

プログラミング教育におけるエラー分析の一例 (オ ー 報)

平林 隆一, 菅野 文友, 猪股 邦臣
(東京理科大学 工学部) (菱電サービス)

1. はじめに.

近年、大量の高信頼性ソフトウェアが要求される中で、ソフトウェア技術者の不足が問題となっている。本学工学部II部経営工学科の卒業生も、その多くは卒業後、ソフトウェア関連業種に就職して、社会のニーズに添えている。従って、ソフトウェア教育は本学科の教育の重要な部分を占めている。ここでは、2年次の学生を対象として分析を行なった。したがって、学生の大部分はプログラムを書くのは初めてであり、そうでない学生もフォートラン言語を使用したことはない。

学生に課したプログラムは、自分の学籍番号(4桁~5桁)を時間に変換するもの(以下、プログラムAとする)と、1から100までの整数の平均と分散を求めるもの(以下、プログラムBとする)であり、この2種類について検討した。なお、使用言語はフォートランであり、マークカード記入方式で提出されたものである。

学生が提出したプログラムに対し、1回目と2回目および3回目に修正再提出されたときに発生したエラー内容を、原因別に分類し、その発生傾向がどのように変化したかについて、若干の分析を行なった。

2. エラー発生状況.

表1にプログラムAのエラー内容を、また、表2にプログラムBのエラー内容をまとめた。プログラムBに対しては、学生の再提出率が低かったため、エラー件数が1回目に対して、2, 3回目が増減している。したがって、件

数そのものよりも、発生率、順位の方を重視した。

3. 分析.

図1にエラー発生の特性要因図、図2および図3にプログラムA, Bそれぞれのエラーに対するパレート図を示した。時間的前後関係としては、プログラムAが終ってから、プログラムBが課されたので、両プログラムにおけるエラー内容が多少異なっている。

プログラムAにおいては、マークカード使用上のミスがトップを占めているが、プログラムBでは皆無である。これは、マークカードの使用法を一切教育しないで演習を行なわせたために、プログラムAでのミスが多かったと思われる。A, Bを較べて見ると、このような習熟速度の速いものは、わざわざ教育する必要のないことがわかる。これに対し、FORMAT文に関するエラーは、A, B何れにおいても高い割合を占めており、習熟の効果はあまり見られないといつてよい。これは、通常のプログラムのミスも、I/Oに関するものが多いことに対応している。従って、I/O関係は丁寧に教育しなければならないことがわかる。

次に、図4, 図5および図6にプログラムAおよびBの順位相関図を示した。図4から順位相関をみると、プログラムAでは何れも正であり、前回の順位に引きずられるようではあるが、有意な結果とはなっていない。また、寄与率は逐次的に3割程度とみられ、回数が進むと急減している。つまり、3割程度の者が同じ誤まりを発生させたことになる。これは、プログラミング

表1 時間を求めるプログラム(プログラムA)におけるエラー状況

項番	エラー内容	エラー発生件数 (件)			出現率 (%)			順位		
		1回目	2回目	3回目	1回目	2回目	3回目	1回目	2回目	3回目
1	マークカード 使用上の誤り	13	10	1	25.0	29.4	6.8	1	1	5
2	演算式の誤り	10	3	0	17.2	8.8	0.0	2	6	6
3	FORMAT 構文の誤り	10	6	2	19.2	17.6	13.3	2	2	3
4	予約語 スペルミス	8	4	5	15.4	11.8	33.3	4	4	1
5	プログラムが 開始行で始まる	5	0	0	9.6	0.0	0.0	5	9	6
6	文番号定義の誤り	2	5	2	3.8	14.7	13.3	6	3	3
7	データ誤り	2	4	5	3.8	11.8	33.3	6	4	1
8	EOF 誤り	1	1	0	1.9	2.9	0.0	8	7	6
9	その他 構文の誤り	1	1	0	1.9	2.9	0.0	8	7	5
合計		52	34	15	100	100	100	—	—	—

表2(1) 平均と分散を求めるプログラム(プログラムB)におけるエラー状況
(提出回で集計した場合)

項番	エラー内容	エラー発生件数 (件)			出現率 (%)			順位		
		1回目	2回目	3回目	1回目	2回目	3回目	1回目	2回目	3回目
1	FORMAT 構文の誤り	17	2	2	28.8	11.1	28.6	1	4	2
2	予約語 スペルミス	14	1	0	23.7	5.6	0.0	2	6	5
3	演算式の誤り	11	5	1	18.6	27.8	14.3	3	1	3
4	データ誤り	4	4	3	6.8	22.1	42.8	5	2	1
5	マークカード 使用上の誤り	3	3	1	5.1	16.7	14.3	6	3	3
6	文番号定義の誤り	3	1	0	5.1	5.6	0.0	6	6	5
7	その他 構文の誤り	7	2	0	11.9	11.1	0.0	4	4	5
合計		59	18	7	100	100	100	—	—	—

表2(2) プログラムBにおけるエラー状況 (提出回数で集計した場合)

項番	エラー内容	エラー発生件数 (件)			出現率 (%)			順位		
		1回目	2回目	3回目	1回目	2回目	3回目	1回目	2回目	3回目
1	FORMAT 構文の誤り	17	4	0	25.4	26.7	0.0	1	1	2
2	予約語 スペルミス	15	0	0	22.4	0.0	0.0	2	7	2
3	演算式の誤り	13	4	0	19.4	26.7	0.0	3	1	2
4	データ誤り	6	3	2	8.9	20.0	100	5	3	1
5	マークカード 使用上の誤り	5	2	0	7.5	13.2	0.0	6	4	2
6	文番号定義の誤り	3	1	0	4.5	6.7	0.0	7	5	2
7	その他 構文の誤り	8	1	0	11.9	6.7	0.0	4	5	2
合計		67	15	2	100	100	100	—	—	—

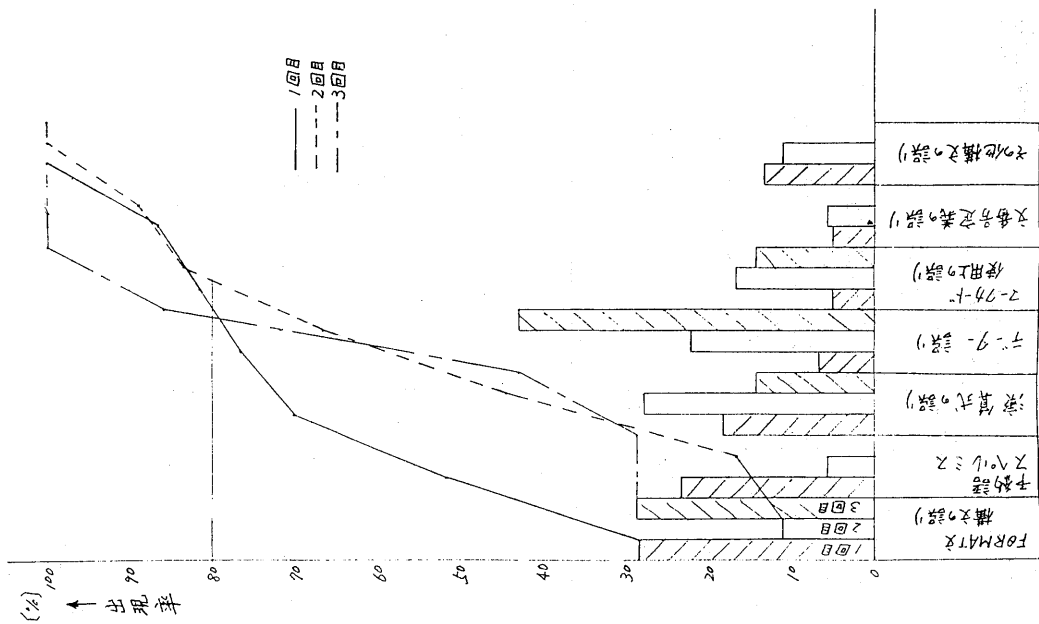


図3. プログラムBにおけるエラーのパーセント図

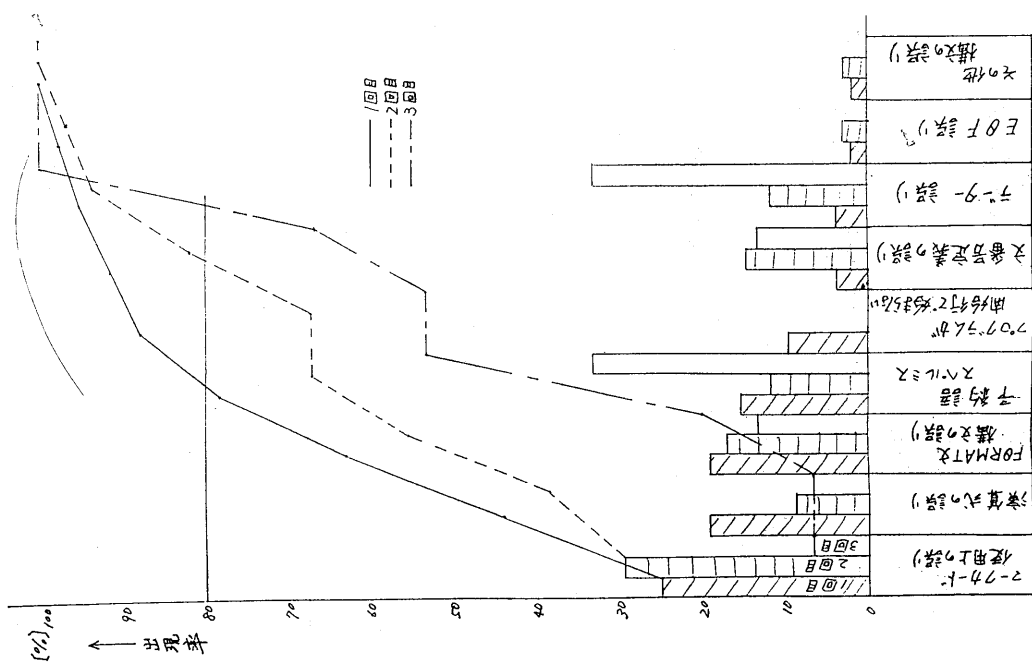


図2. プログラムAにおけるエラーのパーセント図

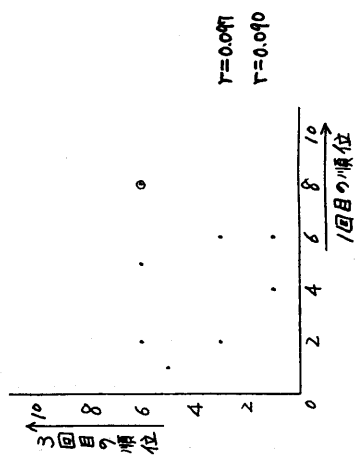
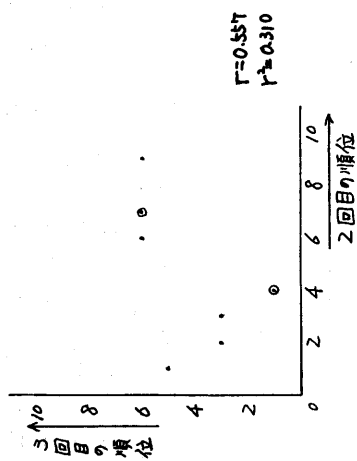
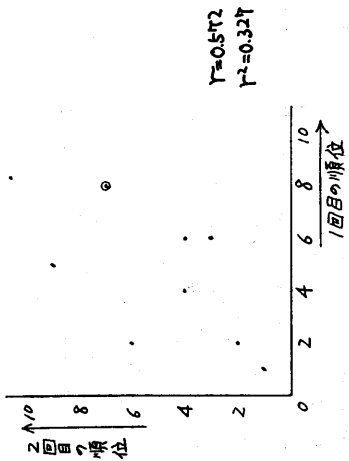


図4 プログラムAの順位相関図

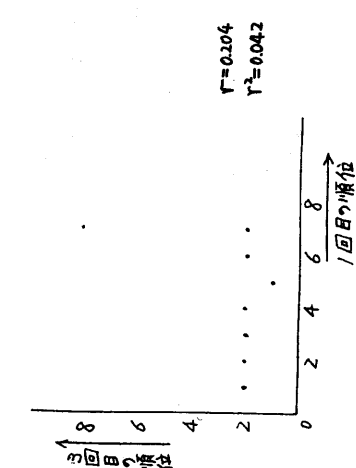
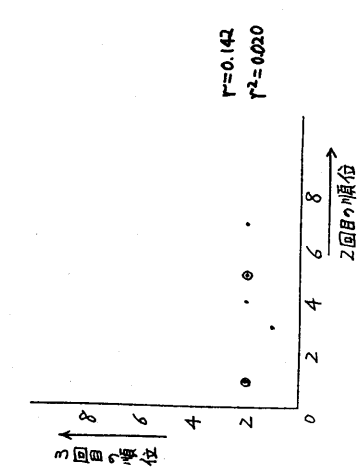
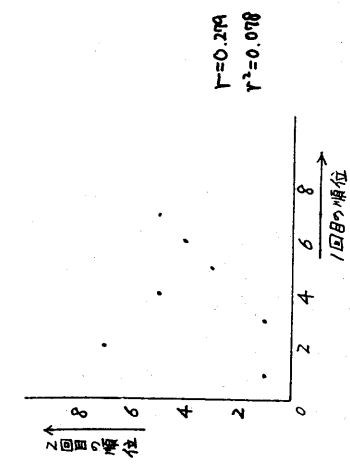


図5. プログラムBの順位相関図
(提出回数で集計した場合)

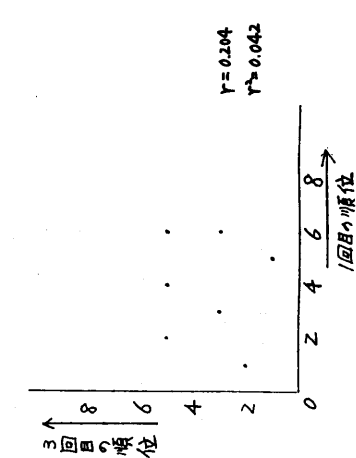
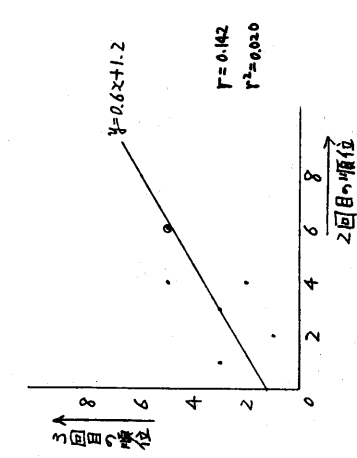
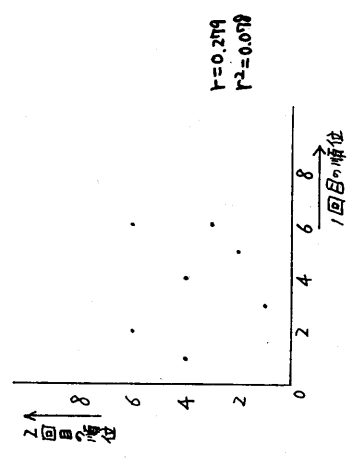


図6. プログラムBの順位相関図
(提出日で集計した場合)

に不慣れで、プログラムまたは同一ステップ中に複数のエラーがあったのを見落したか、あるいは、作製者の性格などから、再び同じ誤りを繰り返したことが考えられる。

プログラムBについては、図5から、各回の相関性は何れも有意でなく、したがって、誤りの設定に関する習熟的影響はみられないことがわかる。また、図6については、2回目と3回目の相関が、有意水準10%で有意となっている。これは、提出期限を過ぎて提出されたものがあったため、2回目としたデータの中に1回目のデータが混入したためである。したがって、図6より図5を重視した方がよいと思われる。

次に、プログラムA, Bの初回時のエラーについての順位相関図を、図7に示す。図7(1)は、プログラムAにあってBにはないエラーは、Bにおいて発生件数0として順位をつけた。また、図7(2)は、Bにおいて発生件数0と

なるエラーをAから除いて順位をつけた。これらの相関性は何れも有意でなく、プログラムA, B間においても、誤りの設定に関する習熟的影響は見られない。

以上のことから、マークカードの記入方法など、マニュアルを読めばすぐ理解可能なことに関しては、習得が速いことがわかる。しかし、プログラム言語の基本的構文(渡算式、予約語のスペルおよびFORMAT文など)は、基本的習得として、特に注意して教育すべきであるといえよう。換言すれば、フォートラン言語におけるFORMAT文は、初学者にとって教育的でない。I/Oに対するわずらわしさのない、ベーシック言語の方が、学生にとって、より速く達成感を味わえるという意味で、教育的であることは既に経験済みである。

4. おわりに

作成したプログラムが1回で実行できることは、ごく稀なことである。裏を返せば、プログラム作成のプロセス上、エラー発生の要因が無数にあるといえる。どんなに注意深くコーディングしても、マークカード記入の際に、うっかりとしたエラーが起こることもありうる。本報告は、ほんの一例にすぎないが、ヒューマンエラーのメカニズムの一端を探る上で、有効な手段であるといえよう。

なお、今後の課題としては、巷間よくいわれる、血液型によるエラーの出方の差、あるいは、男女間の差について、調査をする必要がある。また、マイコン世代が入学してくることにより、ベーシックに既に習熟している学生とそうでない学生との間で、コンパイル言語に対する差が、どの程度現れわけてくるのかについても、今後調査をすすめる必要がある。

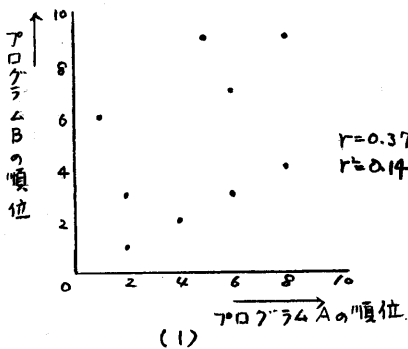
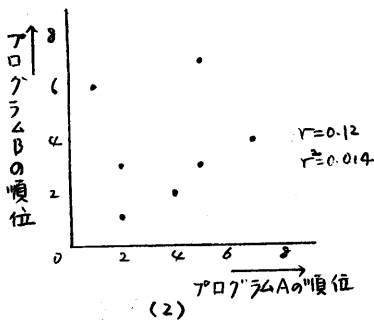


図7. プログラムA, Bの順位相関図