

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Андрей Драгомирович Хлутков
Должность: директор
Дата подписания: 02.12.2024 23:48:09
Уникальный программный ключ:
880f7c07c583b07b775f6604a630281b15ca9fd2

1

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

Северо-Западный институт управления – филиал РАНХиГС

Кафедра бизнес-информатики
(наименование кафедры)

УТВЕРЖДЕНО
Директор СЗИУ РАНХиГС
А.Д.Хлутков

**ПРОГРАММА БАКАЛАВРИАТА
«Бизнес-аналитика»**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
реализуемой без применения электронного (онлайн) курса**

Б1.О.14 Объективно-ориентированный анализ и программирование
(индекс, наименование практики (научно-исследовательской работы), в соответствии с учебным планом)

38.03.05 Бизнес-информатика
(код, наименование направления подготовки)

очная
(форма обучения)

Год набора – 2024

Санкт-Петербург, 2023 г.

Автор–составитель:

доцент кафедры бизнес-информатики Рассказов Владимир Александрович

Заведующий кафедрой бизнес-информатика

д.в.н., профессор

Наумов Владимир Николаевич

РПД по дисциплине Б1.О.14 Объективно-ориентированный анализ и программирование одобрена на заседании кафедры бизнес-информатики. Протокол от 04.07.2022г. №9

В новой редакции РПД одобрена на заседании кафедры бизнес-информатики. Протокол от 27.06.2024 г. № 10.

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы
2. Объем и место дисциплины в структуре образовательной программы
3. Содержание и структура дисциплины
4. Материалы текущего контроля успеваемости обучающихся и фонд оценочных средств промежуточной аттестации по дисциплине
5. Оценочные материалы промежуточной аттестации по дисциплине
6. Методические материалы по освоению дисциплины
7. Учебная литература и ресурсы информационно-телекоммуникационной сети "Интернет", учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине
 - 7.1. Основная литература
 - 7.2. Дополнительная литература
 - 7.3. Учебно-методическое обеспечение самостоятельной работы
 - 7.4. Нормативные правовые документы
 - 7.5. Интернет-ресурсы
 - 7.6. Иные источники
8. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения программы

1.1. Дисциплина «Объектно ориентированный анализ и программирование» обеспечивает овладение следующими компетенциями:

Таблица 1.1

Код компетенции	Наименование компетенции	Код Компонента компетенции	Наименование компонента компетенции
ОПК-3	Способность управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3.1	Способность разрабатывать алгоритмы и программы, проектирует базы данных с целью использования на практике основных методов управления процессами создания продуктов и услуг ИКТ.
		ОПК-3.2	Способность разрабатывать и программы с целью использования на практике основных принципов и методов управления процессами создания продуктов и услуг в сфере информационно-коммуникационных технологий

В результате освоения дисциплины у студентов должны быть сформированы:

Таблица 1.2

ОТФ/ТФ (при наличии профстандарта)/ профессиональные действия	Код компонента компетенции	Результаты обучения
Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы.	ОПК-3.1	на уровне знаний: – основные понятия и методы программирования; – основные синтаксические конструкции, основные структурами данных и типовые методы обработки этих структур. на уровне умений: – разрабатывать и реализовывать алгоритмы на языке высокого уровня; – кодировать на языках программирования, тестировать результаты кодирования, выполнять отладку программ – разрабатывать программное обеспечение с использованием современных инструментальных средств программирования.
Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ОПК-3.2	на уровне знаний: – инструментальные системы, используемые для описания и анализа бизнес-процессов. – теоретические и практические основы технологии объектно-ориентированного анализа и программирования. на уровне умений: – проводить анализ предметной области, выявлять информационные потребности и разрабатывать требования к ИС – разрабатывать программное обеспечение с использованием современных инструментальных средств программирования

2. Объем и место дисциплины в структуре ОП ВО

Объем дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы /144 академ. часов.

Дисциплина реализуется частично с применением дистанционных образовательных технологий (далее – ДОТ).

Доступ к системе дистанционных образовательных технологий осуществляется каждым обучающимся самостоятельно с любого устройства на портале: <https://lms.ranepa.ru/>. Пароль и логин к личному кабинету / профилю предоставляется студенту в деканате.

Таблица 2

Вид работы	Трудоемкость в акад. часах ауд./ЭО, ДОТ	Трудоемкость в астрон. часах ауд./ЭО, ДОТ
Общая трудоемкость	144	108
Контактная работа с преподавателем	50	38
Лекции	20	15
Практические занятия	28	21
Лабораторные занятия		
Практическая подготовка		
Самостоятельная работа	94	70,5
Контроль		
Формы текущего контроля	УО/ПКЗ/Т	
Форма промежуточной аттестации	<i>Зачет с оценкой</i>	

Место дисциплины в структуре ОП ВО

Дисциплина Б1.О.14 «Объективно-ориентированный - анализ и программирование» относится к числу обязательных дисциплин вариативной части учебного плана по направлению «Бизнес-информатика».38.03.05 Преподавание дисциплины «Объектно-ориентированный - анализ и программирование» основано на дисциплине – Б1.О.12 «Программирование». В свою очередь она создаёт необходимые предпосылки для освоения программ таких дисциплин, как Б1.В.ДВ.07.01 «Сетевые технологии», Б1.В.11 «Проектирование и разработка web-приложений», Б1.О.22 «Анализ и моделирование бизнес-процессов» и ряда дисциплин по выбору студента.

Дисциплина изучается в 4 семестре 2-го курса обучения.

3. Содержание и структура дисциплины

Таблица 3

№ п/п	Наименование тем	Объем дисциплины, час.						Форма текущего контроля успеваемости**, промежуточной аттестации* **
		Всего	Контактная работа обучающихся с преподавателем по видам учебных занятий			СР (ЭК)		
			Л	ПЗ	КСР	СРО	СП	
Тема 1	Объектно-ориентированный анализ и шаблоны проектирования	36	4	4		28		УО/ПКЗ/Т*
Тема 2	Базовые алгоритмы программирования	50	8	10		32		УО/ПКЗ/Т*
Тема 3.	Разработка мобильных приложений на C#	56	8	14		34		УО/ПКЗ/Т*
Контроль								
Промежуточная аттестация (4 семестр)					2			Зачет с оценкой
Всего (акад./астр. часы):		144/108	20/15	28/21	2/1,5	94/70,5		

Консультация к зачету с оценкой – 2 часа

Т – тестирование;

ПКЗ – выполнение практического контрольного задания;

УО – устный опрос.

Применяемые на занятиях формы интерактивной работы:

- Лекция-визуализация - передача преподавателем информации студентам сопровождается показом различных рисунков, структурно-логических схем, диаграмм, использование среды разработки;

В процессе освоения данной учебной дисциплины используются следующие образовательные технологии:

Лекционные занятия:

- сопровождаются демонстрацией слайдов, подготовленных в среде MS PowerPoint;
- сопровождаются демонстрацией приёмов работы в изучаемых средах программирования;
- сопровождаются элементами дискуссии по рассматриваемым вопросам.
- Практические занятия выполняются в компьютерных классах:
- направлены на закрепление полученных теоретических знаний;
- включают анализ полученных результатов и способов его достижения;
- сопровождаются элементами дискуссии;
- завершается занятие защитой работы.

Для лекционных и практических занятий используются мультимедийное обеспечение, современное компьютерное оснащение. В аудиториях наличие локальной вычислительной сети института и глобальной сети Интернет, лицензионное программное обеспечение.

Применяемые на занятиях формы интерактивной работы:

- Лекция-визуализация - передача преподавателем информации студентам сопровождается показом различных рисунков, структурно-логических схем, диаграмм, использование среды разработки;

В процессе освоения данной учебной дисциплины используются следующие образовательные технологии:

Лекционные занятия:

- сопровождаются демонстрацией слайдов, подготовленных в среде MS PowerPoint;
- сопровождаются демонстрацией приёмов работы в изучаемых средах программирования;
- сопровождаются элементами дискуссии по рассматриваемым вопросам.
- Практические занятия выполняются в компьютерных классах:
- направлены на закрепление полученных теоретических знаний;
- включают анализ полученных результатов и способов его достижения;
- сопровождаются элементами дискуссии;
- завершается занятие защитой работы.

Для лекционных и практических занятий используются мультимедийное обеспечение, современное компьютерное оснащение. В аудиториях наличие локальной вычислительной сети института и глобальной сети Интернет, лицензионное программное обеспечение.

Содержание дисциплины

Тема 1. Объектно-ориентированный анализ и шаблоны проектирования

Введение в объектно-ориентированное проектирование. Основные сведения о *UML*. Аксиомы *UP*. Определение требований. Аналитическая модель. Объекты и классы. Отношения между классами.

Проектная модель. Диаграммы взаимодействий. Диаграммы деятельности. Агрегация и композиция. Интерфейсы и диаграмма компонентов. Конечные автоматы и диаграмма состояний. Обзор *CASE*-средств проектирования *UML*.

Определение примитивов взаимодействия объектов. Рекурсия. Расширение метода. Переадресация. Переадресованная рекурсия. Доверенная переадресация. Одноуровневая переадресация. Делегирование. Примитив «Вернуть метод». Конгломерация. Делегированная конгломерация. Доверенное делегирование. Одноуровневое делегирование.

Элементарные шаблоны проектирования. Паттерны поведения. Шаблон «Стратегия». Паттерн Шаблонный метод. Паттерн Посредник. Паттерн «Итератор». Паттерн Наблюдатель. Паттерн Посетитель.

Порождающие паттерны. Паттерн «Синглтон». Шаблон «Абстрактная фабрика». Паттерн «Фабричный метод». Паттерн «Строитель». Структурные паттерны. Паттерн «Адаптер объектов». Паттерн «Адаптер классов». Паттерн «Фасад». Паттерн «Декоратор». Паттерн «Компоновщик».

Тема 2. Базовые алгоритмы программирования

Деревья, понятия и определения. Основные алгоритмы работы с деревьями.

Основные понятия и определения. Основные операции с бинарными деревьями. Упорядоченные деревья. Поиск по дереву с включением. Удаление из упорядоченного дерева. Пирамидальная сортировка. Основы работы интерпретатора. Пример интерпретатора

Основные алгоритмы работы с графами. Основные определения теории графов. Структура стек для обработки графов. Структура данных для представления графов. Запись и чтение графов. Поиск в глубину. Поиск в ширину. Поиск кратчайшие пути в графе.

Волновой алгоритм. Алгоритм Дейкстры. Алгоритм Форда-Мура-Беллмана. Циклы на графах. Эйлеровы циклы, алгоритм построения. Гамильтонов цикл. Алгоритмы с возвратом. Гамильтоновы циклы и задача коммивояжера. Комбинаторные задачи на графах. Алгоритмы, основанные на связности графа.

Прикладные алгоритмы. Алгоритм поиска простых чисел: наивный алгоритм и решето Эратосфена. Алгоритмы поиска простых чисел НОД и НОК. Алгоритмы поиска чисел Фибоначчи. Алгоритмы проверки на палиндром. Алгоритмы поиска символа, подмножества символов, подстроки. Алгоритм Кнута–Морриса–Пратта.

Алгоритмы численных методов. Алгоритм решения системы линейных уравнений методом Гаусса. Алгоритмы приближенного вычисления производных. Алгоритмы приближенного вычисления интегралов. Алгоритмы приближенного решения линейных дифференциальных уравнений.

Тема 3. Объектно-ориентированное программирование, разработка мобильных приложений

Работа с *Activity*. Управление жизненным циклом. Атрибут [*Activity*] и основные свойства активности. Намерения и передача данных между *Activity*. Получение результата из *Activity*. Сохранение настроек *Activity*. Взаимодействие между *Activity*. Диалоговые *Activity*

Выбор и использование управляющих компонентов. *Layouts* в *Android*. Контейнер *ScrollView*. Элемент *TextView*. Элемент *EditText*. Элемент *Button*. Элемент *Checkbox*. Элемент *RadioButton*. Элемент *SeekBar*. Элемент *ImageView*. Анимация свойств. Адаптеры и списки. Выбор даты и времени.

Создание меню. Элемент *Toolbar*. Контекстная панель действий. Введение во фрагменты. Элемент *SwipeRefreshLayout*. Перелистывание страниц и *ViewPager*.

Запись и чтение данных. Чтение и сохранение файлов. Основы работы с *SQLite*. Добавление, удаление и обновление данных в *SQLite*. Динамический поиск по базе данных *SQLite*. Введение в веб-сервисы.

Использование основных функций *Android*. Телефонные звонки. Отправка СМС. Работа с контактами. Определение местоположения. Захват изображения и работа с камерой. Распознавание речи. Разработка экранных виджетов. Сборка, распространение и монетизация приложений.

4. Материалы текущего контроля успеваемости обучающихся

4.1. Формы и методы текущего контроля успеваемости обучающихся

В ходе реализации дисциплины «Объективно ориентированный анализ и программирование» используются следующие методы текущего контроля успеваемости обучающихся

Таблица 4.1

Тема	Формы (методы) текущего контроля успеваемости
Тема 1. Объектно-ориентированный анализ и шаблоны проектирования	Устный опрос, тестирование, ПКЗ
Тема 2 Базовые алгоритмы программирования	Устный опрос, тестирование, ПКЗ
Тема 3 Объектно ориентированное программирование, разработка мобильных приложений	Устный опрос, тестирование, ПКЗ

4. 2. Материалы текущего контроля успеваемости обучающихся.

4.2.1 Типовые оценочные материалы

4.2.1.1 Примеры типовых заданий для практических работ.

Тема 1. Основы объектно-ориентированного анализа

Практическая работа 1. Основы объектно-ориентированного анализа

Руководителем банка поставлена задача аналитическому отделу разработать инструментальное средство (программу), которое хранит сведения о клиентах банка, имеющих кредитные карты и рассчитывает текущую задолженность.

Необходимо произвести объектно-ориентированный анализ и проектирование разрабатываемого инструментального программного средства путем построения следующих *UML* диаграмм: диаграмма прецедентов (со спецификацией каждого прецедента), диаграмма классов, диаграмма последовательностей, диаграммы деятельности (по необходимости для различных методов), диаграмма состояний.

Исходные данные.

Кредитная карта имеет лимит (S , рублей), которую клиент банка может использовать по своему усмотрению. Кредитная карта имеет определенную длительность беспроцентного периода (T , дней) льготной процентной ставкой в беспроцентный период (k_1 , %). Длительность льготного периода для каждой покупки индивидуальная, исчисляется с даты покупки. По окончании беспроцентного периода, процентная ставка увеличивается и составляет (k_2 , %).

На первое число каждого текущего месяца предусмотрена обязательная выплата составляющая (p , %) долга. Обязательная должна быть внесена клиентами до последнего числа текущего месяца. Если обязательная выплата не произведена клиентом, то процентная ставка повышается и составляет (k_3 , %).

Программа должна предусматривать ввод, вывод (на экран и в файл), хранение и корректуру данных клиентов (Id – идентификационный номер клиента, ФИО, номер телефона, адрес, данные кредитной карты, сведения по задолженности), производить расчет следующих параметров:

- 1) Величину долга клиента (сумма основного долга и процентной выплаты), величину процентной выплаты, величину основного долга на текущий месяц.
- 2) Величину долга клиента (сумма основного долга и процентной выплаты), величину процентной выплаты, величину основного долга на любой выбранный месяц с учетом ежемесячной выплаты клиентом (M , рублей).

Тема 2. Базовые алгоритмы программирования

Практическое контрольное задание 2. Реализация шейкерной сортировки.

Создать класс, содержащий метод *CocktailSort*, реализующий алгоритм шейкерной (коктейльной) сортировки. Шейкерная (коктейльная) сортировка - модифицированный и немного улучшенный алгоритм пузырьковой сортировки. Начинается с прохода слева направо, как стандартная пузырьковая сортировка, сравнивая каждую пару соседних элементов и, при необходимости, меняя их местами. После достижения конца массива, сортировка меняет направление и выполняет проход справа налево, снова сравнивая и меняя элементы. Это продолжается до тех пор, пока не будет сделан проход без каких-либо обменов, что означает, что массив отсортирован. Обмен выполняется в двух направлениях – наибольшие элементы перемещаются в правую сторону, а во время обратного движения наименьшие движутся в левую сторону.

Практическое контрольное задание 3. Реализация сортировки Шелла.

Создать класс, содержащий метод *CocktailSort*, содержащий реализацию сортировки Шелла. Сортировка Шелла, представляет собой алгоритм, который упорядочивает элементы в массиве, сравнивая их не только с соседними элементами, как в сортировке вставками, но и с элементами, стоящими на определенном расстоянии друг от друга. Это расстояние называется интервалом, и оно уменьшается с каждым проходом алгоритма.

Принцип работы сортировки Шелла заключается в следующем: сначала выбирается начальный интервал, часто равный половине длины массива. На каждом шаге элементы, находящиеся на расстоянии интервала друг от друга, сравниваются и, если нужно, меняются местами. После завершения первого прохода интервал уменьшается, например, вдвое, и процесс повторяется, пока интервал не станет равен 1. На последнем этапе сортировка Шелла сводится к обычной сортировке вставками.

Практическое контрольное задание 4. Генерация простого двоичного дерева.

Создать приложение, в котором генерируется последовательность из 1000 случайных целых чисел в диапазоне $[0, 99]$, сохраняемых в двоичном дереве поиска. Выполнить обход дерева и сформировать массив из 10 чисел, каждое из которых кратно числу, равного индексу этого элемента.

В качестве варианта задания 2 рассмотреть возможность сохранения в узлах дерева отдельных слов из заданного текстового файла.

Практическое контрольное задание 5. Генерация простого двоичного дерева.

Используйте двоичное дерево для представления файловой системы: каждый каталог содержит не более двух наследников – файлов и/или папок. Для файла, кроме имени, в виде целого числа указано время его создания. Выполнить последующий обход дерева с удалением всех файлов, созданных ранее указанной даты.

Практическое контрольное задание 6. Разработка алгоритма проверки графа на односвязность.

Ориентированный граф $G = (V, E)$ обладает свойством односвязности, если имеется не более одного пути для каждой пары вершин этого графа. Разработайте эффективный алгоритм для определения, является ли ориентированный граф односвязным.

Практическое контрольное задание 7. Решение простой задачи динамического программирования.

На числовой прямой сидит кузнечик, который может прыгать вправо на одну или на две единицы. Первоначально кузнечик находится в точке с координатой 1. Определить количество различных маршрутов кузнечика, приводящих его в точку с координатой n .

Подсказка. Обозначим количество маршрутов кузнечика, ведущих в точку с координатой n , как $K[n]$. Существует ровно один маршрут из точки 1 ($K[1]=1$) в точку 1 — он не содержит ни одного прыжка. В точку 2 можно прыгнуть единственным способом — из точки 1 ($K[2]=1$).

В точку n кузнечик может попасть двумя способами — из точки при помощи прыжка длиной 2 и из точки прыжком длины 1. То есть число способов попасть в точку n равно сумме числа способов попасть в точку $(n-1)$ и $(n-2)$, что позволяет выписать рекуррентное соотношение: $K[n] = K[n-1] + K[n-2]$.

Практическое контрольное задание 9. Решение задачи динамического программирования.

Вариант 1. Решить задачу о количестве способов достичь точки n из точки 1, если кузнечик умеет прыгать +1, +2 и +3.

Вариант 2. Решить задачу о количестве способов достичь точки n из точки 1, если кузнечик умеет прыгать +1, +2 и *3.

Тема 3. Разработка мобильных приложений

Практическое занятие 8. Разработать простейший приложение для отправки СМС и определения координаты текущего местоположения.

Практическое контрольное занятие. Создать простой виджет приложения, который отображает значок, текст и расходы на покупки за текущий день. Создать базу данных SQLite, в которую добавляются сведения по расходам. При нажатии на виджет

производится вставка новой записи в таблицу базы данных, затем подсчитывается сумма затрат и обновляется текстовое поле, в котором отображаются расходы.

4.2.1.2 Типовые вопросы для устного опроса

Тема 1. Объектно-ориентированный анализ и шаблоны проектирования

1. Перечислить основные принципы ООА.
2. Дать определения элементов объектной модели.
3. Сделать обзор истории появления *UML*.
4. Определить место *UML* в процессе разработки ИС.
5. Определить использование *UML* на разных этапах разработки ПО.
6. Сделать обзор и объяснить назначение диаграмм *UML*.
7. Сделать обзор диаграммы классов.
8. Дать определение классу.
9. Рассмотреть в деталях виды связей классов

Тема 2. Базовые алгоритмы программирования

1. Дать определение понятию инструментальной среды разработки.
2. Зачем применяют сортировку данных?
3. Какие алгоритмы сортировки данных считаются эффективными?
4. Какие алгоритмы сортировки данных считаются глупыми (неэффективными)?
5. В чем суть алгоритма пузырьковой сортировки?
6. В чем суть алгоритма быстрой сортировки?
7. В чем суть алгоритма сортировки вставками?
8. В чем суть алгоритма сортировки Тима?
9. Какой поиск эффективнее, бинарный или линейный?
10. Что такое ассоциативные массивы?
11. Зачем применяется хеширование?
12. Что такое хеш функция, хеш таблица?
13. Где применяется структура данных дерево?
14. Какие специальные термины связаны с программной реализацией деревьев?
15. Назовите основные операции с бинарными деревьями?
16. Возможно ли представить дерево посредством массива?
17. В чем суть алгоритма обхода дерева?
18. Какие существуют алгоритмы обхода деревьев?
19. В чем суть алгоритма обхода в глубину?
20. В чем суть алгоритма обхода в ширину?
21. Что такое упорядоченные деревья?
22. Какие виды сортировки используют структуру данных дерево?
23. В чем отличие структуры данных графа от дерева?
24. Определите структуру данных для представления графа.
25. В чем суть алгоритмов поиска кратчайшего пути?
26. В чем суть волнового алгоритма.
27. В чем суть алгоритма Дейкстры.
28. В чем суть алгоритм Форда-Мура-Беллмана.
29. Используется ли структура данных граф для сортировки?
30. Что такое минимальное остовное дерево? Существуют ли алгоритмы поиска минимального остовного дерева?
31. В чем суть задачи раскраски графов?
32. В чем суть алгоритма Краскала?
33. Какие алгоритмы нахождения НОК (НОД) существуют? В чем их суть?
34. Что такое числа Фибоначчи? Какие алгоритмы для их нахождения существуют?

35. Что такое простые числа? Какие алгоритмы нахождения простых чисел существуют?
36. Какие алгоритмы приближенных вычислений производных (интегралов) существуют?
37. В чем суть алгоритма Кнута–Морриса–Пратта?

Тема 3. Разработка мобильных приложений

1. Перечислите основные папки проекта *Xamarin*, объясните их назначение?
2. Что такое жизненный цикл приложения *Android*, перечислите основные состояния жизненного цикла приложения *Android*?
3. Как происходит компоновке пользовательского интерфейса приложения *Xamarin Android*? Какие контейнеры компоновки применяются наиболее часто?
4. Какие элементы управления применяются в приложениях *Xamarin Android*? Какие основные их свойства и методы?
5. Определите основные функции для работы с *Android*?
6. Как производится разработка экранных виджетов?
7. Как производится разработка меню и контекстных панелей действий?
8. Как производится разработка многостраничных приложений?

4.2.1.3 Примеры тестовых заданий.

Тема 1. Объектно-ориентированный анализ и шаблоны проектирования

- 1) Идентифицировать процессы, реализуемые некоторым классом**
 - a) Атрибут
 - b) Операция
 - c) Сообщение
 - d) Имя

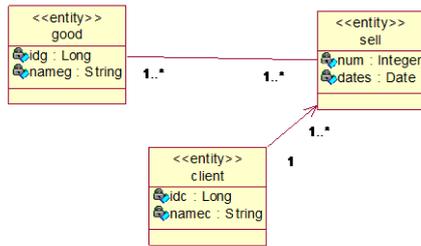
- 2) Определить свойство моделируемой сущности, которым обладают все объекты данного класса**
 - a) Атрибут
 - b) Операция
 - c) Сообщение
 - d) Имя

- 3) Идентифицировать хранение в одной структуре и данных, и методов**
 - a) свойством
 - b) полиморфизмом
 - c) наследованием
 - d) инкапсуляцией

- 4) Идентифицировать упорядочение абстракций, расположение их по уровням**
 - a) наследование
 - b) инкапсуляция
 - c) модульность
 - d) иерархия

- 5) Идентифицировать скрытие информации, т.е. маскировка всех внутренних деталей, не влияющих на внешнее поведение**
 - a) наследование
 - b) абстрагирование
 - c) полиморфизм
 - d) инкапсуляция

- 6) Идентифицировать выделение существенных характеристик объекта, которые отличают его от всех других видов объектов и таким образом чётко определяются особенности данного объекта с точки зрения дальнейшего его рассмотрения**
- a) полиморфизм
 - b) инкапсуляция
 - c) абстрагирование
 - d) наследование
- 7) Определить методология, при которой требования формируются на основе понятий классов и объектов, составляющих словарь предметной области**
- a) объектно-ориентированный анализ
 - b) функциональный анализ
 - c) структурный анализ
 - d) предметный анализ
- 8) Определить связь между целым и его частью**
- a) Агрегация
 - b) Ассоциация
 - c) Обобщение
 - d) Параметризация
- 9) Идентифицировать структурное отношение, показывающее, что объекты одного типа связаны с объектами другого типа**
- a) Агрегация
 - b) Обобщение
 - c) Ассоциация
 - d) Композиция
- 10) Определить отношение типа «является частью» («is-part-of»), когда объект-целое состоит из нескольких объектов-частей и когда объект-часть может существовать без объекта-целого**
- a) Агрегация
 - b) Обобщение
 - c) Ассоциация
 - d) Композиция
- 11) Определить отношение, когда время жизни частей и целого совпадают и, когда объект-часть не может существовать без объекта-целого**
- a) Агрегация
 - b) Обобщение
 - c) Ассоциация
 - d) Композиция
- 12) Определить какая диаграмма представлена на рисунке**



- диаграммы классов
- диаграммы активности
- диаграммы вариантов использования
- диаграмма последовательности

1	2	3	4	5	6	7	8	9	10	11	12
b	a	c	d	d	c	a	a	c	a	d	a

Тема 2. Базовые алгоритмы программирования

1. В чем суть пузырьковой сортировки?

- Выполняются итерации по списку, сравнивая соседние элементы попарно и меняя их местами, если предыдущее оказывается больше последующего.
- Массив просматривается справа налево и на каждую очередную позицию выбирается наименьший (или самый большой) из оставшихся элементов и меняется местами с тем, что находится в этой позиции сейчас.
- Элементы просматриваются по одному и ставятся на место в соответствии с уже упорядоченным массивом.

2. Какие алгоритмы сортировки считаются эффективными?

- Пирамидальная сортировка
- Пузырьковая сортировка
- Сортировка Тима
- Сортировка выбором
- Сортировка вставками
- Быстрая сортировка

3. Какие алгоритмы сортировки считаются глупыми (неэффективными)?

- Пирамидальная сортировка
- Пузырьковая сортировка
- Сортировка Тима
- Сортировка выбором
- Сортировка вставками
- Быстрая сортировка

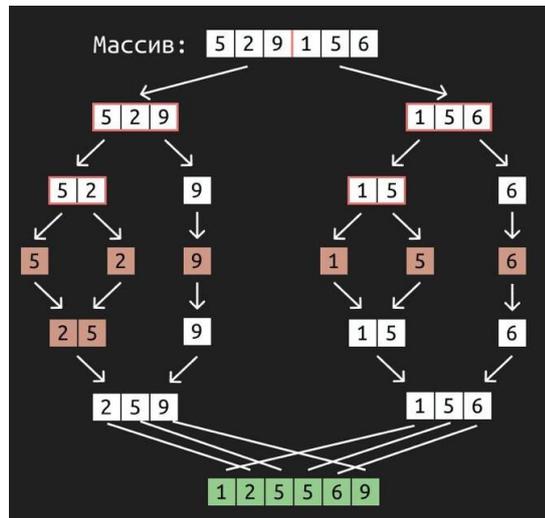
4. В чем суть сортировки вставками?

- Выполняются итерации по списку, сравнивая соседние элементы попарно и меняя их местами, если предыдущее оказывается больше последующего
- Массив просматривается справа налево и на каждую очередную позицию выбирается наименьший (или самый большой) из оставшихся элементов и меняется местами с тем, что находится в этой позиции сейчас
- Элементы просматриваются по одному и ставятся на место в соответствии с уже упорядоченным массивом

5. Оцените сложность алгоритма бинарного поиска по времени выполнения?

- а) $O(1)$
- б) $O(n)$
- в) $O(n^2)$
- г) $O(\log n)$

6. Алгоритм какой сортировки показан на рисунке?

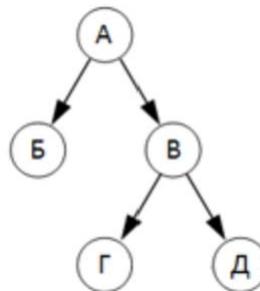


- а) Быстрая сортировка
- б) Сортировка слиянием
- в) Сортировка Тима
- г) Пирамидальная сортировка

7. Что такое дерево

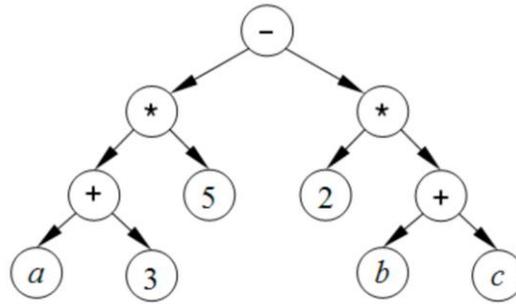
- а) Граф без циклов
- б) Граф только с одним циклом
- в) Взвешенный граф

8. В каких отношениях состоят узлы А и Г?



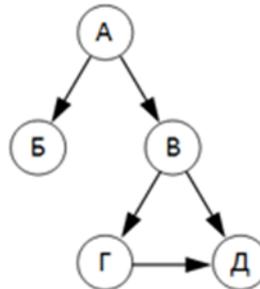
- а) Узел А - родитель для узла Г
- б) Узел А - предок для узла Г
- в) Узел Г - потомок для узла А
- г) Узел Г - сын для узла А
- д) Это некорректный вопрос

9. Используя дерево, показанное на рисунке, вычислите значение арифметического выражения при $a=1$, $b=6$ и $c=3$. Напишите программу анализа дерева и вычисления выражения.



- а) 3
- б) 6
- в) 2
- г) 5
- д) Нет ответа

10. Перечислите листья этого дерева



- а) А, В, Д
- б) Б, Г, Д
- в) В, Г, Д
- г) А, Б, Г
- д) это некорректный вопрос

11. Какой граф называется взвешенным

- а) Граф, в котором его вершины или ребра характеризуются некоторой дополнительной информацией — весами
- б) Граф, в котором все ребра равны
- в) Граф, в котором четное число вершин и ребер

12. Как называется граф с циклом?

- а) Генеалогический
- б) Сеть
- в) Взвешенный

13. Как называется линия без стрелки, соединяющая вершины графа?

- а) Дуга
- б) Линия
- в) Ребро

15. Перечислите ограничения алгоритма Дейкстры:
- а) График должен быть взвешенным.
 - б) График не должен быть взвешенным
 - в) Весы должны быть неотрицательными.
 - г) Весы должны быть отрицательными.
16. Будут ли равными количества способов добраться из клетки (0, 0) в клетку (12, 6) и из клетки (2, 2) в клетку (8, 14) в задаче о Кузнечике?
- а) Да
 - б) Нет
17. Сколько чисел, меньших 23, будет зачеркнуто в ходе работы решета Эратосфена?
- а) 12
 - б) 13
 - в) 14
 - г) 15
18. Сколько чисел, меньших 23, не будет зачеркнуто в ходе работы решета Эратосфена?
- а) 5
 - б) 6
 - в) 7
 - г) 8
19. Выберите число с наибольшим количеством делителей.
- а) 25
 - б) 12
 - в) 14
 - г) 13
20. Укажите пару взаимно простых чисел.
- а) 11 и 44
 - б) 17 и 28
 - в) 50 и 60
 - г) 68 и 3044

Тема 3. Разработка мобильных приложений

1. Какой метод в приложении для Android является точкой входа в приложение.
- а) Метод *Main*
 - б) Запуск программы будет осуществлен, когда система создает экземпляр одного из его классов
 - в) Метод *Start*
2. Какое назначение имеет метод *OnCreate*?
- а) Обработчик события жизненного цикла деятельности, возникающего при создании экрана.
 - б) Ищет элемент управления по его идентификатору и возвращает его экземпляр.
 - в) Метод используется для выполнения действия в потоке UI пользовательского интерфейса.

3. Какое назначение имеет метод *FindViewById*?
 - а) Обработчик события жизненного цикла деятельности, возникающего при создании экрана.
 - б) Ищет элемент управления по его идентификатору и возвращает его экземпляр.
 - в) Метод используется для выполнения действия в потоке UI пользовательского интерфейса.
4. Что происходит при переходе приложения в состояние *Stopped*?
 - а) освобождаются ресурсы
 - б) сохранение состояния приложения
 - в) переводит *Activity* в состояние ожидания
5. Как располагает дочерние элементы контейнер *LinearLayout*?
 - а) Такого контейнера не существует
 - б) расположение дочерних компонентов в одном направлении: вертикально или горизонтально
 - в) расположение дочерних элементов в создаваемой табличной структуре
6. Что представляют собой адаптеры в *Xamarin*?
 - а) Особые элементы, предназначенные для связывания источника данных с компонентом списка
 - б) Особые элементы, предназначенные для фильтрации данных источника.
 - в) Особые элементы, предназначенные для трансформации и создания различных представлений источника данных
7. Что такое фрагменты в *Android*?
 - а) Часть активности, обладающая собственным жизненным циклом
 - б) Элемент управления графическим пользовательского интерфейса, делящий его на участки с возможностью отображения или скрытия
 - в) Элемент управления, сохраняющий состояние части экрана
8. Что содержится в папке *Assets* проекта *Xamarin*?
 - а) В эту папку включаются файлы, требуемые в процессе выполнения приложения, такие как шрифты, локальные файлы данных и текстовые файлы
 - б) В данном каталоге содержатся ресурсы программы
 - в) Эта папка включает сборки, требуемые для разработки и запуска программы
9. Как происходит возврат результата из *Activity*?
 - а) С помощью ключевого слова `return`
 - б) С помощью вызова метода *StartActivity()*
 - в) С помощью вызова метода *StartActivityForResult(Intent intent, int requestCode)*
10. Как производится взаимодействие между активностями?
 - а) Посредством использования объектов класса намерения *Intent*
 - б) Посредством использования буфера обмена
 - в) Взаимодействие между активностями невозможно

Экзамен проводится с применением следующих методов (средств) :

Экзамен проводится в компьютерном классе. Во время экзамена проверяется освоение компетенции ПК-1.1.

Оценивается правильность ответов на поставленные вопросы, степени их полноты и обоснованности.

Промежуточная аттестация может проводиться устно в ДОТ/письменно с прокторингом/ тестирование с прокторингом. Для успешного освоения курса учащемуся рекомендуется ознакомиться с литературой, размещенной в разделе 6, и материалами, выложенными в ДОТ.

Таблица 4.2

Код компетенции	Наименование компетенции	Код этапа освоения компетенции	Наименование этапа освоения компетенции
ОПК-3	Способность управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3.1	Способность разрабатывать алгоритмы и программы, проектирует базы данных с целью использования на практике основных методов управления процессами создания продуктов и услуг ИКТ.
		ОПК-3.2	Способность разрабатывать и программы с целью использования на практике основных принципов и методов управления процессами создания продуктов и услуг в сфере информационно-коммуникационных технологий

Показатели и критерии оценивания компетенций с учетом этапа их формирования

Этап освоения компетенции	Показатель оценивания	Критерий оценивания
ОПК-3.1	<ol style="list-style-type: none">1. Демонстрирует способность разрабатывать и реализовывать алгоритмы.2. Демонстрирует умение разрабатывать пользовательский интерфейс ИС	<ol style="list-style-type: none">1. Полнота и качество выполнения задания.2. Продемонстрирована модель на языке программирования3. Правильное решение задачи. Описан путь ее решения, используемые теоретические положения.4. Правильные ответы на поставленные вопросы
ОПК-3.2	<ol style="list-style-type: none">1. Демонстрирует умение выполнять базовые трудовые функции, связанные с анализом и описанием основных бизнес-процессов предприятия (фирмы).2. Демонстрирует способность самостоятельно решать поставленные задачи	<ol style="list-style-type: none">1. Полнота и качество выполнения задания.2. Продемонстрирована модель на языке программирования3. Правильное решение задачи. Описан путь ее решения, используемые теоретические положения.4. Правильные ответы на поставленные вопросы

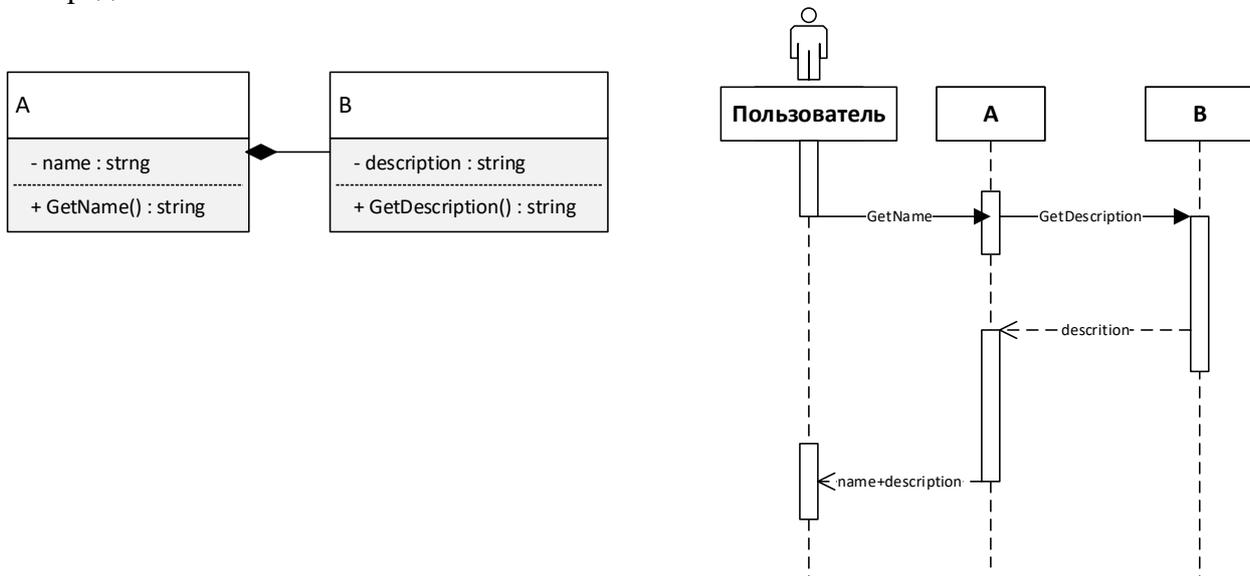
Для оценки сформированности компетенций, знаний и умений, соответствующих данным компетенциям, используются контрольные вопросы, а также выполнение практических заданий.

Типовые вопросы, выносимые на экзамен:

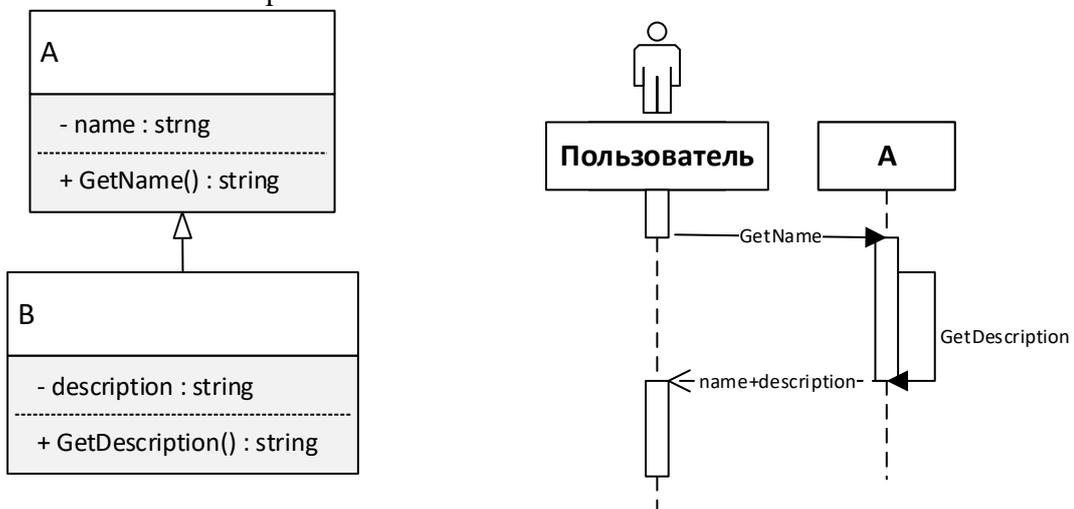
1. Сформулировать принципы объектно –ориентированного анализа.
2. Дать определение инструментальному средству, перечислить инструментальные средства объектно-ориентированного анализа и программирования.
3. Сделать обзор истории появления *UML* и определить место *UML* в процессе разработки ИС.
4. Дать определение классу, рассмотреть в деталях виды связей классов
5. Какие контейнеры компоновки существуют в *Xamarin*?

Типовые задания, выносимые на экзамен:

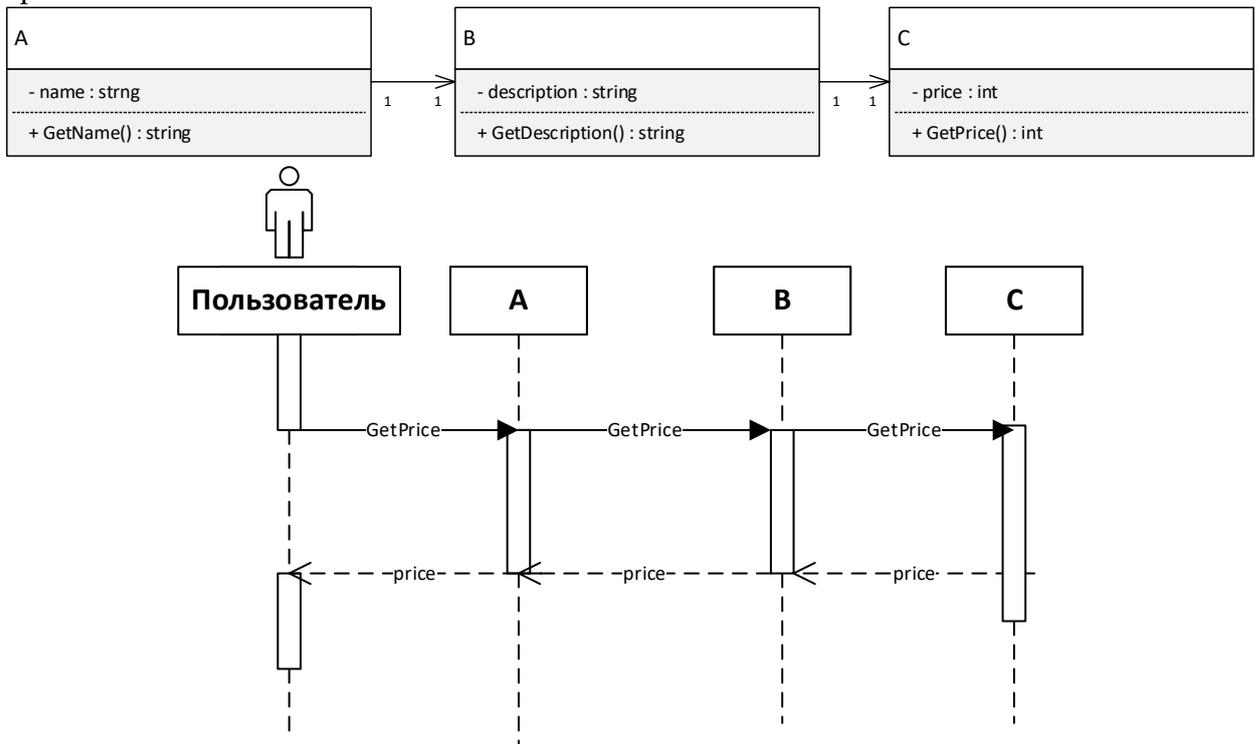
1. Реализуйте программу, на основе диаграммы классов и диаграммы последовательностей, представленных ниже.



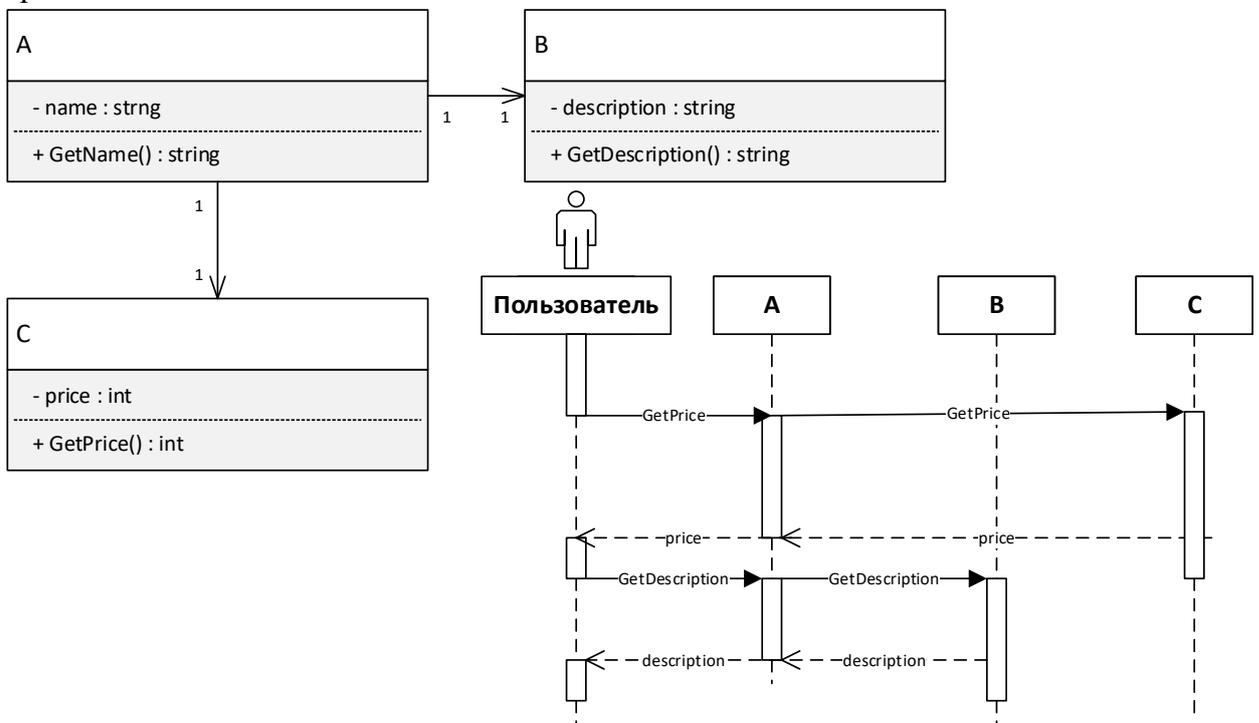
2. Реализуйте приложение, на основе диаграммы классов и диаграммы последовательностей, представленных ниже. Приложение реализуется с использованием шаблона консольного приложения *C#*.



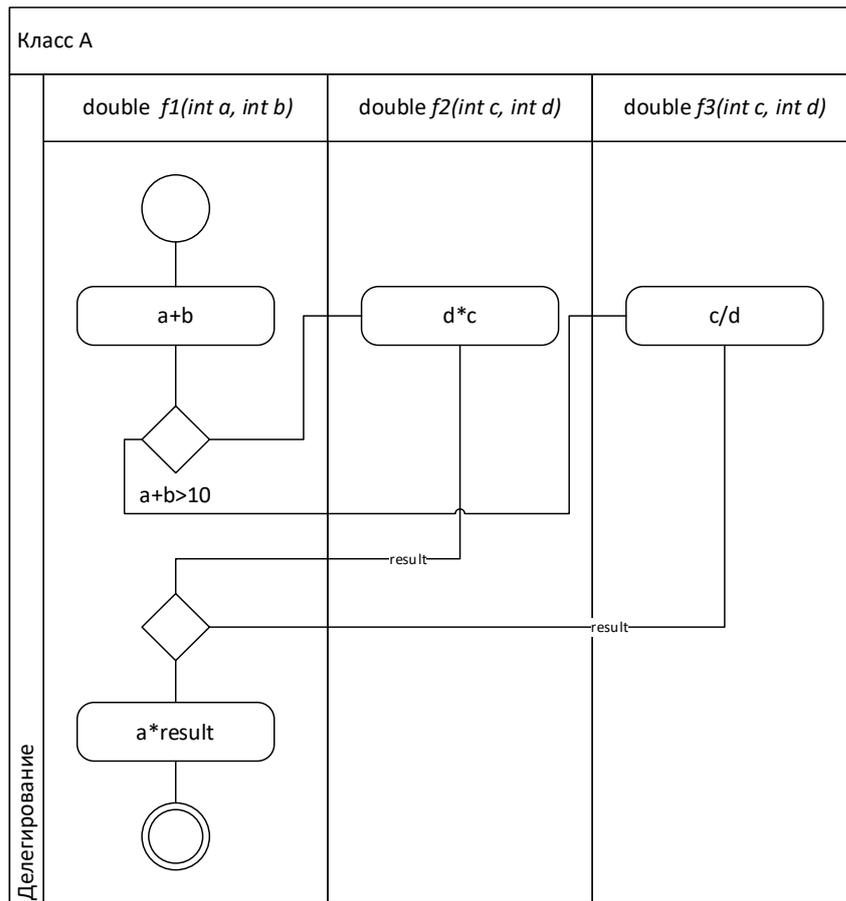
3. Реализуйте программу, на основе диаграммы классов и диаграммы последовательностей, представленных ниже. Приложение реализуется с использованием шаблона консольного приложения *C#*.



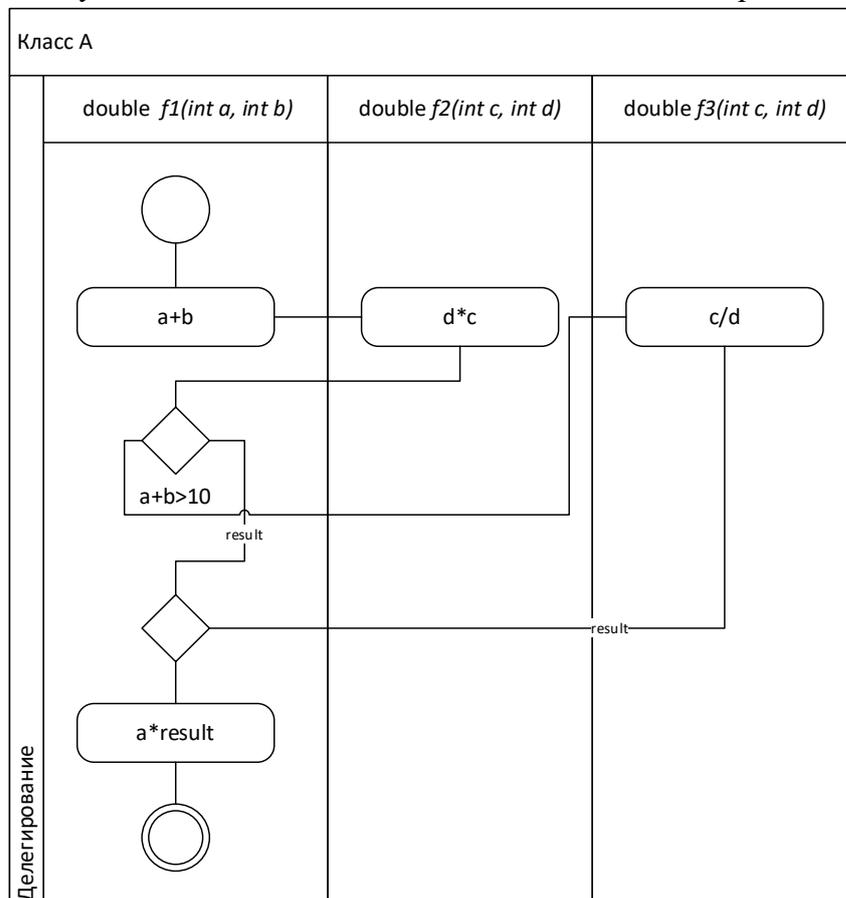
4. Реализуйте программу, на основе диаграммы классов и диаграммы последовательностей, представленных ниже. Приложение реализуется с использованием шаблона консольного приложения *C#*.



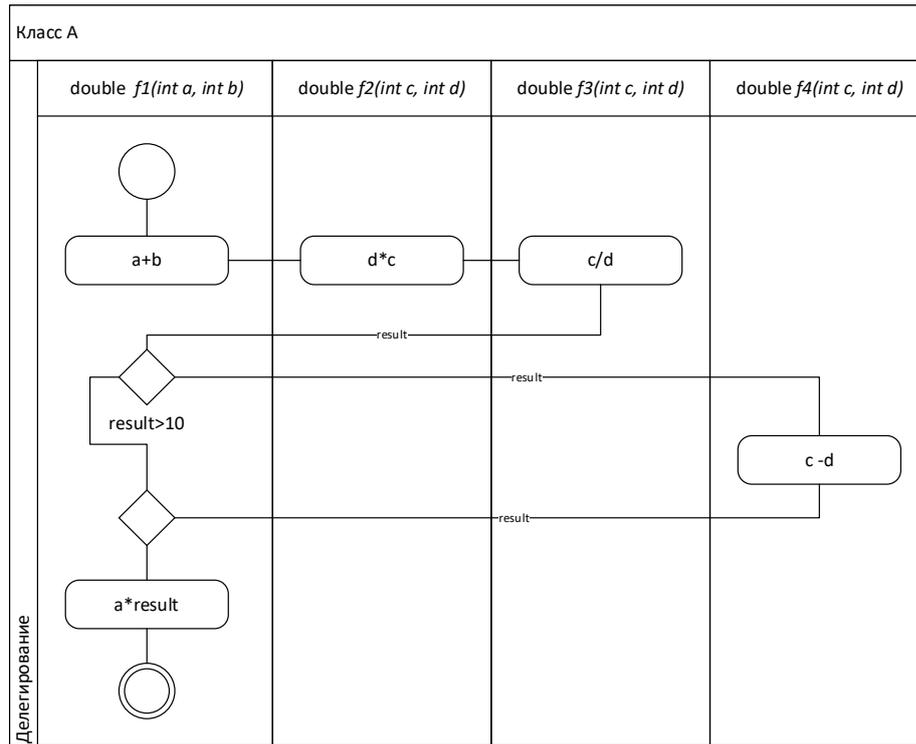
5. Реализуйте программу, на основе диаграммы деятельности, представленной ниже. Приложение реализуется с использованием шаблона консольного приложения *C#*.



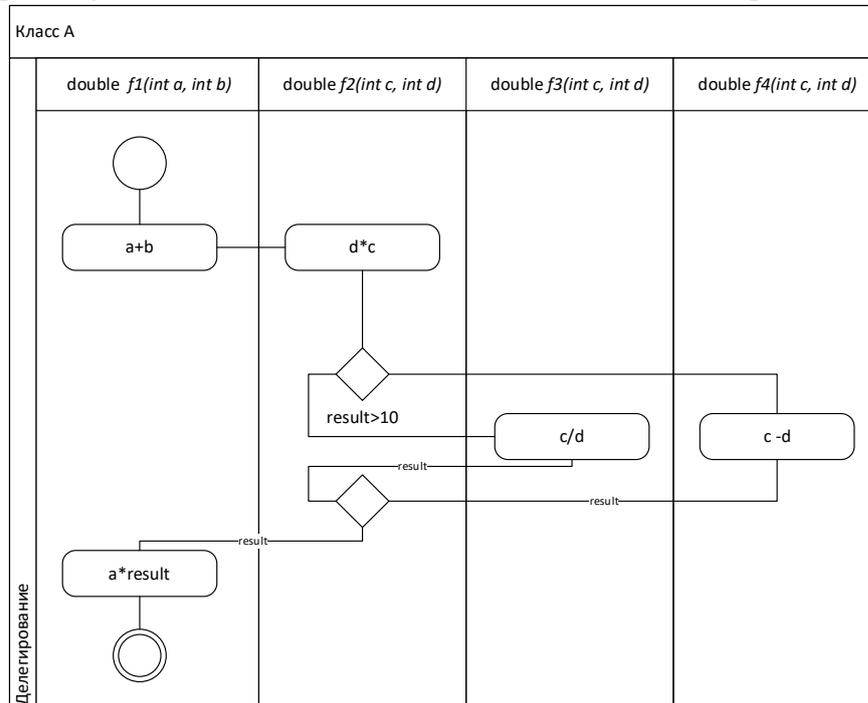
6. Реализуйте программу, на основе диаграммы деятельности, представленной ниже. Приложение реализуется с использованием шаблона консольного приложения *C#*.



7. Реализуйте программу, на основе диаграммы деятельности, представленной ниже. Приложение реализуется с использованием шаблона консольного приложения C#.



8. Реализуйте программу, на основе диаграммы деятельности, представленной ниже. Приложение реализуется с использованием шаблона консольного приложения C#.



9. Создайте мобильное приложение для расчета индекса массы тела $I = m/h^2$.

10. Создайте мобильное приложение конвертер валют.

11. Создайте мобильное приложение конвертер единиц длины.

Описание системы оценивания

Оценочные средства (формы текущего и промежуточного контроля)	Показатели оценки	Критерии оценки
Опрос	Корректность и полнота ответов	<p>Сложный вопрос: полный, развернутый, обоснованный ответ – 4 балла Правильный, но не аргументированный ответ – 2 балла Неверный ответ – 0 баллов</p> <p>Обычный вопрос: полный, развернутый, обоснованный ответ – 4 балла Правильный, но не аргументированный ответ – 2 балла Неверный ответ – 0 баллов.</p> <p>Простой вопрос: Правильный ответ – 2 балла; Неправильный ответ – 0 баллов</p>
Тест	Корректность ответов	В зависимости от семестра максимальное количество баллов за один тест составляет 5 или 10 баллов
Задание	1) полнота выполнения задания; 2) корректность выводов; 3) обоснованность решений.	<p>Выполнена обязательная и самостоятельная часть, даны развернутые ответы на вопросы – 5 баллов</p> <p>Выполнена обязательная часть, даны развернутые ответы на вопросы – 4 баллов</p> <p>В обязательной части допущены ошибки, формальные ответы на вопросы -3 балла</p> <p>В обязательной части допущены ошибки, нет ответов на контрольные вопросы -2 балла</p> <p>Имеются множественные ошибки и нет ответов на контрольные вопросы -1 балл</p> <p>Работа, представленная для защиты позже установленного срока, оценивается с понижением баллов. Просроченные работы и представленные на последнем практическом занятии оцениваются максимум на 1 балл.</p>

Оценивание студентов на экзамене по дисциплине «Проектирование и разработка web-приложений»

Баллы, %	Критерии
100-85 «отлично»	Оценка «отлично» на экзамене выставляется обучающемуся, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение.
84-70 «хорошо»	Оценка «хорошо» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения, допускает неточности в увязывании теории с практикой.
69-55	Оценка «удовлетворительно» выставляется обучающемуся, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно

«удовлетворительно»	правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при установлении связи теории и практики.
Менее 55 «неудовлетворительно»	Оценка «неудовлетворительно» выставляется обучающемуся, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями устанавливает связь теории и практики.

Шкала оценивания.

Оценка результатов производится на основе балльно-рейтинговой системы (БРС). Использование БРС осуществляется в соответствии с приказом от 06 сентября 2019 г. №306 «О применении балльно-рейтинговой системы оценки знаний обучающихся».

Схема расчетов сформирована в соответствии с учебным планом направления, согласована с руководителем научно-образовательного направления, утверждена деканом факультета.

Схема расчетов доводится до сведения студентов на первом занятии по данной дисциплине, является составной частью рабочей программы дисциплины и содержит информацию по изучению дисциплины, указанную в Положении о балльно-рейтинговой системе оценки знаний обучающихся в РАНХиГС.

В случае если студент в течение семестра не набирает минимальное число баллов, необходимое для сдачи промежуточной аттестации, то он может заработать дополнительные баллы, отработав соответствующие разделы дисциплины, получив от преподавателя компенсирующие задания.

В случае получения на промежуточной аттестации неудовлетворительной оценки студенту предоставляется право повторной аттестации в срок, установленный для ликвидации академической задолженности по итогам соответствующей сессии.

Обучающийся, набравший в ходе текущего контроля в семестре от 51 до 70 баллов, по его желанию может быть освобожден от промежуточной аттестации.

Количество баллов	Оценка	
	прописью	буквой
96-100	отлично	А
86-95	отлично	В
71-85	хорошо	С
61-70	хорошо	Д
51-60	удовлетворительно	Е

Перевод балльных оценок в академические отметки «отлично», «хорошо», «удовлетворительно»

- «Отлично» (А) - от 96 по 100 баллов – теоретическое содержание курса освоено полностью, без пробелов необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено максимальным числом баллов.

- «Отлично» (В) - от 86 по 95 баллов – теоретическое содержание курса освоено полностью, без пробелов необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

- «Хорошо» (С) - от 71 по 85 баллов – теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками.

- «Хорошо» (D) - от 61 по 70 баллов – теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, большинство предусмотренных программой обучения учебных заданий выполнены, качество выполнения ни одного из них не оценено

минимальным числом баллов, некоторые виды заданий выполнены с ошибками.

- «Удовлетворительно» (Е) - от 51 по 60 баллов – теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий выполнены с ошибками.

7. Методические материалы по освоению дисциплины.

Описание учебной дисциплины и методика выполнения практических занятий имеются в ресурсах сети факультета Учебные материалы\Объектно ориентированный анализ и программирование. Также методические указания по выполнению работ находятся на странице дисциплины по адресу <http://sziu-de.ranepa.ru>

С целью контроля сформированных компетенций разработан фонд тестовых вопросов. В назначенное преподавателем время студент проходит тест, вопросы теста формируются случайным образом из банка вопросов.

Методические указания для обучающихся по освоению дисциплины

Рабочей программой дисциплины предусмотрены следующие виды аудиторных занятий: лекции, практические занятия, лабораторные работы. На лекциях рассматриваются наиболее сложный материал дисциплины. Лекция сопровождается презентациями, компьютерными текстами лекции, что позволяет студенту самостоятельно работать над повторением и закреплением лекционного материала. Для этого студенту должно быть предоставлено право самостоятельно работать в компьютерных классах в сети Интернет.

Практические занятия предназначены для углубленного изучения дисциплины. На этих занятиях идет осмысление теоретического материала, приобретаются навыки программирования.

Лабораторные работы позволяют объединить теоретико-методологические знания и практические навыки учащихся в процессе научно-исследовательской деятельности.

Все практические и лабораторные работы проводятся в компьютерных классах с использованием среды MS Office. Каждая работа должна быть защищена, т.е. студент должен ответить на вопросы преподавателя о ходе выполнения работы, а также на вопросы теоретического характера.

С целью контроля сформированности компетенций разработан фонд контрольных заданий. Его использование позволяет реализовать балльно-рейтинговую оценку, определенную приказом от 28 августа 2014 г. №168 «О применении балльно-рейтинговой системы оценки знаний студентов».

С целью активизации самостоятельной работы студентов в системе дистанционного обучения *Moodle* разработан учебный курс «Объектно ориентированный анализ и программирование», включающий набор файлов с текстами лекций, заданиями для выполнения практических и лабораторных работ.

Для активизации работы студентов во время контактной работы с преподавателем отдельные занятия проводятся в интерактивной форме. В основном, интерактивная форма занятий обеспечивается при проведении занятий в компьютерном классе. Интерактивная форма обеспечивается наличием разработанных файлов с заданиями, наличием контрольных вопросов, возможностью доступа к системе дистанционного обучения, а также к тестеру.

Для работы с печатными и электронными ресурсами СЗИУ имеется возможность доступа к электронным ресурсам. Организация работы студентов с электронной библиотекой указана на сайте института (странице сайта – «Научная библиотека»).

7. Учебная литература и ресурсы информационно-телекоммуникационной сети "Интернет", включая перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

7.1. Основная литература.

1. Бабушкина, Ирина Анатольевна. Практикум по объектно-ориентированному программированию : учебное пособие / И.А. Бабушкина, С.М. Окулов. - 5-е изд., электрон. - Москва : Лаборатория знаний, 2020. - 367 с. - Текст : электронный. - URL: <https://e.lanbook.com/book/135561> (дата обращения: 24.12.2020).
2. Биллиг, В.А. Основы объектного программирования на C# : учебное пособие / В.А. Биллиг. - 3-е изд. (электрон.). - Москва : ИНТУИТ [и др.], 2021. - 573 с. - Текст : электронный. - URL: <http://www.iprbookshop.ru/102033.html> (дата обращения: 05.03.2021)
3. Умрихин Е. Д. Разработка Android-приложений на C# с использованием Xamarin с нуля. — СПб.: БХВ-Петербург, 2021. — 336 с.
4. Горелов, Сергей Витальевич. Современные технологии программирования: разработка Windows-приложений на языке C# : в 2 т. : учебник / С.В. Горелов ; под ред. Лукьянова Павла Борисовича. - Москва : Прометей, 2019. - 378 с.Т. 2. - Текст : электронный. - URL: <http://www.iprbookshop.ru/94533.html> (дата обращения: 01.10.2020).
5. Залогова, Любовь Алексеевна. Основы объектно-ориентированного программирования на базе языка C# : учебное пособие / Л.А. Залогова. - Изд. 2-е, стер. - Санкт-Петербург [и др.] : Лань, 2020. - 191 с. - (Бакалавриат) (Серия "Учебники для вузов. Специальная литература") . - Текст : электронный. - URL: <https://e.lanbook.com/book/126160> (дата обращения: 24.12.2020).
6. Казанский, Александр Анатольевич. Программирование на Visual C# : учебное пособие для вузов / А. А. Казанский. - 2-е изд., перераб. и доп. - Москва : Юрайт, 2020. - 192 с. - (Высшее образование) . - Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/451467> (дата обращения: 21.09.2020).
7. Кариев, Ч.А. Разработка Windows-приложений на основе Visual C# : учебное пособие / Ч.А. Кариев. - 3-е изд. (электрон.). - Москва : ИНТУИТ [и др.], 2021. - 978 с. - Текст : электронный. - URL: <http://www.iprbookshop.ru/102057.html> (дата обращения: 03.03.2021).

Все источники основной литературы взаимозаменяемы

7.2.Дополнительная литература.

1. Александров, Э.Э Программирование на языке C в Microsoft Visual Studio 2010 : учебное пособие / Э.Э Александров, В.В. Афонин. - 3-е изд. (электрон.). - Москва : ИНТУИТ [и др.], 2021. - 569 с. - Текст : электронный. - URL: <http://www.iprbookshop.ru/102050.html> (дата обращения: 03.03.2021).
2. Кудрина, Елена Вячеславовна. Основы алгоритмизации и программирования на языке C# : учебное пособие для вузов / Е. В. Кудрина, М. В. Огнева. - Москва : Юрайт, 2020. - 322 с. - (Высшее образование) . - Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/456182> (дата обращения: 21.09.2020)

7.3. Учебно-методическое обеспечение самостоятельной работы.

1. Положение об организации самостоятельной работы студентов федерального государственного бюджетного образовательного учреждения высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» (в ред. приказа РАНХиГС от 11.05.2016 г. № 01-2211);

7.4. Нормативные правовые документы.

Не используются

7.5. Интернет-ресурсы.

СЗИУ располагает доступом через сайт научной библиотеки <http://nwapa.spb.ru/> к следующим подписным электронным ресурсам:

Русскоязычные ресурсы

Электронные учебники электронно - библиотечной системы (ЭБС) «Айбукс»

Электронные учебники электронно – библиотечной системы (ЭБС) «Лань»

Рекомендуется использовать следующий интернет-ресурсы

<http://serg.fedosin.ru/ts.htm>

<http://window.edu.ru/resource/188/64188/files/chernyshov.pdf>

7.6. Иные источники.

Не используются.

8. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

Все практические занятия проводятся в компьютерном классе. Учебная дисциплина включает использование программного обеспечения Microsoft Visual Studio.

Методы обучения с использованием информационных технологий (компьютерное тестирование, демонстрация мультимедийных материалов).

В кабинетах должны быть доступны ресурсы Интернет, портал с образовательными ресурсами РАНХиГС.

Интернет-сервисы и электронные ресурсы (поисковые системы, электронная почта, профессиональные тематические чаты и форумы, системы аудио и видео конференций, онлайн энциклопедии, справочники, библиотеки, электронные учебные и учебно-методические материалы).

Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

№ п/п	Наименование
1.	Компьютерные классы с персональными ЭВМ, объединенными в локальные сети с выходом в Интернет
2.	<i>MS Visual Studio</i>
3.	Мультимедийные средства в каждом компьютерном классе и в лекционной аудитории
4.	Браузер, сетевые коммуникационные средства для выхода в Интернет
5.	Облачные технологии <i>Google Collab, Loginom</i>

Компьютерные классы из расчета один ПЭВМ для одного обучаемого. Каждому обучающемуся должна быть предоставлена возможность доступа к сетям типа Интернет в течение не менее 20% времени, отведенного на самостоятельную подготовку.