

## Задания с DOXYGEN:

1) В Doxygen команда ... (@brief) служит для вставки краткого описания

В Doxygen команда @brief служит для вставки краткого описания.

2) В Doxygen команда ... (@return) служит для вставки описания возвращенного значения функцией или методом

В Doxygen команда @return служит для вставки описания возвращаемого значения функцией или методом.

@brief	краткое описание (класса, метода и т. п.);
@details	полное описание (класса, метода и т. п.);
@param foo	описание параметра foo функции или метода;
@return	описание возвращаемого функцией (методом) значения;
@file	описание конкретного файла;
@throw Foo	метод возбуждает исключение Foo;
@warning	предупреждение;
@authors	указывает автора или авторов;
@version	используется для указания версии;
@date	предназначена для указания даты разработки;
@bug	перечисление известных ошибок;
@copyright	используемая лицензия;

## Задания с ЖЦ ПО

1) Укажите для каждого процесса ЖЦ ПО его тип:

<b>Вопрос 2</b> Выполнен Баллов: 0,60 из 1,00 Отметить вопрос	<p>Укажите для каждого процесса ЖЦ ПО его тип:</p> <table><tr><td>валидация</td><td>вспомогательный ▼</td></tr><tr><td>управление конфигурацией</td><td>организационный ▼</td></tr><tr><td>обеспечение качества</td><td>вспомогательный ▼</td></tr><tr><td>управление</td><td>основной ▼</td></tr><tr><td>разработка</td><td>основной ▼</td></tr></table>	валидация	вспомогательный ▼	управление конфигурацией	организационный ▼	обеспечение качества	вспомогательный ▼	управление	основной ▼	разработка	основной ▼
валидация	вспомогательный ▼										
управление конфигурацией	организационный ▼										
обеспечение качества	вспомогательный ▼										
управление	основной ▼										
разработка	основной ▼										

Укажите для каждого процесса ЖЦ ПО его тип:

документирование	вспомогательный ▼
разработка	основной ▼
аудит	вспомогательный ▼
приобретение	основной ▼
обучение	организационный ▼

Укажите для каждого процесса ЖЦ ПО его тип:

обеспечение качества	вспомогательный ▼
верификация	вспомогательный ▼
сопровождение	основной ▼
создание инфраструктуры	организационный ▼
приобретение	основной ▼

Укажите для каждого процесса ЖЦ ПО его

приобретение	основной ▼
разработка	основной ▼
документирование	вспомогательный ▼
обучение	организационный ▼
аудит	вспомогательный ▼

## Задания с GIT:

1) Клонирование репозитория осуществляется командой git ... (clone)

Клонирование репозитория осуществляется командой `git clone`

2) Запрос обновления локального репозитория из внешнего репозитория осуществляется командой git ... (update)

Запрос обновления локального репозитория из внешнего репозитория осуществляется командой `git update`

3) Для передачи изменений локального репозиторий используется команда git ... (push)

Для передачи изменений локального репозитория во внешний репозиторий используется команда `git push`

4) Добавление новых файлов в индекс репозитория осуществляется командой git ... (add)

Добавление новых файлов в индекс репозитория осуществляется командой `git add`

5) Запрос обновления локального репозитория из внешнего репозитория осуществляется командой git ... (pull)

Запрос обновления локального репозитория из внешнего репозитория осуществляется командой `git pull`

6) Сохранить изменения в репозитории необходимо командой git ... (commit)

Сохранить изменения в репозитории необходимо командой `git commit`

7) Инициализация нового репозитория осуществляется командой git ... (init)

Инициализация нового репозитория осуществляется командой `git init`

## Создать новый репозиторий

```
git init # создать новый проект в текущей директории
git init folder-name # создать новый проект в указанной директории
```

## Клонирование репозитория

```
# клонировать удаленный репозиторий в одноименную директорию
git clone https://github.com/cyberspacedk/Git-commands.git

# клонировать удаленный репозиторий в директорию «FolderName»
git clone https://github.com/cyberspacedk/Git-commands.git FolderName

# клонировать репозиторий в текущую директорию
git clone https://github.com:nicothin/web-design.git .
```

## Просмотр изменений

```
git status # показать состояние репозитория (отслеживаемые, измененные, не отслеживаемые файлы)
git diff # сравнить рабочую директорию и индекс (неотслеживаемые файлы)
git diff --color-words # сравнить рабочую директорию и индекс, показать отличия
git diff index.html # сравнить файл из рабочей директории и индекс
git diff HEAD # сравнить рабочую директорию и коммит, на который указываем
git diff --staged # сравнить индекс и коммит с HEAD
git diff master feature # посмотреть что сделано в ветке feature по сравнению с master
git diff --name-only master feature # посмотреть что сделано в ветке feature с момента создания ветки
git diff master...feature # посмотреть что сделано в ветке feature с момента создания ветки
```

## Добавление изменений в индекс

```
git add . # добавить в индекс все новые, изменённые, удалённые файлы из текущей директории
git add text.txt # добавить в индекс указанный файл (был изменён, был удалён или это новый файл)
git add -i # запустить интерактивную оболочку для добавления в индекс только выбранные файлы
git add -p # показать новые/изменённые файлы по очереди с указанием их изменений
```

## Удаление изменений из индекса

```
git reset          # убрать из индекса все добавленные в репозиторий файлы  
git reset readme.txt # убрать из индекса изменения указанные файла
```

## Отмена изменений

```
git checkout text.txt # ОПАСНО: отменить изменения в файле  
git reset --hard      # ОПАСНО: отменить изменения; вернуть индекс в состояние  
git clean -df         # удалить неотслеживаемые файлы
```

## Коммиты

```
git commit -m "Name of commit" # зафиксировать в коммите  
git commit -a -m "Name of commit" # проиндексировать отслеживаемые файлы
```

## Удаление файла

```
git rm text.txt      # удалить отслеживаемый файл  
git rm -f text.txt   # удалить отслеживаемый файл  
git rm -r log/        # удалить всё содержимое папки  
git rm ind*          # удалить все отслеживаемые файлы  
git rm --cached readme.txt # удалить из индекса файл
```

## Начало работы

Создание нового репозитория, первый коммит, приём изменений в удалённый репозиторий.

```
# указана последовательность действий:
# создана директория проекта, мы в ней
git init                                # создаём репозиторий
touch readme.md                        # создаем файл readme.md
git add readme.md                      # добавляем файл в индекс
git commit -m "Старт"                  # создаем коммит
git remote add origin https://github.com:nicothin/
git push -u origin master              # отправляем данные
```

## «Внесение изменений» в коммит

Только если коммит ещё не был отправлен в удалённый репозиторий.

```
# указана последовательность действий:
subl inc/header.html                  # редактируем и сохраняем
git add inc/header.html               # индексируем изменения
git commit -m "Убрал телефон из шапки" # делаем коммит
# ВНИМАНИЕ: коммит пока не был отправлен в удалённый репозиторий
# осознаём, что нужно было еще что-то сделать в этом коммите
subl inc/header.html                  # вносим изменения
git add inc/header.html               # индексируем изменения
git commit --amend -m "«Шапка»: выполнена задача №1"
```

## Задания с MAKE файлами:

1) В приведенном make файле восполните недостающий пропуск

В приведенном make-файле восполните недостающий пропуск:

```
.PHONY: all clean format
CXX=x86_64-w64-mingw32-g++-win32
OFLAGS=-O3 -Wall -DNDEBUG
LDFLAGS= -static-libgcc -static-libstdc++ -lbcrypt
SOURCES=hash.cpp cipher.cpp main.cpp
HEADERS=hash.h cipher.h
EXECUTABLE=crypt.exe
$(EXECUTABLE): $(SOURCES)

#build $(SOURCES) -o $(EXECUTABLE) $(OFLAGS) $(LDFLAGS)
```

В приведенном make-файле восполните недостающий пропуск:

```
.PHONY: all clean
CXX=g++
CFLAGS=-c -Wall -std=c++11 -O2 -DNDEBUG
SOURCES=hash.cpp main.cpp
OBJECTS=$(SOURCES:.cpp=.o)
EXECUTABLE=demo-hash
all: $(SOURCES) $(EXECUTABLE)
$(EXECUTABLE): $(OBJECTS)
# Linking
$(CXX) $(OBJECTS) -o $@

clear:

rm -f $(OBJECTS) $(EXECUTABLE) *.orig
```

Активация Windows

Чтобы активировать Windows, перейдите в разд

В приведенном make-файле восполните недостающий пропуск:

```
.PHONY: all debug release clean
RELEASE_FLAGS = -O2 -Wall -DNDEBUG
DEBUG_FLAGS   = -g -O0 -Wall
RELEASE_EXEC   = hello
DEBUG_EXEC     = hello-dbg
SOURCE         = hello.cpp
all: debug release
debug: $(DEBUG_EXEC)
release: $(RELEASE_EXEC)
$(DEBUG_I) : $(SOURCE)

g++ $(DEBUG_FLAGS) $(SOURCE) -o $(DEBUG_EXEC)
$(RELEASE_EXEC): $(SOURCE)
g++ $(RELEASE_FLAGS) $(SOURCE) -o $(RELEASE_EXEC)
clean:
rm -f $(RELEASE_EXEC) $(DEBUG_EXEC)
```



В приведенном make-файле восполните недостающий пропуск:

```
.PHONY: all debug release clean
RELEASE_FLAGS = -O2 -Wall -DNDEBUG
DEBUG_FLAGS   = -g -O0 -Wall
RELEASE_EXEC   = hello
DEBUG_EXEC     = hello-dbg
SOURCE         = hello.cpp
all: debug release
debug: $(DEBUG_EXEC)
release: $(RELEASE_EXEC)
$(DEBUG_EXEC): $(SOURCE)
    g++ $(DEBUG_FLAGS) $(SOURCE) -o $(DEBUG_EXEC)
$(RELEASE_EXEC): $(SOURCE)
    g++ $(RELEASE_FLAGS) $(SOURCE) -o $(RELEASE_EXEC)
clean:
    rm -f $(RELEASE_EXEC) $(DEBUG_EXEC)
```

## Задания с FOO

1) Из модуля А вызывается функция foo модуля В. Определите тип сцепления модулей в данном случае, если foo имеет вид

Из модуля **А** вызывается функция **foo** модуля **В**. Определите тип сцепления модулей в данном случае, если функция **foo** имеет вид

```
int foo(int a, int b)
{
    return a*b;
}
```

Выберите один ответ:

- ☐ a. по образцу
- ☒ b. по данным
- ☐ c. по общей области
- ☐ d. по управлению



2) Из модуля А вызывается функция foo модуля В. Определите тип сцепления модулей в данном случае, если foo имеет вид

Из модуля А вызывается функция foo модуля В. Определите тип сцепления модулей в данном случае, если функция foo имеет вид

```
int foo(int *m, int size)
{
    int sum = 0;
    for (int i = 0; i < size; i++)
        sum = sum + m[i];
    return sum;
}
```

Выберите один ответ:

- ☐ a. по данным
- ☐ b. по общей области
- ☐ c. по управлению
- ☒ d. по образцу

3) Из модуля А вызывается функция foo модуля В. Определите тип сцепления модулей в данном случае, если переменная s объявлена на уровне модуля как char s[128], а функция foo имеет вид

Из модуля А вызывается функция foo модуля В. Определите тип сцепления модулей в данном случае, если переменная s объявлена на уровне модуля как char s[128], а функция foo имеет вид

```
char * foo(bool is_time, bool is_long)
{
    if (is_long)
        strcpy(s, "long ");
    else
        strcpy(s, "short ");
    if (is_time)
        strcat(s, "time");
    else
        strcat(s, "distance");
    return s;
}
```

Выберите один ответ:

- ☒ a. по общей области
- ☐ b. по данным
- ☐ c. по образцу
- ☐ d. по управлению

## Задания с пропуском слов

1) Проверку связи между компонентами, а также взаимодействия с различными частями системы выполняется с помощью ... (системное) тестирования.

Проверку связи между компонентами, а также взаимодействия с различными частями системы выполняют с помощью системное тестирования

2) Документ, на основании которого определяются требования к компоненту или системе, и базируются тестовые сценарии называют тестовым ... (планом)

Документ, на основании которого определяются требования к компоненту или системе, и базируются тестовые сценарии называют тестовым планом .

3) При ... (цикличной ИЛИ последовательной) связности функцией модуля выход одной функции служит исходными данными для другой функции

При Цикличной связности функций модуля выход одной функции служит исходными данными для другой функции.

4) При ... (последовательной) связности функций модуля выход одной функции служит исходными данными для другой функции

При последова связности функций модуля выход одной функции служит исходными данными для другой функции.

## Про связности

### Функциональная связность

При этой связности все объекты модуля предназначены для выполнения одной функции. Модуль, элементы которого связаны таким образом, имеет четко определенную цель, при его вызове выполняется одна задача, например, подпрограмма поиска минимального элемента массива. Такой модуль имеет максимальную связность, следствием которой являются его хорошие технологические качества: простота тестирования, модификации и сопровождения. Именно с этим связано одно из требований структурной декомпозиции «один модуль - одна функция»

### Последовательная связность

При этой связности функций выход одной функции служит исходными данными для другой. Как правило, такой модуль имеет одну точку входа, т.е. реализует одну подпрограмму, выполняющую две функции. Модуль с такой связностью функций можно разбить на два или более модулей. Технологичность такого модуля хуже: сложнее организовать тестирование, а при выполнении модификации мысленно приходится разделять функции модуля

### Информационная связанность

Так связанными считают функции, обрабатывающие одни и те же данные. Таким образом связанный модуль имеет неплохие показатели технологичности. Это объясняется тем, что все функции, работающие с некоторыми данными, собраны в одно место, что позволяет при изменении формата данных корректировать только один модуль

### Процедурная связанность

Так связаны функции или данные, которые являются частями одного процесса. Обычно модули с такой связанностью функций получают, если в модуле объединены функции альтернативных частей программы. При такой связанности отдельные элементы модуля связаны крайне слабо, так как реализуемые ими действия связаны лишь общим процессом, следовательно, технологичность данного вида связи ниже, чем предыдущего

### Временная связанность

Такая связанность функций подразумевает, что эти функции выполняются параллельно или в течение некоторого периода времени. Например, такую связанность имеют функции, выполняемые при инициализации некоторого процесса. Отличительной особенностью этой связанности является то, что действия, реализуемые такими функциями, могут выполняться в любом порядке. Содержание модуля с такой связанностью функций имеет тенденцию меняться: в него могут включаться новые действия и/или исключаться старые. Большая вероятность модификации функции еще больше уменьшает показатели технологичности модулей данного вида

#### Логическая связь

Такая связь базируется на объединении данных или функций в одну логическую группу. Модуль с такой связностью функций часто реализует альтернативные варианты одной операции, например, сложение целых чисел и сложение вещественных чисел. Из такого модуля всегда будет вызываться одна какая-либо его часть, при этом вызывающий и вызываемый модули будут связаны по управлению. Понять логику работы модулей, содержащих таким образом связанные компоненты, как правило, сложнее, чем модулей, использующих временную связность, следовательно, их показатели технологичности еще ниже

#### Случайная связность

В том случае, если связь между элементами мала или отсутствуют, считают, что они имеют такую связность. Модуль, элементы которого связаны так, имеет самые низкие показатели технологичности, так как элементы, объединенные в нем, вообще не связаны

5) Проверку отдельно взятых модулей, функций или классов выполняют с помощью ... (модульного) тестирования

Проверку отдельно взятых модулей, функций или классов выполняют с помощью модульного тестирования

6) Проверку того, что существует ранее функциональность работает как и прежде после изменений, сделанных в приложении или окружающей среде, выполняют с помощью ... (функционального) тестирования

Проверку того, что существующая ранее функциональность работает как и прежде после изменений, сделанных в приложении или окружающей среде, выполняют с помощью функционального тестирования

7) Набор входных значений, предусловий выполнения, ожидаемых результатов и постусловий выполнения, разработанный для определенной цели или тестового условия, таких как выполнения определенного пути программы или же для проверки соответствия определенному требованию называют тестовым ... (сценарием)

Набор входных значений, предусловий выполнения, ожидаемых результатов и постусловий выполнения, разработанный для определенной цели или тестового условия, таких как выполнения определенного пути программы или же для проверки соответствия определенному требованию называют тестовым сценарием .

8) Процесс определения полноты соответствия заданных требований и созданной системы их конкретному функциональному назначению называется ... (аттестация)

Процесс определения полноты соответствия заданных требований и созданной системы их конкретному функциональному назначению называется аттестацией .

9) Документ, описывающий цели, подходы, ресурсы и график запланированных тестовых активностей называется тестовым ... (планом)

Документ, описывающий цели, подходы, ресурсы и график запланированных тестовых активностей называется тестовым планом .

10) Для защиты заголовочного файла от повторного включения можно использовать директиву #programa ... (once)

Для защиты заголовочного файла от повторного включения можно использовать директиву #pragma once .

11) Документ, на основании которого определяются требования к компоненту или системе, и базируются тестовые сценарии называют тестовым ... (базисом)

Документ, на основании которого определяются требования к компоненту или системе, и базируются тестовые сценарии называют тестовым базисом .

## Задания с диаграммами:

1) Какая диаграмма изображена на рисунке

Какая диаграмма изображена на рисунке?

```
sequenceDiagram
    participant User as Пользователь
    participant Button as Кнопка "Старт игры"
    participant Game as Игра
    participant Loss as Проигрыш
    participant Exit as Выход

    User->>Button: Нажата
    activate Button
    Button->>Game: Запуск
    deactivate Button
    activate Game
    Game->>Loss: Игрок убит
    deactivate Game
    activate Loss
    Loss-->>Game: Начало раунда
    deactivate Loss
    activate Game
    Game->>Exit: Нажато «выход»
    deactivate Game
    activate Exit
    deactivate Exit
```

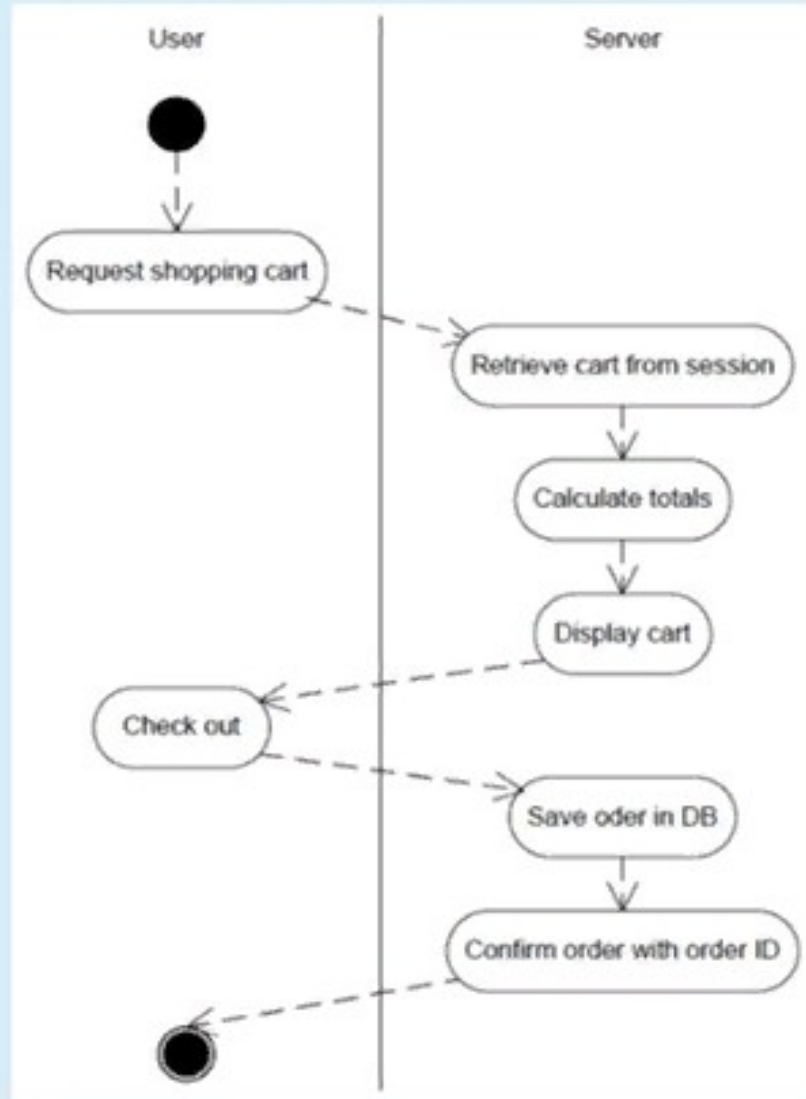
Выберите один ответ:

- ☐ a. диаграмма классов
- ☐ b. диаграмма деятельности
- ☒ c. диаграмма последовательностей
- ☐ d. диаграмма развертывания

Активация )  
Чтобы активир



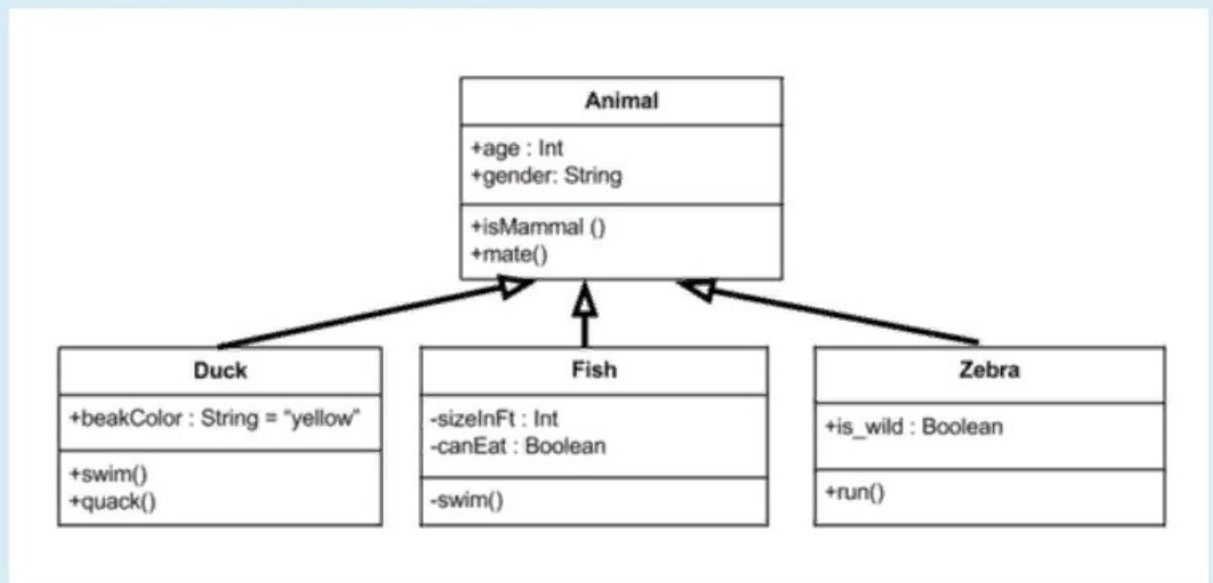
Какая диаграмма изображена на рисунке?



Выберите один ответ:

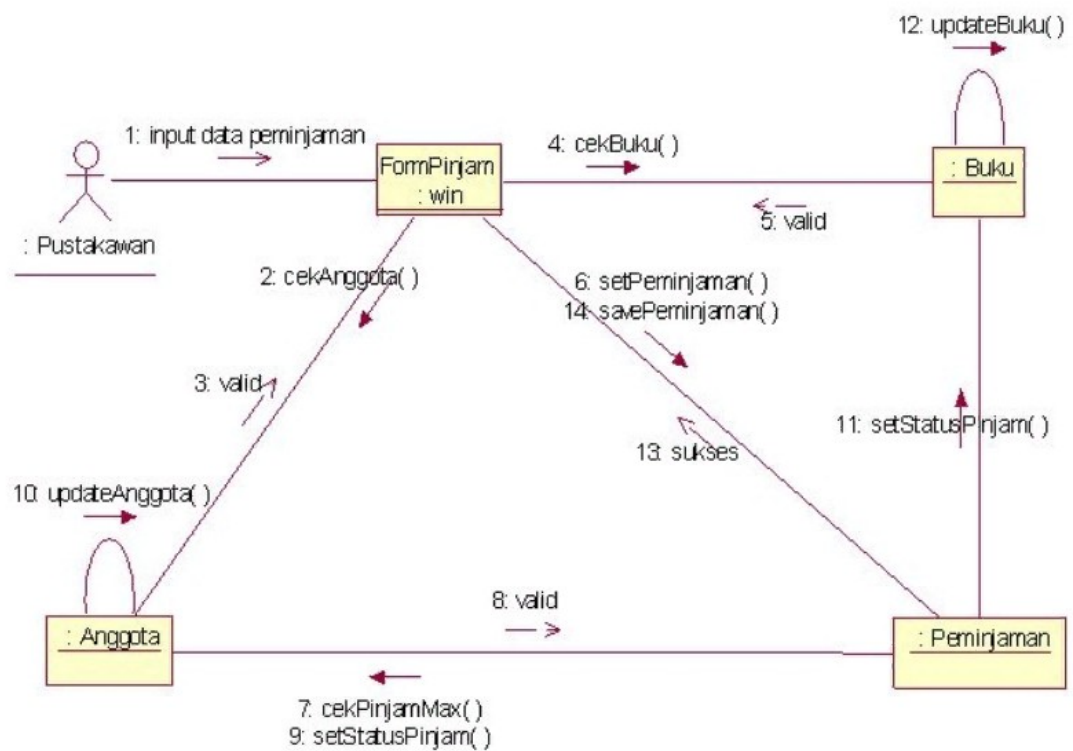
- ☐ a. диаграмма вариантов использования
- ☒ b. диаграмма деятельности
- ☐ c. диаграмма развертывания
- ☐ d. диаграмма кооперации
- ☐ e. диаграмма классов
- ☐ f. диаграмма состояний
- ☐ g. диаграмма последовательностей
- ☐ h. диаграмма компонентов

Какая диаграмма изображена на рисунке?



Выберите один ответ:

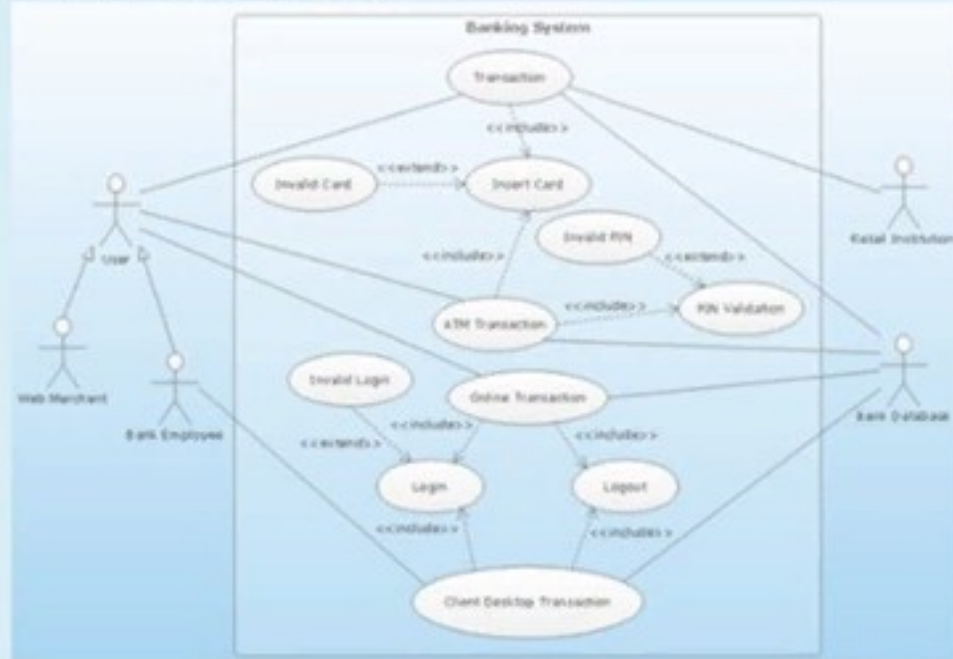
- ☐ a. диаграмма компонентов
- ☐ b. диаграмма деятельности
- ☐ c. диаграмма кооперации
- ☒ d. диаграмма классов
- ☐ e. диаграмма последовательностей
- ☐ f. диаграмма состояний
- ☐ g. диаграмма развертывания
- ☐ h. диаграмма вариантов использования



Выберите один ответ:

- ☐ a. диаграмма состояний
- ☐ b. диаграмма компонентов
- ☐ c. диаграмма последовательностей
- ☐ d. диаграмма вариантов использования
- ☐ e. диаграмма развертывания
- ☒ f. диаграмма кооперации

Какая диаграмма изображена на рисунке?



Выберите один ответ:

- ☒ a. диаграмма вариантов использования
- ☐ b. диаграмма компонентов
- ☐ c. диаграмма деятельности
- ☐ d. диаграмма последовательностей
- ☐ e. диаграмма классов
- ☐ f. диаграмма развертывания
- ☐ g. диаграмма кооперации
- ☐ h. диаграмма состояний

## 19 ЗАДАНИЯ

1) Составьте тестовый сценарий для тестирования модуля содержащего следующие функции

Составьте тестовый сценарий для тестирования модуля, содержащего следующие функции:

```
/*транслитерация строки на русском языке английскими символами */
```

```
string transRuEn(const string)
```

```
/*восстановление транслитерированной строки с английского на русский алфавит*/
```

```
string detransEnRu(const string)
```

Правила транслитерации представлены в таблице ниже:

рус	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О
англ	A	B	V	G	D	E	JO	ZH	Z	I	J	K	L	M	N	O
рус	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ,Ь	Ы	Э	Ю	Я
англ	P	R	S	T	U	F	H	TS	CH	SH	SCH	','	Y	EH	JU	JA

Регистр букв при транслитерации сохраняется. Присутствие английских букв, кавычек и апострофов в русском тексте приводит к исключению. Остальные нетранслитерируемые символы остаются неизменными.

При возникновении ошибок возбуждаются исключения TransError при транслитерации и DetransError при восстановлении.

**Тест      Ключ   Ожидаемый результат**

КОД      001    KOD

код      001    kod

КОД КОД'      TransError

Код      001    Kod

Регист букв

К (р) - 11    О(р) - 15    Д - 5

К (е) - 11    О(е) - 15    Д - 4

Активация Windows

Чтобы активировать Windows, перейдите в разд.  
"Параметры"

Составьте тестовый сценарий для тестирования модуля, содержащего следующие функции:

```
/*кодирование одной цифры в виде строки сегментами почтовой цифры */
```

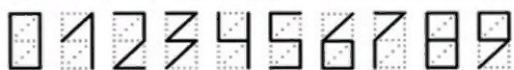
```
string codePostDigit(const int)
```

```
/*декодирование строки с сегментами почтовой цифры в обычную цифру */
```

```
int decodePostDigit(string)
```

Образец написания почтовых цифр представлен на рисунке ниже

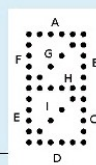
Образец написания цифр индекса



Обозначение сегментов почтовых цифр показано на рисунке справа. Например, цифра 0 будет кодироваться строкой ABCDEF, а цифра 1 — строкой BCG.

Названия сегментов в строке должны быть в верхнем регистре и расположены в алфавитном порядке.

При возникновении ошибок при кодировании должны возбуждаться исключения CodePostError, а при возникновении ошибок декодирования — DecodePostError.



№	Название функции	Условие	Входные данные	Ожидаемый результат	Полученный результат	Комментарий
1.1	codePostDigit	Проверка работы функции кодирования	0	ABCDEF	ABCDEF	Успешно!
1.2	codePostDigit	Ввод пустой строки	""	CodePostError	CodePostError	Ошибка!!! Введена пустая строка
1.3	codePostDigit	Ввод более одной цифры	12	CodePostError	CodePostError	Ошибка!!! Введено более одной цифры
1.4	codePostDigit	Ввод спец.символа или буквы	a	CodePostError	CodePostError	Ошибка!!! Неверный тип вводимых данных
1.5	codePostDigit	Ввод нескольких параметров через пробел	1 b	CodePostError	CodePostError	Ошибка!!! Неверное количество передаваемых параметров
2.1	decodePostDigit	Проверка работы функции декодирования	ABCDEF	0	0	Успешно!
2.2	decodePostDigit	Ввод пустой строки	""	DecodePostError	DecodePostError	Ошибка!!! Введена пустая строка
2.3	decodePostDigit	Ввод любой строки в верхнем регистре	SFGBSD	DecodePostError	DecodePostError	Ошибка!!! Неверная последовательность букв (не в алфавитном порядке)
2.4	decodePostDigit	Ввод любой строки с наличием посторонних символов, не относящихся к буквам верхнего регистра	AbC4E@	DecodePostError	DecodePostError	Ошибка!!! Наличие в строке посторонних символов
2.5	decodePostDigit	Ввод строки в верхнем регистре и в алфавитном порядке	ABCD	DecodePostError	DecodePostError	Ошибка!!! Отсутствие данной последовательности в таблице декодирования
2.6	decodePostDigit	Ввод				

### Задания с выбором ответа

1) Использование модулей в программировании имеет следующее преимущество:

Использование модулей в программировании имеет следующие преимущества:

Выберите один или несколько ответов:

- ☐ а. возможность реализации графического интерфейса
- ☐ б. возможность создания сетевых приложений
- ☒ в. возможность разделения интерфейса и реализации
- ☒ г. возможность создания библиотек функций и классов
- ☐ д. возможность отдельной компиляции