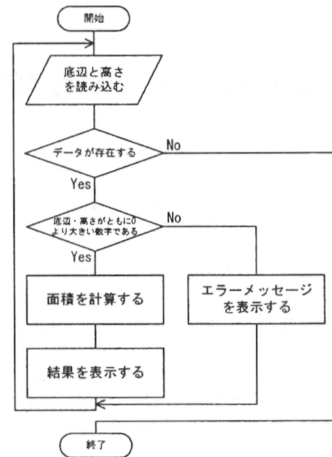


図3 三角形の面積を計算するプログラムのフローチャート

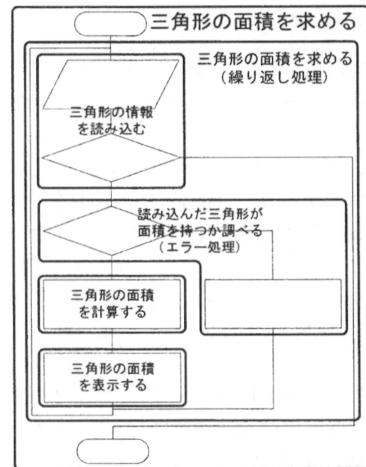


図4 三角形の面積を計算するプログラムのフローチャート (処理を目的でまとめた場合)

まず、図1と同様のプログラムを表現したフローチャートを図3に示す。

図3では、プログラムの制御構造は把握しやすいが、「それぞれの処理の目的は何か?」という論理構造が明確でない。そこで、図3に記述されている処理を目的という観点からまとめると、図4のようになる。

全ての処理をまとめる一番大きな目的は、「三角形の面積を求める」というプログラム全体の目的である。その内部には、繰り返し処理で実現される「三角形の面積を求める」という目的がある。さらにその内部には、「三角形の情報を読み込む (入力)」、「三角形の面積を計算する (処理)」、「三角形の面積を表示する (出

力)、「読み込んだ三角形が面積を持つか調べる(エラー処理)」という4つの目的があると考えることができる。

このように、プログラムにおける処理を目的という観点からまとめていくと、それらは階層構造を形成する。あるまとまりは、上位の目的を達成するための手段であると同時に、下位の手段が達成すべき目的でもある。つまり、まとまりが目的か手段かは相対的な事柄である。HCPチャートでは、プログラムにおける目的/手段の階層構造を、項目を横方向にインデントすることで表現している。

図4のフローチャートでまとめた目的/手段の階層構造をHCPチャートの記法と対応付けると、図2の最下位レベルの記述を省略したHCPチャートと等しくなる。

図1が表現しているのはプログラムの制御構造であって、論理構造ではない。これは、図1と図3を比較すれば明らかである。制御構造を分かりやすく表現するためには、フローチャートで十分である。

2.4 誤解の原因についての考察

図1が掲載されている教科書に限らず、他の書籍やWebで見かけるHCPチャートは、プログラムの制御構造を表現したものばかりである。目的と手段の概念は特殊な考え方ではないが、プログラムにおいて「目的/手段の構造を整理する」ことは簡単ではない。これまで述べてきた誤解は、その難しさに起因しているものであると推察される。

また、HCPチャートでは目的の階層をインデントで表現する。そのため、ソースコードに現れる制御の構造と、チャートの形が似ている場合が多い。このことが、HCPチャートに制御構造を記述してしまう一因と考えられる。

3. ま と め

処理を目的という観点からまとめ、目的/手段の階層構造として表現しなければ、HCPチャートを利用する意味がない。分かりやすいHCPチャートについて議論することによって、プログラムを制御構造から捉える思考(フローチャートを記述すること)だけでなく、プログラムを論理構造から捉える思考を養うことができる。

プログラムの制御構造は、(GOTO文を使わなければ)ソースコードから容易に読み取る

ことができる。逆に、ソースコードという手段(処理)から目的を推測し、プログラムの論理構造を読み取ることは難しい。プログラムの論理構造は、処理の目的をコメントに記述したり、目的でまとめた処理をモジュール分割することによって表現できる。これが人にわかりやすいプログラムを作ることに繋がる。

筆者らが、HCPチャートをプログラミング教育に採用しているのは、このような理由からである。

参 考 文 献

- 1) 花田收悦. プログラム設計図法. 企画センター, 1983.
- 2) 長野宏宣, 浅見秀雄, 忠海均. 階層化プログラム設計図法 -HCPチャート-. 企画センター, 1992.
- 3) 竹田尚彦, 大岩元. プログラム開発経験に基づくソフトウェア技術者育成カリキュラム. 情報処理学会論文誌, Vol. 33, No. 7, pp. 944-954, 1992.
- 4) 松澤芳昭, 青山希, 杉浦学, 川村昌弘, 大岩元. 「目的の表現」に注目したオブジェクト指向プログラミング教育とその評価. 情報処理学会研究会報告, Vol. CE-72, No. 15, pp. 77-84, 2003.
- 5) 松澤芳昭, 杉浦学, 大岩元. プログラム設計教育におけるHCPチャートのレビュー手法. 情報処理学会第66回全国大会, Vol. 1C, No. 2-4, pp. 343-344, 2004.
- 6) 松澤芳昭, 杉浦学, 大岩元. 学習者同士の相互レビューを通じたプログラム設計教育. 情報教育シンポジウム, pp. 189-193, 2004.
- 7) 佐藤匡正, 浅見秀雄. 処理の論理構造と階層化チャート技法HCP. 情報処理学会第23回全国大会, Vol. 5J, No. 8, pp. 407-408, 1981.
- 8) 花田收悦, 佐藤匡正, 松本匡通, 長野宏宣. コンパクト・チャートを用いたプログラム設計法. 情報処理学会論文誌, Vol. 22, No. 1, pp. 44-50, 1981.
- 9) 遠藤 裕香. 構造化プログラム設計図法SPD わかりやすいプログラムへの招待状. 共立出版, 1992.

本 PDF ファイルは 2005 年発行の「第 46 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場（＝情報処理学会電子図書館）で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>