

# 資料1：IT系人材派遣会社による求人数集計結果

●出典：[http://www.workport.co.jp/news/news\\_090213.html](http://www.workport.co.jp/news/news_090213.html)

●概要：IT系人材派遣会社(株)ワークポートによる、プログラム言語別求人数の集計結果。今日のIT技術動向を押し量る材料のひとつと考えられる。

順位	言語	オブジェクト指向技術	WEBプログラミング技術	2008年求人数	2007年求人数	求人数全体に対する比率(2008)
1位	JAVA	○	○	683	783	23.2%
2位	C			653	764	22.2%
3位	C++	○		345	314	11.7%
4位	PHP	△	○	272	270	9.2%
5位	C#	○	○	223	172	7.6%
6位	VisualBasic	△		189	148	6.4%
7位	Perl		○	162	205	5.5%
8位	JavaScript	△	○	110	111	3.7%
9位	COBOL			60	62	2.0%
10位	VisualBASIC.NET	○	○	54	23	1.8%
11位	ruby	○	○	58	43	2.0%
12位	ActionScript	○	○	37	30	1.3%
13位	ASP.NET	○	○	37	24	1.3%
14位	PL/SQL			31	20	1.1%
15位	VBA			29	30	1.0%

※○は関連性の高さ、△は条件付きで関連性が高いことを意味する

※全体に対する比率は求人総数を 2943 件として計算

※「ASP.NET」はプログラム言語ではないが、プログラム言語と同様に求人の条件として含まれているため、他の言語と同様に扱われている。

●ポイント

①オブジェクト指向技術はソフトウェア開発プロセスの全体（主に分析、設計、実装）をカバーする1方法論であり、WEBプログラミング技術は、オンラインショッピングサイト、掲示板、ブログ、SNSといったWEBアプリケーションを開発するための製品技術（目的をもった技術）

②2008年オブジェクト指向技術に関連する求人数合計の全体に対する比率 68.2%

③2008年WEBプログラミング技術に関連する求人数合計の全体に対する比率 55.6%

## 20 資料2：専門学校における教育動向（2008年度調査）

21

22

23

24

25

26

27

28

29

30

- 概要：2008年度の都内の主要な情報系専門学校のカリキュラム（ネット上やパンフレットなどで公開されているもの）に含まれる分野別授業時間数を独自に集計した結果。情報系専門学校の教育動向を推し量る材料と考えられる。ここで、「オブジェクト指向実装」は、JavaやC#といったオブジェクト指向言語によるプログラミング技術。「オブジェクト指向設計」はUML、オブジェクト指向設計の原則、ソフトウェアパターンを駆使した設計技術。「オブジェクト指向分析」は、OOSE法やICONIX法に基づいたドメイン分析、要求分析、ロバストネス分析などの上流工程を構成する技術。「WEBプログラミング」は、PHP、Perl、ASP.NET、J2EE、Flash、JavaScriptなどのWEBアプリケーション・RIA (Rich Internet Application) 関連技術を意味する。

学校	学科	年制	オブジェクト指向 実装系科目	オブジェクト指向 設計系科目	WEBプログラミン グ系科目	データベース 系科目	総時間数
A校	WEB系学科	2	240/11.8%	30/1.5%	120/5.9%	60/2.9%	2,040
	情報処理系学科	3	240/7.9%	30/1.0%	30/1.0%	90/3.0%	3,030
	情報処理系学科	2	240/11.6%	0	0	90/4.3%	2,070
B校	WEB系学科	2	345/17.3%	300/15%	125/6.3%	60/3%	2,000
	情報処理系学科	2	300/15%	225/11.3%	60/3.0%	180/9.0%	2,000
C校	WEB系学科	2	360/20.7%	0	150/8.6%	0	1,740

31

32

※45分授業を1時間に換算

33

※授業内容は、授業タイトルおよびシラバスより判定

34

※総時間数は、選択科目、必修科目、卒業制作などの時間の総数として算出

35

36

## ●ポイント

37

- ①オブジェクト指向実装に関する授業が、カリキュラム全体の10%～20%程度(300時間前後)。

38

カリキュラム中の主要要素となるべきプログラミング技術の比率がこの程度で、本当に十分なプログラマー育成教育が可能なのか。WEBプログラミングに関する授業時間はさらに少ない。

39

40

- ②オブジェクト指向設計に関する授業時間の比率はさらに低く、また、オブジェクト指向分析に関する授業はほとんど見られないため、オブジェクト指向技術を駆使するシステムエンジニアや上流工程に関わるアーキテクトは育成できない。

41

42

43

- ③「基本情報技術者試験」への対応などが影響して、専門学校のカリキュラムは、今日のIT技術動向から大きく外れているのではないか。

44

## 45 資料3：大学における教育動向（2008年度調査）

46

47 ●概要：有名大学理工系学部情報系学科におけるオブジェクト指向実装（Java）科目内容の実例。  
48 全部で11コマの授業からなる。以下、授業各コマのテーマ。

49

50

1. 講義概要、Javaとは、計算、変数と型

51

2. 条件分岐、繰り返し、メソッド

52

3. 配列、上巻のまとめ

53

4. オブジェクト指向とは、クラス、オブジェクト、属性

54

5. 継承、インタフェース、例外処理

55

6. ガーベッジコレクション、デバッグ

56

7. プログラミング演習(I)

57

8. GUI、コレクション

58

9. マルチスレッド

59

10. ファイル入出力、ネットワーク

60

11. プログラミング演習(II)

61

62 ※それぞれ1回の時間は90分

63

64

65 ●ポイント

66 ①オブジェクト指向実装に関する科目の時間数は、45分換算でわずか22時間。授業時間が不足  
67 しているのではないかと危惧される情報系専門学校におけるオブジェクト指向実装に関する授  
68 業時間のさらに10分の1を下回る。

69 ②オブジェクト指向設計・オブジェクト指向分析は、別途、「ソフトウェア工学」科目で扱われ  
70 る。授業規模はオブジェクト指向実装と同程度と思われる。

71 ③WEBプログラミング技術に関する授業の存在は確認できない。少なくとも、学科内のカリキュ  
72 ラムではカバーしていない。

73 ④仮に、オブジェクト指向設計・オブジェクト指向分析に関する授業が内容的に充実していたと  
74 しても、実際にプログラムを作成する技術（実装技術）やWEBプログラミングといった製品  
75 技術に関する知識が希薄となるため、オブジェクト指向設計・オブジェクト指向分析といった  
76 技術の意義やメリットを認識しにくいのではないかと。また、知識と職業との関連性に関する認  
77 識も希薄になるのではないかと。

78

79 資料4：全国スキル調査2008の結果に見られる情報教育の"効果"

80

81

●出典：<http://itpro.nikkeibp.co.jp/article/COLUMN/20090121/323151/>

82

●概要：ITスキル研究フォーラム（iSRF/アイサーフ）主催の調査。3万人を対象に、ITSSに準拠したスキルチェックテスト（ITSS-DS）をWEB上で受験した結果を集計したもの。ITSSに準拠して、スキルレベルが未経験から7までの8段階で判定されている。受験者の48.3%がソフトウェア開発に関わるエンジニアである（アプリケーションスペシャリスト：37.3%、ソフトウェア開発：6.3%、ITアーキテクト：4.7%）。

83

84

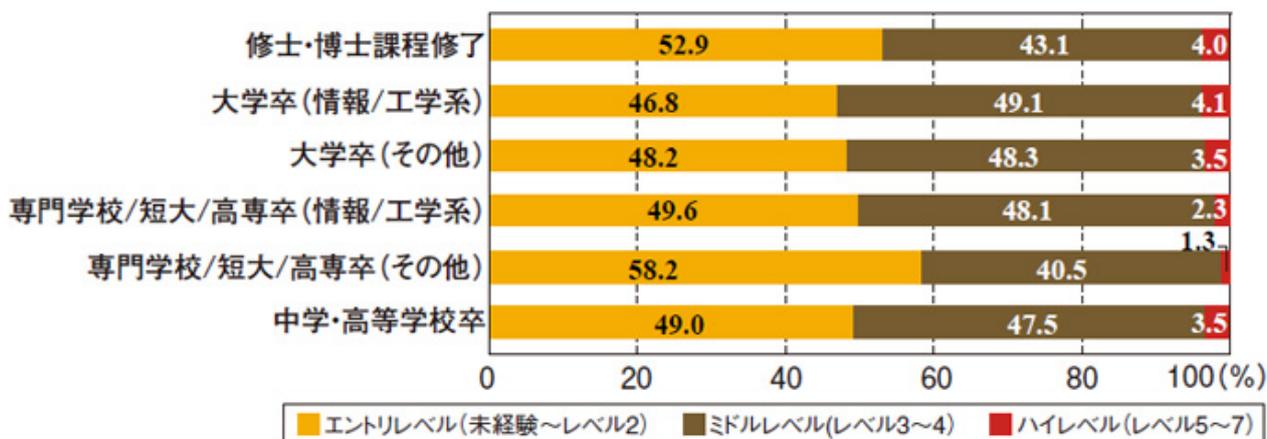
85

86

87

88

1) 最終学歴別に見た、スキルレベルの分布

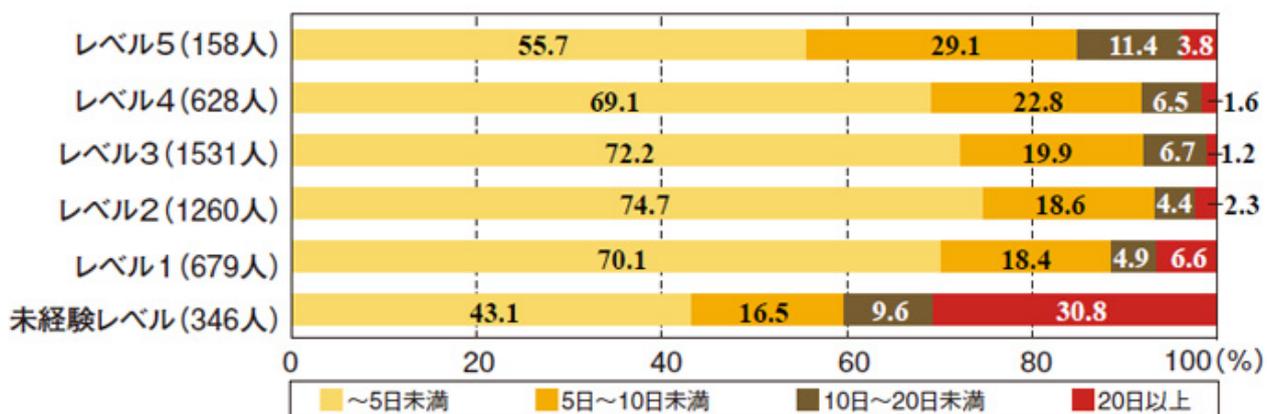


89

90

91

2) スキルレベル別に見た、会社で1年間に受ける教育・研修の日数



\*回答者数が全体の0.1%(30人)を下回ったレベル6とレベル7については省いた

92

93

●ポイント

94

①大学・専門学校における情報教育は、その後のエンジニアのスキルレベルにほとんど影響していない（中学・高等学校卒エンジニアと優位な差が見られない）。

95

96

②このような結果になる一因は、現場の技術動向と、大学・専門学校における教育動向の不一致にあるのではないか。

97

98

③しかも、企業入社後の教育・研修も充実しているとは言い難い。

99

④学生時代の教育も入社後の教育も期待できないエンジニアは、どうやってエンジニアとして成長していくのか。

100

101 資料5：企業における人材育成実態調査2008の結果

- 102 ●出典：<http://help.goo.ne.jp/info/detail/1237/>
- 103 ●概要：NTT レゾナント株式会社が、国内の株式公開・非公開企業138社（有効回答者数）の人
- 104 事・人材育成部門の管理職以上を対象に、企業における人材育成の実態調査を行った結果。
- 105 対象業種の内訳は、製造業28%、運輸・情報通信業15%、サービス業23%、商業17%、
- 106 金融・保険業7%、建設業7%、不動産業7%、電気・ガス業1%。

107 1) 研修テーマと研修形式

	実施形式						平均	
	集合研修 (n=576)	合宿型研修 (n=247)	eラーニング (n=198)	OJT (n=253)	自己啓発=ビ ジネスス クール・セ ミナー (n=210)	自己啓発=書 籍・教材 (n=234)		
研修 テ マ	新人研修 (n=375)	88.4%	40.6%	26.8%	60.1%	24.6%	31.2%	45.3%
	若手研修 (n=251)	53.6%	26.1%	19.6%	31.9%	24.6%	26.1%	30.3%
	中堅研修 (n=235)	54.3%	23.9%	18.8%	23.9%	23.9%	25.4%	28.4%
	主任・係長研修 (n=203)	50.7%	21.0%	15.2%	16.7%	21.7%	21.7%	24.5%
	新任管理職研修 (n=233)	68.1%	28.3%	15.9%	17.4%	18.1%	21.0%	28.1%
	管理職研修 (n=226)	59.4%	26.8%	16.7%	17.4%	21.0%	22.5%	27.3%
	全員研修 (n=195)	42.8%	12.3%	30.4%	15.9%	18.1%	21.7%	23.6%
	平均	59.6%	25.6%	20.5%	26.2%	21.7%	24.2%	29.6%

109 2) 研修に対する（管理職から見た）満足度

	実施形式						平均	
	集合研修	合宿型研修	eラーニング	OJT	自己啓発 ビジネスス クール ・セ ミナー	自己啓発 書籍・教材		
研修 テ マ	新人研修	50.0%	54.5%	29.7%	39.6%	22.1%	31.4%	41.5%
	若手研修	39.2%	44.4%	27.8%	37.5%	19.1%	25.0%	33.7%
	中堅研修	34.0%	34.8%	25.0%	28.6%	19.7%	22.8%	28.7%
	主任・係長研修	41.4%	37.9%	23.8%	21.7%	18.3%	21.7%	30.5%
	新任管理職研修	44.7%	38.5%	27.3%	31.3%	16.0%	25.9%	35.2%
	管理職研修	37.2%	35.5%	30.4%	20.6%	22.4%	24.2%	31.0%
	全員研修	35.6%	23.5%	33.3%	20.5%	20.0%	23.3%	28.5%
	平均	41.3%	41.1%	28.6%	31.6%	19.6%	25.2%	32.6%

112 3) 研修に対する現場の認識

113 この調査の報告書では、「現場の不満は実務に沿っていない（『研修効果が不明確（61.6%）』『即  
114 戦力の育成に役立たない（44.9%）』『研修内容が不足している（40.6%）』）ことがうかがえ、  
115 現場の求めている研修と実際に実施されている研修との間のズレが生じていて、そのズレが不  
116 満の原因となっている可能性があります。」と指摘されている。

117 ●ポイント

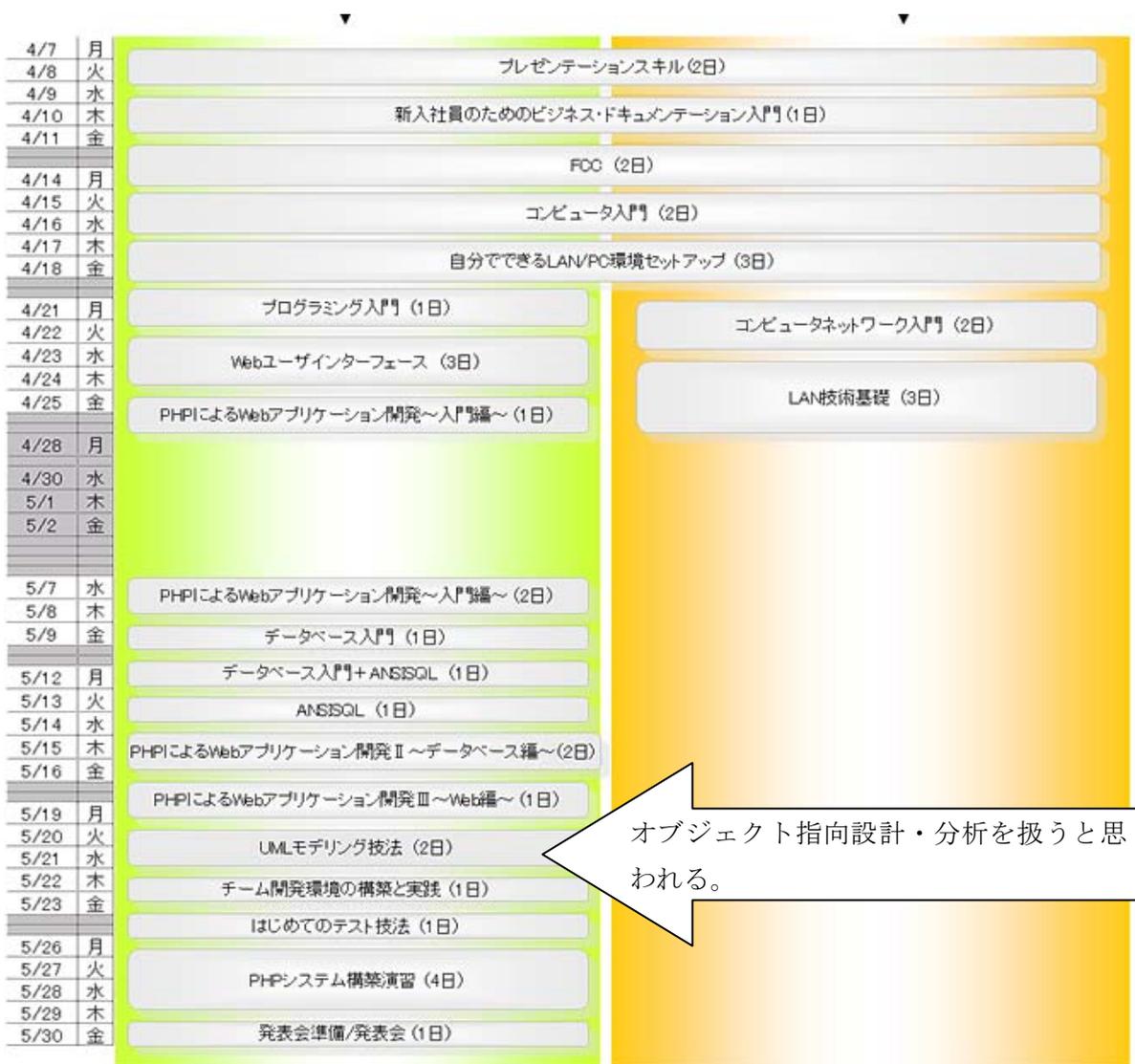
- 118 ①この調査は、IT企業のみを対象としたものではないが、集合研修型の新人教育を始めとする企  
119 業内教育・研修が、あまり充実したものとなっていない（質的にも量的にも）ことが伺える。

124 資料 6 : IT 企業向け集合型新人研修の事例

125

126 ●概要 : 代表的な IT 系研修サービス企業の新人向け研修サービスの事例 (WEB プログラミング系  
127 コースのひとつと思われる)

128



129

130

131 ※1日の研修時間は、推定で6時間(昼食時間などを除外)。45分換算で8時間。

132

133 ●ポイント

134 ①実装技術に関しては、正味 11-14 日程度しか行われていない。45 分換算で 88-112 時間 (2  
135 年制専門学校の 3分の1 程度か)。

136 ②オブジェクト指向設計や分析に関しては2日しか時間がとられていない。45分換算で16時間。

137 ③この事例は PHP を扱うコースであり、オブジェクト指向実装技術 (Java や C#) を扱うコース  
138 ではないが、事例として紹介されている点から、新人研修コース全体の推定材料となる。

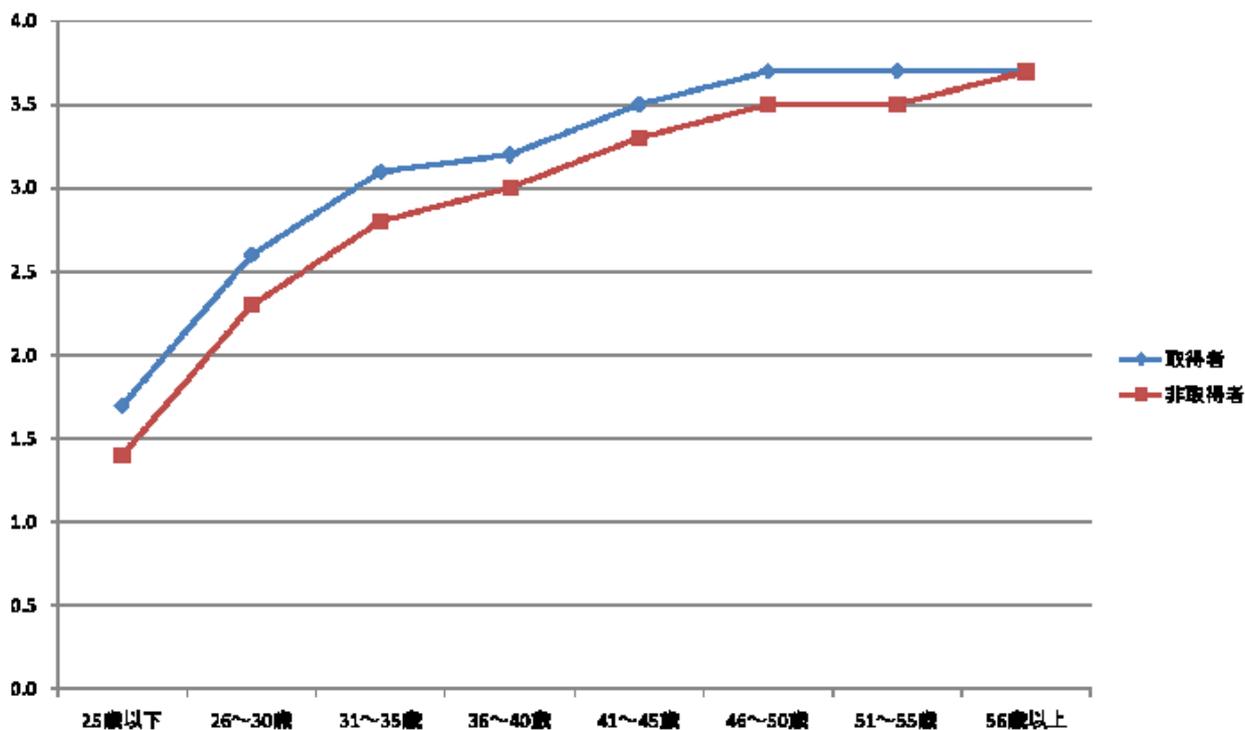
139 資料7:基本情報処理資格のITスキルの関係(全国スキル調査2007)

140  
141  
142  
143  
144  
145

●出典：<https://www.isrf.jp/home/event/chousa.asp>

●概要:資料4に挙げた調査の2007年度版の報告書に含まれる。IT企業の経営者や人事関係者が、就職希望者や採用内定者、情報系教育機関関係者に勧める(採用試験エントリの条件となることさえある)「基本情報技術者試験」の取得とITSSスキルレベルとの関係を示すデータ。

## 基本情報技術者試験の 取得者と非取得者のレベル



146

147

●ポイント

- 149 ①25才以下(おそらくほぼ新人と考えられる)では、「基本情報技術者」取得によって得られた
- 150 スキルの上乘せは、ITSSスキルレベル換算で0.3ポイント程度。初級シスアドと基本情報技
- 151 術者のスキルレベルの差が1.0であり、さらに、その3分の1の実力差しか認められない。
- 152 ②情報系教育機関は、IT企業の経営者や人事関係者に勧められるがまま、「基本情報技術者」取
- 153 得に全力を傾けるほど、この資格に価値を見いだして良いのか。
- 154 ③もっと、今日のIT業界の(開発現場の)技術動向を見極めて、カリキュラムの構成を検討す
- 155 るべきではないか。

156 資料 8 : オブジェクト指向技術 4年制カリキュラムの全体像

157 ●概要 : 本研究における 4年制カリキュラムの全体像と東京工科専門学校 WEBプログラミング科  
 158 (2年制)カリキュラムとの比較 (表中の矢印の番号は「ポイント」の番号と対応)

分野	主な学習要素	2年制 分野別 合計 履修 時間	4年制 分野別 合計 履修 時間	増加率	4年制 対総 時間比
一般教養	時事問題 外国語	120	300	250.0%	8.4%
IT基礎知識	プログラム言語の歴史 データベースの役割 コンピュータのアーキテクチャ コンピュータ数学 インターネット・ネットワーク	195	225	115.4%	6.3%
オブジェクト指向実装	Javaの文法 Javaのライブラリ 実装パターン Java仮想マシン	495	915	184.8%	25.6%
オブジェクト指向設計	UML オブジェクト指向の原則 ソフトウェアパターン フレームワークの開発・利用	345	510	147.8%	14.3%
オブジェクト指向分析・ 開発プロセス	ドメイン分析 要求分析 ロバストネス分析 開発プロセス	0	270	—	7.6%
データベース	SQLによるオペレーション データベースの作成・管理 パフォーマンスチューニング 概念・論理・物理設計	300	705	235.0%	19.7%
WEBプログラミング	HTML Flash・JavaScript J2EE WEBサービス	255	375	147.1%	10.5%
資格対策	SJC-P Oracle Master 10g Bronze OCUP Fundamental XMLマスター ベーシック	210	270	128.6%	7.6%
総履修時間		1,920	3,570		

●ポイント

- ①IT 基礎知識は、「基本情報技術者試験」の出題傾向に合わせるのではなく、各分野にとって本当に必要な内容に限定している。
- ②オブジェクト指向実装・設計・分析で、カリキュラム全体の 47.5%を占める
- ③従来、専門学校が扱ってこなかった実装パターン (資料 p.9 参照) とオブジェクト指向設計 (資料 p.10 参照)、オブジェクト指向分析、データベースに関するパフォーマンスチューニング・物理設計を含む。
- ④2年制カリキュラムでは、オブジェクト指向分析、データベースに関する高度技術 (設計・チューニング) を欠き、プログラマー・SE 止まりだったが、4年制となることで、アーキテクトの育成が可能になる。
- ⑤資格試験は重要視していない。カリキュラム全体の 7.6%。

## 171 資料9：実装パターンとは何か

172

173 ●概要：実装パターンこそ、プログラミング経験に依存せずに、教育や学習によってコーディング  
 174 の質を向上させる有力な手段の1つである。デザインパターンを云々する以前に、実装パター  
 175 ンをまず教育すべき。以下は、実装パターンの1つである Composed Method Pattern の  
 176 紹介。

177

## 178 ●Composed Method Pattern適用前

```
179 class Product{
180     private int _price; private double _tax;
181     Product(int price, double tax) { _price=price; _tax=tax;}
182     public void showTaxedPrice() {
183         int taxedPrice=(int) (_price+(_price*_tax)); //税込価格の計算
184         System.out.println(taxedPrice);
185     }
186     public void showAll() {
187         int taxedPrice=(int) (_price+(_price*_tax)); //税込価格の計算
188         System.out.println(_price+¥t+_tax+¥t+taxedPrice);
189     }
190 }
```

## 191 ●Composed Method Pattern適用後

```
192 class Product{
193     private int _price; private double _tax;
194     Product(int price, double tax) { _price=price; _tax=tax;}
195     public void showTaxedPrice() {
196         int taxedPrice = getTaxedPrice();
197         System.out.println(taxedPrice);
198     }
199     public void showAll() {
200         int taxedPrice = getTaxedPrice();
201         System.out.println(_price+¥t+_tax+¥t+taxedPrice);
202     }
203     //税込価格を計算する処理を独立させる。
204     private int getTaxedPrice() {
205         int taxedPrice=(int) (_price+(_price*_tax));
206         return taxedPrice;
207     }
208 }
```

たったこれだけの工夫で、コードの可読性、再利用性、柔軟性が向上する。これを知っているだけで、経験に依存せずに、コーディング能力を向上させることができる。

209 資料 10：オブジェクト指向設計技術として教えるべき内容

210

211

●概要：“オブジェクト指向教育”と言えば、「カプセル化」、「継承」、「多態性」で済ませる傾向があるが、それは Java などのオブジェクト指向言語の基本文法程度（オブジェクト指向実装）の教育でしかない。オブジェクト指向が価値を発揮する分析・設計について教育するためには、“オブジェクト指向設計教育”に、「開放-閉鎖原則」（Open-Closed Principle）を始めとする Robert C. Martin の「オブジェクト指向設計の原則」は欠かせない。以下、オブジェクト指向設計の原則についてのコマシラバスの一部。

212

213

214

215

216

217

<p>オブジェクト指向の原則 1</p>	<p>①オブジェクト指向設計原則とは、再利用可能なシステムを構築するための設計原則であること解説する。オブジェクト指向による設計のメリットである再利用性について確認した後、オブジェクト指向設計原則の適用時にシステムの再利用性が確保されることを、原則に準拠している設計と違反している設計を用いて、システムに対して機能の変更や追加が発生した場合に、必要になる対応の違いを比較することで解説する。本科目では、設計内容を解説する際にUMLについての知識が不可欠となるため、原則に準拠している設計と違反している設計をUMLで提示して読み方の確認もしておくこと。</p> <p>②Robert C. Martinがまとめたオブジェクト指向設計における5つの原則(オブジェクト指向設計5原則)を一覧として提示して、全体像を紹介する。</p> <p>③5原則の1つである、開放-閉鎖原則(OCP:The Open-Closed Principle)について解説する。Robert C. Martinが著書で定義している(※)OCPを紹介し、その意味を解説する。この際、ソフトウェアの構成要素や拡張、修正といった言葉が何を指しているのかについて解説することで原則の意味を理解させる。ソフトウェアの構成要素についてはクラスやインターフェイス等、具体的かつ身近な例をあげるだけでよいが、拡張と修正については別途解説が必要になる。“拡張”とは、ソースコードの修正を伴わずにシステムの機能を追加・変更することを指し、“修正”とは、ソースコードを修正することでシステムの機能を追加・変更することを指していることを、UMLやソースコードを使って、システムに対して“拡張”するケースと、“修正”するケースの具体例をあげ、解説する。また、ソースコードを修正せずに機能の追加・修正を行う“拡張”を可能にするためには、次コマ以降に解説する他の原則に準拠することが必要であることを解説しておく。なお、本コマでは、拡張によって機能の追加・変更に対応することのメリットからOCPの意味について解説することを趣旨とし、OCPの実現方法については、他の原則を理解する必要があるため詳しくは触れない。</p>
<p>オブジェクト指向の原則 2</p>	<p>①5原則の1つである、依存関係逆転の原則(DIP:The Dependency Inversion Principle)について解説する。Robert C. Martinが著書で定義しているDIPを紹介し、その意味を解説する。また、階層構造を設計に取り込む際に、レイヤ間(上位レイヤと下位レイヤ)の依存関係における、守るべき法則を定めたものがDIPであることを解説する。他の科目(オブジェクト指向設計2(レイヤリング))で、Layersパターンとして階層構造を設計に取り込む意味について解説しているが、再度、確認として簡単に解説をしてもよい。その後、原則に準拠している(クライアントが抽象に依存している)設計と、違反している(クライアントが具象サブライアに依存している)設計を、UMLとソースコードで提示し、システムに対して機能の追加・修正が発生した場合の再利用性や保守性の違いを解説する。特に、違反している設計に発生する問題として、クライアントが具体的なサブライアに依存することによってサブライアの変更がクライアントまでもが影響を受け、修正を余儀なくされることをあげ、クライアントとサブライアの依存関係に抽象を挟むことでクライアントの再利用が可能になっていることを解説しておくこと。また、この際、上位レイヤ(抽象)と下位レイヤの依存関係が逆になっていることを取り上げ、原則の名前の意味も解説する。</p> <p>②DIPをより深く理解するために、クライアントが依存すべき抽象について「上位レベルの方針」という面から解説する。上位レベルの方針とは、クライアントが要求しているインターフェイスであり、それに下位レベルのレイヤが従う形になることが依存関係の逆転に繋がり、クライアントが求めている抽象的な要求をクライアントとサブライアの間で挟むことが必要であることを解説する。また、定義する抽象の名前の留意点として、クライアントやサブライアに依存してしまうような名前(Client ServerやAbstract Suplier)にするのではなく、様々なクライアントの抽象的な要求を表す名前(生成を要求するのであれば、Creatable Deviceなど)にし、別パッケージで管理することで抽象を再利用できることを解説しておく。</p> <p>③DIPについて、設計実習を行い理解を確認する。①で提示したような原則に違反している設計を複数種類、UMLとソースコードで提示した後に、一定の時間を与え、原則に準拠した設計になるように修正させる。その後、修正した例を提示して解説をしていく。この際、原則に違反している設計の</p>

218

219

## 220 資料 11 : 5 学期制のスケジュール

221

222 ●概要：本カリキュラムでは1年5学期制を前提としている。一般的な大学・専門学校の前・後期  
 223 制はもちろん、中学・高等学校の3学期制よりも粒度の小さな学期構成となっている。5学  
 224 期制では、どの科目も、履修判定試験による実力チェックを含め7週間程度で終了する。こ  
 225 の場合、長期にわたる科目運営による授業脱落者の発生率を抑制できる、科目の粒度が小さ  
 226 いため学生にとって科目全体の学習内容を把握しやすい、短いスパンで設置される履修判定  
 227 試験の受験機会により、教員および学生の授業に対する緊張感を維持しやすいといったメ  
 228 リットがある。5学期制の具体的なスケジュール例は、以下の通りである。

229

230

学期	スケジュール例
1	4月1日から5月下旬の5週間
2	6月上旬から7月中旬の7週間
3	8月下旬から10月中旬の7週間
4	10月下旬から12月中旬の7週間
5	1月中旬から3月31日の7週間

231

232

233 ※1 学期は、入学式や新入生向けのオリエンテーションなどの行事が多数あることを想定し、他  
 234 学期よりも短めに設定している。

235

236 資料12:カリキュラムの進行

237 ●概要:1・2年次はオブジェクト指向実装、3・4年次はオブジェクト指向設計、4年次にオブジェ  
 238 クト指向分析・開発プロセスを、それぞれ主題とする。WEBプログラミング技術に関して  
 239 は4年間を通じて扱われるが、主要部分は3年次に教育する。

240

年次	分野	履修時間	分野別 時間 比率	年次別 合計履修 時間	到達点
1	一般教養	75	8.1%	930	<ul style="list-style-type: none"> <li>・2進数、16進数による演算、論理演算、配列を含むサンプルプログラムを作成することができる。</li> <li>・一定の機能を、Javaの文法に基づいて、独立したモジュールとして表現することができる。</li> <li>・リレーショナルデータベースのデータ構造の修正、集合論理に基づいた特定データの抽出、データを加工するための操作を行うことができる。</li> <li>・静的なホームページを作成し、WEBサーバを通して公開することができる。</li> </ul>
	IT基礎知識	225	24.2%		
	オブジェクト指向実装	315	33.9%		
	オブジェクト指向設計	-	-		
	オブジェクト指向分析・開発プロセス	-	-		
	データベース	180	19.4%		
	WEBプログラミング	75	8.1%		
資格対策	60	6.5%			
2	一般教養	75	8.1%	930	<ul style="list-style-type: none"> <li>・一定の論理モデルを、Javaによって表現することができる。</li> <li>・Javaの実装モデルに基づいて、再利用性・保守性に優れたモジュールを作成することができる。</li> <li>・リレーショナルデータベースの論理構造を調整し、アプリケーションから適切に利用することができる。</li> <li>・JavaScriptやFlashを用いた動的なホームページを作成することができる。</li> <li>・「Oracle Master 10g Bronze」を取得することができる。</li> <li>・「SJC-P」を取得することができる。</li> </ul>
	IT基礎知識	-	-		
	オブジェクト指向実装	450	48.4%		
	オブジェクト指向設計	-	-		
	オブジェクト指向分析・開発プロセス	-	-		
	データベース	225	24.2%		
	WEBプログラミング	60	6.5%		
資格対策	120	12.9%			
3	一般教養	75	8.1%	915	<ul style="list-style-type: none"> <li>・様々なソフトウェアパターンを適用することにより、アプリケーションを、最適化されたモジュール集合として設計・実装することができる。</li> <li>・XMLデータの論理構造を調整し、アプリケーションから適切に利用することができる。</li> <li>・開発プロセス全体のフローを理解している。</li> <li>・データベースを利用したWEBアプリケーションを設計・実装することができる。</li> </ul>
	IT基礎知識	-	-		
	オブジェクト指向実装	135	14.5%		
	オブジェクト指向設計	255	27.4%		
	オブジェクト指向分析・開発プロセス	75	8.1%		
	データベース	135	14.5%		
	WEBプログラミング	210	22.6%		
資格対策	30	3.2%			
4	一般教養	75	8.1%	795	<ul style="list-style-type: none"> <li>・WEBアプリケーションの開発にフレームワークを導入することにより、開発効率を促進することができる。</li> <li>・様々なソフトウェアパターンを適用することにより、WEBアプリケーションの全体に対して、最適な設計を行うことができる。</li> <li>・WEBアプリケーションの開発について、オブジェクト指向の方法論に即して、分析から実装まで一貫したプロセスを実現することができる。</li> <li>・高可用性、高パフォーマンスを実現するために、WEBアプリケーションで利用するリレーショナルデータベースを最適化することができる。</li> <li>・「XMLマスターベーシック」を取得することができる。</li> </ul>
	IT基礎知識	-	-		
	オブジェクト指向実装	15	1.6%		
	オブジェクト指向設計	255	27.4%		
	オブジェクト指向分析・開発プロセス	195	21.0%		
	データベース	165	17.7%		
	WEBプログラミング	30	3.2%		
資格対策	60	6.5%			

総履修時間 3570

241  
242

243 資料 13 : 反復的学習

学期	オブジェクト指向実装	オブジェクト指向設計	オブジェクト指向分析・開発プロセス	データベース	WEBプログラミング	
1年次	概要：ソースコードの保守性の向上というテーマが、実装パターン（第一段階）、オブジェクト指向設計の原則とレイヤリング（第二段階）、WEBアプリケーションの設計（第三段階）へと規模を拡大しながら反復的に学習される。他の分野も同様に同一テーマを多段階的に扱う。			Oracle SQL入門	HTML入門	
				Oracle SQL応用	HTML応用	
				Oracleオブジェクト入門	HTTP概論	
				Oracleオブジェクト応用		
				データベースの概念・論理設計		
2年次	1 実装パターン1	← 第一段階		Oracle DBA1 (アーキテクチャ)		
	2 実装パターン2		Oracle DBA2 (データベースの論理構造)			
	3		Oracle DBA3 (オブジェクトの構造)	XML概論	WEBアプリケーション概論	
	4		Oracle DBA4 (ユーザー管理・権限)		J2EE概論	クライアントサイドプログラミング入門
	5				J2EE1 (リクエスト・レスポンス)	クライアントサイドプログラミング応用
3年次		UML概論	開発プロセス		J2EE2 (JSPと式言語)	
		オブジェクト指向設計の原則	ドメイン・要求分析入門		J2EE3 (MVCモデル2)	
	3	レイヤリングの理論と方法	デザインパターン1	ユースケース分析(システム分析入門)	XML1 (スキーマ定義)	J2EE4 (フィルタ・イベント処理)
	4	例外設計	デザインパターン2	設計・実装・テスト・保守入門	XML2 (XPath・XSLT)	J2EE5 (タグライブラリとカスタムタグ)
	5	WEBアプリケーションの設計		ソフトウェア品質概論	XML3 (DOM・SAX)	
	1	J2EEパターン	ドメイン分析応用	データベースの物理設計		
	2	Strutsに基づく設計と実装	エンタープライズアプリケーションアーキテクチャパターン	要求分析応用	Oracle DBA6 (パフォーマンスチューニング)	
		JSFに基づく設計と実装	O/Rマッピング	ユースケース分析(システム分析応用)	Oracle DBA7 (RAC)	
	4	Springと依存性注入		アーキテクチャ設計・J2EE実装	Oracle DBA8 (ストアドプログラム・パッケージ)	WEBサービス
	5	ポストオブジェクト指向プログラミング		レビュー、テスト、デバッグ	Oracle DBA5 (ネットワーク)	

## 245 資料 14：コマシラバスの構成

246

247

●概要：以下、コマシラバスを構成するフォーマット。

248

項目	記述内容
シラバス	その科目で扱う技術に関して、その技術の背景、科目の内容、科目の目標について説明する。
授業形態	「講義 80%、実習 20%」のように、その科目のコマ授業全体に対し、講義と実習の比率を示す。従来、講義一辺倒となりがちな科目であっても、できる限り実習を盛り込むように作成している。これは、実装、実装パターン、設計、分析を分離しないで学習するための工夫でもある。
シラバスとの関係	シラバスに記述された科目の目的や趣旨のなかで、そのコマがどのような位置づけとなっているかを示す。
コマ主題	そのコマの主題を示す。
コマ主題細目	そのコマの主題が、どのようなサブテーマに分節されるかを示す。必ず三つ以上存在する。
コマ主題細目深度	サブテーマごとに、どのような内容をどのような方法で授業するかについて詳細な指示を記述する。コマシラバスの〈再現性〉のための最も重要な情報がここに記述される。
次コマとの関係	次のコマに対して、そのコマではどのような事後条件を満たさなければならないかを記述する。
教材・教具	そのコマの内容に関する主要な参考文献を挙げる。
まとめ	履修判定試験の前に、それまでの授業内容をまとめて確認するためのコマの内容を指示する。科目によって、「まとめ」コマは、ひとつあるいはふたつ存在する。
履修判定試験	学期末に実施される履修判定試験における出題内容を指示する。

249

250 資料 15 : コマシラバスのサンプル

251  
252  
253  
254  
255  
256  
257

●概要 : コマシラバスは、科目の担当教員に対する授業内容に関する指示である。何を教えるか (授業対象の明確化)、どこまで教えるか (授業レベルの明確化)、どのように教えるか (授業方法のノウハウの提供) を示すものである。これにより、カリキュラム責任者が、授業 (常勤・非常勤教員による) を制御する。以下、オブジェクト指向設計分野 3 年次 3 学期 「レイヤリングの理論と方法」 の 5 コマ目のコマシラバス。

レイヤーの粒度と分割3	5_1	シラバスとの関係	エンタープライズアプリケーションの開発に適したレイヤー分割の設計・実装方法について解説する。
	5_2	コマ主題	プレゼンテーションレイヤー、ビジネスロジックレイヤー、インテグレーションレイヤーに分割した場合の、それぞれのレイヤーの設計・実装の仕方について解説する。
	5_3	コマ主題細目	①プレゼンテーションレイヤーの設計と実装 ②ビジネスロジックレイヤーの設計と実装 ③インテグレーションレイヤーの設計と実装 ④サンプルコードの提示
	5_4	コマ主題細目深度	①簡単なコマンドラインアプリケーション(スケジュール管理プログラムなど)を例にとり、プレゼンテーションレイヤーの設計と実装について解説する。プレゼンテーションレイヤーのクラスはユーザーからの入力(たとえば、新しい予定など)を受け取るメソッドやユーザーに処理結果(たとえば、予定の一覧など)を出力するメソッドを持っていること、内部でビジネスロジックレイヤーに対する関連もしくは依存の関係を持っていることを解説する。スケジュール管理プログラムなどの簡単なプログラムを例にとり、クラス図とソースコードを挙げながら解説すること。 ②ビジネスロジックレイヤーの設計と実装について解説する。ここでも、スケジュール管理プログラムなどの簡単なプログラムを例にとり、クラス図とソースコードを挙げながら解説すること。ビジネスロジックレイヤーのクラス(たとえば、予定クラスなど)は、プレゼンテーションレイヤーから入力内容(たとえば、新しい予定に関するデータなど)を受け取り妥当性(たとえば、日付データとして妥当かどうかなど)をチェックするメソッドやプレゼンテーションレイヤーに業務処理の結果(たとえば、検索された予定の情報など)を返すメソッドを持っていること、インテグレーションレイヤーのクラスに対する関連もしくは依存の関係を持ち、インテグレーションレイヤーの永続化処理(たとえば、新しい予定の記録や不要な予定の削除など)に必要なデータを渡すことができるようになっていることを解説する。また、プレゼンテーションレイヤーが複数のデータの表示のために(たとえば、すべての予定の表示など)、ビジネスロジックレイヤーのクラス(たとえば、予定クラスなど)のコレクションや配列を必要とする場合があり、ビジネスロジックレイヤーはこれを提供するのためのメソッドが要求されることも解説しておく。 ③インテグレーションレイヤーの設計と実装について解説する。インテグレーションレイヤーはビジネスロジックレイヤーからデータを受け取り、JDBC APIなどの低レベルなAPIを利用してリソースレイヤーにアクセスしデータの書き込み処理を行うこと、逆リソースレイヤーから取得したデータをコレクションや配列へと変換し、ビジネスロジックレイヤーに返すといった実装をもつことを解説する。ここでもクラス図とソースコードを挙げながら解説する。 ④ここではまだ疎結合については扱っていないため、疎結合を無視したサンプルコードとシーケンス図を提示し、それぞれのレイヤーが協調して動作することを示す。
	5_5	次コマとの関係	次コマでは、アプリケーションを3つのレイヤーに分割する実習を行うため、ここでは、3つのレイヤーの債務について解説しておく。

258  
259  
260  
261  
262  
263  
264

●ポイント  
①何を教えるかだけでは、カリキュラムの責任者と科目担当教員間のコンセンサスは十分ではない。必ず、どこまで教えるのかを明確にする必要がある。また、それにより、「何を」に関する不要な誤解を避けることもできる。また、「どのように」というノウハウも担当教員にとって重要な情報。