

電算部

群馬
高専

ぐんまこうせんてんさんぶ

2008-2009

総集編



本冊子を手にとっていただき、誠にありがとうございます。

初出

平成 20 年度関東信越地区文化発表会号（2008/09）

平成 21 年度関東信越地区文化発表会号（2009/09）

本冊子は、上記の冊子の総集編として、加筆・訂正を行ったものです。

したがって、当時の状況での内容となっている場合があります。

群馬高専電算部は、本冊子の内容を実行したいかなる結果につきましても、責任を負わないものとします。

もくじ

電算部チラシギャラリー（2009・akina&master）	04
第 20 回プロコンレポート（書き下ろし・tk）	06
プロコンアルゴリズム概略（2009・JJ）	07
D 言語（2008・EDolphin）	10
格安で美しいフォント（2008・akina）	12
俗・格安で美しいフォント（2009・akina）	14
ゲーム制作を語る（2008・山田店長）	16
応用情報処理技術者試験受験記（2009・jinkai）	17
いろいろなプログラム言語（2009・tk）	20
付録 CD について	24

電算部チラシギャラリー

illustration:master/design:akina

今年度の入学式と新入生歓迎会で新入生に配布したチラシをご紹介します。



入学式配布チラシ

群馬高専 電算部

赤の広場プレハブ二階学生会本部隣

特集

プログラミングコンテスト
プロコン '08

予選グループ一位なのに
通過せず! (泣) しかし、
運営の都合で試合は持ち
越しネット対戦へ……

新入生歓迎会記念特別号
定価 0円

トピックス

AMD Phenom X3

噂のトリプルコアプロセッサ搭載パソコンを
導入。その他デュアルコア PCも有。メモリは
大台 3GB or 4GB へ増設。Microsoft Visual
Studio .NET 2003 で快適プログラミング。
やさしい先輩の指導とあわせて上達間違い
なし! プロ仕様ペンタブレット intuos3
+ Adobe Photoshop で CG に挑戦。
有線/無線 LAN 高速インターネット完備。

ゲームプログラミング指南 特別付録 オリジナル CD-ROM



新入生歓迎会配布冊子

どちらも、ますたー先生のイラストに、筆者 (akina) が文字を加えて製作しました。左のチラシは Adobe Photoshop CS2、右の冊子は手描き + ライトトレースで作画されています。全体のデザインは Adobe Illustrator CS2 を用いています。

ますたー先生のイラストは、群馬高専電子情報工学科の牛田啓太助教が著した『教える・学ぶ「C 言語入門」』(技術評論社刊・ISBN 978-4-7741-3941-8・定価 (本体 2,480 円 + 税)) のカバー下イラストとして商業出版されています。

第 20 回プロコンレポート

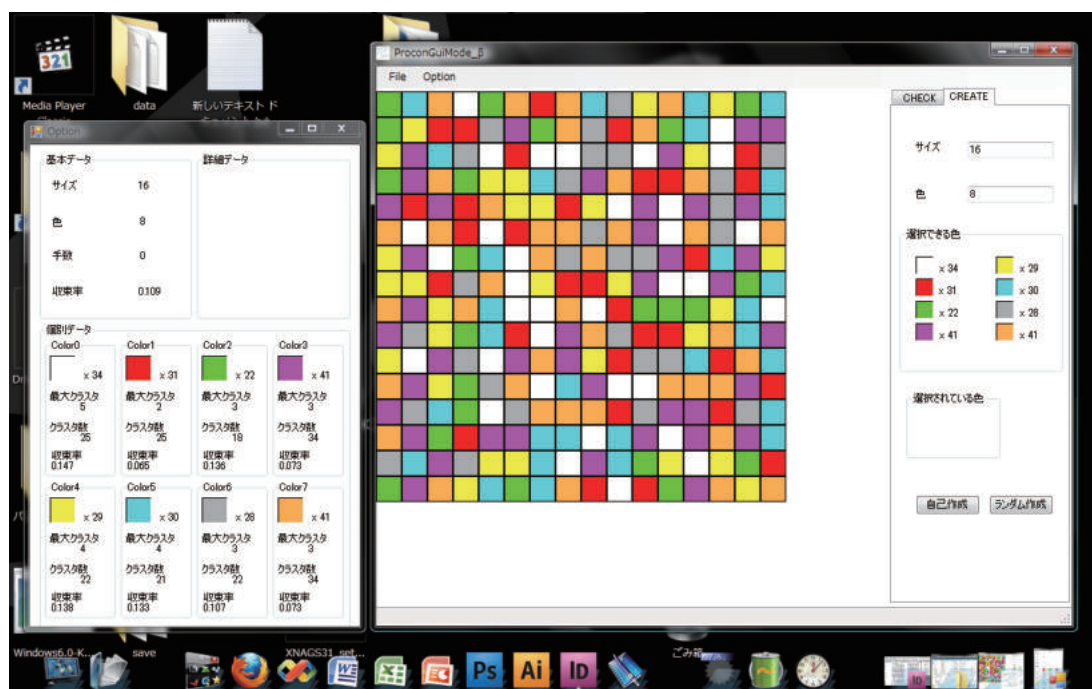
tk

皆さんはプログラミングコンテストというものを知っていますか？プロコンの知名度の低さは、公欠で参加できなかった進路説明会をサボった人と同じ扱いにされたことから分かります。遺憾です。

プログラミングコンテストは競技・自由・課題の3部門から成り立っています。毎年大きな大会となっており、開催する高専は学校をあげて取り組んでいることでしょう。今年の競技部門はボード上のセルを回しながら色をそろえていくというもので、単純ながら奥の深いものとなっていました。課題部門は去年に引き続き「ゆとり」をテーマにした作品、自由部門は独創性あふれる作品が出展されていました。

少し専門的な話になりますが、GPGPU を駆使してきた高専や遺伝的アルゴリズムを用いてきた高専など、初めて出場した私には驚きの連続でした。特に優勝した大阪府立高専のアルゴリズムには目を見張るものがありました。

今回出場したメンバーの一人は来年も出るそうなので、きっと今回の経験をいかして今年以上の結果を残してくれると思います。



スクリーンショット

プロコンアルゴリズム概略

JJ

今回の競技は色を並べかえるパズル。

んでもってかなり制約が多い。

常に $4n^2$ ($n=1,2,3\cdots$) のセルが移動する上、その移動パターンを表現するのが難しい。

というわけで完全読みは捨てる方向に決定。

さて、それではどうやって色をそろえよう。

自分が手動でやるならどうするかな。

たぶん青を左上、赤を右上、黄 (r y
って集めていくだろうな。

というわけで 4 隅にそれぞれの色を集めていく方向で決定。

集める方法だが、セルを 1 つ 1 つ集めていくのでは明らかに効率が悪い。

複数のセルが同時に動くのだから、赤を右上に回すと同時に黄を左下に持っていくべきだ。

そこで思いついたのが「パラメータ」。

たとえば、チェスゲームでは盤のマスそれぞれにパラメータがある。

隅は 100 点、辺は 30 点、他は 10 点…のように。

これを応用する。(なんか他校でも同じような事してそうだが)

パズルの盤の一番左上は青 100 点、離れるに従って 90、80…とする。

また一番右上は赤 100 点、離れるに従って 9 (r y

また一番左下は黄 100 (r y

ま (r y

そして点数の上昇量が最も大きくなるように回す。

回す、回す、回す！！！！

アラ不思議、完成しちゃいました。

……なーんてうまくはいかないんですがね。

特に盤が小さいとセルが引っかかったようになって上手くいかない。
読み入れてないんで当然っちゃ当然なのだが。

まあこれが基本の考え方。

「完成形にする」より「完成形に近づける」を狙った感じ。

なんといっても速さがいい。

20 x 20を10秒で9割完成させる。って書くと凄いプログラムのように見えるが、むしろ速さしかないとも言えるんだよな・・・

読み入れてないから step 数が多いし、しかも完成形には持っていけないという。

そこで現在（8月末）考えているのが「2段構え」。

まず前半、前述のプログラムで7割方完成形に近づける。

そして後半、稼いだ時間を利用し、読みを入れたプログラムで完成させる。

問題はいまだに後半プログラムが1バイトも存在しないこと（オイ

おまけ

15,4

↓

0,2,3,1,0,3,0,1,0,0,1,0,1,3,2

2,2,3,1,2,0,3,0,3,1,3,1,3,3,1

2,2,3,0,3,3,0,1,3,3,3,0,0,1,1

2,3,0,0,1,0,2,0,1,1,2,0,1,1,2

2,1,2,3,1,2,3,0,3,3,0,2,1,1,1

1,1,0,1,2,0,2,0,2,1,2,1,1,1,2

2,1,3,3,2,2,0,3,0,2,1,1,3,2,3

2,2,1,3,0,3,3,1,1,3,3,1,0,2,1

2,1,3,1,1,1,3,0,1,2,0,2,2,3,0

0,3,0,2,2,3,3,2,1,2,1,0,2,2,1

2,3,3,2,0,3,1,2,1,1,0,3,1,2,0

3,0,0,0,2,0,1,0,2,3,0,0,1,2,2

2,0,1,1,2,0,3,0,3,2,1,0,1,1,2

3,2,1,0,2,0,3,2,3,0,0,1,1,0,1

0,3,0,2,1,1,1,1,1,3,2,2,3,1,2

000000030033333

000003303033333

000000003333333

000000030333333

000000330333333

000000033333333

110100003333323

111110102222222

111111112222222

111111122222222

111111112222222

111111112222222

111111112222222

111111121222222

1 1 1 1 1 1 2 2 2 1 2 2 2 2

(7, 4) 1 270°

time=1.17sec

STEP 数 : 283

20,8

4,7,5,2,3,0,3,4,4,7,2,3,2,5,4,0,6,1,3,6

0,5,7,6,4,0,7,2,2,5,2,6,1,0,0,5,0,7,6,4

5,4,1,0,1,1,7,2,2,5,4,5,6,7,0,3,1,5,7,4

0,4,3,5,1,6,7,1,0,7,0,6,1,4,4,4,7,1,2,6

1,0,0,5,3,5,7,0,7,1,0,3,4,0,1,6,1,5,1,5

5,6,4,2,2,5,0,1,4,7,2,6,7,3,0,6,6,0,2,4

6,5,4,5,3,0,5,7,5,7,6,0,0,1,2,7,0,3,7,6

3,7,7,2,3,2,6,1,5,0,1,0,3,2,5,6,7,3,0,2

2,5,0,0,5,7,2,1,1,4,4,2,7,1,3,7,1,4,3,3

0,7,3,3,6,6,1,7,1,0,1,3,0,1,1,4,6,3,0,7

1,5,2,4,0,4,7,2,3,0,3,2,2,2,2,7,5,1,7,0

2,4,6,1,7,4,7,5,3,6,5,2,0,6,7,1,1,3,5,3

5,0,4,6,3,0,5,1,7,5,7,0,7,1,0,5,0,0,6,4

4,1,6,1,7,7,7,0,7,2,5,7,5,1,2,7,7,3,4,0

1,3,4,1,7,6,2,0,0,2,2,2,0,5,1,0,7,4,4,1

4,3,1,4,4,6,6,6,6,2,4,7,7,0,2,2,5,5,5,2

0,3,1,5,1,0,7,4,7,3,5,4,5,2,5,2,0,2,2,7

5,7,0,4,4,2,6,0,2,2,4,1,3,1,3,1,2,3,2,1

0,6,4,3,2,5,2,7,3,1,5,3,2,0,6,4,0,2,7,6

5,0,5,4,3,2,7,6,6,3,5,6,2,3,7,1,5,0,3,7



0 0 0 0 0 0 0 0 0 0 0 3 3 3 3 3 3 3 3 3

0 0 0 0 0 0 0 0 0 4 4 4 4 3 3 3 3 3 3 3

0 0 0 0 0 0 0 0 4 4 4 4 4 3 3 3 3 3 3 3

0 0 0 0 0 0 0 0 4 4 4 4 4 4 3 3 3 3 3 3

0 0 0 0 0 0 0 4 4 4 4 4 4 4 3 3 3 3 3 3

0 0 0 0 0 0 4 4 4 4 4 4 4 4 4 7 3 3 3 3

0 0 0 0 0 5 4 4 4 4 4 4 4 4 7 7 3 3 3 3

0 0 0 0 5 5 5 5 4 4 4 4 4 7 7 7 7 7 7 7

0 5 5 5 5 5 5 5 2 2 2 2 2 2 2 2 2 2 6 1

5 5 5 5 5 5 5 5 2 2 2 2 2 2 2 2 2 2 6 1

1 5 5 5 5 5 5 5 2 2 2 2 2 2 2 6 6 6 6 6

0 5 5 5 5 5 5 5 2 2 2 2 2 2 2 6 6 6 6 6

1 1 1 5 5 5 5 5 2 2 2 2 2 2 6 6 6 6 6 6

1 1 1 1 5 5 5 5 2 2 2 2 2 7 7 6 6 6 6 6

1 1 1 1 1 5 5 5 2 2 2 2 2 7 7 6 6 6 6 6

1 1 1 1 1 5 6 6 2 2 7 7 7 7 7 6 6 6 6 6

1 1 1 1 1 1 1 6 2 7 7 7 7 7 7 7 6 6 7 6

1 1 1 1 1 1 1 1 2 7 7 7 7 7 7 7 7 7 7 4

1 1 1 1 1 1 1 1 7 7 7 7 7 7 7 7 7 7 5 5

1 1 1 1 1 1 1 1 7 7 7 7 7 7 7 7 7 4 6 5

(13,13) 6 180°

time=11.79sec

STEP 数 : 695

うちのペット (2 歳、8 万円)

Intel Core(TM)2 Duo CPU 2.20GHz

D 言語

EDolphin

1 前書きみたいなもの

この部誌ではこの記事がコンストラクタじゃね？ なぜ部誌の題名が D 言語なのかは知らないけど、書けるようなことがなかったので記事にしてみる。というか使ってみたレビューみたいなものだ。どんな特徴があるかの紹介みたいなもの、というか今回初めて使う。

2 D 言語ってどんなもの？

2007 年 1 月 3 日にコンパイラの Ver1.0 がリリースされた、比較的新しい言語で、C 言語の後継を目指して作られたコンパイラ言語だ。文法は C++ に似ている。Digital Mars 社のウォルター・ブライト氏（←と言われても自分もよく知らない）が作った。C 言語のライブラリがそのまま使え、C++ のものも一部互換性がある。数多い C 言語の資産を使用できるのは、大きな強みだと思う。Java や .Net の言語のような GC を実装している。ちなみに拡張子は「.d」。

3 D 言語のコンパイラ

主に 3 つのコンパイラがある。本家「DMD」と、GCC 系の「GDC」、DMD に手を加えた「dli」だ。今回は DMD を使用した。

4 D 言語のプログラムと言語仕様

定番の Hello, world! プログラムは次のようなコードになる。

```
import std.stdio;
void main()
{
    writeln("Hello, world!");
}
```

「import」の部分は Java のようだ。このプログラムだけ見ると、C との違いはそれだけである。移行しやすいように「writeln」関数と同じ機能で「printf」関数も用意されている。実際にコンパイルした実行ファイルのサイズを調べてみたいと思い、「D 言語」と「C 言語」、「C#」で作成した Hello, world! プログラムのサイズを比較してみた。コンパイラはそれぞれ「DMD」、「CL」、「CSC」。

言語	D	C++	C#
サイズ	183KB	47KB	3.5KB

試してから気づいたのだが、C# は比較対照として間違っていた。（CLR 上で動くから）まあ当たり前だが、実行ファイルのサイズは D 言語が大きい。

実行ファイルに GC のコードも含まれるので、大規模になればなるほどさらに差が開くと思われる。実行速度では、C 言語や C++ よりも若干劣る程度だということ。D 言語にもポインタは存在する。C 言語と比べて D 言語で新たに使えるようになった文は、「foreach」や「static if」がある。「foreach」は List や配列の要素一つ一つに同じ処理を行うときに使う。「static if」は、宣言しかできない場所で「if」文を使うことができる。つまり、条件にしたがって、宣言するものを変えることができるということだ。

5 GUI

D 言語で GUI プログラムを作成するには、GUI ライブラリを用意する必要がある。GUI ライブラリは、統一されていないので、自分で選ばなければならない。C 言語用のライブラリも使えるが、D 言語専用のライブラリを使用したほうが簡単だ。今回は、「DFL」というライブラリを試用した。全体的に .Net によく似た仕様になっている。専用のデザイナーもある。メジャーだが、今のところ Windows でしか使えないらしい。右がサンプル。

これを実行すると、「DFL Example」というウィンドウに「Hello, DFL World」という文字が表示される。というかサンプルそのままです。

```
import dfl.all;
int main()
{
    Form myForm;
    Label myLabel;
    myForm = new Form;
    myForm.text = "DFL Example";
    myLabel = new Label;
    myLabel.font = new Font("Verdana",14f);
    myLabel.text = "Hello, DFL World!";
    myLabel.location = Point(15, 15);
    myLabel.autoSize = true;
    myLabel.parent = myForm;
    Application.run(myForm);
    return 0;
}
```

6 使用した感想

GC を使える言語は折衷型言語が多いが、純粋にコンパイルされるということを知って、少し驚いた。まだまだマイナーだが、扱いやすい言語だと思う。

7 用語集

Java Sun のあれ。言語として認識されているけど、もっと広い意味を持ってるらしい。まあ JRuby とかもあるし。

.Net Microsoft のあれ。.Net 構想とか。主に VB.Net、C#、J#、C++/CLI とかがある。みんな折衷型言語。

GCC オープンソースの C コンパイラ。UNIX 系でよく使われてる。

GC ガーベージコレクタ。メモリの自動管理をする。C でも自力で実装できるらしい。

格安で美しいフォント

akina

日頃、フォントについて気にしてみたことがあるだろうか。

読者のみんなが普段使っているフォントといえば、MS ゴシック、MS 明朝、Windows Vista/7 ユーザーであれば、さらにメイリオ程度であろう。

今回は、出版業界やデザインの分野で使われているフォントについて少し解説してみよう。

○ MORISAWA Font Pack for Vista （モリサワ）

デジタルフォントの最大手、モリサワが出している格安フォントパックである。通常、モリサワのフォントは1書体で2万円以上するのだが、このフォントパックは7書体で定価 12,600 円（税込）と超格安である。Vista 専用。

・新ゴ **あいうアイウ永国東 abcABC123**

日本で生活していて、新ゴを見ない日はないだろう。このフォントは「新世代ゴシック体」と呼ばれるゴシック体のなかで一番使われている。使われている例を挙げると、本や雑誌の本文から食品パッケージの栄養表示など、いたるところで使われている。ちなみに、この部誌は見出しから本文まで新ゴを使用している。ウェイト（太さ）は、EL/L/R/M/DB/B/H/U の8ウェイト中、L/M/B の3ウェイトが収録されている。

・ゴシック MB101 **あいうアイウ永国東 abcABC123**

このフォントも非常によく使われている。新ゴと比べ、このフォントは縦画の入りやハネもしっかり表現されている伝統的なゴシック体の部類に入る。雑誌の見出しなどに使われている。ウェイトは、L/R/M/DB/B/H/U の7ウェイト中、R の1ウェイトが収録されている。

・リュウミン **あいうアイウ永国東 abcABC123**

このフォントはスタンダードな明朝体である。日本で最初期に発売されたDTP（デスクトップ・パブリッシング＝大まかにはコンピュータでの出版作業）用のフォントであり、さまざまところで使われている。使われている例を挙

げると、本の本文に多い。高専の授業で使われる数学の教科書もこの書体である。ウェイトは、L/R/M/B/EB/H/EH/U の8ウェイト中、L/R/M の3ウェイトが収録されている。

○ Adobe Type Classics for Learning (アドビ)

フォトショップやイラストレータで有名なアドビのフォントパックである。欧文書体 (Myriad などの有名書体も含む) のパッケージであるが、日本語書体も 26 書体含む約 400 書体を収録。ここでは解説しないが、平成丸ゴシックの太いサイズ (W8) が手に入る数少ないパッケージである。定価は 99 アメリカドル。海外製品であるが、ビックカメラで取り寄せできる。名前に Learning とあるように、アカデミック (学生) 専用パッケージである。ちなみに、単品で下記の2書体をそろえると軽く 10 万円を超える。なお、一部は Adobe の各製品 (フォトショップやイラストレータ) に付属している。

・ 小塚明朝 **あいうアイウ永国東 abcABC123**

リュウミンに比べ、ふところ (字の画線と画線の間) が広がっている、現代的な明朝体である。

雑誌の本文などで使われることが多い。

ウェイトは、EL/L/R/M/B/H の6ウェイト中、スタンダード版が全ウェイト、収録文字数の多いプロ版が L/B の2ウェイト、が収録されている。

・ 小塚ゴシック **あいうアイウ永国東 abcABC123**

このフォントのデザインを担当したのはモリサワの新ゴのデザインを担当した人物と同じである。したがって、新ゴと同じようなデザインだと感じるが、かなの設計が違っている。

小塚明朝と同じく雑誌の見出しや本文で使われることが多い。

ウェイトは、EL/L/R/M/B/H の6ウェイト中、スタンダード版が全ウェイト、収録文字数の多いプロ版が M/H の2ウェイト、が収録されている。

俗・格安で美しいフォント

akina

前項で、「格安で美しいフォント」と題して、出版・デザイン業界で使われている日本語書体について解説した。今回は欧文書体（アルファベット）について解説する。

なお、今回紹介するフォントはすべてインプレス社発行の『改訂 4 版・TrueType フォント・パーフェクトコレクション』（編：深沢英次&インプレス編集部・ISBN 978-4-8443-2514-7）に収録されている。この本は、よくあるフリーフォント集などではなく、業界では有名な米 BitStream 社製の欧文書体が 500 書体も収録されている。しかも「本」なので書店で購入可能で、お値段はなんと定価 2,980 円（本体）！ 通常、BitStream 社のフォントは 1 書体 5,000 円以上もザラなので、安いにもほどがあるといっても過言ではない。

各書体の見本として、小文字の abc、大文字の ABC、数字の 123 と欧文書体を組んだ（並べた）ときに見本となるといわれる Handgloves を示した。

○ Cooper ファミリー

Handgloves abcABC123

セリフ系書体（日本語書体で明朝体にあたる書体）のなかでひととき目立つ存在なので、どこかで見たことがあるだろう。組んだだけでロゴにも使えるほどのデザイン性である。

この書体は、今年度 10/31・11/1 に群馬高専で開催される文化祭「^{こうかさい}工華祭」のテーマ「Trick or Create」のロゴとして、ポスターや校内で発行されている工華祭新聞に用いられている（私が作ったのだが）。

収録されている Light/Medium/Bold/Black の 4 ウェイト（太さ）うち、見本は Black。

○ Garamond ファミリー

Handgloves abcABC123

オールドタイプのセリフ書体。英文の文学本などでよく使われる。昔の（りんごマークが虹色だった頃）Apple 社のロゴや説明文には必ずこの書体が使われていた。

この書体にはかなりの亜種（のようなもの）があり、たとえばこの本に収録されているバージョンのほか、Microsoft Office をインストールすれば必ず一緒に入る Monotype Garamond や現在無料でダウンロードできる Apple

Garamond、昨年の部誌で述べた Adobe Type Classics for Learning に収録されている Adobe Garamond などがある。

収録されている Roman/Bold の 2 ウェイトのうち、見本は Roman。

○ **Times Roman ファミリー** (BitStream 名 Dutch801) Handgloves abcABC123
モダンタイプのセリフ書体。いろいろなところで使われている、セリフの標準書体である。実はたいていの Windows マシンには Times New Roman の名前でインストールされている。

欧文書体の名前は、見出しに「BitStream 名 Dutch 801」と書いたように、メーカーによって異なることが多い。これは、書体自体に著作権が主張できない(仮に主張できたとすれば、すべての印刷物に著作権が主張できることになってしまう)かわりに、書体名を商標登録していることに由来する。

ウェイトやバリエーションはかなりあるので省略するが、見本は Roman。

○ **Helvetica ファミリー** (BitStream 名 Swiss 721) Handgloves abcABC123
サンセリフ書体(日本語書体でゴシック体にあたる書体)の標準。道路標識から商品パッケージまでいたるところで使われている。この部誌の本文書体は日本語が新ゴで、アルファベットはソースコードをのぞいてすべてこれ。

実は、Windows には太めだが、標準で Arial という名前が入っている。

これもかなりのウェイト、バリエーションがあるが、見本は Light。

○ **Frutiger ファミリー** (BitStream 名 Humanist 777) Handgloves abcABC123
Helvetica よりモダンなサンセリフ書体。サイン(表示)用書体として世界中の空港で使われている。日本でも東京メトロが採用しているほか、JR 東日本が番線の表示に用いている(番線以外の部分は Helvetica)。

Light/Roman/Bold/Black/Extra Black の 5 ウェイト中、見本は Roman。

○ **Futura ファミリー** Handgloves abcABC123
サンセリフ系書体の中ではかなり目立つ存在で、まるっこいのが特徴的。これもデザイン性が高く、有名なところではフォルクス・ワーゲンやルイ・ヴィトンがそのロゴに用いている。この部誌の本文は新ゴ + Helvetica と前述したが、実は数字だけはこれを使っている。

同様にたくさんのウェイト、バリエーションがあるが、見本は Medium。

○ **番外：Myriad ファミリー** Handgloves abcABC123
これはこの本には収録されていないのだが、かなり有名なのでここで紹介する。Frutiger をさらにモダンに発展させたサンセリフがこの書体。Apple や Adobe がコーポレートフォントとして広告や取扱説明書に用いている。

基本的に有料のフォントなのだが、Adobe Reader をインストールしてアレ(詳細は ggrks)すると入手できる。見本は Regular。ロゴの使用例：iPod

ゲーム製作を語る

山田店長

STG（編注：シューティングゲーム）の難しさは異常。
クリアじゃなくて、製作する難しさが異常。

ゲームプログラミングの中では STG なんて初歩みたいなモノだと言うけれど、
難しいものは難しい。

まず敵の行動パターン。

敵がいつ出現して、どんな行動をとるかっていうのを決めるわけだけど、そこからして異常。

単純すぎるパターンだとつまらないし、複雑な動きをしても STG として面白みがなくなる。

敵が多く出現すると頭が疲れるし、少ないと画面がスカスカで楽しくない。
体力が多すぎるとストレスだし、逆に即撃破できるようでは全体のバランスが悪くなる。

これだけでも苦労してるのに、さらに頭を抱えるような問題も発生する。

STG の華といえば、弹幕。

当然、弹幕には気合いいれて製作する。

だけど、その弹幕が、どうしても他所の STG に似て見えてしまう。

もちろん、似せようと思って似せているわけじゃない。なのに似てしまう。
完全に同じではない。似ているだけ。動きの一部に共通点が見られるだけ。
だけど、だからこそ、一部に共通点があるからこそ、何か納得がいかない。
珍しくスラスラと作れた弹幕も、実際に見てみるとよくある洗濯機であったり、
「面白い動きを」と考えて作っていた弹幕が「これ、東方な西行がドリキャスじゃん」と途中で気付いたり、そのまんまスカーレットシュートであったり。
逆に、他と全くもって違う弹幕を作ることは相当難しいことなんじゃないかと、途中で悟りました。この際、動きを参考にした方が綺麗に作れるんじゃないかって。そう悟りました。いやあ、STG は奥が深い。

応用情報処理技術者試験受験記

jinkai (塵芥)

4月19日、応用情報技術者試験を受験。結果は合格(午前72.5 午後68.0)。これは、勉強を始めた2月ごろから受験当日、そして二〇からの手紙(※合格証書のこと)が届くまでの記録。あまり参考にならない。

/* テキストはじめました */

2月はじめ。受験を申し込んだ日に買った応用情報技術者試験(以下、応用)のテキストを読み始める。ほとんどわからない。かろうじてアルゴリズムだけは一通り勉強すればできそう。困った。

わからないのはしょうがないので、本屋さんで基本情報技術者試験(以下、基本)のテキストを立ち読みする。読み始めて5分ほど、あることに気づく。「応用と中身一緒じゃね？」

なんと、出てくる単語、計算式などなどほとんど一緒じゃないか！(※後から思えば全然違いました。ただ当時は一緒に思えたのです。)これはつまり、基本テキストを読めば、応用テキストがわかるのではないか。希望が見えた。まずは基本テキストを読み終わることが目標になった。

/* たぶん合格のために */

基本テキストが読み終わったのは2月中旬。このころには、ただ覚えればいいことが多いことに気がついた。覚えるものをリストアップする作業が始まる。並行して応用テキストを読み始める。やはり基本と中身は一緒だった。ほとんどの内容に見たことある！という感じがした。

「意外といけるんじゃない？」

これは、もしかしたら。後から思えば、とんでもない思い違いだった。

旧制度でいうソフトウェア開発技術者試験、これに合格するために必要な勉強時間は450時間という。逆算して、それだけ確保するのは無理だった。新制度での救いは、午後問題が選択問題になっていて12問中6問回答すればよいことだ。合格するためには効率よく勉強しなければ。つまり、山を張るべきだ。(※非推奨です)

/* 期末、襲来 */

期末テストの季節。ぜんぜん勉強できなかった。ちょっとでもやっておけば……。

/* ドキドキ 勉強は終わらない！ */

3月はじめ。応用テキストを読み終わる。それと同時に、過去問題集、予想問題集、SQL問題集を購入。過去問題集は出題傾向を分析するため、解説を読むため。予想問題集は午後対策として一応。SQL問題集はデータベース分野での出題率10割というデータを知って、午後の点稼ぎのため。

出題傾向の分析は次のように行った。

1. 問題に関連のある重要単語を書き出す。[]内は分野。()は出題。

例) IPsec[SC] (20. 秋 .68)(…)(…)

2. これを過去問題集、ネットでの過去問で行う。

3. できた単語集を切り取る。

4. 切り取った単語ごとに、別の紙に書いて意味を書き足す。

大変な作業量だ。3月中に終わらすことを目標に行う。実際に終わったのは4月に入ってからだった。予想問題集は、まだ手をつけない。午前を完璧にしてから午後の勉強をしようと思う。これは間違った判断だった。SQL問題集は、応用テキストよりずいぶん難しい。しかし、過去問のSQLに限り、4月には完璧に解けるようになった。

/* 諦めたらそこで試合終了ですよ */

勉強もまとまらないまま4月に突入。一緒に受ける同級どもは「忘れてた(笑)」などとぼざく始末。幸い、部活の先輩に数人受験者がいたので、情報交換。いい感じで焦る。午後の勉強はデータベース分野(SQL)を除いてしてない。ネットワークと情報セキュリティ必須。しかし、午前の勉強が長引いて結果としてその勉強は2日しかできなかった。

/* 戦場のリアリスト */

試験当日。午前は80問4択問題。技術系の問題はいい感じで解けていく。時間を費やしたのだから、当然。これは余裕かもしれないと思いつつ、ちょっとした不安が芽生え始める。過去問では分野の出る順番はほぼ決まっていて、今回もそのとおりだったが、ちょっとばかりそれぞれの分野での問題数が減っているんじゃないか？ いや、マネジメント・ストラテジ系の問題が増えたんだ、その分だろう……。

予想は的中した。予感というべきか、悪い予感。80問中30問がマネジメント・

ストラテジ系だった。私はその分野の勉強を怠っていたので、絶望した。なぜその分野の勉強をしなかったかという、次のような推測による(ばかばかしく間違ってます)。

過去のソフトウェア開発技術者試験を受けて、落ちた人が今回受ける場合、新たな範囲を勉強しなければ受からない、ということがあってはならない。

実際、確かに技術系だけで問題数の6割超えているけれど、基本すら持っていない私は対策をとってしかるべきだった。点が多いほうがいいのだ。

さて、午後はあらかじめ受けると決めていた分野のひとつ(情報システム開発)が、ちょうど勉強してない部分、クラスが出て、しかたなく勉強してない分野(プロマネ)を選択。

オブジェクト指向を勉強しないで応用を受けたのは私くらいであろう。手ごたえなど微塵も感じなかった。データベースは割りとできた。ネットワークは半分適当、情報セキュリティは6割適当。そんなかんじ。

/* そんなことよりサメの話しようぜ */

試験後は、日常だった。誰も試験の話をしていない。無理に振っても、「あんなの無駄だよ」など。事実そのとおりだと思っていた。

そんな私に、合格の知らせが届いたのはしばらく後のことだった。ホームページで確認したのだが、番号があるではないか！ えっ、って感じだった。そして2週間くらい後、担任から〇階からの手紙、合格証書が手渡されたのであった。

/* 奇跡の価値は */

応用情報技術者試験の前身はソフトウェア開発技術者試験であるが、試験範囲の拡大にともなって、試験の傾向も大きく変わった。今までは多くて数問であったストラテジ系の問題が20問と拡大されていた。みんなが口をそろえて、無駄だったのはそのためで、技術系の問題にしか興味がなかった。

今回の試験では、合格率は26.1%。ソフトウェア開発技術者試験は高くても18%程度である。これを考えると、もしかして簡単な資格に成り下がった？ そんなこともあって、やはり取る価値なし、と判断した人も多い。今後の難易度が上がることを期待？

さまざまなところで合格体験談を読んでいると、やはりその職の人でほとんど無勉強でも受かったという人がいる。今回、知識詰め込みに大変な時間を割いてしまったのは私の知識不足によるものであり、試験への効率のよい勉強法がわからなかったからだろう。もっと計画的に勉強したいものだった。

最後に、部活の先輩も受けたと書いたが、それはつまりこの記を読むかもしれないということで、なんというか、ごめんなさい><。。

いろいろなプログラム言語

tk

何を書けばいいのかわからないので、とりあえず電算部っぽくプログラムの話でも書こうかと思います。（間違えているところもあるはずなので叩かないでくださいネ！）

C

J 科（編注：群馬高専電子情報工学科のこと）愛用の言語ですね。この言語の一番楽しい点はポインタでしょう。ここまで好き勝手に操作できる言語はなかなか無いです。オブジェクト指向が台頭している現在では少し劣勢ですが、組み込みや小規模開発には重宝する言語だと信じています。

C++

C 言語をベースに Class に対応させた言語です。オーバーロードやテンプレート、例外処理など楽しい事がたくさんできます。他言語にありがちな GC は無く、他言語ではあまり見ない多重継承が許されています。非常に複雑でうまく使いこなすのに難しい言語の一つなのではないでしょうか。C++0x が待ち遠しいですね。

Java

上の二つと併せて最も使われているような気がする言語の一つです。仮想マシン（JavaVM と呼ばれています）を通すことで再コンパイルせずにマルチプラットフォームに対応させるという素敵な仕様です。

当時はパフォーマンスが厳しいと敬遠されていたこともありますが、パソコンの処理速度が上がった現在では大きな問題ではありません。個人的に堅実なイメージがあり、C# に大きな影響を与えていると思います。

Ruby

簡潔で拡張性の高い言語です。軽量プログラミング言語と呼ばれる言語の一種なのでプログラマに優しいです。

ブロックという構文が特徴的で、クロージャやイテレータを使用します。メ

タブプログラミングやダックタイピングなどの動きは非常に面白く、是非一度触ってみてはどうでしょうか。最近はこの種類の言語が人気な気がします。

D

部誌のタイトルです。D1.0 と D2.0、ライブラリには Phabos と Tango があり、何を選択するかで分かります。さらに様々な言語から良いところを集めたような言語仕様で、何が何だかわからなくなります。

SDL や OpenGL、Lua などと組み合わせてゲーム開発に用いると、とても楽しく作ることができるのでお勧めです。

JavaScript

主に Web で用いられているスクリプト言語です。

言語仕様が気持ち悪いなどと言われることもありますが、ほぼ全ての Web ブラウザに処理系が組み込まれています。実は世界で最も普及したプログラミング言語の一つであり、最も誤解されたプログラミング言語でもあります。クラスベースではなく、プロトタイプベースなオブジェクト指向言語なので、組み込み系として使うのに非常に有効です。

Web2.0 の時代になって必須とも言えるようになった、Ajax 技術を支えています。

ActionScript3.0

Flash に使われる言語です。

これもオブジェクト指向で、Flash を効率よく、楽しく作ることができます。

ただ個人的に言語仕様が落ち着かない感じがします。ちょっとした Flash を作るのに最適です。

Processing

グラフィックに特化している言語・総合開発環境です。

Java をベースにしており、簡単に書くことができるのでちょっとしたグラフィック作成には最適です。

Haskell

純粋関数型言語です。関数型言語でお馴染みの高階関数やカリー化、パターンマッチング、遅延処理といった機能があります。特徴的なのはモナドで、入

出力にも使われる身近なものですが、難解な機能でもあります。うまく使うと手続き型よりも短いコードで複雑な処理ができて楽しいです。

Scala

紹介した言語の中では最も新しいオブジェクト指向・関数型言語です。JavaVM 上で動作し、Java との相性も抜群です。シンプルで強力なコードを書くことができるのでこれから今以上に知名度が上がっていくことを期待しています。

Twitter は Ruby On Rails で構築されていますが、最近ではこの Scala での置き換えが行われているようです。

Brainf*ck

難解プログラミング言語の一種です。名称がアレなので一部隠しています。実用性はほぼありません。冗談で作られた言語ですね。同じ難解プログラミング言語の一種である Whitespace のソースはとても美しいので一度見てみると良いでしょう。

次のページには、それぞれの言語で書かれた、いわゆる「Hello,world!」のプログラムです。コピペしてコンパイラもしくはインタプリタで処理すれば Hello,world! が表示されます。言語の名前はコメントの形式で示しました。ただし、Brainf*ck にコメントはありませんので一行目は消してください。一部処理系依存するかもしれません。

ソースコード書体見本

Consolas (Vista 標準フォント)

abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

() _ - / * + @ ! " # \$ % & ' { } [] | ~ =

付録 CD について

CD には以下のファイルが含まれております。

○新作プログラム（コミックマーケット 76 出展作品）

クイズの国のアリスたん 作：しまぱん、からぼん（グラフィック：ますたー）

○旧作プログラム

シューティング 作：JJ

リバーシ（オセロ） 作：塵芥

東方工華鏡（シューティング） 作：akina（グラフィック：ますたー）

○卒業生製作プログラム

から風（落ち物系） 作：輝凜

TypeHell（タイピング） 作：Skyblue

流れ星（シューティング） 作：いちご、よもぎ、霧ヶ峰ジュン、さっちゃん

○画像など 作：ますたー、EXILIAS

○その他 本誌 PDF 版、文発 2008 部誌 PDF 版など

収録されているプログラムは Microsoft Windows XP Professional SP3、Windows Vista Business SP2 で動作確認を行っております。なお、すべてのコンピュータにおいて動作を保証するものではありません。また、各プログラムの著作権はそれぞれの制作者にあります。無断使用・転載は大歓迎します。バグの報告、ご感想などありましたら、下記メールアドレスまでお待ちしております。

メール gnct.densan@gmail.com

編集後記

最後までお読みくださりましてありがとうございました。付録 CD のゲームでも遊んでいただけましたら幸いです。（編集担当 副部長 akina）

装丁 akina / 表紙イラスト ますたー

電算部部誌 2008-2009 総集編

著 者 群馬高専電算部

発行日 平成 21 年 10 月 31 日 初版

発行所 群馬工業高等専門学校
