日本工業規格　　JIS X 3014：2003

プログラム言語 C++

正　誤　票

区分	位　置	誤	正
本体	**2.2** 第 1 段落 6 行目	_ { } [ ] # ( ) < > % : ; . ? * + - / ^ & | ~ = , \ " ’	_ { } [ ] # ( ) < > % : ; . ? * + - / ^ & | ~ ! = , \ " ’
	2.10 第 2 段落 1 行目	(**17.4.3** 及び **1.2** 参照)	(**17.4.3.1.2** 参照)
	3.4 第 3 段落 1 行目	補正クラス名も,	補整クラス名も,
	3.4.2 第 2 段落 6 行目	関連クラスは，そのクラスそのもの 並びにその直接及び間接の基底クラスとする。	関連クラスは，そのクラスそのもの，それがメンバになっているクラス，並びに その直接及び間接の基底クラスとする。
	3.4.3.2 第 2 段落 4 行目	<<using 宣言>>を無視する。	<<using 指令>>を無視する。
	3.4.3.2 第 5 段落 4 行目	同一の名前空間にある場合だけとする。	同一の名前空間にある場合だけとする。そうでない場合 (宣言が他の名前空間に由来する場合)，そのプログラムは不適格とする。
	3.5 第 6 段落 1 行目	大域的有効範囲で宣言してある関数の名前，及び大域的有効範囲の extern 宣言で宣言してあるオブジェクトの名前は，結合をもつ。	ブロック有効範囲で宣言してある関数の名前, 及びブロック有効範囲の extern 宣言で宣言してあるオブジェクトの名前は，結合をもつ。
	3.5 第 6 段落 3 行目	その大域的有効範囲での宣言は,	そのブロック有効範囲での宣言は,
	3.5 第 6 段落 5 行目	その大域的有効範囲の実体は,	そのブロック有効範囲の実体は,
	3.6.3 第 1 段落 1 行目	(大域的有効範囲又は名前空間有効範囲	(ブロック有効範囲又は名前空間有効範囲
	3.9 第 7 段落 6 行目	(“境界値不明の配列 *T*”	(“要素数未知の配列 *T*”
	3.9 第 7 段落 **例** 14 行目	ここで arr 型は,	ここで arr の型は,

区分	位　置	誤	正
本体	**5.** 第 4 段落 2 行目	相続く二つの副作用完了点の間では，一つの式の評価によってスカラオブジェクトの格納値が変わるのは，高々1 回でなくてはならない。	相続く二つの副作用完了点の間では，一つの式の評価によって一つのスカラオブジェクトの格納値が変わるのは，高々1 回でなければならない。
	5.16 第 2 段落 4 行目	例外送出（**15.1**）となる。結果は，他方の型の右辺値となる。	送出式（**15.1**）の場合，結果は，他方の型の右辺値となる。
	6.6.3 第 2 段落 6 行目	未定とする。	未定とする。
	8.3.4 第 1 段落 12 行目	“上限不明の配列 *T*”	“要素数未知の配列 *T*”
	8.3.5 第 6 段落 1 行目	“*T* の上限不明の配列へのポインタ”…“*T* の上限不明の配列への参照”	“*T* の要素数未知の配列へのポインタ”…“*T* の要素数未知の配列への参照”
	注([87]) 1 行目	ただし，…“*T* の上限不明の配列へのポインタ”	ただし，…“*T* の要素数未知の配列へのポインタ”
	8.5 第 14 段落 21 行目	**3)**　そうでない場合…その関数がコンストラクタの場合，元の型の一時変数を初期化する。	**3)**　そうでない場合…その関数がコンストラクタの場合，目的の型の一時変数を初期化する。
	8.5.1 第 1 段落 1 行目	次のいずれをももたない配列又はクラス（**9.**）を，集成体と呼ぶ。	配列，又は次のいずれをももたないクラス（**9.**）を，集成体と呼ぶ。
	8.5.1 第 4 段落 6 行目	大きさが不明な配列	要素数未知の配列
	10.3 第 5 段落 **例** 1 行目	class B {}	class B {};
	11.4 第 2 段落 **例** 5 行目	A::B は，X の基底箇条	A::B は，X の基底節
	13.3.1.2 第 3 段落 7 行目	－　非メンバ候補の…修飾なし検索をした結果とする。	－　非メンバ候補の…修飾なし検索をした結果とする。ただし，クラス型の演算対象がない場合には，名前検索集合の中のメンバ関数のうち，第 1 仮引数として型 *T*1 若しくは *T*1 が列挙型のときには（cv 修飾されているかもしれない）*T*1 への参照をもつメンバ関数，又は（右演算対象がある場合であるが）第 2 仮引数として型 *T*2 又は *T*2 が列挙型のときには（cv 修飾されているかもしれない）*T*2 への参照をもつメンバ関数だけが，候補関数となる。
	13.3.3.2 第 4 段落 7 行目	－　クラス B がクラス C の直接又は間接の派生クラスであり	－　クラス B がクラス A の直接又は間接の派生クラスであり

区分	位　置	誤	正
本体	**13.6** 第 2 段落 4 行目 **参考**	列挙型の演算対象は，汎整数昇格によってアクセス可能となる。	汎整数昇格によって，列挙型の演算対象を用いることができるようになる。
	14.6 第 8 段落 **例** 16 行目	上の例で，i は，printcall で宣言された局所変数 i であり，	上の例で，i は，printall で宣言された局所変数 i であり，
	14.6.1 第 1 段落 3 行目	その補正クラス名は，	その補整クラス名は，
	14.6.1 第 1 段落 4 行目	その補正クラス名は，	その補整クラス名は，
	14.6.1 第 2b 段落 3 行目	補正クラス名は，	補整クラス名は，
	16.3 **注**([141]) 1 行目	識別子含みうる文字の列ではなくなっている（**2.1.1.2** の翻訳段階参照。）から，	識別子を含みうる文字の列ではなくなっているから，
	17.3.2.1.3.1 第 2 段落 1 行目	終端ヌル文字より前	終端ナル文字より前
	17.3.2.1.3.1 第 3 段落 1 行目	終端ヌル文字まで（終端ヌル文字も含む。）の要素	終端ナル文字まで（終端ナル文字も含む。）の要素
	20.3.5 第 1 段落 1 行目	否定子（nagator）	否定子（negator）
	21.3.5.6 第 14 段落 1 行目 **参考**	文字列の長さは，	文字列の長さの差分は，
	21.3.5.6 第 16 段落 1 行目 **参考**	文字列の長さは，	文字列の長さの差分は，
	21.3.5.6 第 18 段落 1 行目 **参考**	文字列の長さは，	文字列の長さの差分は，
	21.3.5.6 第 20 段落 1 行目 **参考**	文字列の長さは，	文字列の長さの差分は，

区分	位　置	誤	正
本体	**21.3.5.6** 第 22 段落 1 行目 **参考**	文字列の長さは,	文字列の長さの差分は,
	21.3.6.2 第 1 段落 2 行目	－　pos <= xpos	－　xpos <= pos
	22.2.2.2.2 第 2 段落 2 行目	－　段階 1　pritf	－　段階 1　printf
	23.2.1 第 2 段落 58 行目	// 23.2.1.3 修飾子	// 23.2.1.3 変更子
	23.2.2 **注**(247) 1 行目	メンバ関数は,	これらのメンバ関数は,
	23.2.2 第 2 段落 55 行目	// 23.2.2.3 修飾子	// 23.2.2.3 変更子
	23.2.4 第 2 段落 61 行目	// 23.2.4.3 修飾子	// 23.2.4.3 変更子
	23.2.5 第 1 段落 68 行目	// 修飾子	// 変更子
	23.3.1 第 2 段落 64 行目	// 修飾子	// 変更子
	23.3.2 第 2 段落 63 行目	// 修飾子	// 変更子
	23.3.3 第 2 段落 50 行目	// 修飾子	// 変更子
	23.3.4 第 2 段落 52 行目	// 修飾子	// 変更子
	27.6.2.6 第 1 段落 2 行目	書式付き出力関数は,	書式なし出力関数は,
	27.8.1.4 細分箇条 項目名	**多重定義された仮想関数**	**上書きされた仮想関数**
附属書 B	**附属書 B** 第 2 段落 22 行目	－　一つのメンバ，構造体又は共用体内のデータメンバの個数（16 384）	－　一つのクラス, 構造体又は共用体内のデータメンバの個数（16 384）

平成 22 年 2 月 1 日作成