問8　クーポン券発行システムの設計に関する次の記述を読んで，設問1～3に答えよ。

X社は，全国に約400店のファミリーレストランを展開している。X社の会員向けWebサイトでは，割引料金で商品を注文できるクーポン券を発行しており，会員数は1,000万人を超える。このたび，会員の利便性の向上や店舗での注文受付業務の効率向上のために，会員のスマートフォン宛てにクーポン券を発行することになった。

スマートフォン宛てにクーポン券を発行する新しいシステム（以下，新システムという）は，スマートフォン向けアプリケーションソフトウェア（以下，スマホアプリという）とサーバ側のWebアプリケーションソフトウェア（以下，Webアプリという）から構成され，Webアプリの開発は情報システム部門のY君が担当することになった。

〔新システムの利用イメージ〕

X社の会員は，事前に自分のスマートフォンにX社のスマホアプリをダウンロードし，インストールしておく。会員がクーポン券を利用する際は，スマホアプリに会員IDとパスワードを入力してログインする。ログインが完了すると，おすすめ商品と利用可能なクーポン券の一覧が表示される。会員が利用したいクーポン券を選択すると，QRコードを含むクーポン券画面が表示される。X社店舗の注文スタッフがQRコードを注文受付端末で読み取ると，割引料金での注文ができる。

〔Webアプリの処理方式の調査〕

Y君がWebアプリの実現方式を検討したところ，X社のWebサイトで利用しているブロッキングI/O型のWebサーバソフトウェア（以下，サーバソフトという）では，スマホアプリからの同時アクセス数が増えると対応できないことが分かった。

ブロッキングI/O型のサーバソフトでは，ネットワークアクセスやファイルアクセスなどのI/O処理を行う場合，CPUは低速なI/O処理の完了を待って次の処理を実行する。例えば，表1に示す，QRコードを作成するために必要なWebアプリの処理（以下，QRコード作成処理という）の場合，全体の処理時間の［　a　］%がI/O処理の完了待ち時間となる。

表1　QR コード作成処理

処理番号	処理内容	開始条件	処理時間（ミリ秒）	処理区分
1	会員のログイン状態を確認	なし	0.02	CPU 処理
2	スマホアプリからクーポン券番号を取得	処理 1 の完了	10	I/O 処理
3	クーポン券番号から QR コードの画像データをメモリに作成	処理 2 の完了	0.06	CPU 処理
4	QR コードの画像データを画像ファイルに書き出し	処理 3 の完了	3	I/O 処理
5	クーポン券番号を発行履歴としてデータベースに書き込み	処理 2 の完了	15	I/O 処理
6	スマホアプリに QR コードの画像ファイルを返信	処理 4 の完了	5	I/O 処理
7	QR コードの画像ファイルを削除	処理 6 の完了	2	I/O 処理

このため，ブロッキング I/O 型のサーバソフトでは，複数のスマホアプリにサービスを提供するために，プロセスやスレッドを複数生成している。しかし，プロセスやスレッドの数が増えると，プロセスやスレッドの切替え処理である［ b ］スイッチがボトルネックとなり，CPU やメモリを追加してもスマホアプリからの同時アクセスへの対応は困難となる。

そこで Y 君は，多数のスマホアプリからのアクセスを効率よく処理できるノンブロッキング I/O 型のサーバソフトの利用を検討した。ノンブロッキング I/O 型のサーバソフトでは，一つのプロセスやスレッドの中で，CPU は I/O 処理の完了を待たずに，実行可能なほかの処理を実行する。その結果，Web サーバは複数のプロセスやスレッドを生成する必要がなく，スマホアプリへも効率的にサービスを提供できる。

〔リアクタパターンの調査〕

ノンブロッキング I/O 型のサーバソフトで，Web アプリを動作させるためには，非同期処理の考え方に基づいたソフトウェア設計が必要である。そこで，Y 君は，ノンブロッキング I/O 型の処理を実現するデザインパターンの一つであるリアクタパターンについて調査した。図 1 に Y 君が調査したリアクタパターンの処理の流れを示す。

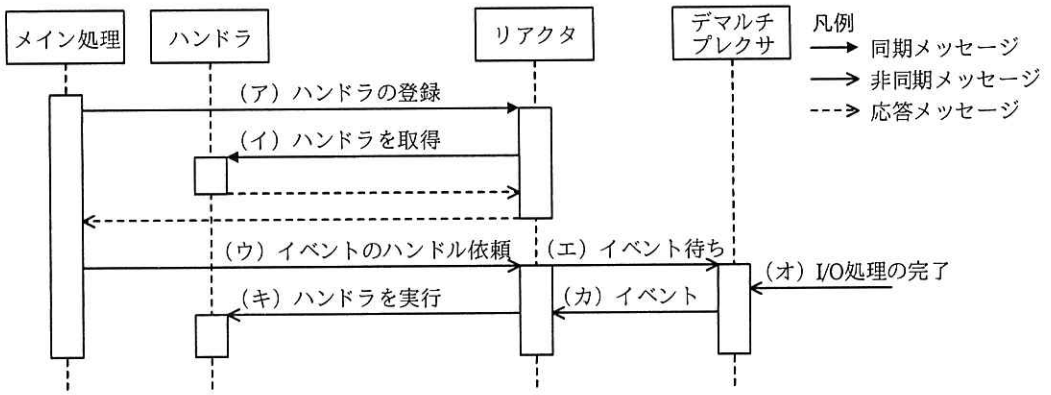


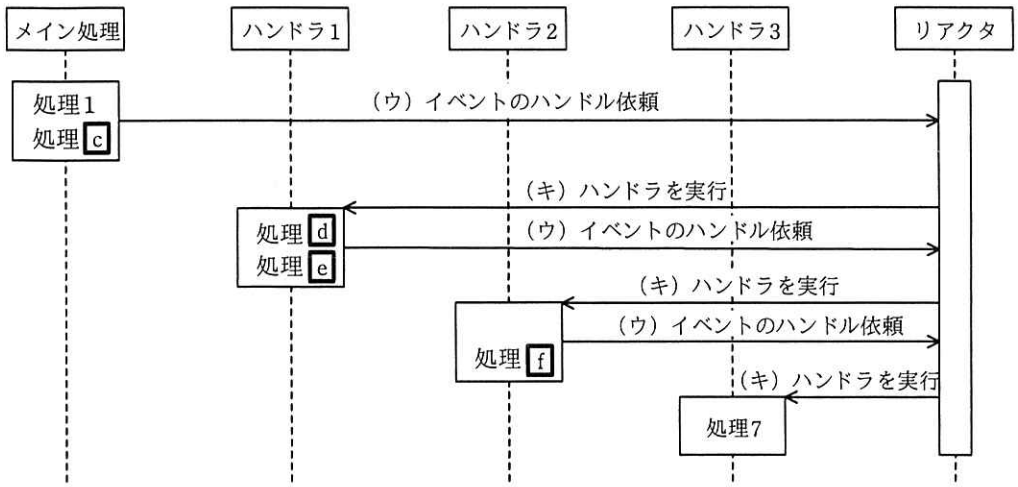
図1　Y君が調査したリアクタパターンの処理の流れ

I/O 処理の処理結果を利用する処理をハンドラとして定義する。次に，I/O 処理の完了待ちを依頼したいメイン処理が，①I/O 処理完了後に実行するハンドラ名を引数に“(ア) ハンドラの登録”を行うと，リアクタは“(イ) ハンドラを取得”する。次に，メイン処理がリアクタに“(ウ) イベントのハンドル依頼”を行うと，リアクタはデマルチプレクサに“(エ) イベント待ち”を指示する。デマルチプレクサは複数の I/O 処理の完了を一括して監視し，“(オ) I/O 処理の完了”を検知した場合には，対応する“(カ) イベント”をリアクタに発行する。イベントを受け取ったリアクタは“(キ) ハンドラを実行”する。メイン処理は，イベントのハンドル依頼を行った後は，I/O 処理の完了を待たずにほかの処理を実行できる。

リアクタパターンを適用する場合は，遅い I/O 処理の次に実行される処理をハンドラとして分割するのがよい。しかし，リアクタパターンに基づき設計されたプログラムは，②保守性が下がるおそれがある。

〔QR コード作成処理の設計〕

Y 君は，リアクタパターンを用いて，スマホアプリからのアクセスに対する応答時間が最小になるように，QR コード作成処理を設計した。図 2 に Y 君が設計した QR コード作成処理の流れを示す。



注記1　図1中の（ア），（イ）の同期メッセージは省略
注記2　図1中の（エ）～（カ）の非同期メッセージは省略
注記3　図1中の応答メッセージは省略

図2　Y君が設計したQRコード作成処理の流れ

その後Y君は，新システムのWebアプリの開発を完了させ，X社の会員はスマートフォンを通じてクーポン券を利用することが可能となった。

設問1　〔Webアプリの処理方式の調査〕について，(1)，(2)に答えよ。

(1)　本文中の［　a　］に入れる適切な数値を答えよ。答えは，小数第3位を四捨五入して，小数第2位まで求めよ。

(2)　本文中の［　b　］に入れる適切な字句を答えよ。

設問2　〔リアクタパターンの調査〕について，(1)，(2)に答えよ。

(1)　本文中の下線①について，関数呼出しの引数として渡される関数のことを何というか，解答群の中から選び，記号で答えよ。

解答群

ア　callback関数　　イ　static関数

ウ　template関数　　エ　virtual関数

(2)　本文中の下線②について，プログラムの保守性が下がる理由を15字以内で述べよ。

設問3　〔QRコード作成処理の設計〕について，(1)，(2)に答えよ。

(1)　図2中の［　c　］～［　f　］に入れる適切な処理番号を，表1中の

処理番号を用いて答えよ。ただし，複数ある場合は全て答えよ。

(2) 図 2 のように QR コード作成処理を設計した場合，処理 4～7 はどのような順序で完了するか，処理が早く完了する順にコンマ区切りで答えよ。