

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МУРМАНСКИЙ АРКТИЧЕСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «МАГУ»)

Методическая разработка практического занятия

по профессиональному модулю ПМ.03 «Участие в интеграции программных модулей», МДК 03.02. «Инструментальные средства разработки программного обеспечения»

09.02.03 Программирование в компьютерных системах
очной формы обучения

Авторы: преподаватель дисциплин профессионального цикла Ломова Л.А.

Оглавление

Ключевые слова	3
Аннотация.....	3
Основная часть	4
Список литературы	16

Ключевые слова

Проектирование, activity diagram, UML, технологическая документация.

Аннотация

Методическая разработка практического занятия по теме: «Разработка и построение диаграммы деятельности» составлена в соответствии с требованиями Федерального государственного образовательного стандарта СПО.

В методической разработке практического занятия даны обоснования развивающего обучения, способствующие формированию у студента знаний и умений по дисциплине.

На практическом занятии используются приемы, средства и методы обучения, активизирующие мыслительную деятельность, воспитывающие у студентов устойчивый познавательный интерес, а также умение осмысливать и применять имеющиеся знания в различной практической деятельности.

При изучении темы используются элементы беседы, самостоятельной работы, самостоятельное выполнение практических работ, работа в группах.

Методическая разработка практического занятия включает себя:

- методическую разработку для преподавателя;
- методическую разработку практического занятия для студентов;
- дидактический материал по данной теме.

Основная часть

1. МЕТОДИЧЕСКАЯ РАЗРАБОТКА ПРАКТИЧЕСКОГО ЗАНЯТИЯ ДЛЯ ПРЕПОДАВАТЕЛЯ

Курс 4, специальность 09.02.03 «Программирование в компьютерных системах»

Тема: «Разработка и построение диаграммы деятельности»

Цели занятия:

1. Дидактические:

– формирование профессиональных компетенций в соответствии с требованиями ФГОС:

ПК 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

ПК 3.6. Разрабатывать технологическую документацию.

– контроль и коррекция знаний по теме «Разработка и построение диаграммы деятельности».

2. Развивающие:

– развивать способность осуществлять поиск информации, использовать информацию, необходимую для эффективного выполнения профессиональных задач, профессионального и личностного развития (ОК.4);

– использовать информационно-коммуникационные технологии в своей деятельности (ОК.5);

– развивать способность организовывать свою деятельность, выбирать методы и способы решения поставленных задач (ОК.2);

– развивать способность работать в команде (ОК.6);

– развивать способность принимать решение в стандартных и нестандартных ситуациях (ОК.3).

3. Воспитательные:

– воспитывать устойчивый интерес к своей будущей профессии (ОК.1);

– воспитывать чувство ответственности за результаты своей работы, работы членов команды (ОК.7).

В соответствии с требованиями ФГОС студенты

должны знать:

– основные принципы процесса разработки программного обеспечения;

– основные методы и средства эффективной разработки.

должны уметь:

– владеть основными методологиями процессов разработки программного обеспечения;

использовать приобретенные знания и умения в практической деятельности для:

– участия в выработке требований к программному обеспечению;

– участия в проектировании программного обеспечения с использованием специализированных программных пакетов.

Тип занятия: формирование умений.

Вид занятия: практическое занятие.

Методы обучения: инструктаж и самостоятельная работа, проблемный, репродуктивный.

Метод контроля знаний: защита отчета (в группе); тестирование.

Продолжительность занятия: 180 минут.

Межпредметные связи:

- Информационные технологии;
- ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем МДК 01.02 Прикладное программирование.

Распределение времени практического занятия представлено в таблице 1.

Таблица 1 – Хронокарта практического занятия

№	Элементы занятия	Время, мин.
1	Организационный момент	2
2	Мотивация	3
3	Контроль исходного уровня знаний	15
4	Методические инструкции	10
5	Работа студентов в группах по заданиям	90
6	Представление результатов работы	40 (по 10 мин на гр.)
7	Тестирование	15
8	Подведение итогов занятия	3
9	Сообщение домашнего задания	2
Итого:		180

Характеристика отдельных элементов занятия:

1. Приветствие. Контроль внешнего вида студентов, отсутствующих студентов, готовности аудитории к занятию.

2. Постановка целей и задач. Создание мотивационного пространства.

Преподаватель четко называет тему занятия, цель занятия, этапы занятия.

Совместно со студентами формируется значение и место данной темы в будущей профессии. Обращается внимание на то, что этап проектирования является важнейшим этапом в разработке программ и автоматизированных систем, от результата которого зависит дальнейшая разработка. При создании мотивационного пространства используются межпредметные связи, показывается значение данной темы при изучении профессиональных модулей.

3. Контроль исходного уровня знаний.

Используется лекционный материал по теме.

Вопросы:

- 1) Что такое диаграмма деятельности?
- 2) Что такое деятельность?
- 3) Перечислите основные нотации, используемые для построения диаграммы.

4) Какие Case – средства можно использовать для построения этого типа диаграмм?

5) Почему для проектирования данного этапа конкретизации системы нельзя обойтись другими типами диаграмм? (**проблематика занятия**)

4. Инструктаж к выполнению практической работы.

Преподаватель совместно со студентами разбирают предстоящую практическую работу в соответствии с методическими указаниями. Преподаватель обращает внимание на наиболее сложные моменты, на требования к оформлению отчетов и соответствию структуры диаграмм результатам практической работы № 1 «Разработка и построение диаграммы вариантов использования». Работа выполняется в группах по 2 человека, особое внимание уделяется коллективной работе над поставленной задачей.

5. Выполнение практической работы

Студенты выполняют практическую работу в парах в соответствии с методическими указаниями и рекомендациями, данными преподавателем. Преподаватель в процессе выполнения работы консультирует студентов, направляет их при возникновении затруднений.

6. Представление результатов работы.

Студенты представляют результаты работы, поясняя основные компоненты и сопоставляя с этапом общего проектирования (практическая работа № 1), отвечают на вопросы.

7. Тестирование.

Студенты выполняют тест по теме, подтверждая уровень знаний.

Критерий оценки:

80 – 100 баллов – «5» (отлично);

70 – 60 баллов – «4» (хорошо);

50 – 60 баллов – «3» (удовлетворительно);

менее 50 баллов – «2» (не удовлетворительно).

8. Подведение итогов занятия.

Преподаватель обобщает результаты работы, достижение целей занятия, комментирует работу на занятии отдельных групп студентов и всей группы в целом. Выставление итоговых оценок интегративно с учётом вводного контроля, проделанной самостоятельной работы и тестирования.

9. Сообщение домашнего задания.

Преподаватель сообщает тему следующего занятия: «Инструментальные средства проектирования и анализа требований к программному обеспечению. UML: диаграмма последовательности».

2. МЕТОДИЧЕСКАЯ РАЗРАБОТКА ПРАКТИЧЕСКОГО ЗАНЯТИЯ ДЛЯ СТУДЕНТОВ

Практическая работа.

Тема: Разработка и построение диаграммы деятельности.

Цель: приобрести умение создавать диаграммы для отображения состояний объектов системы, имеющих сложную модель поведения.

Время выполнения: 180 минут.

Ход работы:

1. Ознакомиться с кратким теоретическим материалом.
2. Выполнить индивидуальное задание:
 - разработать диаграммы деятельности для проектируемой подсистемы, согласно варианту индивидуального задания, используя инструментальные средства (для 4 прецедентов из диаграммы прецедентов, разработанной в практической работе 1), используя сервис (на выбор).
3. Оформить отчет по практической работе¹.
4. Представить результат работы в группе.
5. Выполнить тест по теме.

Методические указания по выполнению работы

1.1. Диаграмма активности

Диаграммы активностей (Activity Diagrams) являются представлением алгоритмов неких действий (активностей), выполняющихся в системе.

Деятельность – это протяженное по времени составное действие (т.е. составленное из более простых действий).

Диаграммы активностей являются представлением алгоритмов неких действий (активностей), выполняющихся в системе.

Диаграммы деятельности описывают переход от одной деятельности к другой.

Разработка диаграммы деятельности преследует цели:

- детализировать особенности алгоритмической и логической реализации выполняемых системой операций и прецедентов;
- выделить последовательные и параллельные потоки управления;
- подготовить детальную документацию для взаимодействия разработчиков системы с ее заказчиками и проектировщиками.

Самым большим достоинством диаграмм деятельности является поддержка *параллелизма* протекающих в системе процессов.

Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия или состояния деятельности, а дугами – переходы от одного состояния действия/деятельности

¹ Оформление учебных документов. Методические указания. – Режим доступа: <http://mshu.edu.ru/fk/sveden/programmy/vsem/mu/mu-oformlenie-uchebnyh-dokumentov.pdf>

к другому. Каждая диаграмма деятельности должна иметь единственное начальное и единственное конечное состояния.

Основные элементы диаграммы деятельности:

Начальное и конечное состояния на диаграммах деятельности изображаются как закрашенный кружок и закрашенный кружок внутри окружности соответственно, рисунок 1.

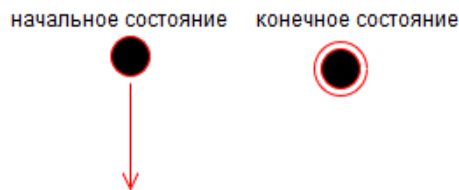


Рисунок 1 – Графические нотации. Начало и конец диаграммы

Переход (Transitions) — отношение между двумя состояниями, показывающее, что объект, находящийся в первом состоянии, должен выполнить некоторые действия и перейти во второе состояние. Когда действие или деятельность в некотором состоянии завершается, поток управления сразу переходит в следующее состояние действия или деятельности (представляется простой линией со стрелкой).

Ветвления или множественный переход со сторожевыми условиями. Ветвление описывает различные пути выполнения в зависимости от значения некоторого булевского выражения. В точку ветвления может входить ровно один переход, а выходить — два или более. Для каждого исходящего перехода задается булевское выражение, которое вычисляется только один раз при входе в точку ветвления. Ни для каких двух исходящих переходов сторожевые условия не должны одновременно принимать значение «истина», иначе поток управления окажется неоднозначным. Условия должны покрывать все возможные варианты, иначе поток остановится. Альтернатив может быть две или больше, и они должны полностью покрывать все возможные варианты разрешения условия, рисунок 2:

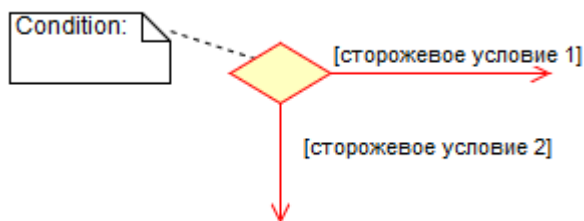


Рисунок 2 – Графические нотации. Ветвление

Разделения и слияния. Используются для изображения параллельных потоков, что характерно для моделирования бизнес-процессов. В UML для обозначения разделения и слияния таких параллельных потоков выполнения используется синхронизационная черта, которая рисуется в виде жирной вертикальной или горизонтальной линии. При этом *разделение (concurrent fork)* имеет один входящий переход и несколько выходящих, *слияние (concurrent*

join), наоборот, имеет несколько входящих переходов и один выходящий, как показано на рисунке 3.

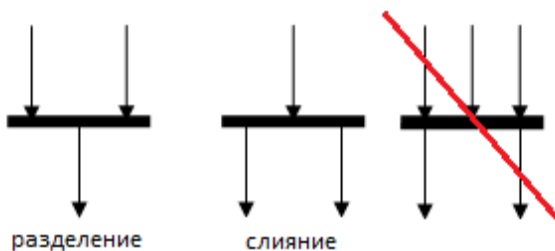


Рисунок 3 – Графические нотации. Разделения и слияния

Дорожки – это разновидность пакетов, описывающих связанную совокупность работ. Каждой присутствующей на диаграмме дорожке присваивается уникальное имя. Никакой глубокой семантики дорожка не несет и может отражать некоторую сущность реального мира (каждая дорожка представляет сферу ответственности за часть всей работы, изображенной на диаграмме). На диаграмме деятельности, разбитой на дорожки, каждая деятельность принадлежит ровно одной дорожке, но переходы могут пересекать границы дорожек.

В потоке управления, ассоциированном с диаграммой деятельности, могут участвовать объекты.

К *объектам* относятся непосредственно объекты (object) в традиционном понимании UML, отправка сигнала (send signal), прием сигнала (accept signal) и событие времени (time event).

Некоторые виды деятельности будут порождать объекты-экземпляры этих классов (например, деятельность «Обработать заказ» создаст объект Заказ), тогда как другие виды деятельности будут модифицировать эти объекты (например, «Отгрузить заказ» может изменить состояние объекта Заказ на «выполнен»).

Пример оформления объекта представлен на рисунке 4.

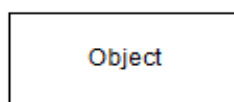


Рисунок 4 – Графические нотации. Объекты

Сигналы и события

Шаг в процессе, отправляющий сигнал, можно показать как действие отправки сигнала в активности. Шаг, ожидающий конкретного сигнала или события перед продолжением работы, можно показать как действие принятия события.

Например, можно показать шаг, отправляющий заказ, и другой шаг, который должен получить заказ, прежде чем его обработать.

– Отправка сигнала

Действия отправки сигнала позволяют указать, что какой-либо сигнал или сообщение отправлено в другие активности или процессы. Тип отправляемых сообщений указывается в имени действия.

Особенности:

а) Управление немедленно передается следующему действующему в потоке управления (если есть).

б) Действие отправки сигнала нельзя использовать для описания реакции процесса на возвращаемую информацию. Для этого нужно использовать отдельное действие приема события.

– Ожидание сигнала или события

Действия приема события позволяют указать, что данное действие ожидает некоторого внешнего события или входящего сообщения. Тип ожидаемых событий указывается в имени действия.

Особенности:

а) Чтобы показать, что действие ожидает внешнего события или сообщения в определенной точке потока, в подходящем месте активности необходимо изобразить действие приема события с входящим потоком.

б) Чтобы показать, что действие может отвечать на внешнее событие или сообщение в любое время, необходимо изобразить действие приема события без входящего потока. При возникновении указанного внешнего события в активности начнется новый поток, который начинается с действия приема события.

в) Действия приема события нельзя использовать для описания значений, возвращаемых отправителю сигнала (используется отдельное действие отправки сигнала).

г) Отобразив исходящие из действия потоки данных, можно продемонстрировать, каким образом активность обрабатывает данные, получаемые с сигналом.

Примеры объектов на рисунке 5:

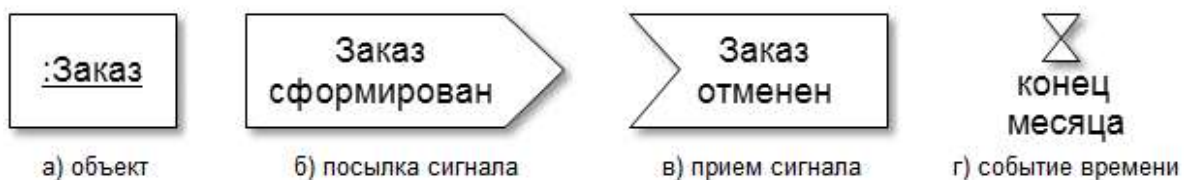


Рисунок 5 – Графические нотации. Разновидности объектов

1.2. Пример построения диаграммы

Построить диаграмму активности, описывающую прецедент «Продажа товара»:

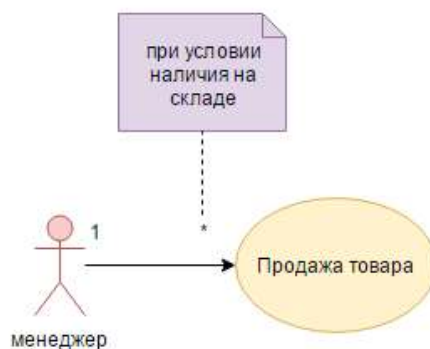


Рисунок 6 – Описание прецедента «Продажа товара»

В данном случае уровень конкретизации процесса «Продажа товара» позволяет включить нескольких актеров: сотрудника отдела оформления заказа (для принятия заказа и отправки), сотрудника отдела продаж (получение оплаты и подготовки товара к отправке) и складского работника (для поиска заданного товара и перенаправлении его в отдел продаж).

Совокупность работ каждой сущности объединена в дорожки (имеют те же имена).

Диаграмма активности, описывающая прецедент «Продажа товара», учитывает спецификацию исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов, как показано на рисунке 7.

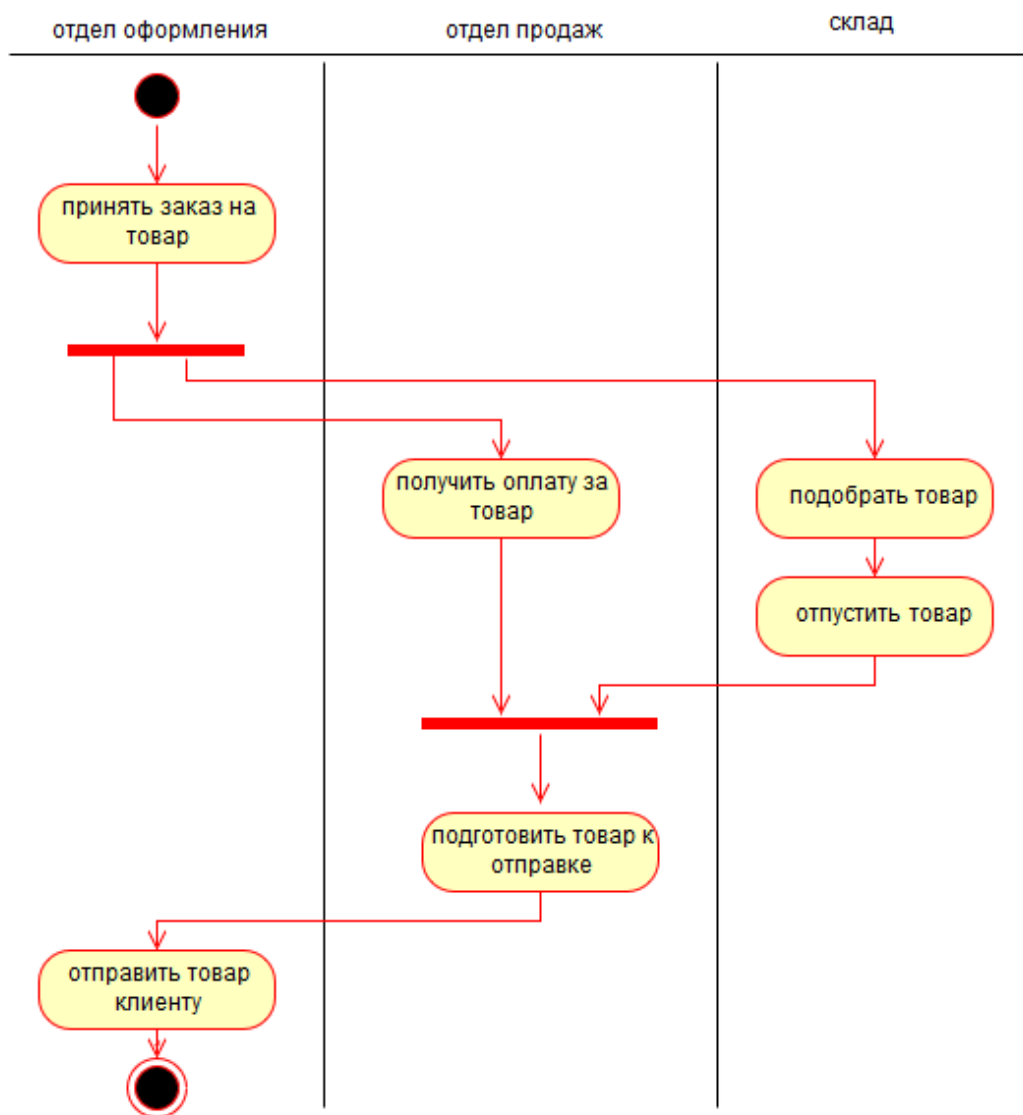


Рисунок 7 – Диаграмма активности

1.3. Инструментарий для создания диаграмм

- 1) <https://creately.com/ru/инструмент-создания-диаграмм-UML-онлайн>
- 2) <https://www.draw.io/>
- 3) <https://www.genmymodel.com/activity-diagram-online>
- 4) <https://online.visual-paradigm.com/solutions/free-activity-diagram-tool/>

1.4. Задания:

№	Вариант
1.	АИС ГИБДД. Подсистема фиксации нарушений.
2.	АИС ГИБДД. Авторизация + ведение БД сотрудников.
3.	АИС ГИБДД. Постановка транспорта на учет.

4.	АИС ГИБДД. Плановые мероприятия.
5.	АИС ГИБДД. Статистика + единая стилистика проекта.
6.	АИС ГИБДД. Подсистема Личный кабинет пользователя.
7.	АИС Отель. Подсистема регистрации заездов /выбытия /бронь /дополнительные услуги.
8.	АИС Отель. Клининг.
9.	АИС Отель. Статистика + единая стилистика проекта.
10.	АИС Отель. Авторизация + ведение БД сотрудников, клиентов, номерного фонда.
11.	АИС Отель. Подсистема управления персоналом.

Вопросы по теме «Разработка и построение диаграммы деятельности»:

- 1) Что такое диаграмма деятельности?
- 2) Что такое деятельность?
- 3) Перечислите основные нотации, используемые для построения диаграммы.
- 4) Какие Case – средства можно использовать для построения этого типа диаграмм?
- 5) Почему для проектирования данного этапа конкретизации системы нельзя обойтись другими типами диаграмм?
- 6) Перечислите основные ошибки, допускаемые при построении диаграмм.

Критерии оценки практической работы:

«5» (отлично): выполнены все задания практической работы, студент ориентируется в предметной области, четко и без ошибок отвечает на все контрольные вопросы.

«4» (хорошо): выполнены все задания практической работы; студент отвечает на все контрольные вопросы с замечаниями; диаграмма(ы) содержит незначительные ошибки.

«3» (удовлетворительно): выполнены все задания практической работы с замечаниями; студент не уверено ориентируется в предметной области; диаграмма (ы) содержит незначительные ошибки; отвечает на контрольные вопросы с замечаниями.

«2» (не зачтено): не выполнены или выполнены неправильно задания практической работы; студент не ориентируется в предметной области; не владеет теоретическим материалом в должном объеме.

3. ДИДАКТИЧЕСКИЙ МАТЕРИАЛ ПО ТЕМЕ

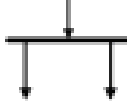


1. Моделирование в UML позволяет решить задачи*:

- а) анализа и синтеза систем управления;
- б) разработки и отладки программного обеспечения;
- в) визуализации системы в ее текущем или желательном состоянии;
- г) проведения тестирования разработанного программного обеспечения;
- д) описания структуры или поведения системы и получения шаблона, позволяющего сконструировать систему;
- е) моделирования разрабатываемой информационной системы в целом;
- ж) документирования принимаемых решений, используя полученные модели;
- з) расчёта экономической эффективности от внедрения программного обеспечения.

4. Что описывает диаграмма активности?

- а) последовательность реализации системы;
- б) последовательность операций, выполняемых системой;
- в) последовательность ввода системы в эксплуатацию;
- г) последовательность действий по разработке системы.



3. Каким символом изображается распараллеливание процесса на диаграмме деятельности?


а)	
б)	
в)	

4. Какое определение дорожки (swimlane) правильно?

- а) специальный случай состояния деятельности с несколькими входными переходами и, по крайней мере, одним выходящим из состояния переходом;
- б) отдельная диаграмма деятельности, которая служит для представления реализации операций бизнес-процесса;
- в) графическая область диаграммы деятельности, содержащая элементы модели, ответственность за выполнение которых принадлежит отдельным подсистемам.

5. Каким символом изображается объект на диаграмме деятельности?

а)	
б)	

в)	
----	---

6. Допустимо ли изображение триггерных переходов на диаграмме деятельности?

- а) да, но с дополнительными сторожевыми условиями;
- б) да;
- в) нет.

7. Какое определение состояния под-деятельности (subactivity state) правильно?

- а) произвольное состояние с дополнительным стереотипом «subactivity»;
- б) состояние в графе деятельности, которое служит для представления неатомарной последовательности шагов процесса;
- в) состояние в графе деятельности, которое служит для представления действий, выполняемых мгновенно.

8. Допустимо ли изображение нетриггерных переходов на диаграмме деятельности?

- а) нет;
- б) да, но с дополнительными ограничениями;
- в) да;
- г) да, но только при моделировании параллельных процессов.

Ключи к тесту:

№\ варианты ответа	1	2	3	4	5	6	7	8
а			+					
б		+			+		+	
в	+			+		+		+
г								
д	+							
е								
ж	+							
з								

Критерий оценки:

- 80 – 100 баллов – «5» (отлично);
- 70 – 60 баллов – «4» (хорошо);
- 50 – 60 баллов – «3» (удовлетворительно);
- менее 50 баллов – «2» (не удовлетворительно).

Список литературы

Для студентов:

- 1) Виды диаграмм UML – Режим доступа: <https://www.intuit.ru/studies/courses/1007/229/lecture/5954?page=1>
- 2) Инструкции для практических работ по МДК 03.02 «Инструментальные средства разработки программного обеспечения» (сборник МУ по выполнению практических работ), филиал МАГУ, 2018;
- 3) Рудаков А.В. Технология разработки программный продуктов: учебник для студ. учреждений сред. проф. образования / А.В. Рудаков. – 9-е изд., стер. – М: Издательский центр «Академия», 2014.
- 4) Теория и практика UML. Диаграмма деятельности. – Режим доступа: http://it-gost.ru/articles/view_articles/96
- 5) Чертова, Е. А. Программная инженерия. Визуальное моделирование программных систем: учебник для СПО / Е. А. Чертова. – 2-е изд., испр. и доп. – М.: Издательство Юрайт, 2019. – 147 с. – (Серия: Профессиональное образование); [Электронный ресурс]. – Режим доступа: <https://www.biblio-online.ru/book/programmaya-inzheneriya-vizualnoe-modelirovanie-programmnyh-sistem-441255>

Для преподавателей:

- 1) Рудаков А.В. Технология разработки программный продуктов: учебник для студ. учреждений сред. проф. образования / А.В. Рудаков. – 9-е изд., стер. – М: Издательский центр «Академия», 2014.
- Перечень ресурсов информационно-телекоммуникационной сети «Интернет»:
- 5) Гниденко, И. Г. Технология разработки программного обеспечения: учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – М.: Издательство Юрайт, 2019. – 235 с. – (Серия: Профессиональное образование). – ISBN 978-5-534-05047-9; [Электронный ресурс]. – Режим доступа: <https://biblio-online.ru/book/tehnologiya-razrabotki-programmnogo-obespecheniya-438444>
 - 6) Диаграмма деятельности [Электронный ресурс]. – Режим доступа: <https://www.intuit.ru/studies/courses/941/229/lecture/5958>
 - 7) Теория и практика UML. Диаграмма деятельности [Электронный ресурс]. – Режим доступа: http://www.it-gost.ru/articles/view_articles/96
 - 8) Чертова, Е. А. Программная инженерия. Визуальное моделирование программных систем: учебник для СПО / Е. А. Чертова. – 2-е изд., испр. и доп. – М.: Издательство Юрайт, 2019. – 147 с. – (Серия: Профессиональное образование); [Электронный ресурс]. – Режим доступа: <https://biblio-online.ru/book/programmaya-inzheneriya-vizualnoe-modelirovanie-programmnyh-sistem-441255>.