【１】プログラム言語の分類　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

□機械語

：機械が唯一理解できるハード固有の言語である。

　０と１の記述の為プログラミングが困難である。

□低水準言語

□アセンブラ言語

：機械語と１：１に対応させた記号で記述する。

　実行するとき機械語に変換する必要がある。

□手続き型言語

：COBOL、Cなど

□高水準言語

□非手続き型言語

：C++、Javaなど

□第４世代言語

：プログラム開発の経験や知識のない人でも、簡単にプログラムを作成できる言語である。４ＧＬともいう。

□スクリプト言語

：JavaScript、Python、Ruby、など

□マークアップ言語

：SGML、HTML、XML、CSSなど

１．Javaについて

　オブジェクト指向型プログラム言語であるJavaは、プログラムが扱う対象をオブジェクト（データとそれに対する処理をまとめたもの）としてとらえ、オブジェクトの組み合わせでプログラムを作成する。

|  |  |
| --- | --- |
| Java言語の頻出用語 | |
| [①　　　　　　　　　　　　］ | Javaで書かれたプログラム  独立した単体のプログラムとして実行 |
| ［②　　　　　　　　　（VM）］ | Javaのインタプリタ |
| [③　　　　　　　　　　　　] | サーバからダウンロードされ、ブラウザ側で実行するJavaプ  ログラム |
| [④　　　　　　　　　　　　] | サーバ上で実行されるJavaプログラム |
| [⑤　　　　　　　　　　　　] | Javaのプログラムにおいて、よく使う機能などを部品化し、  再利用できるようにコンポーネント化（クラス化）する仕様 |
| Eclipse | Javaでの開発によく利用されるツール。コンパイラ、エディタ、デバッグなどの機能が組み込まれている |

２．マークアップ言語について

　文書中の指定部分の前後にタグという記号を挿入し、データの種類やレイアウト情報などの属性を記

述するための言語。

このマークアップ言語において、文書の要素や定義した内容、文書構成のルールを記述した部分を、

DTD(Document Type Definition)と呼ぶ。

３．Ajax(Asynchronous javaScript + XML：エイジャックス、アジャックス)について

　JavaScriptとXMLを利用し、Webページの再読み込みや別のページへの遷移を行うことなくWebサ

ーバとやり取りを行い、表示内容を変更するプログラミング手法。GoogleMapなどで使われている。

≪範例≫

プログラム言語に関する記述のうち、Javaの説明として適切なものはどれか。

ア　1970年代に開発されたインタプリタ型のオブジェクト指向言語であり、エディタやデバッガな

どの統合開発環境やOSの機能などを含む。

イ　Cにクラスやインヘリタンスといったオブジェクト指向の概念を取り入れたものであり、Cとの

上位互換性をもつ。

ウ　Webで用いられているマーク付け言語であり、タグによって文書の構造を記述する。テキストや

動画などを関連付けたハイパテキストを作成できる。

エ　ブラウザで動作するアプレットなどを作成できる。このアプレットは、仮想マシンを実装した環

境上であれば、どこでも実行できる。

≪解答≫　エ

　ア　Smalltalkに関する記述です。

　イ　C++に関する記述です。

　ウ　HTMLに関する記述です。

【２】言語プロセッサ　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

　アセンブラ言語や高水準言語で記述したプログラムは、そのままコンピュータで実行することはできず、機械語に変換する必要がある。これを行うのが言語プロセッサである。

原始プログラム

目的プログラム

**言語プロセッサ**

１．言語プロセッサの種類

|  |  |
| --- | --- |
| 種類 | 変換の特徴 |
| [①　　　　　　　　　] | アセンブラ言語を目的プログラム（機械語）に翻訳する。 |
| [②　　　　　　　　　] | 高水準言語を一括して目的プログラム（機械語）に翻訳する。 |
| [③　　　　　　　　　] | 高水準言語を1行ずつ翻訳し実行する。 |
| [④　　　　　　　　　] | パラメタを指定することにより自動的に目的プログラムを生成する。 |
| プリプロセッサ | 擬似命令言語を含んだ原始プログラムを翻訳可能な原始プログラムに変換する。プリコンパイラともいう。 |
| クロスアセンブラ  クロスコンパイラ | 原始プログラムを他の機種用の目的プログラム（機械語）に変換する。アセンブラ用とコンパイラ用に分類される。 |
| トランスレータ | 原始プログラムを他の機種用の原始プログラムに変換する。 |
| シミュレータ  エミュレータ | 目的プログラムを他の機種用の機械語に変換しながら実行する。  前者はソフトウェアで､後者はハードウェアで変換する。 |

擬似命令言語

目的プログラム

原始プログラム

実行

目的プログラム

原始プログラム

実行

コンパイラ

アセンブラ

クロスコンパイラ

クロスアセンブラ

シミュレータ

エミュレータ

トランスレータ

プリプロセッサ

〔Ｂ社システム〕

〔Ａ社システム〕

　≪範例≫

各種言語プロセッサの説明のうち、適切なものはどれか。

ア　アセンブラは、ある処理系用に書かれた原始プログラムを、ほかの処理系用の原始プログラムに変換する。

イ　インタプリタは、ほかのコンピュータ用のプログラムを解読し、実行するマイクロプログラムである。

ウ　ジェネレータは、入力・処理・出力などの必要な条件をパラメタで指示することによって、処理目的に応じたプログラムを生成する。

エ　トランスレータは、高水準言語で書かれたプログラムを、解釈しながら実行する。

　≪解答≫　ウ

　ア　トランスレータに関する記述です。

　イ　エミュレータに関する記述です。

　エ　インタプリタに関する記述です。

【３】プログラムの実行手順　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

目的プログラムは、リンカ（リンケージエディタ、連係編集プログラム）によってライブラリのプログラムが複数組み合わされて実行可能プログラム（ロードモジュール）に変換され、補助記憶装置に格納されます。その後、ローダによって取り出されて、主記憶装置の指定された番地に格納され、実行されます。

[①　　　　　　　]

プログラム

ライブラリ

原始プログラム

目的プログラム

実行可能プログラム

実行

[②　　　　　　　]

[③　　　　　　　]

図　プログラムの実行手順

≪範例≫

リンカの機能として、適切なものはどれか。

ア　作成したプログラムをライブラリに登録する。

イ　実行に先立ってロードモジュールを主記憶にロードする。

ウ　相互参照の解決などを行い、複数の目的モジュールなどから一つのロードモジュールを生成する。

エ　プログラムの実行を監視し、ステップごとに実行結果を記録する。

　≪解答≫　ウ

　ア　コンパイラの機能です。

　イ　ローダの機能です。

　エ　デバッガの機能です。

【４】コンパイラの翻訳過程　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

[①　　　　　　　　　]

[②　　　　　　　　　]

[③　　　　　　　　　]

[④　　　　　　　　　]

[⑤ 　　　　　　　　]

：原始プログラムの文字列を言語として意味のもつ最小単位（トークン）に分解する。

：各言語の構文に従って分解された原始プログラムが､文法的に正しいかを検査する。（シンタックスエラー）

：各言語の原始プログラムについて､意味的に誤りがないかどうかを検査する。

：不要な処理をオプティマイザ（最適化プログラム）で取り除き､実行効率のよいプログラムを生成する。

：機械語レベルのコードを生成する。

≪範例≫

手続型言語のコンパイラが行う処理のうち、最初に行う処理はどれか。

ア　意味解析　　　イ　構文解析　　　ウ　最適化　　　エ　字句解析

　≪解答≫　エ

【５】オープンソースソフトウェア（OSS）　　　　　　　　　　　　　　　　　…テキストP.（　　　）

オープンソースソフトウェアは、ソースコードが公開されているソフトウェア群のことで、誰でも改良や再配布が可能です。代表的なオープンソースソフトウェアに、Linux（OS）、BIND（DNSサーバ）、Postfix（メールサーバ）、Apache（Webサーバ）、PostgreSQL（DBMS）、Eclipse（統合開発環境）、Thunderbird（メールソフト）、Firefox（ブラウザ）などがあります。

また、Linuxなどのように、改変したものを販売している業者（ディストリビュータ）もあります。ただし、改良や再配布に際しては原著者の著作権表示を義務付けています。

OSSの概念をまとめたものに、OSI（Open Source Initiative）が公表したOSD（The Open Source Definition）があり、次のような基準があります。

①自由な再配布ができる。

②ソースコードを入手できる。

③派生物が存在でき、派生物に同じライセンスを適用できる。

④差分情報の配布を認める場合には、同一性の保持を要求してもかまわない。

⑤個人やグループを差別しない。

⑥適用領域に基づいた差別をしない。

⑦再配布において追加ライセンスを必要としない。

⑧特定製品に依存しない。

⑨同じ媒体で配布される他のソフトウェアを制限しない。

⑩技術的な中立を保っている。

OSSに関するライセンスとしては、FSF（Free Software Foundation）が提唱したGPL（General Public License）や、カリフォルニア大学バークレー校（University of California, Berkeley, UCB）が開発・配布したBSD（Berkeley Software Distribution）ライセンスがあります。