

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«Новосибирский национальный исследовательский государственный
университет»
(Новосибирский государственный университет, НГУ)

Факультет естественных наук

УТВЕРЖДАЮ
Декан ФЕН
д.х.н. проф. В.А. Резников

« _____ » _____ 201__ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

по дисциплине

Информационные технологии и языки программирования-II

Направление подготовки: 06.04.01 БИОЛОГИЯ, МАГИСТРАТУРА

Кафедра информационной биологии

Новосибирск 2020

Фонд оценочных средств промежуточной аттестации по дисциплине является **Приложением 2** к рабочей программе дисциплины «**Информационные технологии и языки программирования-II**», реализуемой в рамках основной образовательной программы высшего профессионального образования по направлению подготовки: 06.04.01 Биология, магистратура

Фонд оценочных средств промежуточной аттестации по дисциплине утвержден решением ученого совета Факультета естественных наук № _____ от _____.

Разработчики:

к.б.н., доцент кафедры информационной биологии

Лашин С.А.

Ф.В. казанцев

Ответственный за образовательную программу:

Доцент кафедры информационной биологии,

доктор биологических наук,

Д.П. Фурман

1. Содержание и порядок проведения промежуточной аттестации по дисциплине

1.1. Общая характеристика содержания промежуточной аттестации

Промежуточная аттестация по дисциплине «**Информационные технологии и языки программирования-II**» проводится по завершению периода освоения образовательной программы 2 семестра для оценки сформированности компетенций в части следующих укрупненных характеристик результатов обучения (таблица П1.1).

Таблица П1.1

Код	Компетенции, формируемые в рамках дисциплины « Информационные технологии и языки программирования-II »	Семестр 2
		дифференцированный зачет
ОПК-4	Способность самостоятельно анализировать имеющуюся информацию, выявлять фундаментальные проблемы, ставить задачу и выполнять полевые, лабораторные биологические исследования при решении конкретных задач с использованием современной аппаратуры и вычислительных средств, нести ответственность за качество работ и научную достоверность результатов.	
	Уметь проектировать и реализовывать программы на языке Java, владеть современными методами программирования и некоторыми методами проектирования программных продуктов	+
ОПК-7	Готовность творчески применять современные компьютерные технологии при сборе, хранении, обработке, анализе и передаче биологической информации для решения профессиональных задач	
	Знать основные форматы представления биологических данных (FASTA, Genbank и другие), - принципы построения программных конвейеров (pipe-line) и используемые при этом технологии, владеть приемами моделирования биологических процессов, а также способами оценки валидности разработанных моделей	+
ПК-1	Способность творчески использовать в научной и производственно-технологической деятельности знания фундаментальных и прикладных разделов дисциплин (модулей), определяющих направленность (профиль) программы магистратуры	
	Уметь использовать теоретические знания и экспериментальные навыки для самостоятельного планирования и проведения эксперимента, анализа и оформления полученных результатов	+
ПК-4	Способность генерировать новые идеи и методические решения	
	Уметь творчески представлять результаты биоинформатического анализа с помощью современных средств визуализации,	+

	владеть приемами и методами для выполнения и решения новых идей	
--	---	--

Компетенции оцениваются дифференцированным зачетом, который включает в себя решения задач из всех разделов дисциплины **«Информационные технологии и языки программирования-II»**:

1.2. Порядок проведения аттестации по дисциплине

По дисциплине «Информационные технологии и языки программирования-II» проводится текущая и промежуточная аттестация.

Текущая аттестация

Текущий контроль успеваемости осуществляется путем контроля выполнения практических заданий, на базе которых создаётся оценочное портфолио для каждого обучающегося.

Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется в форме дифференцированного зачёта и проводится в два этапа:

1) Оценочное портфолио по результатам работы в семестре, которое включает выполненные практические задания.

2) Устный ответ на вопросы экзаменационного билета.

Оценка **"удовлетворительно"** выставляется, если студент уверенно отвечает на один вопрос из экзаменационного билета и имеет в своём оценочном портфолио не менее 60% решённых задач, включая задачу из Темы 3. Решение этих задач позволяют проверить знание и умение работать с базовыми типами данных, структурами языка программирования Java и основными концепциями Объектно-Ориентированного Программирования.

Оценка **"хорошо"** выставляется, если студент уверенно отвечает на вопросы из экзаменационного билета и имеет в своём оценочном портфолио не менее 80% решённых задач, включая задачу из Темы 3. Это свидетельствует об умении структурного проектирования алгоритмов с вовлечением различных аспектов возможностей языка программирования Java;

Оценка **"отлично"** выставляется, если студент уверенно отвечает на вопросы из экзаменационного билета и имеет в своём оценочном портфолио 100% решённых задач. Это свидетельствует об умении структурировать решение задач и умение работать со сторонними библиотеками, проводить анализ документации этих библиотек.

2. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Описание критериев оценивания индикаторов достижения результатов обучения по дисциплине Информационные технологии и языки программирования-II

Код компетенции	Результат обучения по дисциплине	Оценочное средство
ОПК-4	Умение проектировать и реализовывать программы на языке Java	Оценочное портфолио Дифференцированный зачет

	Владение современными методами программирования и некоторыми методами проектирования программных продуктов	Оценочное портфолио Дифференцированный зачет
ОПК-7	Знание основных форматов представления биологических данных (FASTA, Genbank и другие)	Оценочное портфолио Дифференцированный зачет
	Знание принципов построения программных конвейеров (pipe-line) и используемых при этом технологий.	Оценочное портфолио Дифференцированный зачет
	Владение приемами моделирования биологических процессов, а также способами оценки валидности разработанных моделей	Оценочное портфолио Дифференцированный зачет
ПК-1	Умение использовать теоретические знания и экспериментальные навыки для самостоятельного планирования и проведения эксперимента, анализа и оформления полученных результатов	Оценочное портфолио Дифференцированный зачет
ПК-4	Умение творчески представлять результаты биоинформатического анализа с помощью современных средств визуализации	Оценочное портфолио Дифференцированный зачет
	Умение комбинировать известные алгоритмы для получения качественно новых решений	Оценочное портфолио Дифференцированный зачет
	Владение приемами и методами для выполнения и решения новых иде	Оценочное портфолио Дифференцированный зачет

Описание шкал оценивания индикаторов достижения результатов обучения по дисциплине Информационные технологии и языки программирования-II

Критерии оценивания результатов обучения	Шкала оценивания
<p><u>Оценочное портфолио</u> – 100% решённых задач.</p> <p><u>Дифференцированный зачет</u> – полнота понимания и изложения причинно-следственных связей, – самостоятельность, осмысленность, структурированность, логичность и аргументированность изложения материала, отсутствие затруднений в его объяснении, а также при формулировке собственных суждений, – точность и корректность применения терминов и понятий, – наличие исчерпывающих ответов на дополнительные вопросы.</p>	<i>Отлично</i>
<p><u>Оценочное портфолио</u> – 80% решённых задач, , включая задачу из Темы 3..</p> <p><u>Дифференцированный зачет</u> – полнота понимания и изложения причинно-следственных связей, – самостоятельность, осмысленность, структурированность, логичность и аргументированность изложения материала, наличие затруднений в объяснении отдельных его частей, а также при формулировке собственных суждений, – точность и корректность применения терминов и понятий при наличии незначительных ошибок, – наличие полных ответов на дополнительные вопросы с возможным присутствием ошибок.</p>	<i>Хорошо</i>

<p><u>Оценочное портфолио</u> – не менее 60% решённых задач, включая задачу из Темы 3.</p> <p><u>Дифференцированный зачет</u> – студент уверенно отвечает на один вопрос из экзаменационного билета , – частичное понимание и неполное изложение причинно-следственных связей, – корректность применения терминов и понятий при наличии незначительных ошибок, – наличие неполных и/или содержащих существенные ошибки ответов на дополнительные вопросы.</p>	<i>Удовлетворительно</i> 0
<p><u>Оценочное портфолио</u> – менее 60% решённых задач, – нерешенная задача из Темы 3.</p> <p><u>Дифференцированный зачет</u> – студент не отвечает на один вопрос из экзаменационного билета – непонимание причинно-следственных связей, – отсутствие осмысленности, структурированности, логичности и аргументированности в изложении материала, – грубые ошибки в применении терминов и понятий, – отсутствие ответов на дополнительные вопросы.</p>	<i>Неудовлетворительно</i>

2.1. Критерии выставления оценок по результатам промежуточной аттестации по дисциплине

Результаты промежуточной аттестации по «Информационные технологии и языки программирования-1» в 1 семестре определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Оценка «отлично» соответствует продвинутому уровню сформированности компетенции.

Оценка «хорошо» соответствует базовому уровню сформированности компетенции.

Оценка «удовлетворительно» соответствует пороговому уровню сформированности компетенции.

Оценка «неудовлетворительно» выставляется, если компетенция не сформирована.

2.2.. Типовые контрольные задания и иные материалы, необходимые для оценки результатов обучения по дисциплине «Информационные технологии и языки программирования-1»

Примеры задач:

Тема: Основы программирования на языке Java.

Задача 1: Создание проекта программы в среде разработки Eclipse. Цель – освоить базовые настройки среды разработки, освоить правила структурирования файлов проекта, освоить инструменты запуска программ, их отладки и способы вывода информации исполнения.

Задача 2: Вывод в консоль последовательностей символов в заданном порядке. Цель - отработка навыка работы с базовыми типами данных, операторами ветвления и циклами. Освоить создание и использование методов, передачи параметров в методы и получение значений из них. Освоить работу с консолью программы, аргументами командной строки.

Тема: Формализация алгоритмов в виде блок-схем.

Задача 1: создание серии алгоритмов обработки данных. Формализация алгоритмов в виде блок-схемы. Цель – получить навыки планирования решения задачи.

Тема: Основы объектно-ориентированного программирования. Классы и объекты.

Задача 1: Разобрать отношения в рамках выбранной предметной области в концепции ООП (выбор предметной области обсуждается на занятии). Описать классы, формирующие предметную область. Задать отношения между ними, применяя концепции инкапсуляции, наследования и полиморфизма. Выбрать методы классов, которые могут быть переопределены. Сформулировать в виде блок-схемы варианты алгоритмов обработки данных предметной области и реализовать. Цель – освоить концепцию объектно-ориентированного дизайна

Тема: Массивы и контейнеры. Работа со строками.

Задача 1: В рамках созданной структуры классов на предыдущих занятиях переопределить/доопределить свойства и методы класса с использованием типов данных «массив», «контейнер» и «строки». Цель – научиться выбирать подходящие типы данных для моделируемых объектов.

Тема: Работа с файлами, форматы биоинформатических данных.

Задача 1: Чтение текстовых файлов с геномными данными. Разбор данных в файле с использованием инструментов обработки строк, их переработка и сохранение в новых файлах. Инструменты поиска в строковых данных. Цель – понять устройство файлов и научиться самим разбирать и искать информацию в структурированных текстовых файлах.

Задача 2: Чтение геномных файлов с использованием профильных инструментов (BioJava). Цель – освоить использование сторонних библиотек, организованных в терминах ООП.

Тема: Работа с сетью и REST-запросы к биоинформатическим ресурсам.

Задача 1: Подключение к REST API банка данных KEGG и извлечение информации по выбранным организмам (Выбор организма обсуждается на занятии). Цель – освоить принципы обмена данными через REST API.

Тема: Визуализация в анализе биоинформатических данных.

Задача 1: Визуализация результатов вычислений программ, выполненных на предыдущих занятиях в рамках инструмента Excel, Python, Matplotlib, GnuPlot.

Тема: Организация программных пакетов и запуск сторонних программ.

Задача 1: построить конвейер обработки данных из разработанных на предыдущих занятиях инструментов, с привлечением сторонних программных инструментов. Цель – освоить запуск серии программ из консоли. Научится формулировать задачи обработки данных в рамках последовательных этапов. Понять, что для решения задачи возможно использовать готовые программы, внедряя свои инструменты в ключевых местах обработки данных.

Тема: Подходы автоматического тестирования программных инструментов биоинформатики и системной биологии.

Задача 1: реализовать серию тестов для разработанных на предыдущих занятиях программ.

Тема: Высокопроизводительные вычисления и многопоточность.

Задача 1: Формулирование блок-схемы алгоритма, который может быть переработан в концепцию «высокопроизводительных вычислений».

Задача 2: Реализация алгоритма параллельного исполнения в рамках потоков.

Типовые вопросы и задачи дифференцированного зачета:

1) Введение, рассказ об языках программирования: какие бывают, чем отличаются, почему выбран именно Java, и чем он может быть полезен в решении задач биоинформатики и системной биологии.

2) Установка и настройка версии языка программирования, среды разработки программ, запуск первых приложений.

3) Рассказ о базовых типах данных в Java (числа, строки и т.д.), ключевых особенностях и потенциальных возможностях при работе с ними. Способы ввода данных в программу.

4) Операторы языка Java. Логические операторы, ветвление, циклы, присваивание.

5) Введение в построение блок-схем: что такое блок-схемы, и зачем их создавать до начала программирования. Правила построения.

6) Классические алгоритмы биоинформатики: разбор блок-схем алгоритмов биоинформатического анализа

7) Что такое парадигма объектно-ориентированного программирования (ООП). Концепции абстракции, инкапсуляции, модульности, полиморфизма и наследования.

8) Время жизни и область видимости переменных.

9) Сборка мусора.

10) Документирование Javadoc.

11) Тип данных «массив», инициация, индексация и базовые методы работы с массивами.

12) Контейнеры Java и библиотека Java Collections. Принципы устройства и встроенные методы работы с коллекциями. Интерфейсы и содержимое контейнеров.

13) Тип данных «строка». Методы работы со строками и примеры использования в работе геномными данными.

14) Типы файлов. Инструменты чтения/записи файлов. Поток ввода/вывода в Java. Форматы файлов, используемых инструментами биоинформатики и системной биологии.

15) Исключения в Java. Генерация, перехват и обработка исключений.

16) Протоколы обмена данными в глобальной сети Интернет. Протокол REST как стандартный инструмент обращения к банкам данных.

17) Методы HTTP запроса и принципы организации простого HTTP-клиента. Инструменты библиотеки Java Net.

18) REST API биоинформатических банков данных.

19) Типичные варианты визуализации данных биоинформатики и системной биологии.

20) Форматы визуализации.

21) Инструменты визуализации биоинформатических данных.

22) Подход к решению задач биоинформатики с привлечением сторонних программ.

23) Оформление программы как исполняемого файла или библиотеки. Интеграция программ на уровне консоли. Запуск любых программ из кода языка Java.

24) Подходы и инструменты автоматического тестирования программ. Концепция модульного тестирования.

25) Библиотека JUnit.

26) Технология разработки программного обеспечения на основе модульного тестирования (TDD).

27) понятие «высокопроизводительные вычисления». Поддерживающие такие вычисления платформы и принципы организации вычислений.

28) Параллелизация вычислений на уровне данных и на уровне исполнения программ. Диаграммы потока исполнения программы.

29) Инструменты Java для параллелизации вычислений.

Задача 1: Создание проекта программы в среде разработки Eclipse.

Задача 2: Вывод в консоль последовательностей символов в заданном порядке.

Задача 3: создание серии алгоритмов обработки данных. Формализация алгоритмов в виде блок-схемы.

Задача 4: Разобрать отношения в рамках выбранной предметной области в концепции ООП (выбор предметной области обсуждается на занятии). Описать классы, формирующие предметную область. Задать отношения между ними, применяя концепции инкапсуляции, наследования и полиморфизма. Выбрать методы классов, которые могут быть переопределены. Сформулировать в виде блок-схемы варианты алгоритмов обработки данных предметной области и реализовать.

Задача 5: В рамках созданной структуры классов на предыдущих занятиях переопределить/доопределить свойства и методы класса с использованием типов данных «массив», «контейнер» и «строки».

Задача 6: Чтение текстовых файлов с геномными данными. Разбор данных в файле с использованием инструментов обработки строк, их переработка и сохранение в новых файлах. Инструменты поиска в строковых данных.

Задача 7: Чтение геномных файлов с использованием профильных инструментов (BioJava).

Задача 8: Подключение к REST API банка данных KEGG и извлечение информации по выбранным организмам (Выбор организма обсуждается на занятии).

Задача 9: Визуализация результатов вычислений программ, выполненных на предыдущих занятиях в рамках инструмента Excel, Python, Matplotlib, GnuPlot.

Задача 10: построить конвейер обработки данных из разработанных на предыдущих занятиях инструментов, с привлечением сторонних программных инструментов.

Задача 1: реализовать серию тестов для разработанных на предыдущих занятиях программ.

Задача 1: Формулирование блок-схемы алгоритма, который может быть переработан в концепцию «высокопроизводительных вычислений».

Задача 2: Реализация алгоритма параллельного исполнения в рамках потоков.