

問2 ソフトウェア製品の品質特性に関する次の記述を読んで、設問に答えよ。

JIS X 0129-1では、ソフトウェア製品の品質について、表1に示す六つの品質特性を定めている。

表1 六つの品質特性（JIS X 0129-1）

品質特性	ソフトウェア製品の能力の概要	品質副特性（一部）
機能性	指定された条件下で利用されるとき、明示的及び暗示的必要性に合致する機能を提供する。	合目的性、正確性、セキュリティ、相互運用性
使用性	指定された条件下で利用するとき、理解、習得、利用でき、利用者にとって魅力的である。	運用性、習得性、魅力性、理解性
信頼性	指定された条件下で利用するとき、指定された達成水準を維持する。	回復性、障害許容性、成熟性
効率性	明示的な条件下で、使用する資源の量に対比して適切な性能を提供する。	時間効率性、資源効率性
保守性	修正のしやすさ	安定性、解析性、試験性、変更性
移植性	ある環境から他の環境に移すことができる。	環境適応性、共存性、設置性、置換性

これらの品質特性のうち、コーディングの段階では、信頼性、効率性、保守性、移植性を考慮することが大切である。

あるソフトウェア開発会社では、開発するソフトウェア製品の品質向上を図るため、品質特性を考慮したプログラム開発の社内標準を制定し、作成したプログラムのコードレビュー体制を確立した。

表2は、最近のコードレビューで新人のプログラム開発担当者が受けた指摘の例である。

表2 新人のプログラム開発担当者が受けた指摘の例

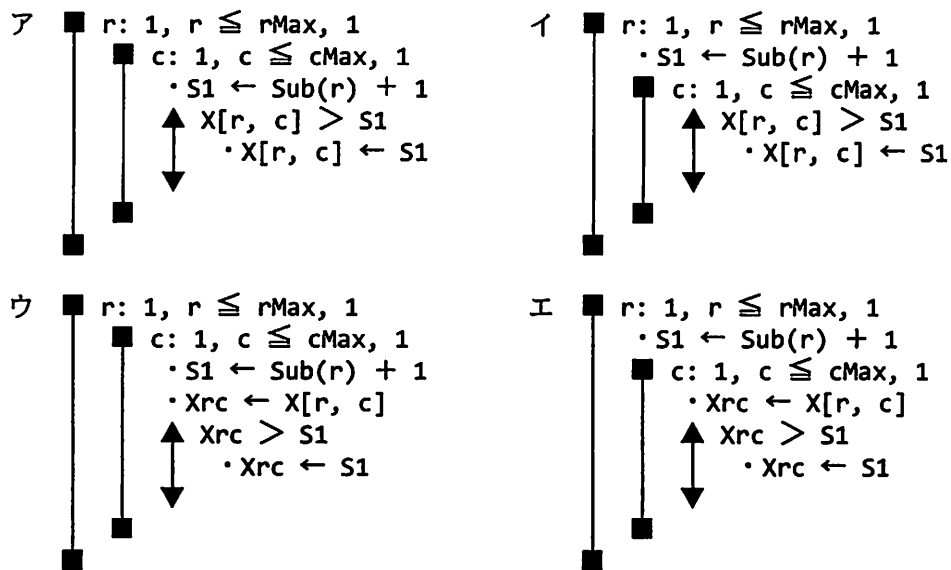
ソースコード	指摘の内容	主な品質特性 (品質副特性)
○実数型: Ave, Count, Total ⋮ ・Ave ← Total ÷ Count ⋮	左の処理を次のように変更すること。 ⋮ ↑ Count > 0 ・Ave ← Total ÷ Count ↓ ・Ave ← 0 ⋮	<div>a</div>
⋮ ■ r: 1, r ≤ rMax, 1 ■ c: 1, c ≤ cMax, 1 ▲ X[r, c] > Sub(r) + 1 ▼ ・X[r, c] ← Sub(r) + 1 ⋮	左の処理で、関数 Sub は計算時間は長い、 返却値は引数だけに依存する。 次のように最適化すること。 ⋮ <div>b</div> ⋮	効率性 (時間効率性)
⋮ /* 主記憶の動的取得 */ ・GetMain(Addr, Len) ⋮ /* 主記憶の動的開放 */ ・FreeMain(Addr) ⋮	主記憶の動的取得と開放で、システム標準 の関数を使用している。一般に、取得した 範囲外や開放済の記憶域を誤って更新する などの障害は、 <div>c</div> 。 次のように、デバッグ機能のある社内で開 発した同機能の関数を使用すること。 ⋮ ・X_GetMain(Addr, Len, …) ⋮ ・X_FreeMain(Addr, …) ⋮	保守性 (試験性)
○整数型: P1, P2, Ans ○整数型関数: Fn(P1, P2) ⋮ ・Ans ← Fn(P1, P2) ⋮	このプログラムは複数の機種で汎用的に使 われる。機種の違いによって <div>d</div> が異なることがあるので、次のように宣言 の記述形式を変更すること。 ○32ビット整数型: P1, P2, Ans ○32ビット整数型関数: Fn(P1, P2) ⋮	<div>e</div>

設問 表2中の に入れる正しい答えを、解答群の中から選べ。

a, eに関する解答群

- ア 移植性（環境適応性） イ 効率性（資源効率性） ウ 信頼性（成熟性）
エ 保守性（解析性） オ 保守性（変更性）

bに関する解答群



cに関する解答群

- ア 更新した時点で障害と分かるが、ログを記録する機能のある OS は少ない
イ 更新した時点で障害と分かるが、ログを記録する機能のあるハードウェアは少ない
ウ 更新内容を後で参照したときに障害となることが多く、原因箇所の特定が困難である
エ 取得可能な主記憶域が残っている間は、障害を検知できない

dに関する解答群

- ア 指定できる変数や関数の個数
イ 変数や関数の型宣言で省略した場合のビット数
ウ リンカで扱える関数のビット数
エ ロードで扱える関数の個数