

Федеральное государственное бюджетное образовательное учреждение
высшего образования
Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики



УТВЕРЖДАЮ

Декан факультета ВМК МГУ

И.А.Соколов/

_____ 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Наименование дисциплины (модуля):

Объектно-ориентированный анализ и проектирование

Уровень высшего образования:

бакалавриат

Направление подготовки (специальность):

02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль) ОПОП:

дисциплина относится к вариативной части программы

Форма обучения:

очная

Москва 2023

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ бакалавриата, магистратуры, реализуемых последовательно по схеме интегрированной подготовки по направлениям 02.03.02, 02.04.02 «Фундаментальная информатика и информационные технологии» в редакции приказа МГУ от 30 декабря 2016 г.

1. Дисциплина относится к вариативной части ОПОП ВО.

2. Входные требования для освоения дисциплины (модуля): учащиеся должны владеть знаниями поддисциплинам базовой части «Системы программирования» и «Объектно-ориентированное программирование» в объеме, соответствующем программе первых двух лет обучения основных образовательных программ бакалавриата по укрупненному группам направлений и специальностей 02.00.00 «Компьютерные и информационные науки».

3. Результаты обучения по дисциплине (модулю), соотнесенные с требуемыми компетенциями выпускников.

Компетенции выпускников, частично формируемые при реализации дисциплины (модуля):

- **ПК-7.Б** Способность разрабатывать и реализовывать процессы жизненного цикла информационных систем, программного обеспечения, сервисов систем информационных технологий, а также методы и механизмы оценки и анализа функционирования средств и систем информационных технологий.

Планируемые результаты обучения по дисциплине (модулю):

Знать:

1. принципы объектно-ориентированного подхода к анализу и проектированию программного обеспечения;
2. графическую нотацию языка Unified Modeling Language¹;
3. текстовую нотацию языка Object Constraint Language²;
4. методы использования языков UML и OCL в разработке программного обеспечения.

Уметь:

1. моделировать структурные и поведенческие аспекты программных систем с помощью диаграмм UML;
2. формально описывать ограничения на языке OCL в моделях программного обеспечения;
3. применять CASE-средства с поддержкой UML в технологическом цикле создания программного обеспечения.

Владеть:

1. навыками анализа и проектирования в соответствии с современной технологией разработки программного обеспечения, использующей языки UML и OCL.

¹ сокращённо UML

² сокращённо OCL

4. Формат обучения: лекции проводятся с использованием мультимедийного проектора; в рамках самостоятельной работы студенты выполняют практические задания для освоения CASE-средства с поддержкой языков UML и OCL.

5. Объем дисциплины (модуля) составляет 3 з.е., в том числе 72 академических часа, отведены на контактную работу обучающихся с преподавателем, 36 академических часа отведены на самостоятельную работу обучающихся.

6. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий.

Наименование и краткое содержание разделов и тем дисциплины (модуля), форма промежуточной аттестации по дисциплине (модулю)	Всего (часы)	В том числе			
		Контактная работа (работа во взаимодействии с преподавателем)		Самостоятельная работа обучающихся, часы	
		лекционные занятия	занятия семинар-кого типа	Всего	(виды самостоятельной работы – эссе, реферат, контрольная работа и пр.)
<i>Тема 1. Объектная модель – основа объектно-ориентированного анализа и проектирования</i> (принципы построения объектной модели; её элементы: объект, класс, атрибут, операция, компонент, интерфейс, пакет, подсистема; виды связей между элементами объектной модели: соединение, ассоциация, обобщение, реализация, зависимость; модели программного обеспечения и их роль в создании систем; архитектура программного обеспечения; модель «4+1»; архитектурные представления)	8	6	0	6	2 (домашнее задание)
<i>Тема 2. Унифицированный язык моделирования</i> (основные виды UML-диаграмм: диаграммы вариантов использования, диаграммы взаимодействия, диаграммы классов, диаграммы состояний, диаграммы деятельности, диаграммы компонентов и диаграммы размещения; типы элементов, типы связей и область применения UML-диаграмм каждого вида; общие языковые механизмы UML: стереотипы, примечания, ограничения; введение в метамодель UML)	16	12	0	12	4 (домашнее задание, практическое задание №1)
<i>Тема 3. Объектный язык ограничений</i> (основные понятия OCL; виды ограничений: инвариант, предусловие, постусловие, тело запроса, правило вывода, дополнительное определение, значение по умолчанию; элементы OCL-	8	6	0	6	2 (домашнее задание)

<p>выражений, навигация по ассоциациям, виды OCL-коллекций: Set, OrderedSet, Bag, Sequence; операции над коллекциями: collect (...), select (...), forAll (...), exists (...), isEmpty (...), notEmpty (...), size (...), sum (...), min (...), max (...), count (...), includes (...), excludes (...), includesAll (...), excludesAll (...), including (...), excluding (...), intersection (...), union (...), closure (...), flatten (...), sortBy (...); операции над упорядоченными коллекциями: append (...), prepend (...), insertAt (...), indexOf (...), at (...), first (...), last (...), reverse (...); конструкции let и ifThenElse; операция iterate (...); n-ка и tuple; операция product (...); операции allInstances (...), ofType (...), ofTypeOf (...), ofTypeOf (...); операции приведения типов: ofType (...), ofTypeOf (...), asSet (...), asOrderedSet (...), asBag (...), asSequence (...); примеры OCL-выражений; наследование ограничений)</p>			0	6		
<p><i>Тема 4. Определение требований к программному обеспечению (понятие требования к программному обеспечению; виды требований: функциональные и нефункциональные; процесс определения требований, его цели, содержание, исполнители и рабочие продукты; варианты использования как способ определения функциональных требований; область действия варианта использования; уровень цели варианта использования; текстовое описание варианта использования; виды сценариев вариантов использования: основной, альтернативный, подчинённый; правила составления текстовых описаний вариантов использования и типичные ошибки в описаниях; модель вариантов использования, её элементы, связи, диаграммы; параметризованные варианты использования и варианты использования CRUD; моделирование сценариев вариантов использования диаграммами деятельности)</i></p>	8	6	0	6		2 (практическое задание №2 этап 1)
<p><i>Тема 5. Архитектурный анализ (цели и содержание архитектурного анализа; исполнители и рабочие продукты; соглашения моделирования; аналитические архитектурные механизмы и их использование; идентификация ключевых абстрактной предметной области; формирование начального архитектурного представления; архитектурные стили систем: «уровни», «модель-представление-управление», «каналы и фильтры»)</i></p>	8	6	0	6		2 (практическое задание №2 этап 2 часть 1)
<p><i>Тема 6. Анализ вариантов использования (цели и содержание анализа вариантов использования; исполнители и рабочие продукты; идентификация классов анализа; определение атрибутов и ассоциаций классов анализа, образцы</i></p>	8	6	0	6		2 (практическое задание №2 этап 1 часть 2)

распределения обязанностей между классами по Ларману: Information Expert, Creator, Low Coupling, High Cohesion; образцы распределения обязанностей между классами по Фаулера: «сценарий транзакции» и «модель предметной области»; примеры использования образцов распределения обязанностей; унификация классов анализа)	4	0	0	0	0	4	0	0	0	4	(подготовка к письменной контрольной работе №1)
Текущий контроль успеваемости: письменная контрольная работа № 1	8	6	0	0	6	8	6	0	6	2	(практическое задание №2 этап 2 часть 1)
Тема 7. Проектирование архитектуры системы (цели и содержание проектирования архитектуры системы; проектные механизмы и механизмы реализации; модель проектных механизмов JDBC и XMLRPCInterchange; выявление проектных элементов системы: классов, пакетов, подсистем и интерфейсов; проектирование структуры потоков управления; проектирование конфигурации системы)	8	6	0	0	6	8	6	0	6	2	(практическое задание №2 этап 2 часть 2)
Тема 8. Проектирование элементов системы (цели и содержание проектирования элементов системы; исполнители и рабочие продукты; проектирование вариантов использования; проектирование подсистем на основе проектных механизмов; проектирование классов: детализация проектных классов, уточнение операций классов, моделирование состояний, уточнение атрибутов классов, уточнение связей между классами; выводимые атрибуты и ассоциации; метаморфоза подтипов)	8	6	0	0	6	8	6	0	6	2	(домашнее задание)
Тема 9. Паттерны проектирования (понятие образца/паттерна проектирования; состав описания паттерна; порождающие паттерны из каталога Gang4: абстрактная фабрика, фабричный метод, одиночка; структурные паттерны из каталога Gang4: адаптер класса, адаптер объектов, компоновщик, мост, фасад, заместитель, декоратор; паттерны поведения из каталога Gang4: цепочка обязанностей, итератор, стратегия, наблюдатель; паттерны Фаулера: особый случай, интерфейс удалённого доступа)	8	6	0	0	6	8	6	0	6	2	(практическое задание №2 этап 2 часть 3)
Тема 10. Объектно-реляционное отображение (модель независимое и модель зависимое ORM; отображение классов в таблицы; отображение атрибутов с разными мощностями, выводимых и статических атрибутов; ORM-стратегии отображения бинарных ассоциаций: слияние таблиц, добавление внешнего ключа, отдельная таблица для связи; отображение квалифицированных ассоциаций, n-арных (n > 2) ассоциаций, классов ассоциаций; ORM-стратегии отображения обобщения: единая таблица для всей иерархии, таблицы для	8	6	0	0	6	8	6	0	6	2	(практическое задание №2 этап 2 часть 3)

каждого класса, таблицы только для конкретных классов, таблицы только для различных конкретных классов; моделирование схем баз данных на диаграммах классов; универсальный мэппинг Амблера)								
<i>Тема 11. UnifiedProcess – технология создания программного обеспечения с использованием языков UML и OCL (принципы технологии UnifiedProcess; её общее представление – «диаграмма с горбами»; содержание жизненного цикла создания программного обеспечения согласно технологии UnifiedProcess: стадии, их содержание, контрольные точки, процессы жизненного цикла, их задачи, их содержание, основные рабочие продукты и задействованные исполнители)</i>	8	6	0	6	0	6	2	(подготовка отчёта по практическому заданию №2)
<i>Текущий контроль успеваемости: письменная контрольная работа № 2</i>	4	0	0	0	0	0	4	(подготовка к письменной контрольной работе №2)
Промежуточная аттестация: зачет	4	0	0	0	0	0	4	
Итого	108	72	0	72	0	72	36	

7. Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине (модулю)

7.1. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости

7.1.А. Варианты домашних заданий

Домашнее задание №1

I. Укажите определение жизненного цикла (ЖЦ) программного обеспечения (ПО) по ISO 12207:1995:

a) ЖЦ – это ряд событий, происходящих с ПО в ходе его создания и использования;

b) ЖЦ – это набор технологических процессов, связанных с разработкой ПО;

c) ЖЦ – это период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

II. Отметьте среди перечисленных пунктов реально существующие группы процессов ЖЦ по стандарту ISO 12207:1995 :

a) основные;

b) второстепенные;

c) организационные;

d) вспомогательные;

e) управляющие;

f) технологические.

Домашнее задание №2

I. Отметьте среди перечисленных пунктов реально существующие модели жизненного цикла:

- a) итерированная;
- b) итерационная;
- c) спиральная;
- d) циклическая;
- e) параболическая;
- f) параноидальная;
- g) иррациональная;
- h) эволюционная.

II. Отметьте три основные категории, на которые делят софтверные проекты создатели CHAOS Report:

- a) успешные;
- b) удачные;
- c) завершённые;
- d) неуспешные;
- e) отменённые;
- f) аннулированные;
- g) просроченные;
- h) ущербные;
- i) убыточные.

Домашнее задание №3

I. Билет на электричку продаётся по какому-то Тарифу. Тариф выбирается в зависимости от того, с какой Станции до какой другой Станции будет поездка по Билету.

Станция

Билет

Тариф

I.A. Предложите, как связать классы Билет, Тариф и Станция при помощи только обычных (т. е. бинарных) ассоциаций. Укажите мощности на концах ассоциаций. [Новые классы заводить не следует.]

Станция

Билет

Тариф

I.B. Предложите, как связать классы Билет, Тариф и Станция при помощи бинарной ассоциации с классом ассоциации, и, возможно, дополнительных бинарных ассоциаций. Укажите мощности на концах ассоциаций. [Новые классы заводить не следует.]

I. C. Предложите, как связать классы Билет, Тариф и Станция при помощи n-арной ($n > 2$) ассоциации, и, возможно, дополнительных бинарных ассоциаций. Укажите мощности на концах ассоциаций. [Новые классы заводить не следует.]

Станция

Билет

I. D. Какое из решений I.A-C Вам кажется самым подходящим?

- A) I.A; B) I.B; C) I.C.

Домашнее задание №4

Рисунок состоит из частей, каждая из которых является либо текстом, либо фигурой, либо группой, либо вложенным рисунком. Группа объединяет две или более части рисунка Части, объединённые в группу, упорядочены. Они могут быть непосредственно включены лишь в одну группу. Нарисуйте связи между классами. Явно укажите композиции и агрегации.

Часть

Текст

Рисунок

Домашнее задание №5

Нарисуйте диаграмму состояний, моделирующий цикл ресурса. Ресурс может быть занят в общем доступе, либо занят в монопольном доступе. Сразу после создания ресурс свободен. Свободный ресурс может быть занят пользователем либо в монопольном доступе, либо в общем доступе. Занятый в монопольном доступе ресурс не может быть занят другими пользователями. Пока он не будет освобождён занявшим его пользователем, попытки занять его другими пользователями ведут к неуспеху. Если ресурс занят в общем доступе, то пользователи могут также занимать его в общем доступе, но не в монопольном. Дважды занять один и тот же ресурс один и тот же пользователь не может, если предварительно его не освободил. Ресурс становится свободен лишь тогда, когда его освободит последний пользователь, занявший его. Свободный ресурс может быть уничтожен. Попытка уничтожить занятый ресурс ведёт к неуспеху. На диаграмме должны быть действия по учету пользователей, занимающих ресурс, и проч. Список действий см. ниже.

Составляя диаграмму, используйте только следующие события и действия:

События вызова: `occureSh(user)` – попытка пользователя user занять ресурс в общем доступе; `occureM(user)` – попытка пользователя user занять ресурс в монопольном доступе, `free(user)` – пользователь user освобождает ресурс, `create()` – создание ресурса, `delete()` – уничтожение ресурса.

Тариф

Действия: `success()` – сообщить, что попытка занять, освободить или уничтожить ресурс успешна, `fail()` – сообщить, что попытка занять, освободить или уничтожить ресурс неуспешна, `add2List(user)` – добавить пользователя `user` в список занявших ресурс; `delFromList(user)` – удалить пользователя `user` из списка занявших ресурс.

Также допускается заводить переменные, осуществлять присваивания им значений, использовать их значения в сторожевых условиях.

Функция проверки: `isInList(user)` – проверить, есть ли пользователь `user` в списке занявших ресурс.

Домашнее задание №6

Смоделируйте на диаграмме деятельности поиск n – номера элемента непустого целочисленного массива $M[i]$ из k элементов с нумерацией элементов, начинающейся с 0, такого что $M[n] \leq M[i]$, для всех $M[i]$ с чётным i , $k > 2$ и n – максимальное среди всех возможных. На диаграмме можно использовать управляющие узлы и узлы действий, но не узлы вызова действий. Допустимые действия: присваивания, вывод/возврат найденного n . Моделировать ввод массива не следует.

Домашнее задание №7

I. Отметьте среди перечисленных пунктов не выдуманные уровни цели вариантов использования:

- a) уровень птицы;
- b) уровень рыбы;
- c) уровень морского змея;
- d) уровень морской звезды;
- e) уровень неба;
- f) уровень воздушного змея;
- g) уровень солнца;
- h) уровень моллюска.

II. Отметьте, какие элементы диаграммы вариантов использования могут быть соединены ассоциацией:

- a) два действующих лица;
- b) два варианта использования;
- c) действующее лицо и вариант использования;
- d) вариант использования и рамки системы;
- e) действующее лицо и рамки системы.

7.1.Б. Вариант практического задания №1

Полное описание практического задания №1 дано в методическом пособии «UML-моделирование системы обработки заказов в среде Visual Paradigm 15.2» [HTML] [<http://sp.cs.msu.ru/ooap/exerb2019.html>]. Выполняя задание, студент осуществляет моделирование системы обработки заказов. Задание состоит из последовательности упражнений:

Упражнение №1
Знакомство с CASE-средой Visual Paradigm, настройка среды для дальнейшей работы, создание проекта на основе предложенного шаблона.
Упражнение №2
Моделирование требований к системе. Создание диаграммы вариантов использования и диаграмм деятельности согласно рисункам 3.1.1-3.4.3 из пособия.
Упражнение №3
Архитектурный анализ системы. Создание в проекте аналитической модели (AnalysisModel), создание заготовок реализаций ключевых вариантов использования, создание диаграммы ключевых абстракций согласно рисункам 4.1.1-4.1.3 из пособия.
Упражнение №4
Анализ ключевых вариантов использования системы («CRUDданных о заказах», «Войти в систему»). Создание диаграмм классов VORСв каждой реализации ключевого варианта использования. Создание диаграмм последовательности, моделирующих сценарии ключевых вариантов использования. Упражнение выполняется согласно рисункам 4.2.1-4.2.10 из пособия.
Упражнение №5
Проектирование архитектуры системы. Создание в проекте проектной модели (DesignModel). Разбиение системы на архитектурные уровни. Наполнение архитектурных уровней пакетами и подсистемами. Создание интерфейсов для подсистем. Распределение проектных элементов по пакетам и подсистемам. Установление связей между пакетами и подсистемами. Моделирование распределённой конфигурации системы.
Упражнение выполняется согласно рисункам 5.1.1-5.1.9 из пособия.
Упражнение №6
Проектирование элементов системы. Уточнение реализаций вариантов использования с учётом созданных подсистем. Проектирование подсистемы DBAccessна основе механизма JDBC. Проектирование подсистемы BSAccessна основе механизма XMLRPCInterchange.
Проектирование классов системы. Упражнение выполняется согласно рисункам 5.2.1-5.2.17 из пособия.
Упражнение №7
Проектирование схемы реляционной базы данныхсогласно рисунку 5.3.1 из пособия. Подготовка к сдаче практического задания № 1.

7.1.В. Вариант письменной контрольной работы № 1

Вопросы-тесты

1. Укажите все типы связей, которые могут быть проведены между классом и интерфейсом на диаграмме классов:

- a) реализация;
- b) агрегация;
- c) соединение;
- d) включение;
- e) переход;
- f) обобщение;

- g) расширение;
- h) разделение;
- i) зависимость.

2. Отметьте все, что можно встретить на диаграммах коммуникации (кооперативных диаграммах):

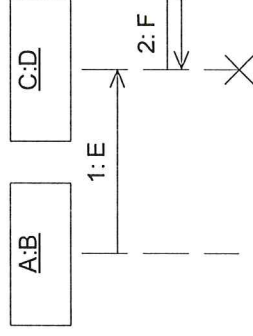
- a) активация;
- b) объект;
- c) триггер;
- d) сообщение;
- e) ассоциация;
- f) зависимость
- g) переход;
- h) соединение;
- i) действие;
- j) сторожевое условие;
- k) принятие решения;
- l) возврат.

3. Отметьте все, что является истинным для приведенной справа диаграммы:

- a) это диаграмма взаимодействия;
- b) это диаграмма объектов;
- c) A и C – объекты;
- d) F – это возврат;
- e) E и F – это вызовы операций класса A;
- f) E и F – это вызовы операций класса B;
- g) E и F – это вызовы операций класса C;
- h) E и F – это вызовы операций класса D.

4. Укажите элементы и связи UML, которые можно встретить на диаграмме вариантов использования:

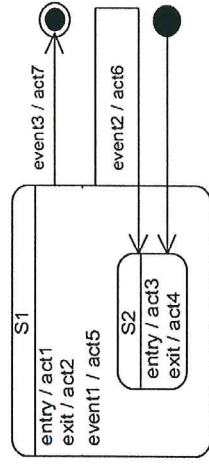
- a) поток управления;
- b) вариант использования;
- c) поток событий;
- d) сценарий;
- e) объект;
- f) действующее лицо;
- g) обобщение;



- h) расширение;
 - i) ассоциация.
5. Укажите роли, з
адействованные в процессе определения требований в технологии RUP:
- a) архитектор;
 - b) писатель вариантов использования;
 - c) исполнитель;
 - d) системный аналитик;
 - e) менеджер проекта;
 - f) системный интегратор;
 - g) технический рецензент;
 - h) рецензент по требованиям.
6. Каково количество всевозможных типов узлов управления, которые используются на диаграммах деятельности:
- a) 3;
 - b) 4;
 - c) 6;
 - d) 7;
 - e) 5;
 - f) >7

7. Укажите связи, которые могут встретиться на диаграмме вариантов использования:

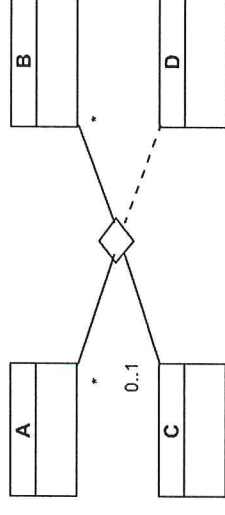
- a) сообщение;
 - b) коммуникация;
 - c) соединение;
 - d) включение;
 - e) переход;
 - f) обобщение;
 - g) расширение;
 - h) разделение;
 - i) поток управления.
8. Отметьте все, что является истинным для приведенной справа диаграммы:
- a) на диаграмме есть внутренний переход;
 - b) если S2 текущее состояние, то по событию event2 осуществляется переход в состояние S2;
 - c) если S2 текущее состояние, то событие event2 игнорируется;



- d) если S2 текущее состояние, то по событию event3 осуществляется переход в финальное состояние;
- e) если S2 текущее состояние, то событие event3 игнорируется;
- f) если S2 текущее состояние, то по событию event1 выполняется действие act5;
- g) если S2 текущее состояние, то событие event1 игнорируется.

9. Укажите верные утверждения о фрагменте диаграммы, приведенном справа:

- a) это фрагмент структурной диаграммы;
- b) это фрагмент поведенческой диаграммы;
- c) это фрагмент диаграммы классов;
- d) это фрагмент диаграммы объектов;
- e) изображена N-арная ассоциация (N = 4);
- f) изображена N-арная ассоциация (N = 3);
- g) изображена агрегация.



10. Каково количество основных процессов жизненного цикла по стандарту ISO 12207:1995?

- a) 8;
- b) 9;
- c) 5;
- d) 6;
- e) 10;
- f) > 10.

Задача 1

Постройте диаграмму классов, имена которых выделены в условии. Для классов выделите в условии. Для классов укажите атрибуты и связи, о которых есть сведения в условии. Явно укажите мощности ассоциаций (в том числе мощность 1 должна быть явно указана!). Создавать дополнительные классы (помимо классов, отмеченных подчеркиванием) не следует.

О любом студенте известны его фамилия, имя и отчество, а также год его поступления в университет. Заканчивая обучение, любой студент сдает госэкзамен по некоторому направлению. Весь список вопросовгосэкзамена разделен на основную и дополнительную части. Вопросы каждой части перенумерованы, начиная с 1. Для любого вопроса определен его текст. Объем и основной, и дополнительной частей не пуст и не превышает 20 вопросов в каждой. Каждый студент на госэкзамене отвечает на билет, который состоит из одного вопроса основной части и одного вопроса дополнительной части. Билетыгосэкзамена различаются по номерам.

Приём госэкзамена у студента осуществляют два профессора. О профессорах известны учёное звание и учёная степень. Каждый из них предлагает свой балл (со значением от 2 до 5) в зависимости от того, доволен ли он ответом студента по билету. Значение итоговой оценкистудента по госэкзамену определяется как округлённое вверх среднее арифметическое значений баллов, предложенных профессорами.

Задача 2

На корпусе чёрного ящика есть две цветные лампы (красная и зелёная), кнопка «вкл/выкл» и два рубильника, различаемые по номерам (первый и второй). Каждый рубильник может находиться либо в положении «включён», либо в положении «выключен». В зависимости от положения рубильников и от того, включено ли (или отключено) питание, лампы чёрного ящика либо горят, либо потушены. Действуют следующие правила:

1) Если питание отключено, то ни одна лампа не горит.

2) Всегда, когда включён первый рубильник и есть питание, горит зелёная лампа. Если первый рубильник выключен, зелёная лампа потушена.

3) Только тогда, когда есть питание и включены первый и второй рубильники, горит красная лампа.

Считайте, что в начальном состоянии питание включается, и лампы ящика начинают гореть по указанным правилам. Отключить питание можно нажатием на кнопку «вкл/выкл» питания включается, и лампы ящика начинают гореть по указанным правилам. Отключить питание можно повторным нажатием на «вкл/выкл», либо оно отключается само через 15 минут после включения.

Нарисуйте диаграмму состояний, моделирующую чёрный ящик, считая, что любые события происходят последовательно. Следует показать события, сторожевые условия и действия на переходах. Набор событий: *onOff* – нажатие на кнопку «вкл/выкл»; *1stOn* – включение первого рубильника; *1stOff* – выключение первого рубильника; *2ndOn* – включение второго рубильника; *2ndOff* – выключение второго рубильника; *after(n мин)* – событие времени, *n* – его параметр. Можно заводить переменные и связывать с их значениями *события изменения*.

Допустимые действия: *ItRd* – зажечь красную лампу; *bkRd* – потушить красную лампу; *ItGrn* – зажечь зелёную лампу; *bkGrn* – потушить зелёную лампу. При необходимости допускается *использование переменных, событий изменения и действий по присваиванию значений переменным*. Недопустимы никакие другие действия, в том числе условные: *if b then a1* и циклы! Избегайте повторных действий вроде гашения не светящихся ламп, включения уже горящих ламп и т. п.. За каждое повторное действие будут снижаться баллы.

7.2. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации

7.2.А. Требования к составлению отчёта по практическому заданию № 2

Отчет по заданию пишется на русском языке, предоставляется в электронном виде преподавателю (верстка в формат А4, "портрет", pdf, doc или odt). Отчет состоит из следующих частей:

- Титульный лист, с «шапкой» – «Московский государственный университет имени М. В. Ломоносова, факультет Вычислительной математики и кибернетики». Далее следует заголовок: «Отчёт по объектно-ориентированному анализу», тема задания, сведения об исполнителе (фамилия, имя и отчество полностью, номер группы) и преподавателе, принявшем задание. Внизу титульного листа указывается город и год. Нелишне обратить внимание на то, что точки после заголовков не ставятся.
- Содержание состоит из перечня названий глав и подглав, сопровождаемых указанием номеров страниц, с которых они начинаются. Нумеруются все страницы, за исключением титульного листа. Номер страницы с содержанием: 2.
- Первая глава, названная «Постановка задачи» содержит формулировку задания. Каждую главу следует начинать с новой страницы.

- Вторая глава, названная «Определение требований» содержит глоссарий, UML-диаграмму вариантов использования, описания действующих лиц и ключевых вариантов использования. Для одного из ключевых вариантов использования приводится UML-диаграмма деятельности.
- Третья глава, названная «Анализ вариантов использования» содержит UML-диаграмму классов Key Abstractions, UML-диаграммы последовательности, описывающие взаимодействия между объектами в рамках потоков событий трёх ключевых вариантов использования, UML-диаграммы классов VOPC трёх ключевых ВИ. UML-диаграммы следует сопроводить пояснениями, указывающими, какому потоку событий они соответствуют (если это не ясно из их названия), и комментариями об объектах (классах), присутствующих на диаграммах.
- Четвёртая глава, названная «Проектирование архитектуры системы» содержит UML-диаграмму с указанием уровней системы, перечень пакетов и подсистем с указанием их назначения, UML-диаграммы пакетов для уровней, состоящих из более чем одного пакета. Также приводится UML-диаграмма размещения с пояснениями о каждом узле на ней. Для встроенных систем диаграмма размещения должна изображать связи между процессором и устройствами, а также узлами внешних систем. В остальных вариантах диаграмма размещения показывает узлы вычислительной среды системы, узлы внешних систем, связи между ними и размещение процессов разрабатываемой системы по узлам.
- Пятая глава, названная «Проектирование элементов системы» содержит описание проектных классов системы (всех ключевых абстракций и всех классов, участвующих в смоделированных реализациях вариантов использования), сгруппированных по пакетам. Сведения о классе включают в себя: краткое описание – ответственность класса; описание атрибутов и операций в виде таблицы из 2-х столбцов: полная сигнатура атрибута или операции, его или её назначение. Также приводятся UML-диаграммы проектных классов системы, отображающие связи между классами, диаграммы состояний (не менее 1), описывающие поведение экземпляров отдельных классов (как правило, классов-контроллеров или классов-сущностей), и диаграммы деятельности реализации реализации нетривиальных операций классов. Нетривиальной считается реализация с логическим ветвлением потока и циклом. Для одной смоделированной подсистемы приводится описание её интерфейса (полные сигнатуры операций и описания), диаграмма классов подсистемы (вид подсистемы изнутри) и диаграммы последовательности, описывающие реализацию операций интерфейса подсистемы (достаточно описать 3–4 реализации разнородных операций, если в интерфейсе их больше). Во всех вариантах UML-диаграмма классов, моделирующая схему БД, также должна быть включена в отчёт.
- Если выполнялось моделирование дополнительных ВИ (дополнительной подсистемы), то сведения о нём могут быть включены в отчёт по усмотрению исполнителя задания. Если сведения будут включены, то могут быть начислены дополнительные баллы за это.
- Заключение (оно не нумеруется), в котором подводится общий итог работы, завершает отчёт. В заключении можно указать характеристики трудоёмкости созданной модели: общее количество вариантов использования; количество проектных классов и подсистем; количество UML-диаграмм разных видов.

7.2.Б. Вариант письменной контрольной работы № 2

Тесты

1. Укажите процессы жизненного цикла по стандарту ISO 12207:1995, не являющиеся организационными:

- a) обучение;
- b) документирование;
- c) управление;
- d) приобретение;
- e) усовершенствование;
- f) создание инфраструктуры.

II. Добавление в модель действующих лиц (акторов) происходит:

- a) при анализе вариантов использования;
- b) при архитектурном анализе;
- c) при определении требований;
- d) при проектировании архитектуры системы.

III. Укажите те виды поведенческих UML-диаграмм, которые могут содержать в архитектурном представлении варианты использования в модели «4+1»:

- a) диаграммы вариантов использования;
- b) диаграммы действующих лиц;
- c) диаграммы классов;
- d) диаграммы деятельности;
- e) диаграммы действий;
- f) диаграммы составной структуры.

IV. На каких UML-диаграммах могут быть элементы и/или связи, в обозначении которых используется подчёркнутый текст как специальное обозначение UML:

- a) на диаграммах последовательности;
- b) на диаграммах классов;
- c) на коммуникационных диаграммах;
- d) на диаграммах объектов;
- e) на диаграммах вариантов использования;
- f) на диаграммах Венна.

V. Связь включения (include) может связывать между собой:

- a) два действующих лица;
- b) два интерфейса;
- c) два варианта использования;
- d) два узла действия;
- e) два класса;

f) два состояния.

VI. Укажите согласующиеся с диаграммой классов сочетания количеств экземпляров классов A, B, C и D. Считайте, что $|A|$ – это количество экземпляров класса A, $|B|$ – B, ...:

- a) $|A|=1, |B|=4, |C|=3, |D|=3$;
- b) $|A|=4, |B|=3, |C|=3, |D|=5$;
- c) $|A|=4, |B|=1, |C|=3, |D|=4$;
- d) $|A|=4, |B|=2, |C|=3, |D|=5$.

VII. Укажите всё то, что **не является** принципами объектной модели по Бучу:

- a) иерархия;
- b) полиморфизм;
- c) параллелизм;
- d) устойчивость;
- e) инкапсуляция;
- f) модульность;
- g) абстрагирование;
- h) наследование.

VIII. Назначение приоритетов вариантам использования осуществляет:

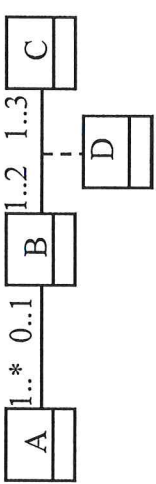
- a) архитектор системы;
- b) рецензент по требованиям;
- c) системный аналитик;
- d) бизнес-аналитик;
- e) заказчик;
- f) менеджер проекта.

IX. Укажите те виды UML-диаграмм, которые **не могут** содержаться в представлении вариантов использования в модели «4+1»:

- a) диаграммы компонентов;
- b) диаграммы деятельности;
- c) диаграммы размещения;
- d) диаграммы вариантов использования;
- e) диаграммы Венна.

X. Укажите выражения OCL, результатом вычисления которых является **истина**:

- a) $(\text{Set } \{1, 2, 1, 2\} \rightarrow \text{size}()) = (\text{Bag } \{1, 2, 1, 2\} \rightarrow \text{count}(1))$
- b) $(\text{Set } \{1, 2, 1, 2\} \rightarrow \text{count}(1)) = (\text{Bag } \{1, 2, 1, 2\} \rightarrow \text{size}())$
- c) $(\text{Set } \{1, 2, 1, 2\} \rightarrow \text{count}(1)) = (\text{Bag } \{1, 2, 1, 2\} \rightarrow \text{asSet}() \rightarrow \text{size}())$



d) \square (Set {1, 2, 1, 2} -> count(1)) \Leftrightarrow (Bag {1, 2, 1, 2} -> asSet() -> size())

Задача 1

Постройте диаграмму классов. В условии выделены имена классов и/или перечислимых типов. Для классов укажите атрибуты, связи, ограничения, для перечислимых типов – литералы, о которых есть сведения в условии. Полусам ассоциацией укажите мощности (в том числе мощность 1 должна быть явно указана!). Явно укажите агрегации, композиции, квалификации ассоциаций, если есть. Проведите связи к перечислимым типам, помещённым Вами на диаграмму. Создавать дополнительные классы (помимо классов, отмеченных подчеркиванием) не следует.

О каждой книге известен список имён её авторов (от одного до пяти), название издательства, год издания. Структура книги составлена из частей текста, образующих иерархию. У любой части известен порядковый номер, заголовок и количество страниц в ней. Самые крупные части – это тома. В любой книге их один или более. У каждого тома есть предисловие. Любой том в свою очередь делится на одну или несколько более мелких частей – на главы. Для каждой главы тома может быть указан номер страницы, с которой она начинается. Разные главы одного тома не могут начинаться с одной и той же страницы. Крупная глава может быть разделена на самые мелкие части текста – на параграфы. Для каждого параграфа может быть указан номер страницы, с которого он начинается. Разные параграфы одной главы могут начинаться на одной странице. Любая книга относится к одному из жанров: детектив, нонфикшен, фэнтези, учебник.

Задача 2

Траволатор служит для передвижения пассажиров в прямом и в обратном направлении. Когда траволатор выключен, его ступени неподвижны, все индикаторы выключены. Если траволатор включить, по умолчанию, он переходит в режим ожидания. В этом режиме горит индикатор «Жду», и траволатор ждёт сигнал либо от датчика прямого хода, либо от датчика обратного хода. Если приходит сигнал от датчика прямого хода, то загорается индикатор «Прямо», гаснет индикатор «Жду», и лента траволатора начинает двигаться в прямом направлении. Движение ленты и индикация продолжаются в течение 300 секунд с момента прихода сигнала от датчика прямого хода, затем прекращаются. Если за это время датчик прямого хода сработает ещё раз, траволатор продолжает работу в прямом ходе в течение ещё 300 секунд, отсчитывая время от момента последнего прихода сигнала. Во время работы в прямом ходе сигналы от датчика обратного хода игнорируются.

Работа в обратном ходе организована похоже. Если в режиме ожидания приходит сигнал от датчика обратного хода, загорается индикатор «Обратно», гаснет индикатор «Жду», и лента траволатора начинает двигаться в обратном направлении. Обратный ход ленты и индикация продолжаются в течение 300 секунд с момента прихода сигнала от датчика обратного хода, затем прекращаются. Если за это время датчик обратного хода сработает ещё раз, траволатор продолжает работать в обратном направлении в течение ещё 300 секунд, отсчитывая время от момента последнего срабатывания датчика обратного хода. Во время обратного хода сигналы от датчика прямого хода игнорируются.

По окончании работы в любом направлении (прямом или обратном), траволатор тушит соответствующий индикатор, останавливает ленту и зажигает индикатор «Жду». Выключить траволатор можно только в режиме ожидания.

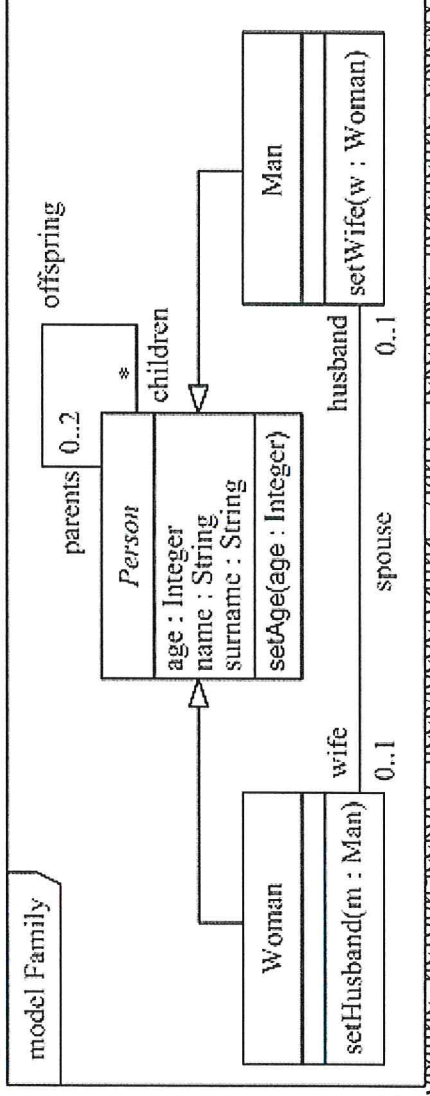
Для предотвращения несчастных случаев в траволатор встроены переключатель для аварийной остановки. При работе в любом направлении (прямом или обратном) поворот этого переключателя временно останавливает движение. При обратном его переключении траволатор

возвращается в тот режим, в котором был остановлен, возобновляет движение и действует так, как если бы получил сигнал от датчика хода (для отсчитывания 300 секунд).

Нарисуйте диаграмму состояний, описывающую траволатор, считая, что любые события происходят последовательно. Следует показать события, сторожевые условия и действия на переходах. Набор допустимых событий: *forward* – сигнал датчика прямого хода, *backward* – сигнал датчика обратного хода; *onOff* – включение/выключение траволатора; *raise* – поворот аварийного переключателя; *after(n s)* – событие времени с любым нужным Вам *n*; *when(exp)* – событие изменения с любым нужным Вам логическим выражением относительно переменных, заведённых Вами. Можно заводить переменные, менять их значения, использовать их в сторожевых условиях, событиях изменений. Необходимые действия: *gotoFw* – включить прямой ход ленты; *gotoBw* – включить обратный ход ленты; *stopFw* – выключить прямой ход ленты; *stopBw* – выключить обратный ход ленты; *lampW* – зажечь «Жду»; *blankW* – погасить «Жду»; *lampFw* – зажечь «Прямо»; *blankFw* – погасить «Прямо»; *lampBw* – зажечь «Обратно»; *blankBw* – погасить «Обратно». Допускаются действия по присваиванию значений переменным. Недопустимы другие события, другие действия (в том числе условные, циклы, вычисление условных выражений и т. п.)! Избегайте повторных действий вроде гашения ранее погашенного индикатора и/или включения движения уже движущейся ленты. За каждое запрещённое событие, запрещённое или избыточное действие будут снижаться баллы. То же относится к избыточным действиям.

Вопрос 1. Какую задачу помогает решить образец «Адаптер» («Adapter»)? Опишите словами предлагаемое этим образцом решение в случае адаптера объекта. Приведите диаграмму классов и диаграмму взаимодействия объекта «Адаптер».

Вопрос 2. Ниже дана диаграмма классов. Осуществите её отображение в реляционную модель. Представьте предлагаемую Вами схему в виде диаграммы классов со стереотипами из профиля для моделирования баз данных. Для каждой таблицы на диаграмме должно быть указано её имя, список имён её столбцов, её первичный ключ и её внешние ключи, если таковые есть. Опишите, какие стратегии объектов-реляционного отображения Вы применили, и в чём они состоят. Обратите внимание: имя класса *Person* указано курсивом.



Вопрос 3. Расскажите про историю использования исторического псевдосостояния в каком-то другом состоянии? пример диаграммы состояний, на которой уместно использовать историческое псевдосостояние. Дайте название состоянию своему примеру. Что обозначает на диаграммах состояний переход из исторического псевдосостояния в какое-то другое состояние?

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине (модулю)

Оценка	«незачёт»	«зачёт»
<p>РО и соответствующие виды оценочных средств</p> <p>Знания (тесты и вопросы в составе контрольных работ; ответы на вопросы при сдаче практического задания №1)</p> <p>Умения (задачи в составе контрольных работ, зачет, оформление отчёта по практическому заданию №2)</p> <p>Навыки (владения, опыт деятельности) (выполнение практического задания №2 по индивидуальному варианту)</p>	<p>отсутствие знаний или фрагментарные знания</p> <p>отсутствие умений или не вполне успешное, не систематическое умение</p> <p>отсутствие навыков (владений, опыта) или наличие отдельных навыков (наличие фрагментарного опыта)</p>	<p>общие, но не структурированные знания или сформированные систематические знания</p> <p>в целом успешное, но содержащее отдельные пробелы умение (с неточностями непринципиального характера) или успешное и систематическое умение</p> <p>в целом, сформированные навыки (владения), используемые не в активной форме или вполне сформированные навыки (владения)</p>
Соответствие результатов обучения и компетенций, в развитии которых участвует дисциплина (модуль)		
<p>Результаты обучения</p>		<p>Компетенция, с частичным формированием которой связано достижение результата обучения</p>
<p>Знать:</p> <ol style="list-style-type: none"> 1) принципы объектно-ориентированного подхода к анализу и проектированию программного обеспечения; 2) графическую нотацию языка UML; 3) текстовую нотацию языка OCL; 4) методы использования языков UML и OCL в разработке программного обеспечения. <p>Уметь:</p> <ol style="list-style-type: none"> 1) моделировать структурные и поведенческие аспекты программных систем с 		<p align="center">ПК-7.Б</p>

	<p>помощью диаграмм UML;</p> <p>2) формально описывать ограничения на языке OCL в моделях программного обеспечения;</p> <p>3) применять CASE-средства с поддержкой UML в технологическом цикле создания программного обеспечения.</p> <p>Владеть:</p> <p>1) навыками анализа и проектирования в соответствии с современной технологией разработки программного обеспечения, использующей языки UML и OCL.</p>
--	--

8. Ресурсное обеспечение:

Основная литература:

1. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка. 2-е изд.: Пер. с англ. - СПб.: Питер, 2006.
2. Буч Г., Якобсон И., Рамбо Дж. UML. Классика CS. 2-е изд.: Пер. с англ. - СПб.: Питер, 2006.
3. Малышко В. В. UML-моделирование системы обработки заказов в среде Visual Paradigm 15.2 - М: ВМК МГУ, 2019. - [HTML] (<http://sp.cs.msu.ru/ooap/exerb2019.html>).

Дополнительная литература:

1. Буч Г., Рамбо Дж., И. Якобсон И. Введение в UML от создателей языка.: Пер. с англ. - М.: ДМК Пресс, 2015.
2. Гамма Э., Хэлм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования.: Пер. с англ. - СПб.: Питер, 2016.
3. Ларман К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку.: Пер. с англ. - М.: Вильямс, 2013.
4. Фаулер М. UML. Основы. 3-е издание. Краткое руководство по стандартному языку объектного моделирования.: Пер. с англ. - СПб: Символ-Плюс, 2011.
5. Фримен Эр., Фримен Эл., Сьерра К. и др. Паттерны проектирования.: Пер. с англ. - СПб.: Питер, 2016.
6. Арлоу Дж., Нейштадт А. UML 2 и унифицированный процесс. Практический объектно-ориентированный анализ и проектирование: Пер. с англ. - СПб.: Символ-Плюс, 2007.

Информационные справочные ресурсы:

1. <http://sp.cs.msu.ru/ooap> – веб-страница в поддержку обучения по дисциплине
2. https://vk.com/cmc_ooad – группа в социальной сети Вконтакте в поддержку обучения по дисциплине

3. <http://www.uml.org> – сайт ObjectManagementGroup с публикациями стандартов языков UML и OCL
4. <https://www.uml-diagrams.org> – справочник по стандарту UML 2.5
5. <https://www.visual-paradigm.com/download/community.jsp> – Страница загрузки актуальной версии CASE-среды Visual Paradigm Community Edition

Материально-техническое обеспечение:

Для лекционных занятий требуется аудитория, оборудованная мультимедийным проектором и экраном. Для самостоятельной работы студентов требуется персональный компьютер с подключением к сети Internet, на который следует установить CASE-среду, поддерживающую работу с языками UML и OCL (в течение ряда последних лет предлагается среда Visual Paradigm Community Edition, использование которой в целях обучения бесплатно). Для доступа к учебным материалам в WWW требуется установить веб-браузер. Для просмотра материалов в формате PDF требуется установить программное обеспечение Adobe Reader. Для архивации/разархивации файлов проектов по моделированию требуется установить программное обеспечение 7-zip. Для подготовки отчёта по практическому заданию требуется установить программное обеспечение Apache OpenOffice.

9. Язык преподавания – русский.
10. Преподаватели: доцент факультета ВМК МГУ В. В. Малышко
11. Авторы программы: доцент факультета ВМК МГУ В. В. Малышко