

ФГБОУ ВО Новосибирский ГАУ
Институт цифровых технологий
Кафедра прикладной биоинформатики



ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Методические указания
по выполнению практических занятий, контрольной
и самостоятельной работы

Новосибирск 2026

УДК 004.652.5(07)

ББК 16.2,я7

О 294

Кафедра прикладной биоинформатики

Составители: д.б.н., доцент Е.В.Камалдинов, М.П.Меркушкина

Рецензент: к.тех.н. А.А.Диденко

Объектно-ориентированное программирование: методические указания по выполнению практических занятий, контрольной и самостоятельной работы / Университет биотехнологий, Институт цифровых технологий; составитель: Е.В.Камалдинов, М.П.Меркушкина – Новосибирск, 2026. – 25 с.

Методические указания предназначены для студентов магистратуры, обучающихся по направлению подготовки 09.04.03 Прикладная информатика, профиль Прикладная биоинформатика, и устанавливают порядок проведения практических занятий, а также самостоятельной подготовки студентов по разделам дисциплины.

Утверждены и рекомендованы к изданию учебно-методическим советом Института цифровых технологий (протокол № 5 от 20.01.2026 г.).

Содержание

Введение.....	4
1. Общие методические указания по самостоятельному изучению разделов дисциплины.....	5
2. Методические указания по самостоятельному изучению отдельных тем дисциплины и вопросы для проверки знаний.....	6
3. Выполнение контрольной работы	14
4. Промежуточная аттестация.....	18

Введение

На изучение дисциплины «Объектно-ориентированное программирование» запланировано 144 ч, в том числе 44 аудиторных и 100 (69,4%) отведено для самостоятельного освоения. Учебным планом предусмотрены экзамен и контрольная работа.

Основная цель дисциплины – формирование у студентов теоретических знаний и практических навыков объектно-ориентированного проектирования и реализации программного обеспечения на современных языках программирования.

В процессе изучения дисциплины решаются следующие **задачи**:

- изучение принципов ООП и их отличий от процедурного и функционального программирования;
- освоение синтаксиса и возможностей объектно-ориентированных языков программирования;
- разработка программных модулей с использованием классов, объектов, наследования и интерфейсов;
- применение паттернов проектирования для решения типовых задач;
- анализ и оптимизация архитектуры ПО.

Требования к уровню освоения учебной дисциплины. В соответствии с требованиями ФГОС ВО дисциплина «Объектно-ориентированное программирование» направлена на формирование следующих компетенций по направлению подготовки 09.04.03 Прикладная информатика (квалификация (степень) магистр):

- способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач (ОПК-2);
- способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем (ОПК-5);
- способен управлять работами по сопровождению и проектами создания (модификации) информационных систем, автоматизирующих задачи организационного управления и производственных процессов в животноводстве (ПК-1).

1. Общие методические указания по самостоятельному изучению разделов дисциплины

Самостоятельное изучение разделов дисциплины целесообразно начинать с подбора рекомендованной учебной и методической литературы. После изучения необходимо провести самопроверку знаний путем ответов на вопросы, предусмотренные по каждой теме.

Рекомендуемая литература

Основная

1. Барков, И. А. Объектно-ориентированное программирование / И. А. Барков. — 2-е изд., стер. — Санкт-Петербург: Лань, 2023. — 700 с. — ISBN 978-5-507-47113-3. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/329549>
2. Золотарюк, А. В. Язык и среда программирования R: учебное пособие / А.В. Золотарюк. — Москва: ИНФРА-М, 2023. — 162 с. — (Высшее образование). — DOI 10.12737/textbook_5b8fdb0bd795c4.69435980. - ISBN 978-5-16-018723-5. - Текст: электронный. - URL: <https://znanium.com/catalog/product/2049696>
3. Галиновский, Н. Г. Введение в программирование на языке R: учебное пособие / Н. Г. Галиновский. — Гомель: ГГУ имени Ф. Скорины, 2022. — 222 с. — ISBN 978-985-577-826-5. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/320945>. — Режим доступа: для авториз. пользователей.

Дополнительная

4. Гришин, В. А. Методы обработки данных и моделирование на языке R: учебно-методическое пособие / В. А. Гришин, М. С. Тихов. — Нижний Новгород: ННГУ им. Н. И. Лобачевского, 2019. — 54 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/144653>
5. Тагирова, Л. Ф. Программирование и алгоритмизация: учебное пособие / Л. Ф. Тагирова. — Оренбург: ОГУ, 2020. — 370 с. — ISBN 978-5-7410-2515-4. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/422693>.

2. Методические указания по самостоятельному изучению отдельных тем дисциплины и вопросы для проверки знаний

Раздел 1. Введение в ООП. Парадигмы ООП в R (S3, S4, R6).

Основные принципы объектно-ориентированного программирования. Понятия класса и объекта, атрибутов и методов. Преимущества и недостатки ООП. Эволюция ООП в R: общие сведения о S3, S4 и R6. Место R6 в экосистеме R. Сравнение R6 с другими ООП-парадигмами R и других языков (Python). Подготовка рабочей среды: установка R и RStudio, R6 пакета.

Раздел 2. Основы ООП в R6: Классы, объекты, инкапсуляция.

Определение классов в R6: R6Class(). Создание объектов. Метод initialize() (конструктор): передача и инициализация атрибутов. Открытые (public) и закрытые (private) члены класса. Инкапсуляция: цели и методы реализации. Активные привязки (active): реализация геттеров и сеттеров для контроля доступа к атрибутам. Примеры инкапсуляции данных о биологических объектах (например, ДНК-последовательности, белки).

Раздел 3. Наследование и композиция в R6.

Наследование в R6: создание дочерних классов (inherit). Переопределение методов и атрибутов родительского класса. Использование super для доступа к методам родителя. Множественное наследование (концепции). Композиция как альтернатива наследованию: агрегация объектов. Выбор между наследованием и композицией в зависимости от задачи (например, моделирование генеалогического древа или сложных биоинформатических объектов).

Раздел 4. Полиморфизм, абстракция и продвинутые концепции R (S3, S4). Применение ООП в биоинформатике.

Полиморфизм: концепция и реализация в R6. Диспетчеризация методов. Принципы абстракции в ООП. Создание "абстрактных" классов в R6 (через соглашения). Обзор ООП в S3: создание generic-функций, методы для классов. Обзор ООП в S4: setClass, setMethod, валидация слотов. Перегрузка операторов в R (на основе S3/S4). Примеры применения ООП для моделирования биологических данных и процессов (например, генетические варианты, экспериментальные данные, пайплайны анализа) с использованием R6 и других парадигм R.

Текущий контроль успеваемости

Вопросы для коллоквиумов и собеседования

Раздел 1. Введение в ООП. Парадигмы ООП в R (S3, S4, R6).

1. Что такое объектно-ориентированное программирование (ООП)? Объясните его основные преимущества по сравнению с процедурным подходом.
2. Перечислите и кратко охарактеризуйте основные понятия ООП: класс, объект, атрибут, метод.
3. Опишите, какие три основные парадигмы ООП используются в языке R.
4. В чем заключаются фундаментальные различия между S3, S4 и R6 системами ООП в R?
5. Когда целесообразно использовать S3-объекты и методы в R? Приведите примеры встроенных S3-классов.
6. В каких случаях предпочтительнее применять S4-классы и методы? Какова их роль в пакетах Bioconductor?
7. Почему R6-классы часто называют "императивными" или "мутабельными"? В чем их сходство с ООП в Python или Java?
8. Объясните концепцию "ссылочной семантики" в R6. Как она отличается от поведения большинства объектов в базовом R?
9. Приведите примеры задач в биоинформатике, где применение ООП (в целом) может быть особенно эффективным.
10. Какие шаги необходимо выполнить для подготовки рабочей среды для разработки ООП-приложений на R (установка R, RStudio, пакетов)?
11. Каковы основные аргументы в пользу перехода от чисто функционального программирования к ООП при решении сложных биоинформатических задач?
12. Объясните понятие "парадигма программирования". Как ООП вписывается в этот контекст?
13. В чем заключается основной недостаток процедурного программирования, который ООП призвано решить?
14. Что такое "конструктор" класса в ООП? Для чего он используется?
15. Какие существуют подходы к созданию объектов в каждой из трех ООП-систем R (S3, S4, R6)?
16. Как R6-классы облегчают инкапсуляцию по сравнению с S3 или S4?

17. Обсудите, как выбор ООП-парадигмы в R может повлиять на структуру и масштабируемость биоинформатического проекта.
18. Приведите примеры объектов из реального мира (в контексте животноводства или биологии), которые могли бы быть представлены как классы в ООП.
19. Какие основные функции выполняет среда разработки RStudio при работе с R-кодом, написанным в ООП-стиле?
20. Опишите роль `methods` и `base` пакетов в реализации ООП-функциональности в R.

Раздел 2. Основы ООП в R6: Классы, объекты, инкапсуляция.

1. Как создать новый R6-класс с использованием функции `R6Class()`? Приведите базовый синтаксис определения класса.
2. Объясните назначение метода `initialize()` в R6-классах. Какие параметры он может принимать и для чего они используются?
3. Что такое инкапсуляция в ООП? Почему этот принцип является фундаментальным для создания надежного и поддерживаемого кода?
4. Как R6 реализует инкапсуляцию с помощью `public` и `private` членов класса? Приведите примеры их объявления.
5. Объясните разницу между `public` и `private` атрибутами в R6. Когда следует использовать каждый из них?
6. Что такое "активные привязки" (`active`) в R6? Как они отличаются от обычных методов (`public` функции) и публичных полей?
7. Приведите пример использования активной привязки для реализации "геттера" (метода для чтения значения атрибута) в R6.
8. Как реализовать "сеттер" (метод для изменения значения атрибута) с помощью активной привязки в R6? Какую дополнительную логику можно включить в сеттер?
9. Продемонстрируйте создание R6-класса `Gene`, который имеет приватный атрибут `_sequence` (последовательность ДНК/РНК) и публичный `id`. Как вы обеспечите инкапсуляцию для `_sequence`?
10. Какие преимущества дает использование активных привязок (геттеров/сеттеров) по сравнению с прямым доступом к публичным атрибутам?
11. Объясните, как инкапсуляция помогает предотвратить "нежелательное" изменение состояния объекта извне.
12. В чем заключается важность использования конструктора (`initialize()`) для обеспечения корректного начального состояния объекта?

13. Приведите пример ошибки, которая может возникнуть при попытке прямого доступа к `private` члену R6-класса извне.

14. Как R6 позволяет контролировать, какие атрибуты и методы являются доступными для внешнего использования, а какие - только для внутренней логики класса?

15. Какие общие правила именования используются для `private` атрибутов в R6 (и в R в целом)?

16. Объясните, как принцип инкапсуляции способствует модульности и повторному использованию кода в биоинформатических проектах.

17. Может ли `initialize()` быть приватным методом в R6? Объясните свой ответ.

18. Как вы будете отлаживать R6-класс, если подозреваете проблему с инкапсулированным атрибутом?

19. Рассмотрите сценарий, где необходимо контролировать форматирование входных данных для атрибута класса (например, последовательность ДНК должна быть только из ACGT). Как инкапсуляция и сеттеры могут помочь в этом?

20. Объясните, как "защищенные" (`protected`) атрибуты могут быть имитированы в R6, и чем они отличаются от строго "приватных" (`private`) атрибутов.

Раздел 3. Наследование и композиция в R6.

1. Дайте определение наследования в ООП. Какова основная цель использования наследования?

2. Как реализовать наследование в R6? Объясните роль аргумента `inherit` в функции `R6Class()`.

3. Что такое "переопределение метода" (`methodoverriding`) в контексте наследования? Приведите пример, когда это необходимо.

4. Объясните назначение ключевого слова `super` в R6. Когда и для чего его следует использовать?

5. Рассмотрите сценарий: у вас есть R6-класс `BiologicalEntity` и от него наследуется `Animal`. Объясните, как `Animal` может расширять функционал `BiologicalEntity` и переопределять его методы.

6. В чем заключается концепция "композиции" в ООП? Как она отличается от наследования?

7. Приведите примеры отношений "is-a" (наследование) и "has-a" (композиция) в контексте моделирования биологических или животноводческих данных.

8. Когда следует отдавать предпочтение композиции перед наследованием? Приведите конкретные сценарии использования.
9. Обсудите концепцию "множественного наследования". Как R6 подходит к этой проблеме, и какие альтернативы существуют в R?
10. Разработайте концептуальную иерархию R6-классов для генеалогического древа (например, Organism ->Mammal ->Cattle). Какие атрибуты и методы могут быть общими, а какие специфическими?
11. Как композиция может быть использована для создания сложного R6-класса "АналитическийПайплайн", который включает в себя объекты для различных этапов анализа (например, DataReader, Normalizer, Visualizer)?
12. Объясните, как наследование способствует повторному использованию кода и расширяемости системы.
13. Какие проблемы могут возникнуть при чрезмерном использовании наследования (например, "иерархия классов-монстров")? Как композиция помогает избежать этих проблем?
14. Может ли дочерний класс в R6 получить доступ к приватным атрибутам родительского класса? Обоснуйте.
15. Какие механизмы в R6 позволяют дочернему классу вызывать методы родительского класса, которые были переопределены?
16. Обсудите, как проектирование классов с учетом наследования и композиции может улучшить управляемость и тестируемость больших биоинформатических проектов.
17. Приведите пример из реальной биоинформатики, где использование композиции будет более уместным, чем наследования (например, модуль для чтения файлов, модуль для статистического анализа).
18. Какие возможности R6 облегчают интеграцию классов, построенных на принципах наследования и композиции, в существующие R-пакеты?
19. Объясните, как принцип "приоритета композиции перед наследованием" ("prefercompositionoverinheritance") применяется в современном ООП-дизайне.
20. Как R6-классы обеспечивают гибкость при изменении поведения объектов, построенных на принципах наследования и композиции?

Раздел 4. Полиморфизм, абстракция и продвинутые концепции R (S3, S4). Применение ООП в биоинформатике.

1. Что такое полиморфизм в ООП? Объясните, как этот принцип позволяет использовать единый интерфейс для различных типов объектов.

2. Как полиморфизм реализуется в R6-классах? Приведите пример, демонстрирующий, как один и тот же метод может иметь разное поведение в разных классах.
3. Опишите концепцию "generic-функций" в S3 ООП. Как S3-классы и методы взаимодействуют с ними для достижения полиморфизма?
4. В чем заключаются основные отличия между полиморфизмом, реализованным в S3, S4 и R6?
5. Объясните, как `setClass()` и `setMethod()` используются для определения классов и методов в S4 ООП. В чем S4 превосходит S3?
6. Дайте определение "абстракции" в ООП. Почему абстракция важна для управления сложностью в больших системах?
7. В R6 нет прямого механизма "абстрактных классов" как в Java. Как можно имитировать абстрактные классы в R6 для создания общих интерфейсов, которые должны быть реализованы дочерними классами?
8. Приведите пример полиморфизма в биоинформатике, где одна функция (например, `plot()`) может быть использована для визуализации разных типов данных (например, ДНК-последовательности, матрицы экспрессии).
9. Какие преимущества дает использование S4-классов и методов в пакетах Bioconductor для работы с геномными данными?
10. Как принцип "перегрузки операторов" реализуется в R (на основе S3/S4) и как это связано с полиморфизмом?
11. Объясните, как ООП в R (особенно R6) может быть использовано для создания инкапсулированных и модульных биоинформатических пайплайнов.
12. Разработайте концептуальный дизайн R6-класса для представления "Биологического Пути" (например, сигнальный путь, метаболический путь), который абстрагирует детали его компонентов.
13. Приведите пример использования S3-классов для работы с результатом статистического анализа (например, `lm` объект), где различные функции (`summary()`, `plot()`) полиморфно адаптируют свое поведение.
14. Как концепция "слотов" в S4-классах способствует более строгой структурированности данных по сравнению с S3-списками?
15. Какие существуют подходы для обеспечения валидации данных при создании объектов S4-классов?
16. Обсудите, как полиморфизм и абстракция способствуют гибкости при разработке программного обеспечения для анализа больших объемов геномных данных.

17. Как вы будете выбирать между использованием S3, S4 или R6 для конкретной биоинформатической задачи, учитывая их особенности в полиморфизме и абстракции?

18. Приведите пример R-пакета из Bioconductor, который активно использует S4 ООП, и объясните, как это помогает в управлении сложными биологическими данными.

19. Какие общие паттерны проектирования (например, "фабрика", "стратегия") могут быть реализованы с использованием полиморфизма в R6 для биоинформатических приложений?

20. Объясните, как внедрение ООП в R (с учетом различных парадигм) способствует более эффективной коллаборации и обмену кодом в научном сообществе биоинформатиков.

Критерии оценки текущего контроля (коллоквиум, собеседование)

Оценка «отлично» выставляется магистранту, если он демонстрирует высокий уровень освоения материала, проявляя способность:

- Полностью и развёрнуто отвечать на поставленный вопрос, демонстрируя глубокое понимание темы.
- Синтезировать знания из различных разделов дисциплины и смежных областей, выявлять причинно-следственные связи.
- Аргументированно формировать собственное суждение, оценивать информацию, предлагать возможные варианты решения проблем.
- Использовать корректную научную терминологию, излагать материал логично, последовательно, без ошибок, быстро реагировать на уточняющие вопросы.
- Приводить примеры, иллюстрирующие теоретические положения, применять новые научные принципы и методы исследований.

Оценка «хорошо» выставляется магистранту, если он демонстрирует достаточный уровень освоения материала, проявляя способность:

- Полностью отвечать на поставленный вопрос, выделять существенные признаки, причинно-следственные связи.
- Излагать материал структурированно и логично, использовать терминологию науки.
- Допускать незначительные ошибки или недочёты, которые исправляются самостоятельно или с помощью наводящих вопросов.
- Приводить примеры, но их количество или глубина анализа могут быть недостаточными.

Оценка «удовлетворительно» выставляется магистранту, если он демонстрирует пороговый уровень освоения материала, проявляя способность:

- Давать неполный ответ, с существенными нарушениями логики и последовательности изложения.
- Допускать грубые ошибки в определении сущности понятий, демонстрировать непонимание существенных и несущественных признаков.
- Испытывать затруднения в формировании выводов и конкретных проявлений обобщённых знаний.
- Требовать постоянных наводящих вопросов и коррекции.

Оценка «неудовлетворительно» выставляется магистранту, если он демонстрирует недопустимый уровень освоения материала, проявляя:

- Незнание теоретических основ, неспособность выделять аргументированные выводы и приводить примеры.
- Слабое владение монологической речью, отсутствие логичности и последовательности изложения.
- Множественные ошибки, которые не могут быть исправлены даже при помощи преподавателя, отказ отвечать.

3. Выполнение контрольной работы

Контрольная работа включает в себя четыре вопроса – три теоретических и один практический. Контрольная работа оформляется в виде реферата с обязательным приведением списка цитированной литературы. Реферат должен содержать следующие разделы: титульный лист, содержание, 4 раздела (по одному разделу на каждый вопрос варианта), заключение по 3-4 вопросу варианта, список использованной литературы. Оценка «зачтено» ставится, если работа выполнена самостоятельно и содержит авторское изложение и анализ источников литературы.

Вариант 1

1. Опишите принципы объектно-ориентированного программирования (ООП). Объясните, почему ООП считается одной из основных парадигм современного программирования, и в чем его преимущества по сравнению с процедурным подходом.

2. Объясните концепцию инкапсуляции в R6-классах. Опишите, как R6 обеспечивает инкапсуляцию с помощью `public`, `private` и `active` привязок. Приведите пример R6-класса, демонстрирующего использование этих концепций для защиты данных.

3. Что такое наследование в ООП? Опишите, как R6 реализует наследование, и какие возможности (переопределение методов, использование `super`) оно предоставляет. Сравните наследование с композицией и укажите, когда следует использовать каждый подход.

4. Практическое задание: Разработайте и опишите R6-класс `DNASequence` для представления ДНК-последовательностей. Класс должен иметь следующие возможности:

- Приватный атрибут: `sequence` (символьная строка, например, "ATGC").
- Публичный атрибут: `id` (идентификатор последовательности).
- Активная привязка (getter): `length` (длина последовательности).
- Метод: `get_gc_content()` для расчета процентного содержания GC.
- Метод: `reverse_complement()` для получения обратной комплементарной последовательности.

Продемонстрируйте создание объекта класса, доступ к публичным атрибутам и активным привязкам, а также вызов методов. Объясните, как реализована инкапсуляция.

Вариант 2

1. Сравните три основные парадигмы ООП в R: S3, S4 и R6. Опишите их отличительные особенности, синтаксис и типичные сценарии использования. Укажите, в чем R6 отличается от "нативных" S3/S4 систем R.

2. Что такое полиморфизм в ООП? Опишите, как полиморфизм реализуется в R с помощью R6-классов (через общие имена методов). Приведите примеры кода, демонстрирующие полиморфное поведение для различных классов биологических объектов.

3. Объясните понятие абстракции в ООП. Почему абстракция важна для создания гибкого и расширяемого кода? Как можно реализовать "абстрактные" концепции в R6, учитывая отсутствие прямых механизмов абстрактных классов, как в других языках?

4. Практическое задание: Вам необходимо создать систему классов R6 для моделирования данных об экспрессии генов.

Создайте базовый класс `ExpressionData` с атрибутами для матрицы экспрессии (`counts_matrix`, приватный) и метаданных образцов (`sample_metadata`, публичный).

Реализуйте метод `normalize()` в `ExpressionData`, который выполняет базовую нормализацию (например, логарифмирование).

Создайте дочерний класс `RNASeqData`, который наследует от `ExpressionData`.

В `RNASeqData` добавьте специфический метод `get_top_genes(n)` для выбора N наиболее экспрессирующихся генов.

Продемонстрируйте создание объектов обоих классов, вызов унаследованных и специфических методов. Объясните преимущества такого ООП-дизайна для управления данными экспрессии.

Вариант 3

1. Что такое атрибуты (поля) и методы в ООП? Как они используются для описания состояния и поведения объектов? Приведите примеры атрибутов и методов для гипотетического класса "Генетический_вариант".

2. Объясните, почему объектно-ориентированное программирование особенно полезно в биоинформатике. Приведите не менее трех конкретных примеров задач в биоинформатике, где ООП-подход может быть эффективен.

3. Опишите роль конструктора (`initialize()` в R6) при создании объектов. Какие типичные действия выполняются в конструкторе? В чем разница между `public` и `private` полями в R6 и как они влияют на инкапсуляцию?

4. Практическое задание: Разработайте и опишите R6-класс ProteinSequence для представления белковых последовательностей. Класс должен иметь:

- Приватный атрибут: `sequence` (символьная строка, например, "AGCTAGCT").
- Публичный атрибут: `accession_id` (идентификатор доступа к белку).
- Активная привязка (getter): `molecular_weight` (вычисляемое молекулярная масса на основе последовательности, используйте упрощенный расчет).
- Метод: `has_motif(motif)` для проверки наличия заданного мотива.
- Метод: `translate_to_dna()` (для демонстрационных целей, пусть будет заглушка, возвращающая сообщение).

Продемонстрируйте создание объекта, доступ к его свойствам и вызов методов. Объясните, как вы обеспечили инкапсуляцию данных.

Вариант 4

1. В чем заключается роль активных привязок (active) в R6-классах? Объясните, как они позволяют реализовывать геттеры и сеттеры и для чего это необходимо. Приведите примеры их использования.

2. Объясните концепцию композиции в ООП. Чем она отличается от наследования? Приведите пример, когда композиция более предпочтительна для построения сложной структуры данных в R (например, "Геном", состоящий из множества "Хромосом").

3. Опишите процесс отладки ООП-кода в R. Какие инструменты и подходы (например, `debug()`, `browser()`, `RStudiodebugger`) можно использовать для выявления и исправления ошибок в R6-классах?

4. Практическое задание: Создайте систему R6-классов для работы с данными генеалогического древа животных.

Создайте базовый класс `Animal` с атрибутами `name`, `species` (оба публичные).

Создайте дочерний класс `FarmAnimal` с наследованием от `Animal` и добавьте специфические атрибуты `tag_id` (публичный) и `birth_date` (приватный). Реализуйте активную привязку `age` (в годах) для `FarmAnimal`, которая вычисляется на основе `birth_date`.

В `FarmAnimal` реализуйте метод `is_older_than(other_animal)`, который сравнивает возраст текущего животного с другим объектом `FarmAnimal`.

Продemonстрируйте создание объектов, их инициализацию, доступ к атрибутам и вызов методов. Покажите примеры использования метода `is_older_than`.

4. Промежуточная аттестация

Вопросы для зачета с оценкой

1. Дайте определение объектно-ориентированного программирования (ООП). Назовите его основные принципы.
2. Объясните различия между классом и объектом в контексте ООП. Приведите примеры.
3. В чем заключаются основные преимущества ООП по сравнению с процедурным подходом к программированию?
4. Перечислите три основные парадигмы ООП, используемые в языке R.
5. Объясните концепцию "обобщенных функций" (generic functions) в S3 ООП. Как S3-классы и методы взаимодействуют с ними?
6. Какие характеристики присущи S4 ООП, которые отличают его от S3 (например, строгая типизация, слоты)?
7. Почему R6-классы часто называют "ссылочными классами" (reference classes)? В чем заключается их основное отличие от S3/S4 в этом аспекте?
8. Как создать новый R6-класс с использованием функции R6Class()? Приведите базовый синтаксис определения класса.
9. Объясните роль метода initialize() в R6-классе. Для чего он используется и какие параметры может принимать?
10. Дайте определение инкапсуляции как одного из принципов ООП. Почему инкапсуляция важна для создания надежного ПО?
11. Как R6 обеспечивает инкапсуляцию с помощью public и private членов класса? Продемонстрируйте их объявление.
12. В чем основное отличие между public и private атрибутами R6-класса? Когда следует объявлять атрибут private?
13. Что такое "активные привязки" (active) в R6? Как они позволяют реализовывать геттеры (методы чтения) и сеттеры (методы записи)?
14. Приведите пример реализации геттера с использованием active привязки для приватного атрибута в R6-классе.
15. Как реализовать сеттер с помощью active привязки, который будет выполнять валидацию входных данных?
16. Объясните, почему прямой доступ к приватным атрибутам R6-класса извне нежелателен и как он обходится в R.
17. Дайте определение наследования в ООП. Какие отношения между классами оно описывает (например, "is-a")?

18. Опишите, как реализовать наследование в R6, используя аргумент `inherit` в `R6Class()`.
19. Что такое "переопределение метода" (`methodoverriding`) в контексте наследования? Приведите пример.
20. Объясните назначение ключевого слова `super` в R6. Когда и для чего оно используется в дочерних классах?
21. Дайте определение композиции в ООП. Какие отношения между классами она описывает (например, "has-a")?
22. Сравните наследование и композицию. В каких случаях предпочтительнее использовать каждый подход? Приведите примеры.
23. Как R6 подходит к концепции "множественного наследования"? Какие альтернативы существуют в R?
24. Дайте определение полиморфизма в ООП. Как этот принцип позволяет использовать единый интерфейс для различных типов объектов?
25. Как полиморфизм реализуется в R6-классах через использование методов с одинаковыми именами, но разной реализацией в разных классах?
26. Объясните, как S3-механизм полиморфизма работает с помощью `UseMethod()`.
27. В чем заключается основной принцип S4-полиморфизма и как он использует `setMethod()`?
28. Дайте определение абстракции в ООП. Почему абстракция важна для управления сложностью в больших системах?
29. Как можно имитировать "абстрактные классы" или "интерфейсы" в R6, учитывая отсутствие прямых механизмов, как в других языках?
30. Какие преимущества дает использование R6-классов для моделирования биологических объектов, таких как ДНК-последовательности, белки или генетические варианты?
31. Опишите, как R6-классы могут быть использованы для создания инкапсулированных и модульных биоинформатических пайплайнов.
32. Приведите пример применения полиморфизма в биоинформатике (например, для визуализации различных типов данных).
33. Как пакет `Bioconductor` активно использует S4 ООП для создания структурированных объектов, таких как `GenomicRanges` или `SummarizedExperiment`?
34. Объясните, как принцип "перегрузки операторов" реализуется в R (на основе S3/S4) и как это связано с полиморфизмом.
35. Какие инструменты профилирования (например, `Rprof`, `profvis`) доступны в R и как они помогают оптимизировать производительность ООП-кода?

36. Объясните роль документации (например, с использованием roxygen2) для R6-классов и их методов.

37. Как системы контроля версий (например, Git) способствуют эффективной командной разработке ООП-проектов на R?

38. Какие механизмы обработки ошибок в R (tryCatch, stop, warning) следует использовать при разработке ООП-приложений?

39. Объясните, почему ООП в R особенно полезно для разработки сложных информационных систем в животноводстве.

40. Какие преимущества дает использование RStudio как среды разработки при работе с R6-классами (например, отладка, инспектирование объектов)?

41. Как метод \$set() может быть использован для динамического изменения полей R6-объекта после его создания?

42. Опишите метод clone() в R6-классах. Для чего он используется и в чем его важность при работе со ссылочными объектами?

43. Какие общие правила именования используются для приватных атрибутов в R6?

44. Могут ли приватные методы вызывать публичные методы того же R6-класса? Обоснуйте.

45. Объясните, как принцип инкапсуляции помогает предотвратить "нежелательное" изменение состояния объекта извне.

46. В чем заключается важность использования конструктора (initialize()) для обеспечения корректного начального состояния объекта?

47. Как наследование способствует повторному использованию кода и снижает дублирование?

48. Объясните, как композиция может обеспечить большую гибкость по сравнению с наследованием при изменении внутренней структуры компонентов.

49. Какие проблемы могут возникнуть при слишком глубокой или сложной иерархии наследования ("иерархия классов-монстров")?

50. Обсудите, как проектирование классов с учетом наследования и композиции может улучшить управляемость и тестируемость больших биоинформатических проектов.

51. Какие возможности R (помимо ООП) дополняют ООП-подход для работы с большими биоинформатическими данными (например, параллельные вычисления, data.table)?

52. Приведите пример R-пакета из Bioconductor, который активно использует S4 ООП, и объясните, как это помогает в управлении сложными биологическими данными.

53. Объясните, как внедрение ООП в R (с учетом различных парадигм) способствует более эффективной коллаборации и обмену кодом в научном сообществе биоинформатиков.

54. Какие основные шаги необходимо предпринять для создания R-пакета, содержащего R6-классы, для распространения среди других пользователей?

55. Как R6-классы поддерживают принцип "прозрачности" (transparency) при использовании наследования и композиции?

56. В чем заключаются преимущества использования активных привязок (геттеров/сеттеров) по сравнению с прямым доступом к публичным атрибутам?

57. Опишите, как можно обеспечить валидацию данных при создании S4-объектов.

58. Какие преимущества дает модульность, достигаемая благодаря ООП, при разработке сложных аналитических пайплайнов в биоинформатике?

59. Объясните, как `message()`, `warning()` и `stop()` используются для информирования пользователя и обработки исключительных ситуаций внутри методов R6-класса.

60. Как системный подход к разработке информационных систем в животноводстве может быть усилен принципами ООП?

Критерии оценки на зачете

Основные критерии оценки знаний по дисциплине: глубина, систематичность, конкретность, осознанность, логичность и четкость изложения, полнота и прочность знаний программного материала.

- Глубина – характеризует осознание магистрантом связей между изучаемыми объектами при решении проблемной ситуации исследовательского характера.

- Систематичность – предполагает последовательность и логическое построение всей совокупности знаний по изучаемой дисциплине.

- Конкретность — связана с умением конкретизировать задачу, пользуясь обобщенными знаниями.

- Осознанность – восприятие знаний в их логической взаимосвязи.

Оценка «отлично» выставляется магистранту, если он демонстрирует высокий уровень освоения компетенций, проявляя способность:

- Полностью и развёрнуто отвечать на поставленный вопрос, демонстрируя глубокое и осознанное понимание всей совокупности знаний по дисциплине.

- Систематизировать и критически анализировать сложную информацию, выделять ключевые аспекты, выявлять и формулировать актуальные научные проблемы, предлагать стратегии их решения.

- Применять новые научные принципы и методы исследований, оценивать достоверность и воспроизводимость результатов, а также применимость и эффективность информатических методов.

- Излагать материал чётко структурировано, логично, используя литературный язык и научную терминологию, быстро реагировать на уточняющие вопросы, демонстрируя владение культурой научного поиска и критического анализа.

Оценка «хорошо» выставляется магистранту, если он демонстрирует достаточный уровень освоения компетенций, проявляя способность:

- Давать полный, развёрнутый ответ на поставленный вопрос, умение выделить существенные и несущественные признаки, причинно-следственные связи.

- Излагать ответ чётко структурированно, логично, в терминах науки.

- Применять полученные знания для анализа проблемных ситуаций, оценивать альтернативные подходы, но с допущением незначительных ошибок или недочётов, которые исправляются с помощью наводящих вопросов.

- В целом демонстрирует владение необходимыми умениями и знаниями, но с меньшей глубиной анализа или аргументации по сравнению с «отличной» оценкой.

Оценка «удовлетворительно» выставляется магистранту, если он демонстрирует пороговый уровень освоения компетенций, проявляя способность:

- Давать неполный ответ, с существенными нарушениями логики и последовательности изложения.

- Допускать грубые ошибки при определении сущности раскрываемых понятий, явлений, в следствии непонимания их существенных и несущественных признаков и связей.

- Не демонстрирует умения раскрыть конкретные проявления обобщённых знаний, испытывает затруднения в критическом анализе или формировании стратегий.

- Речевое оформление требует поправок, коррекций; проявляет затруднения в применении биоинформатических подходов.

Оценка «неудовлетворительно» выставляется магистранту, если он демонстрирует недопустимый уровень освоения компетенций, проявляя:

– Незнание теоретических основ предмета, не умеет выделять аргументированные выводы и приводить примеры.

– Слабое владение монологической речью, отсутствие логичности и последовательности изложения, делает ошибки, которые не может исправить даже при коррекции преподавателем, отказывается отвечать на занятия.

– Неспособность применять полученные знания для решения проблемных ситуаций, отсутствует понимание роли биоинформатики в области животноводства.

Объектно-ориентированное программирование

Приложение

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФГБОУ ВО Университет биотехнологий

Институт цифровых технологий

Кафедра прикладной биоинформатики

Контрольная работа

по дисциплине: «Объектно-ориентированное программирование»

Тема: «...»

Выполнил(а): студент(ка) ... курса

группы ...

ФИО

Проверил: степень,

ФИО

Новосибирск 202...

Составители:

Камалдинов Евгений Варисович

Меркушкина Мария Петровна

....

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Методические указания по выполнению
практических занятий, контрольной и самостоятельной работы

Компьютерная верстка Меркушкина М.П.

ФГБОУ ВО Университет биотехнологий
630039, г. Новосибирск, ул. Добролюбова, 160