

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
НАУЧНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «СИРИУС»
(АНО ВО «УНИВЕРСИТЕТ «СИРИУС»))

Научный центр информационных технологий и искусственного интеллекта

СОГЛАСОВАНО
Ученым советом

АНО ВО «Университет «Сириус»
(протокол от «03» августа 2021 г. № 05)

УТВЕРЖДАЮ

Руководитель отдела по учебно-методической работе
АНО ВО «Университет «Сириус»

Г.В. Федоров

«04» августа 2021 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Молекулярное моделирование

наименование дисциплины

Уровень высшего образования:
магистратура

бакалавриат, специалист, магистратура, аспирантура

Направление подготовки (специальность):

01.04.02 Прикладная математика и информатика

код и название направления/специальности

Направленность (профиль):

Биоинформатика и математическая биология

код и название направления/специальности

Форма обучения:

очная

очная / очно-заочная / заочная

Федеральная территория «Сириус» - 2021 г.

1. Общая характеристика дисциплины

1.1. Место дисциплины в структуре основной профессиональной образовательной программы высшего образования (ОПОП ВО):

Уровень высшего образования: магистратура;

Направление подготовки: 01.04.02 Прикладная математика и информатика

Направленность (профиль): Биоинформатика и математическая биология (далее – образовательная программа, ОПОП ВО);

Форма реализации (очная, дистанционная): очная с возможностью применения электронного обучения и дистанционных образовательных технологий.

1.2. Место дисциплины в образовательной программе и описание взаимосвязи дисциплины с другими реализуемыми в АНО ВО «Университет «Сириус» образовательными модулями, образовательными программами, научными исследованиями и проектами:

В рамках ОПОП ВО дисциплина реализуется в третьем семестре.

Изучению дисциплины в рамках ОПОП ВО предшествует изучение дисциплин: Молекулярная биология, Структурная биоинформатика и строение белков, Алгоритмы и структуры данных.

Изучение дисциплины необходимо для освоения следующих дисциплин ОПОП ВО: магистерская диссертация.

Дисциплина может быть использована в программах магистратуры 01.04.02 Прикладная математика и информатика в качестве обязательной дисциплины или дисциплины по выбору.

1.3. Входные требования для освоения дисциплины: обучающийся должен освоить дисциплины базовой части образовательной программы специалиста или бакалавриата 1 и 2-го годов обучения, знать элементарные сведения из математического анализа, структурной биологии, общей физики, физической химии, а также уметь работать в среде Linux, программировать простейшие скрипты на Bash и Python..

1.4. Цель дисциплины: сформировать уровень компетенции студентов, достаточный для критического анализа современных работ с применением методов молекулярного моделирования и свободного применения распространенных и доступных подходов для решения задач молекулярной биологии.
1.5. Задачи дисциплины: ознакомить слушателей с основными подходами и научить слушателей приемам молекулярного моделирования от квантовой механики до макромолекулярного докинга. Ключевым моментом практических занятий является использование открытого программного обеспечения.
1.6. Язык преподавания: русский.

1.7. Результаты обучения по дисциплине, соотнесенные с формируемыми компетенциями:

Формируемые компетенции (код компетенции, формулировка)	Планируемые результаты обучения по дисциплине
ПК-1. Способен решать актуальные задачи фундаментальной и прикладной математики	ПК-1.1. Знает основы фундаментальных математических, биологических дисциплин и других естественных наук.

	ПК-1.2. Умеет формулировать, анализировать и решать профессиональные задачи с применением фундаментальных знаний математики, биологии и других естественных наук. ПК-1.3. Имеет практический опыт постановки и решения актуальных задач математической биологии и других наук
ПК-2. Способен реализовывать и совершенствовать существующие и вводить новые математические методы решения прикладных задач	ПК-2.1. Знает основные математические методы решения прикладных задач. ПК-2.2. Умеет осуществлять анализ и выбор методов решения задач профессиональной и научной деятельности на основе теоретических знаний в области математических и компьютерных наук. ПК-2.3. Имеет практический опыт разработки новых методов математического моделирования для решения задач профессиональной и научной деятельности
ПК-3. Способен разрабатывать математические модели и проводить их анализ при решении задач в области профессиональной деятельности	ПК-3.1. Знает основные положения, терминологию и методологию в области математического и компьютерного моделирования. ПК-3.2. Умеет выбирать, использовать и разрабатывать необходимые методы математического и компьютерного моделирования в зависимости от поставленных задач. ПК-3.3. Имеет практический опыт применения методов вычислительной биологии в решении прикладных и управленческих задач
ПК-5. Владеет методами математического и компьютерного исследования при разработке интеграционных решений на основе знаний фундаментальных математических и компьютерных наук и навыками проблемно-задачной формы представления научных знаний	ПК-5.1. Знает теоретические основы фундаментальных методов исследования научных проблем. ПК-5.2. Умеет самостоятельно применять полученные знания для анализа объекта исследования, определения целей и задач исследования, а также выбора корректного метода исследования научной проблемы.

	ПК-5.3. Имеет практический опыт научно-исследовательской деятельности, а именно решения научных задач в соответствии с поставленной целью и выбранной методикой.
ПК-9. Имеет опыт самостоятельного проведения работ по обработке и анализу научно-технической информации и результатов исследования, разработки организационно-распорядительной и отчетной документации	ПК-9.1. Знает особенности поиска научно-технической информации в различных источниках, методов и технологий её обработки и анализа, а также способов представления. ПК-9.2. Умеет самостоятельно организовать целенаправленный поиск информации в различных источниках, выбирать методы и технологии её обработки, анализа и представления, исходя из поставленной задачи. ПК-9.3. Имеет практический опыт поиска и анализа научнотехнической информации в различных источниках для решения стандартных профессиональных задач, а также опыт публичного представления научных результатов.

2. Структура дисциплины

2.1. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся:

объем дисциплины составляет 3 з.е., 108 ак. ч., из которых 48 ак. ч. составляет контактная работа обучающихся с преподавателем (24 ак. ч. занятий лекционного типа, 24 ак. ч. лабораторных, практических занятий, семинаров), 60 ак. ч. составляет самостоятельная работа обучающихся.

2.2. Календарный график реализации дисциплины (в случае реализации дисциплины в формате интенсива):

день	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1 пара	лек	лек	лек	лек	лек	лек	лек	лек	лек	лек	лек	лек	лек	лек	пр	пр	пр	сам	экз
2 пара	лек	лек	лек	лек	пр	пр	пр	пр	пр	пр	пр	сам	сам	сам	сам	сам	сам	сам	экз
3 пара	пр	пр	пр	пр	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам				экз

Введение. Краткая история дисциплины. Общая постановка задачи оптимизации. Примеры	10	2			4			6	4	
Строение атома водорода. Метод Хартри-Фока. Теория функционала плотности. Полуэмпирические расчеты.	14	2			4			6	8	
Молекулярная механика. Главные особенности молекулярно механических силовых полей. Потенциалы взаимодействий через ковалентные связи. Электростатические взаимодействия. Ван дер Ваальсовы взаимодействия. Параметры для описания воды. Создание силового поля	16	4			4			8	8	
Молекулярная динамика. Простые модели. Создание системы и запуск молекулярной динамики. Динамика с ограничениями. Молекулярная динамика при постоянном давлении и температуре. Введение эффектов растворителя. Конформационные изменения. Гибридный	18	4			4			8	10	

[illegible]

Поиск новых биоактивных молекул и хемоинформатика. Молекулярное моделирование для поиска лекарств. Компьютерное представление молекул, SMILES, SMARTS. Источники данных для баз данных структур. Молекулярный докинг. Применение поиска по базам данных структур и докинга. Построение лиганда de novo на основе структуры белка.	18	4	4					8	10	
Промежуточная аттестация										
Итого	108	24	24					48	60	

3. Текущий контроль и промежуточная аттестация

Текущий контроль состоит в проверке выполнения обучающимися контрольных заданий. Промежуточная аттестация проводится в форме экзамена

3.1. Требования к структуре и содержанию фонда оценочных средств текущего контроля и промежуточной аттестации по дисциплине: в соответствии с требованиями для аттестации обучающихся на соответствие их персональных достижений планируемым результатам обучения по дисциплине созданы фонды оценочных средств представленные в виде типовых контрольных заданий и/или иных материалов, необходимых для оценки результатов обучения.

3.2. Критерии и процедуры оценивания результатов обучения по дисциплине:

Критерии выставления экзаменационных оценок.

Оценка «отлично» выставляется при наличии сформированных систематических знаний разделов молекулярного моделирования.

Оценка «хорошо» выставляется при наличии сформированных, но содержащих отдельные пробелы знаний разделов молекулярного моделирования.

Оценка «удовлетворительно» выставляется при наличии общих, но не структурированных знаний разделов молекулярного моделирования.
Оценка «неудовлетворительно» выставляется при отсутствии знаний.

4. Примеры оценочных средств для проведения текущего контроля успеваемости и итоговой аттестации:

Домашнее задание № 1

Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - отчет в формате pdf, docx, Google Docs.

Цель занятия: используя пакет модулей RDKit предложить аналог ибупрофена :

на сайте PubChem найти все радикалы с азидом для Click Chemistry и скачать их SMILES нотации или использовать pubchempy; найти формулу ибупрофена и предложить способ изменения его SMILES для эмуляции продукта Click Chemistry; заменить в найденных радикалах азидную группу на модифицированный ибупрофен; превратить те новые SMILES в объекты-молекулы; отобразить те молекулы, которые удовлетворяют правилу пяти Lipinsky.

Домашнее задание № 2

Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - отчет в формате pdf, docx, Google Docs.
Вычисление атомных орбиталей водорода¶

Цель занятия: опираясь на уравнение построить электронную плотность в одно-электронном атоме и сравнить с известными программами расчёт плотности в Ipython Notebook с сохранением в формате CUBE визуализация в Rmol.

Примерный список экзаменационных вопросов:

1. Волновая функция
2. Точное решение уравнения Шредингера Одно-электронный атом
3. Метод самосогласованного поля, SCF
4. Метод: Хартри-Фока, Подход Рутхана-Хола
5. Базисные наборы

6. Семи-эмпирические методы
7. Описание базисных наборов для программы GAUSSIAN
8. Теория функционала плотности, DFT
9. Молекулярная механика Простое уравнение силового поля
10. Электростатические взаимодействия. Двойное обрезание Суммирование Эвальда
11. Ван-дер-Ваальсовы взаимодействия Взаимодействия между разными типами атомов
12. Минимизация энергии Минимумы, максимумы и стационарные точки Переходные состояния
13. Молекулярная динамика Алгоритмы интегратора
14. Периодические граничные условия
15. Молекулярная динамика Список соседей
16. Ограничения быстрых колебаний
17. Температура Контроль давления в системе
18. Гибридное QM/MM моделирование
19. Метод Монте-Карло расчёт термодинамических свойств
20. Методы выборочного поиска (Biased Monte-Carlo)
21. Дистанционная геометрия и ЯМР
22. Использование МД при оптимизации данных ЯМР
23. Сравнительное моделирование
24. Степень идентичности и сравнительное моделирование
25. Предсказание структуры белка Ab initio
26. Термодинамическая пертурбация
27. Термодинамические циклы

- 28. Потенциал средней силы
- 29. Линейное представление молекул, SMILES
- 30. SMARTS: паттерны для SMILES
- 31. Фрагментарное построение лиганда
- 32. QSAR, количественные соотношения структура/ активность
- 33. Докинг белок-лиганд
- 34. Белковый докинг

5. Методические материалы для обучающихся

Конспект и презентации лекций.

6. Методические материалы для преподавателей

Теоретическое содержание дисциплины состоит в рассмотрении основных положений и теоретических вопросов по предмету дисциплины в соответствии с целями и задачами, указанными в п. 1 рабочей программы дисциплины (РПД). Содержание лекционных занятий конкретизировано в соответствии с элементами теоретического, практического изучения и применения объектов, образующих предмет изучения дисциплины и включающих:

- основные понятия и их определения;
- особенности строения и функционирования объектов, их основные свойства, характеристики, параметры;
- задачи (проблемы) теоретического и/или практического изучения объектов, их создания и применения;
- методы, средства и способы их теоретического и/или практического изучения и совершенствования.

Глубина изложения теоретических вопросов определяется объемом часов, отведенных для занятий лекционного типа (лекции) и на самостоятельную работу обучающихся в рамках проработки материалов лекций.

7. Ресурсное обеспечение

7.1. Кадровое обеспечение

Руководитель программы: А.В.Головин, профессор, д.б.н., Московский государственный университет

Авторы программы:

А.В.Головин, профессор, д.б.н., Московский государственный университет

А. С. Злобин

7.2. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине:

В рамках самостоятельной работы обучающиеся используют интернет-ресурсы:

1. <https://edu.rsc.org/resources/analysis>

2. <https://collection.asdlib.org/>

7.3. Перечень основной учебной литературы

1. Основы клеточной биологии: учебное пособие / Н.Г. Палеев, И.И. Бессчетнов.- Ростовна-Дону: Издательство ЮФУ, 2011. - 246 с. - <http://znanium.com/bookread2.php?book=550792>
2. Биология. В 2 т. Т. 2 [Электронный ресурс] : учебник / под ред. В. Н. Ярыгина. - М. : ГЭОТАР-Медиа, 2015. - <http://www.studmedlib.ru/book/ISBN9785970435656.html>

7.4. Перечень дополнительной учебной литературы:

1. Гистология и эмбриональное развитие органов полости рта человека [Электронный ресурс] / В.Л. Быков - М. : ГЭОТАР-Медиа, 2014. - <http://www.studmedlib.ru/book/ISBN9785970430118.html>
2. Медицинская микробиология, вирусология и иммунология: в 2 т. Том 1. [Электронный ресурс] : учебник / Под ред. В.В. Зверева, М.Н. Бойченко. - М. : ГЭОТАР-Медиа, 2016. - <http://www.studmedlib.ru/book/ISBN9785970436417.html>
3. Некрасова, И.И. Основы цитологии и биологии развития [Электронный ресурс] : учебное пособие / И.И. Некрасова; Ставропольский государственный аграрный университет. - Ставрополь: АГРУС, 2008. - 152 с. - <http://znanium.com/bookread2.php?book=514534>
4. Медицинская биология: Энциклопедический справочник / Смирнов О.Ю. - М.: Форум, НИЦ ИНФРА-М, 2016. - 608 с. - <http://znanium.com/bookread2.php?book=538672>

7.5. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»:

База знаний по биологии человека - <http://humbio.ru>

Биохимия для обучающихся медицинских специальностей - <http://tulpar.kpfu.ru/enrol/index.php?id=948>

Википедия - свободная энциклопедия - <http://ru.wikipedia.org/>

Издательство BioMedCentral - <http://www.biomedcentral.com>

Сайт о химии - <http://www.ximuk.ru/encyklopedia/>

7.6. Описание материально-технической базы:

7.6.1. Аудиторный фонд и оборудование

№ п.п.	Вид аудитории	Технические средства и оборудование (кол-во)	Расходные материалы (кол-во)
1.	Учебная аудитория для проведения лекционных занятий	Компьютер с проектором Меловая или маркерная доска, мел или маркер.	Мел или маркер 15 шт.
2.	Учебная аудитория для проведения практических занятий	Компьютер с проектором Меловая или маркерная доска, мел или маркер.	Мел или маркер 10 шт.

7.6.2. Оборудование для лабораторных и практических занятий:

Оборудование для проведения практических занятий.

7.7. Перечень лицензионного и свободно распространяемого программного обеспечения:

Для проведения всех занятий используется проектор и компьютер для проекции слайдов. Для самостоятельной работы необходимо следующее ПО: пакет библиотек для Python (Anaconda), инструмент для сборки Haskell (Stack), компилятор C++ (clang). Кроме того, для работы будут использованы базы данных PDB, PubChem, OpenFF, а также программы Gromacs, Psi4, Modeller, NGLview, zdock, trRosetta.

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
НАУЧНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «СИРИУС»
(АНО ВО УНИВЕРСИТЕТ «СИРИУС»))

Научный центр информационных технологий и искусственного интеллекта

СОГЛАСОВАНО
Ученым советом

АНО ВО «Университет «Сириус»
(протокол от «03» августа 2021 г. № 05)

УТВЕРЖДАЮ

Руководитель отдела по учебно-методической работе
АНО ВО «Университет «Сириус»

Г.В. Федоров

«03» августа 2021 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Функциональное программирование

наименование дисциплины

Уровень высшего образования:

магистратура

бакалавриат, специалитет, магистратура, аспирантура

Направление подготовки (специальность):

01.04.02 Прикладная математика и информатика

код и название направления/специальности

Направленность (профиль):

Биоинформатика и математическая биология

код и название направления/специальности

Форма обучения:

очная

очная / очно-заочная / заочная

Федеральная территория «Сириус» - 2021 г.

1. Общая характеристика дисциплины

1.1. Место дисциплины в структуре основной профессиональной образовательной программы высшего образования (ОПОП ВО):

Уровень высшего образования: магистратура;

Направление подготовки: 01.04.02 Прикладная математика и информатика

Направленность (профиль): Биоинформатика и математическая биология (далее – образовательная программа, ОПОП ВО);

Форма реализации (очная, дистанционная): очная с возможностью применения электронного обучения и дистанционных образовательных технологий.

1.2. Место дисциплины в образовательной программе и описание взаимосвязи дисциплины с другими реализуемыми в АНО ВО «Университет «Сириус» образовательными модулями, образовательными программами, научными исследованиями и проектами:

В рамках ОПОП ВО дисциплина реализуется в 3 семестре.

Изучению дисциплины в рамках ОПОП ВО предшествует изучение дисциплин: Основы программирования (Python, Unix), C++.

Изучение дисциплины необходимо для освоения следующих дисциплин ОПОП ВО: Магистерская диссертация.

Дисциплина может быть использована в программах магистратуры 01.04.02 Прикладная математика и информатика в качестве дисциплины по выбору.

1.3. Входные требования для освоения дисциплины: знание базовой части дисциплины образовательной программы специалитета или бакалавриата и 2-го годов обучения и элементарных сведений из программирования и умение решать простейшие задачи элементарной (школьной) математики.

1.4. Цель дисциплины: формирование у студентов формирования теоретических знаний и практических навыков программирования на языке Haskell.

1.5. Задачи дисциплины: развитие у студентов практических навыков конструирования конкретных алгоритмов на языке высокого уровня для решения разнообразных биоинформатических задач.

1.6. Язык преподавания: русский.

1.7. Результаты обучения по дисциплине, соотношенные с формируемыми компетенциями:

Формируемые компетенции (код компетенции, формулировка)	Планируемые результаты обучения по дисциплине
ПК-2. Способен реализовывать и совершенствовать существующие и вводить новые математические методы решения прикладных задач	ПК-2.1. Знает основные математические методы решения прикладных задач.

	ПК-2.2. Умеет осуществлять анализ и выбор методов решения задач профессиональной и научной деятельности на основе теоретических знаний в области математических и компьютерных наук. ПК-2.3. Имеет практический опыт разработки новых методов математического моделирования для решения задач профессиональной и научной деятельности
ПК-3. Способен разрабатывать математические модели и проводить их анализ при решении задач в области профессиональной деятельности	ПК-3.1. Знает основные положения, терминологию и методологию в области математического и компьютерного моделирования. ПК-3.2. Умеет выбирать, использовать и разрабатывать необходимые методы математического и компьютерного моделирования в зависимости от поставленных задач. ПК-3.3. Имеет практический опыт применения методов вычислительной биологии в решении прикладных и управленческих задач
ПК-4. Способен комбинировать и адаптировать существующие информационно-коммуникационные технологии для решения задач в области профессиональной деятельности с учетом требований информационной безопасности	ПК-4.1. Знает базовые понятия информатики, информации, ее измерения, кодирования и представления в вычислительных системах, принципы сбора, хранения и обработки информации, а также современные алгоритмы, средства разработки и программные средства. ПК-4.2. Умеет использовать знания, полученные в области компьютерных наук. ПК-4.3. Имеет практический опыт использования информационных технологий, а также создания программных средств для решения задач профессиональной деятельности.
ПК-6. Способен организовывать и поддерживать коллективную научно-исследовательскую деятельность в соответствии с направлением подготовки, организовывать научную и прикладную коммуникацию.	ПК-6.1. Способен организовывать работу научно-исследовательского коллектива по направлению подготовки. ПК-6.2. Умеет поддерживать коллективную научную коммуникацию, организовывать научные мероприятия.

	ПК-6.3. Имеет опыт трансляции методов и практик работы с имеющимися и новыми инструментами в области прикладной математики
ПК-8. Умеет самостоятельно разрабатывать, исследовать, применять программное обеспечение, проводить расчётные работы и исследования, обработку результатов, оформление отчётной документации	ПК-8.1. Знает классические методы решения задач, современные программные комплексы для проведения расчётных исследований. ПК-8.2. Умеет самостоятельно проводить расчётные исследования, выбирать и применять современные программные комплексы, получать, обрабатывать и анализировать результаты исследований. ПК-8.3. Имеет практический опыт применения математического и компьютерного моделирования.

2. Структура дисциплины

2.1. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся:

объем дисциплины составляет 3 з.е., 108 ак. ч., из которых 48 ак. ч. составляет контактная работа обучающихся с преподавателем (24 ак. ч. занятий лекционного типа, 24 ак. ч. лабораторных, практических занятий, семинаров), 60 ак. ч. составляет самостоятельная работа обучающихся.

2.2. Календарный график реализации дисциплины (в случае реализации дисциплины в формате интенсива):

день	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1 пара	лек	лек	лек	лек	лек	лек	лек	лек	пр	пр	пр	пр	сам	сам	сам	сам	сам	сам	ЭКЗ
2 пара	лек	лек	лек	лек	пр	пр	пр	пр	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	ЭКЗ
3 пара	пр	пр	пр	пр	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	ЭКЗ
4 пара																			

2.3. Учебный план дисциплины и её содержание, структурированное по темам (разделам):

	В том числе	
--	-------------	--

Наименование разделов и тем дисциплины	Всего (ак. часы)	Контактная работа (работа во взаимодействии с преподавателем), ак. часы						Самостоятельная работа обучающегося, ак. часы	Форма текущего контроля успеваемости и промежуточной аттестации
		Лекции	Лабораторные занятия	Практические занятия	Семинары	Учебные занятия, направленные на проведение текущего контроля успеваемости, промежуточной аттестации	Всего		
Введение. Краткая история дисциплины. Общая постановка задачи оптимизации. Примеры	2	1		1			2		
Основы функционального программирования: лямбда-исчисление и комбинаторная логика	25	5		5			10	15	

Язык программирования Haskell: синтаксис, семантика, стандартная библиотека	27	6				6				12	15	
Управление эффектами с помощью аппликативных функторов и монад	27	6				6				12	15	
Системы типов функциональных языков	27	6				6				12	15	
Промежуточная аттестация												Экзамен
Итого	108	24				24				48	60	

3. Текущий контроль и промежуточная аттестация

Текущий контроль состоит в проверке выполнения обучающимися контрольных заданий. Промежуточная аттестация проводится в форме экзамена.

3.1. Требования к структуре и содержанию фонда оценочных средств текущего контроля и промежуточной аттестации по дисциплине: в соответствии с требованиями для аттестации обучающихся на соответствие их персональных достижений планируемым результатам обучения по дисциплине созданы фонды оценочных средств представленные в виде типовых контрольных заданий и/или иных материалов, необходимых для оценки результатов обучения.

3.2. Критерии и процедуры оценивания результатов обучения по дисциплине:

Критерии выставления экзаменационных оценок.

Оценка «отлично» выставляется при наличии сформированных систематических знаний специальных разделов программирования.
 Оценка «хорошо» выставляется при наличии сформированных, но содержащих отдельные пробелы знаний специальных разделов программирования.
 Оценка «удовлетворительно» выставляется при наличии общих, но не структурированных знаний специальных разделов программирования.
 Оценка «неудовлетворительно» выставляется при отсутствии знаний.

4. Примеры оценочных средств для проведения текущего контроля успеваемости и итоговой аттестации

Домашнее задание 1

Описание технологии применения домашнего задания как оценочного средства:

- задачи выдаются обучающимся после каждого практического занятия,
- решения задач обучающиеся представляют в электронном виде, в любом из стандартных открытых форматов данных,
- сроки представления решения составляют одну неделю с момента выдачи задания.

Задача 1. Выделите свободные и связанные переменные в термах и выполните указанные подстановки:

$lyz.xyw(zx) [x := \lambda y.yw]$
 $\lambda xy.xy(\lambda x.xy)x [x := \lambda z.z]$
 $xy(\lambda xz.xyz)y [y := xz]$

Определите, возможно ли в получившемся терме выполнить β -преобразование.

(1 балл)

Задача 2. Уберите лишние скобки и при возможности выполните β -преобразование

$(x(\lambda x.((xy)x))y)$
 $((\lambda p.(\lambda q.((q(pr))s))((q(pr))s))$

(1 балл)

Задача 3. Покажите, что для любых M и N выполняется

$\lambda x.MN = S(\lambda x.M)(\lambda x.N)$

(1 балл)

Задача 4. Покажите, что

$$SKK = I$$

$$B = S(KS)K$$

(2 балла)

Задача 5. Реализуйте функцию возведения в степень для чисел Чёрча. (2 балла)

Задача 6. Напишите функции: `minus`, вычитающую числа Чёрча, `equals`, сравнивающую два числа Чёрча на предмет равенства, а также всевозможные неравенства, строгие и нестрогие, `lt`, `gt`, `le`, `ge`. (1 балл)

Задача 7. (1 балл) Постройте функции:

– `sum` суммирующую элементы списка, например

$$\text{sum } [5, 3, 2] = 10$$

– `length` вычисляющую длину списка, например

$$\text{length } [5, 3, 2] = 3$$

Задача 8. (3 балла) Постройте функцию `tail`, возвращающую хвост списка, например

$$\text{tail } [5, 3, 2] = [3, 2]$$

Задача 9. (2 балла) Используя Y -комбинатор, сконструируйте

– «пожиратель», то есть такой терм F , который для любого M обеспечивает $FM = F$.

– терм F таким образом, чтобы для любого M выполнялось $FM = MF$.

– терм F таким образом, чтобы для любых термов M и N выполнялось $FMN = NF(NMF)$.

Задача 10. (2 балла) Пусть имеются взаимно-рекурсивное определение функций f

и g :

$$f = Ffg$$

$$g = Gfg$$

Используя Y-комбинатор, найдите рекурсивные определения для f и g .

Задача 11. (1 балл) Найдите замкнутые термы в нормальной форме, являющиеся обитателями типа

- $\alpha \rightarrow \beta \rightarrow \gamma \rightarrow \alpha$
- $\alpha \rightarrow \beta \rightarrow \gamma \rightarrow \beta$
- $\alpha \rightarrow \beta \rightarrow \alpha \rightarrow \alpha$
- $\alpha \rightarrow \alpha \rightarrow \alpha \rightarrow \alpha$

Сколько разных (с точностью до $\alpha\beta$ -эквивалентности) термов каждого типа вы можете привести?

Задача 12. (2 балла) Найдите замкнутые термы в нормальной форме, являющиеся обитателями типа

- $(\delta \rightarrow \delta \rightarrow \alpha) \rightarrow (\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\delta \rightarrow \beta) \rightarrow \delta \rightarrow \gamma$
- $(\delta \rightarrow \delta \rightarrow \alpha) \rightarrow (\gamma \rightarrow \alpha) \rightarrow (\alpha \rightarrow \beta) \rightarrow \delta \rightarrow \gamma \rightarrow \beta$ (2 штуки)

Задача 13. (2 балла) Типизируйте по Чёрчу

- SKK
- SKI

(Обратите внимание, что в задании НЕТ указания редуцировать терм.)

Задача 14. (2 балла) Сконструируйте замкнутый терм (в нормальной форме) типа

- $(\gamma \rightarrow \delta) \rightarrow ((\gamma \rightarrow \delta) \rightarrow \delta) \rightarrow \delta$
- которому нельзя было бы приписать тип $a \rightarrow (a \rightarrow \delta) \rightarrow \delta$.

Задача 15. (3 балла) Сконструируйте замкнутый терм (в нормальной форме) типа

- $((\alpha \rightarrow \beta) \rightarrow \alpha) \rightarrow (\alpha \rightarrow \alpha \rightarrow \beta) \rightarrow \alpha$
- $((\alpha \rightarrow \beta) \rightarrow \alpha) \rightarrow (\alpha \rightarrow \alpha \rightarrow \beta) \rightarrow \beta$

Критерии оценивания и шкала оценки домашнего задания №1.

Оценка	Критерии выставления оценки
«Отлично» (10)	Все конструкции соответствуют спецификации, а преобразования проведены аккуратно и снабжены комментариями.
«Хорошо» (4-6)	В преобразованиях или реализации есть неточности, переходы в доказательствах не обоснованы.
«Неудовлетворительно» (0)	Неустраняемые ошибки в преобразованиях или реализации.

Домашнее задание 2

Описание технологии применения домашнего задания как оценочного средства:

- задачи выдаются обучающимся после каждого практического занятия,
- решения задач обучающиеся представляют в виде кода на языке Haskell на платформе Stack, в специальном закрытом курсе,
- сроки представления решений составляют одну неделю с момента выдачи задания.

Задача 1. (1 балл) Какому известному вам библиотечному оператору, конструктору или функции эквивалентно выражение `uncurry (flip const)` ?

Задача 2. (1 балл) В модуле `Data.Tuple` стандартной библиотеки определена функция `swap :: (a,b) -> (b,a)`, переставляющая местами элементы пары:

```
GHCi> swap (1,'A')
('A',1)
```

Эта функция может быть выражена в виде `swap = f (g h)`, где `f`, `g` и `h` — некоторые идентификаторы из следующего набора:

```
curry uncurry (,) const flip
```

Укажите через запятую подходящую тройку `f, g, h`.

Задача 3. (1 балл) Определите функцию вычисляющую двойной факториал, то есть произведение натуральных чисел, не превосходящих заданного числа и имеющих ту же четность. Например: $7!! = 7 \cdot 5 \cdot 3 \cdot 1$, $8!! = 8 \cdot 6 \cdot 4 \cdot 2$.

```
doubleFact :: Integer -> Integer
doubleFact n = undefined
```

Предполагается, что аргумент функции может принимать только неотрицательные значения.

Задача 4. (1 балл) Реализуйте функцию, находящую элементы следующей рекуррентной последовательности

$$a_0 = 1; a_1 = 2; a_2 = 3; a_{k+3} = a_{k+2} + a_{k+1} - 2a_k.$$

```
seqA :: Integer -> Integer
seqA n = undefined
```

Найдите эффективное решение со сложностью не хуже линейной по числу вызовов арифметических операторов.

```
GHCI> seqA 7
```

-10

Задача 5. (2 бала) Понятие чисел Фибоначчи можно расширить, потребовав, чтобы рекуррентное соотношение выполнялось для произвольных целых значений аргумента, в том числе и отрицательных. Реализуйте функцию, вычисляющую числа Фибоначчи так, чтобы она удовлетворяла этому требованию.

```
fibonacci :: Integer -> Integer
fibonacci n = undefined
```

Реализация должна иметь сложность не хуже линейной по числу вызовов оператора сложения.

```
GHCI> fibonacci (-99)
218922995834555169026
```

Задача 6. (2 балла) Реализуйте функцию, находящую сумму и количество цифр заданного целого числа.

```
sum'n'count :: Integer -> (Integer, Integer)
sum'n'count x = undefined
```

```
GHCi> sum `n`count (-39)
(12,2)
```

Задача 7. (2 балла) Реализуйте функцию, находящую значение определённого интеграла от заданной функции на заданном интервале методом трапеций.

```
integration :: (Double -> Double) -> Double -> Double -> Double
integration f a b = undefined
```

Используйте равномерную сетку, достаточно 1000 элементарных отрезков.

```
GHCi> integration sin pi 0
-2.0
```

Результат может отличаться от -2.0, но не более чем на 1e-4.

Задача 8. (1 балл) Тип данных Ordering определен в стандартной библиотеке так:

2

```
data Ordering = LT | EQ | GT
```

Он используется при определении функций, сравнивающих элементы каких-либо типов. Если первый аргумент меньше второго, возвращается LT, если равен — EQ, если больше — GT.

Определим тип LogLevel следующим образом

```
data LogLevel = Error | Warning | Info
```

Реализуйте функцию cmp, сравнивающую элементы типа LogLevel так, чтобы имел место порядок Error > Warning > Info.

```
cmp :: LogLevel -> LogLevel -> Ordering
```

```
cmp = undefined
```

Задача 9. (1 балл) Тип данных Person определен как запись:

```
data Person = Person {firstName :: String, lastName :: String, age :: Int}
```

Реализуйте функцию abbrFirstName, которая сокращает имя до первой буквы с точкой, то есть если имя было "John", то после применения этой функции, оно превратится в "J.". Однако если имя было короче двух символов, то оно не меняется.

```
abbrFirstName :: Person -> Person
```

abbrFirstName = undefined

Задача 10. (1 балл) Следующий рекурсивный тип данных задаст бинарное дерево:

```
data Tree a = Leaf | Node (Tree a) a (Tree a)
```

Напишите следующие функции:

- вычисление суммы элементов дерева

```
treeSum :: Tree Integer -> Integer
```

```
treeSum = undefined
```

- вычисление максимальной высоты дерева

```
treeHeight :: Tree a -> Int
```

```
treeHeight = undefined
```

Задача 11. (1 балл) Составить список сумм соответствующих элементов трех заданных списков. Длина результирующего списка должна быть равна длине самого длинного из заданных списков, при этом «закончившиеся» списки не должны давать вклада в суммы.

```
sum3 :: Num a => [a] -> [a] -> [a] -> [a]
```

```
sum3 = undefined
```

Задача 12. (1 балл) Сформируйте список цифр заданного целого числа.

```
digits :: Integer -> [Integer]
```

```
digits = undefined
```

Задача 13. (1 балл) Определите, содержит ли заданное целое число все цифры от 1 до 9. (Используйте функцию digits из предыдущего задания.)

```
containsAllDigits :: Integer -> Bool
```

```
containsAllDigits = undefined
```

Задача 14. (1 балл) Определите, содержит ли заданное целое число все цифры от 1 до 9 в точности по одному разу. (Используйте функцию digits из предыдущего задания.)

```
containsAllDigitsOnes :: Integer -> Bool
```

containsAllDigitsOnes = undefined

Задача 14. (1 балл) Из заданного списка выделите подпоследовательности из элементов с n -го номера по k -ый, считая от нуля. При этом n -ый элемент должен входить в результат, а k -ый — нет.

sublist :: Int -> Int -> [a] -> [a]

sublist = undefined

Постарайтесь обеспечить разумное поведение для произвольных целых индексов и для бесконечных списков.

Задача 15. (1 балл) Повторите каждый элемент списка заданное число раз.

repeatEveryElem :: Int -> [a] -> [a]

repeatEveryElem = undefined

Задача 16. (1 балл) Дан список (возможно бесконечный) $[a_1, a_2, \dots]$ и положительное целое число n . Создайте список «скользящих» подпоследовательностей длины n , то есть список списков следующего вида:

$[[a_1, \dots, a_n], [a_2, \dots, a_{n+1}], [a_3, \dots, a_{n+2}], \dots]$

movingLists :: Int -> [a] -> [[a]]

movingLists = undefined

Критерии оценивания и шкала оценки домашнего задания №2.

Оценка	Критерии выставления оценки
«Отлично» (10)	Все тесты для текущей задачи на платформе <code>stepik</code> успешно пройдены
«Неудовлетворительно» (0)	Решение не проходит полный набор тестов.

Описание технологии применения домашнего задания как оценочного средства:

- задачи выдаются обучающимся после каждого практического занятия,
- решения задач обучающиеся представляют в виде кода на языке Haskell на платформе Stepik, в специальном закрытом курсе,
- сроки представления решений составляют одну неделю с момента выдачи задания.

Задача 1. (1 балл) «Окружите» каждый элемент списка заданными «скобками», используя монаду списка и `do` -нотацию:

```
surround :: a -> a -> [a] -> [a]
surround x y zs = do undefined
```

Проверка:

```
GHCi> surround '{' '}' "abcd"
"{a}{b}{c}{d}"
```

Задача 2. (1 балл) Ассоциативным списком называют список пар (ключ, значение). Реализуйте функцию поиска в таком списке, возвращающую список всех значений с заданным ключом. Используйте монаду списка и `do` -нотацию.

```
lookups :: (Eq a) => a -> [(a,b)] -> [b]
lookups x ys = do undefined
```

Проверка:

```
GHCi> lookups 2 [(1, "one"), (2, "two"), (3, "three"), (2, "two")]
["two", "two"]
```

Задача 3. (2 балла) Разложите положительное целое число на два сомножителя всевозможными способами, используя монаду списка и `do` -нотацию.

```
factor2 :: Integer -> [(Integer, Integer)]
```

```
factor2 n = do undefined
```

Пары должны быть уникальными, первый элемент пары не должен превышать второй, результат следует упорядочить лексикографически, в возрастающем порядке.

Проверка:

```
GHCi> factor2 45
[(1,45),(3,15),(5,9)]
```

Задача 4. (2 балла) Вычислите модули разностей между соседними элементами списка, используя монаду списка и `do` -нотацию.

```
absDiff :: Num a => [a] -> [a]
absDiff xs = do undefined
```

Проверка:

```
GHCi> absDiff [2,7,22,9]
```

[5,15,13]

Задача 5. (2 балла) Для типа данных

```
data OddC a = Un a | Bi a (OddC a) deriving (Eq,Show)
```

(контейнер-последовательность, который по построению может содержать только нечетное число элементов) реализуйте функцию

```
concat3OC :: OddC a -> OddC a -> OddC a -> OddC a
```

конкатенирующую три таких контейнера в один:

```
GHCI> tst1 = Bi 'a' 'b' (Un 'c')
```

```
GHCI> tst2 = Bi 'd' 'e' (Bi 'f' 'g' (Un 'h'))
```

```
GHCI> tst3 = Bi 'i' 'j' (Un 'k')
```

```
GHCI> concat3OC tst1 tst2 tst3
```

```
Bi 'a' 'b' (Bi 'c' 'd' (Bi 'e' 'f' (Bi 'g' 'h' (Bi 'i' 'j' (Un 'k')))))
```

Обратите внимание, что отображения четности запрещают конкатенацию двух контейнеров OddC .

Задача 6. (2 балла) Для типа данных

```
data OddC a = Un a | Bi a (OddC a) deriving (Eq,Show)
```

реализуйте функцию

```
concatOC :: OddC (OddC a) -> OddC a
```

Она должна обеспечивать для типа OddC поведение, аналогичное поведению функции concat для списков:

```
GHCI> concatOC $ Un (Un 42)
```

```
Un 42
```

```
GHCI> tst1 = Bi 'a' 'b' (Un 'c')
```

```
GHCI> tst2 = Bi 'd' 'e' (Bi 'f' 'g' (Un 'h'))
```

```
GHCI> tst3 = Bi 'i' 'j' (Un 'k')
```

```
GHCI> concatOC $ Bi tst1 tst2 (Un tst3)
```

```
Bi 'a' 'b' (Bi 'c' 'd' (Bi 'e' 'f' (Bi 'g' 'h' (Bi 'i' 'j' (Un 'k')))))
```

Задача 7. (2 балла) Сделайте тип данных

```
data OddC a = Un a | Bi a (OddC a) deriving (Eq,Show)
```

представителем классов типов Functor , Applicative и Monad . Семантика должна быть подобной семантике представителей этих классов типов для

списков: монада OddC должна иметь эффект вычисления с произвольным нечетным числом результатов:

```
GHCI> tst1 = Bi 10 20 (Un 30)
```

```
GHCI> tst2 = Bi 1 2 (Bi 3 4 (Un 5))
```

```
GHCI> do {x <- tst1; y <- tst2; return (x + y)}
```

Bi 11 12 (Bi 13 14 (Bi 15 21 (Bi 22 23 (Bi 24 25 (Bi 31 32 (Bi 33 34 (Un 35))))))

```
GHCI> do {x <- tst2; y <- tst1; return (x + y)}
```

Bi 11 21 (Bi 31 12 (Bi 22 32 (Bi 13 23 (Bi 33 14 (Bi 24 34 (Bi 15 25 (Un 35))))))

Функцию fail можно не реализовывать, полагаясь на реализацию по умолчанию.

Задача 8. (1 балл) Используя монаду Writer, напишите функцию правой свертки вычитанием

```
minusLoggedR :: (Show a, Num a) => a -> [a] -> Writer String a
```

```
minusLoggedR = do undefined
```

в которой рекурсивные вызовы сопровождались бы записью в лог, так чтобы в результате получалось такое поведение:

```
GHCI> runWriter $ minusLoggedR 0 [1..3]
```

```
(2, "(1-(2-(3-0)))")
```

Задача 9. (2 балла) Используя монаду Writer, напишите функцию левой свертки вычитанием

```
minusLoggedL :: (Show a, Num a) => a -> [a] -> Writer String a
```

```
minusLoggedL = do undefined
```

в которой рекурсивные вызовы сопровождались бы записью в лог, так чтобы в результате получалось такое поведение:

```
GHCI> runWriter $ minusLoggedL 0 [1..3]
```

```
(-6, "(((0-1)-2)-3)")
```

Задача 10. (1 балл) Напишите функцию, вычисляющую числа Фибоначчи с использованием монады State .

```
fib :: Int -> Integer
```

```
fib n = fst $ execState (replicateM n fibStep) (0,1)
```

```
fibStep :: State (Integer,Integer) ()
```

```
fibStep = do undefined
```

Задача 11. (2 балла) Напишите функцию

```
while :: IORef a -> (a -> Bool) -> IO () -> IO ()
```

```
while ref p action = do undefined
```

позволяющую кодировать «императивные циклы» следующего вида:

```
factorial :: Integer -> IO Integer
```

```
factorial n = do
```

```
  r <- newIORef 1
```

```
  i <- newIORef 1
```

```
  while i (<= n) ( do
```

```

ival <- readIORef i
modifyIORef r (* ival)
modifyIORef i (+ 1)
)
readIORef r

```

Задача 12. (2 балла) Вычислите усреднённый по сериям модуль отклонения количества орлов от своего теоретического среднего значения в серии (предполагается, что эта величина известна и равна половине длины серии n). Используйте глобальный системный генератор случайных чисел.

```

avgdev :: Int -> IO Double
avgdev k n = undefined

```

Задача 13. (1 балл) Вычислите усреднённый по сериям модуль отклонения количества орлов от своего теоретического среднего значения в серии (предполагается, что эта величина известна и равна половине длины серии n). Генератор случайных чисел подракумевается созданным с помощью mkStdGen и передается в процессе вычислений через монаду State (используйте реализованную на практике функцию randomRState).

```

randomRState :: (Random a, RandomGen g) => (a, a) -> State g a
randomRState (x,y) = do undefined
avgdev' :: Int -> Int -> State StdGen Double
avgdev' k n = undefined

```

Задача 14. (1 балл) Вычислите усреднённый по сериям модуль отклонения количества орлов от своего теоретического среднего значения в серии (предполагается, что эта величина известна и равна половине длины серии n). Генератор случайных чисел создайте с помощью mkStdGen. Решите задачу, не используя монад: явно передавая генератор между вычислениями или используя как источник случайности бесконечный список, возвращаемый randomRs.

```

avgdev'' :: Int -> Int -> Double
avgdev'' k n = undefined

```

Задача 15. (3 балла) Для $n = 1000$ и $k = 1000$ запишите в файл в виде гистограммы (ascii-art):

```

...
-5 xxxxxxxxxxxxxxxxxxxx
-4 xxxxxxxxxxxxxxxxxxxx
-3 xxxxxxxxxxxxxxxxxxxx
-2 xxxxxxxxxxxxxxxxxxxx
-1 xxxxxxxxxxxxxxxxxxxx
0 xxxxxxxxxxxxxxxxxxxx

```

1 xxxxxxxxxxxxxx
2 xxxxxxxxxxxxxx
...

абсолютные частоты округленных до целых отклонений количества орлов от среднего значения. Постарайтесь не включать пустые «хвосты» распределения и не допускать неровностей основания гистограммы из-за изменения знака и/или разрядности чисел

Критерии оценивания и шкала оценки домашнего задания №3.

Оценка	Критерии выставления оценки
«Отлично» (10)	Все тесты для текущей задачи на платформе sterik успешно пройдены
«Неудовлетворительно» (0)	Решение не проходит полный набор тестов.

Домашнее задание 4

Описание технологии применения домашнего задания как оценочного средства:

- задачи выдаются обучающимся после каждого практического занятия,
- решения задач обучающиеся представляют в виде кода на языке Haskell на платформе Sterik, в специальном закрытом курсе,
- сроки представления решений составляют одну неделю с момента выдачи задания.

Задача 1. (1 балл) Реализуйте следующий набор вспомогательных функций. Функция freeVars возвращает список свободных переменных термина

freeVars :: Expr -> [Symb]

freeVars = undefined

Функция freeTVars возвращает список свободных переменных типа (в STT все переменные типа свободные)

freeTVars :: Type -> [Symb]

freeTVars = undefined

Функция extendEnv расширяет контекст переменной с заданным типом

extendEnv :: Env -> Symb -> Type -> Env

extendEnv = undefined

Функция freeTVarsEnv возвращает список свободных типовых переменных контекста

freeTVarsEnv :: Env -> [Symb]

```
freeTVarEnv = undefined
```

Задача 2. (1 балл) Реализуйте функцию `appEnv`, позволяющую использовать контекст как частичную функцию из переменных в типы:

```
appEnv :: (MonadError String m) => Env -> Symb -> m Type
appEnv (Env xs) v = undefined
```

Обратите внимание на механизм обработки исключений, который мы будем использовать и в дальнейшем. Информация об исключении передается в виде строки; конкретная монада-обработчик исключений не фиксируется.

Ожидаемое поведение:

```
GHCi> let tx = (TVar "a" :-> TVar "b") :-> TVar "c"
GHCi> let ty = TVar "a" :-> TVar "b"
GHCi> let env = Env [("y",ty),("x",tx)]
GHCi> let Right res = appEnv env "x" in res
(TVar "a" :-> TVar "b") :-> TVar "c"
GHCi> let Right res = appEnv env "y" in res
TVar "a" :-> TVar "b"
GHCi> let Left res = appEnv env "z" in res
"There is no variable \"z\" in the environment."
```

Задача 3. (1 балл) Реализуйте функции, осуществляющие подстановку типов вместо переменных типа в тип (`appSubsTy`) и подстановку типов вместо переменных типа в контекст (`appSubsEnv`):

```
appSubsTy :: SubsTy -> Type -> Type
appSubsTy = undefined
appSubsEnv :: SubsTy -> Env -> Env
appSubsEnv = undefined
```

Задача 4. (2 балла) Реализуйте функцию, выполняющую композицию двух подстановок (носитель композиции является объединением носителей двух этих подстановок):

```
composeSubsTy :: SubsTy -> SubsTy -> SubsTy
composeSubsTy = undefined
```

Используя эту функцию сделайте тип `SubsTy` представителем класса типов `Monoid`.

Задача 5. (2 балла) Реализуйте алгоритм унификации, возвращающий для двух переданных типов наиболее общий унификатор или сообщение об ошибке, если унификация невозможна.

```
unify :: (MonadError String m) => Type -> Type -> m SubsTy
```

```
unify = undefined
```

Ожидаемое поведение:

```
GHCI> let Right sbs = unify (TVar "a" :-> TVar "b") (TVar "c" :-> TVar "d") in sbs
SubsTy [(("a",TVar "c"),("b",TVar "d"))]
GHCI> let Left tst = unify (TVar "a") (TVar "a" :-> TVar "a") in tst
"Can't unify (TVar \"a\") with (TVar \"a\" :-> TVar \"a\")!"
```

Задача 6. (5 баллов) Реализуйте алгоритм построения системы уравнений на типы для заданных контекста, терма и инициализирующего типа для терма:

```
equations :: (MonadError String m) => Env -> Expr -> Type -> m [(Type, Type)]
```

equations = undefined

Возможное поведение:

```
GHCI> let term = Lam "y" $ Var "x"
GHCI> let env = Env [("x",TVar "a" :-> TVar "b")]
GHCI> let Right eqs = equations env term (TVar "o") in eqs
[(TVar "d",TVar "a" :-> TVar "b"),(TVar "c" :-> TVar "d",TVar "o")]
GHCI> let Left err = equations (Env []) term (TVar "o") in err
"There is no variable \"x\" in the environment."
```

Используя equations , реализуйте алгоритм поиска главной пары для терма бестипового лямбда-исчисления:

```
principlePair :: (MonadError String m) => Expr -> m (Env, Type)
```

principlePair = undefined

Возможное поведение (с точностью до имен типовых переменных):

```
GHCI> let Right pp = principlePair (Var "x") in pp
(Env [("x",TVar "a")],TVar "a")
GHCI> let Right pp = principlePair (Var "f" :@ Var "x") in pp
(Env [("f",TVar "a" :-> TVar "b"),("x",TVar "a")],TVar "b")
GHCI> let Right pp = principlePair (Lam "x" $ Lam "y" $ Var "y") in pp
(Env [],TVar "a" :-> (TVar "b" :-> TVar "b"))
GHCI> let Left err = principlePair (Var "x" :@ Var "x") in err
"Can't unify (TVar \"a\") with (TVar \"a\" :-> TVar \"b\")!"
```

Оценка	Критерии выставления оценки
«Отлично» (10)	Все тесты для текущей задачи на платформе sterik успешно пройдены
«Неудовлетворительно» (0)	Решение не проходит полный набор тестов.

Примерный список экзаменационных вопросов

Примерный перечень вопросов к экзамену:

Теория

1. Сравнение функционального и императивного подходов к программированию.
2. Основы λ -исчисления. λ -термы, свободные и связанные переменные.
3. Подстановка λ -терма. Лемма подстановки.
4. α - и β -конверсии. η -конверсия и экстенциональная эквивалентность.
5. Распирения чистого лямбда-исчисления. δ -конверсия.
6. Кодирование булевых значений, кортежей в чистом бестиповом λ -исчислении.
7. Кодирование чисел Чёрча в чистом бестиповом λ -исчислении.
8. Теорема о неподвижной точке. Y -комбинатор.
9. Редексы. Одношаговая и многошаговая редукция. Нормальная форма. Редукционные графы.
10. Теорема Чёрча-Россера и её следствия.
11. Стратегии редукции. Теорема о нормализации. Механизмы вызова в функциональных языках.
12. Функция предшествования для чисел Чёрча. Комбинатор примитивной рекурсии.
13. Просто типизированное λ -исчисление в стиле Карри. Предтермы. Утверждения о типизации. Контексты. Правила типизации.
14. Просто типизированное λ -исчисление в стиле Чёрча. Предтермы. Утверждения о типизации. Контексты. Правила типизации.

15. Свойства систем просто типизированного λ -исчисления.
16. Связь между системами Карри и Чёрча. Проблемы разрешимости. Сильная и слабая нормализация.
17. Свойство универсальности правой свертки. Закон слияния foldr-map.
18. Понятие главного (наиболее общего) типа. Подстановки типа и их композиция. Унификаторы.
19. Алгоритм унификации.
20. Алгоритм построения системы ограничений.
21. Теорема Хиндли-Милнера.
22. Рекурсивные типы. μ -нотация.
23. Кагаморфизмы.
24. Анаморфизмы и гилеоморфизмы.
25. Зипперы. Контексты с дыркой.

Haskell

1. Основы программирования на Haskell. Связывание. Рекурсия. Базовые конструкции языка.
2. Основные встроенные типы языка Haskell. Система модулей. Частичное применение, каррирование.
3. Операторы и их сечения в Haskell. Бесточечный стиль.
4. Ошибки. Основание. Строгие и нестрогие функции. Ленивое и энергичное исполнение.
5. Алгебраические типы данных. Сопоставление с образцом, его семантика.
6. Объявления `type` и `newtype`. Метки полей.
7. Списки, стандартные функции для работы с ними. Генерация (выделение) списков.
8. Специальный полиморфизм. Классы типов. Объявления представителей. Классы типов `Eq`, `Ord`, `Enum` и `Bound`.
9. Внутренняя реализация классов типов.
10. Стандартные классы типов: `Num` и его наследники, `Show` и `Read`.
11. Моноиды. Представители класса типов `Monoid`.
12. Свёртки списков. Правая и левая свёртки. Энергичные версии. Развёртки.
13. Класс типов `Foldable` и его представители.

14. Класс типов Functor и его представители.
15. Класс типов Applicative и его представители.
16. Классы типов Alternative и MonadPlus .
17. Аппликативные парсеры.
18. Класс типов Traversable и его представители.
19. Монады. Класс типов Monad . Законы для монад. do-нотация.
20. Стандартные монады: Maybe и списки.
21. Ввод-вывод в чистых языках. Монада IO. Взаимодействие с файловой системой.
22. Монада Reader .
23. Монада Writer .
24. Монада State .
25. Монада Except .
26. Трансформеры монад. Библиотеки transformers и mtl.
27. Линзы ван Лаарховена.

Примерный список вопросов к зачету:

Описание технологии проведения зачета:

- зачет проводится в формате тестирования в виде модуля на платформе sterik.org,
- время, отводимое на выполнение теста составляет 90 минут,
- тест состоит из 5 вопросов по всем пройденным темам; по каждой из тем обучающемуся выводится пять утверждений, среди которых он должен отметить верные,
- за каждую из тем обучающийся получает 2 балла при правильном выборе всех верных утверждений и только их с первой попытки. Если при первой попытке это условие было нарушено, то можно осуществлять повторные попытки, до тех пор, пока указанный результат не будет достигнут. При этом баллы за задание рассчитываются по формуле $2/N$, где N – число попыток.
- результаты тестирования учитываются платформой sterik.org автоматически.

. Любой оператор является либо левоассоциативным, либо правоассоциативным.

. Функция определена на глобальном уровне через сопоставление с образцом, в виде нескольких равенств. Допустимо использовать общий для этих

равенств идентификатор `foo`, определенный в предложении `where foo = ...`

. Функция определена на глобальном уровне с помощью охранных выражений. Допустимо использовать общий для нескольких охранных выражений идентификатор `foo`, определенный в предложении `where foo = ...`

Функция определена на глобальном уровне с помощью охранных выражений. Допустимо использовать общий для нескольких охранных выражений идентификатор `foo`, определенный в предложении `let foo = ... in ...`

- Выражения `foo bar bas` и `(foo bar) bas` эквивалентны.
- Выражения `foo bar bas` и `foo (bar bas)` эквивалентны.
- Выражения `foo $ bar bas` и `foo (bar bas)` эквивалентны.
- Выражения `(foo $ bar) bas` и `foo bar bas` эквивалентны.
- Если тип всех переменных одинаков, выражение `a : [b,c] ++ d : [e,f]` успешно пройдет проверку типов.
- Если тип всех переменных одинаков, выражение `[a,b] : c ++ d : [e,f]` успешно пройдет проверку типов.
- Уменьшение отступа всегда означает завершение текущего подвыражения.
- Если выражение проходит проверку типов, и некоторое его подвыражение заменится на константу `undefined`, то получившееся выражение тоже будет проходить проверку типов.

• Если для функции `foo` не заданы приоритет и ассоциативность при использовании в операторном синтаксисе, то выражение `x `foo` y `foo` z` недопустимо.

• Если для функции не заданы приоритет и ассоциативность при использовании в операторном синтаксисе, то она имеет приоритет равный 9.

Выражения `foo $ bar bas` и `foo bar bas` эквивалентны.

5. Методические материалы для обучающихся

Конспект и презентации лекций.

6. Методические материалы для преподавателей

Методические материалы для преподавателей

Теоретическое содержание дисциплины состоит в рассмотрении основных положений и теоретических вопросов по предмету дисциплины в соответствии с целями и задачами, указанными в п. 1 рабочей программы дисциплины (РПД). Содержание лекционных занятий конкретизировано в соответствии с элементами теоретического, практического изучения и применения объектов, образующих предмет изучения дисциплины и включающих:

— основные понятия и их определения;

- особенности строения и функционирования объектов, их основные свойства, характеристики, параметры;
- задачи (проблемы) теоретического и/или практического изучения объектов, их создания и применения;
- методы, средства и способы их теоретического и/или практического изучения и совершенствования.

Глубина изложения теоретических вопросов определяется объемом часов, отведенных для занятий лекционного типа (лекции) и на самостоятельную работу обучающихся в рамках проработки материалов лекций.

7. Ресурсное обеспечение

7.1. Кадровое обеспечение

Руководитель программы: А.В. Омельченко - профессор, доктор физико-математических наук, декан факультета Санкт-Петербургская школа физико-математических и компьютерных наук НИУ ВШЭ – Санкт-Петербург, руководитель направления «Вычислительная биология» АНО ВО «Университет «Сириус».

Авторы программы:

А.В. Омельченко: профессор, доктор физико-математических наук, декан факультета Санкт-Петербургская школа физико-математических и компьютерных наук НИУ ВШЭ – Санкт-Петербург, руководитель направления «Вычислительная биология» АНО ВО «Университет «Сириус»

7.2. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине:

В рамках самостоятельной работы обучающиеся используют интернет-ресурсы:

1. <https://edu.rsc.org/resources/analysis>
2. <https://collection.asdlib.org/>

7.3. Перечень основной учебной литературы

1. Learn You a Haskell for Great Good!: A Beginner's Guide / By: Lipovača, Miran. No Starch Press. 2011

2. Haskell Data Analysis Cookbook / By: Nishant Shukla. Packt Publishing. 2014
3. Pearls of functional algorithm design / By: Bird, Richard. Cambridge University Press. 2010 Pearls of functional algorithm design / By: Bird, Richard. Cambridge University Press. 2010

7.4. Перечень дополнительной учебной литературы

1. Beginning Haskell: A Project-Based Approach / By: Mena, Alejandro Serrano. Apress. 2014
2. Кубенский, А. А. Функциональное программирование: учебник и практикум для академического бакалавриата / А. А. Кубенский. — М.: Издательство Юрайт, 2018. — 348 с.

7.5. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»:

1. <https://stepik.org/course/75/promo>

7.6. Описание материально-технической базы:

7.6.1. Аудиторный фонд и оборудование:

№ п.п.	Вид аудитории	Технические средства и оборудование (кол-во)	Расходные материалы (кол-во)
1.	Учебная аудитория для проведения лекционных занятий	Компьютер с проектором Меловая или маркерная доска, мел или маркер.	Мел или маркер 15 шт.
2.	Учебная аудитория для проведения практических занятий	Компьютер с проектором Меловая или маркерная доска, мел или маркер.	Мел или маркер 10 шт.

7.6.2. Оборудование для лабораторных и практических занятий:

Учебная аудитория для проведения практических занятий

7.7. Перечень лицензионного и свободно распространяемого программного обеспечения:

Для проведения всех занятий используется проектор и компьютер для проекции слайдов. Для самостоятельной работы необходимо следующее ПО: пакет библиотек для Python (Anaconda), инструмент для сборки Haskell (Stack), компилятор C++ (clang).

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАУЧНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «СИРИУС»

(АНО ВО «УНИВЕРСИТЕТ «СИРИУС»)

Научный центр информационных технологий и искусственного интеллекта

СОГЛАСОВАНО
Ученым советом

АНО ВО «Университет «Сириус»
(протокол от «03» августа 2021 г. № 05)

УТВЕРЖДАЮ

Руководитель отдела по учебно-методической работе

АНО ВО «Университет «Сириус»

Г.В. Федоров

«03» августа 2021 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Исследование лекарственных средств

наименование дисциплины

Уровень высшего образования:

магистратура

бакалавриат, специалитет, магистратура, аспирантура

Направление подготовки (специальность):

01.04.02 Прикладная математика и информатика

код и название направления/специальности

Направленность (профиль):

Биоинформатика и математическая биология

код и название направления/специальности

Форма обучения:

очная

очная / очно-заочная / заочная
Федеральная территория «Сириус» - 2021 г.

Общая характеристика дисциплины

1.1. Место дисциплины в структуре основной профессиональной образовательной программы высшего образования (ОПОП ВО):

Уровень высшего образования: магистратура;

Направление подготовки: 01.04.02 Прикладная математика и информатика

Направленность (профиль): Биоинформатика и математическая биология (далее – образовательная программа, ОПОП ВО);

Форма реализации (очная, дистанционная): очная с возможностью применения электронного обучения и дистанционных образовательных технологий.

1.2. Место дисциплины в образовательной программе и описание взаимосвязи дисциплины с другими реализуемыми в АНО ВО «Университет «Сириус» образовательными модулями, образовательными программами, научными исследованиями и проектами:

В рамках ОПОП ВО дисциплина реализуется в третьем семестре.

Изучению дисциплины в рамках ОПОП ВО предшествует изучение дисциплин: Основы программирования (Python, Unix) и C++, Компьютерная разработка лекарственных средств и основы хемоинформатики.

Изучение дисциплины необходимо для освоения следующих дисциплин ОПОП ВО: Магистерская диссертация.

Дисциплина может быть использована в программах магистратуры 01.04.02 Прикладная математика и информатика в качестве обязательной дисциплины или дисциплины по выбору.

1.3. Входные требования для освоения дисциплины: знание основ физики и органической химии, умение решать простейшие задачи из курсов физики и органической химии.

1.4. Цель дисциплины: освоение современных методов исследования лекарственных средств.

1.5. Задачи дисциплины: формирование у студентов теоретических знаний и практических навыков, связанных с использованием инструментальных физико-химических методов исследования лекарственных средств и обработкой получаемых с их помощью данных в ходе разработки лекарственных препаратов и при исследовании качества готовых лекарственных форм.

1.6. Язык преподавания: русский.

1.7. Результаты обучения по дисциплине, соотношенные с формируемыми компетенциями:

Формируемые компетенции (код компетенции, формулировка)	Планируемые результаты обучения по дисциплине
ПК-1. Способен решать актуальные задачи фундаментальной и прикладной математики	ПК-1.1. Знает основы фундаментальных математических, биологических дисциплин и других естественных наук. ПК-1.2. Умеет формулировать, анализировать и решать профессиональные задачи с применением фундаментальных знаний математики, биологии и других естественных наук.

ПК-2. Способен реализовывать и совершенствовать существующие и вводить новые математические методы решения прикладных задач	ПК-1.3. Имеет практический опыт постановки и решения актуальных задач математической биологии и других наук ПК-2.1. Знает основные математические методы решения прикладных задач. ПК-2.2. Умеет осуществлять анализ и выбор методов решения задач профессиональной и научной деятельности на основе теоретических знаний в области математических и компьютерных наук. ПК-2.3. Имеет практический опыт разработки новых методов математического моделирования для решения задач профессиональной и научной деятельности
ПК-3. Способен разрабатывать математические модели и проводить их анализ при решении задач в области профессиональной деятельности	ПК-3.1. Знает основные положения, терминологию и методологию в области математического и компьютерного моделирования. ПК-3.2. Умеет выбирать, использовать и разрабатывать необходимые методы математического и компьютерного моделирования в зависимости от поставленных задач. ПК-3.3. Имеет практический опыт применения методов вычислительной биологии в решении прикладных и управленческих задач
ПК-7. Способен самостоятельно анализировать поставленную задачу, использовать корректные методы её решения, применять математически сложные алгоритмы, определять необходимые системные и программные средства для разработки и отладки прикладного программного обеспечения в современных специализированных программных комплексах, реализовывать в них новые алгоритмы	ПК-7.1. Знает теоретические основы и методологию построения решений фундаментальных задач, основы информационных технологий. ПК-7.2. Умеет самостоятельно осуществлять анализ и выбор методов и алгоритмов решения задач профессиональной деятельности.

	ПК-7.3. Имеет практический опыт решения задач в соответствии с выбранным методом и построенным алгоритмом с использованием современных программных комплексов.
ПК-8. Умеет самостоятельно разрабатывать, исследовать, применять программное обеспечение, проводить расчётные работы и исследования, обработку результатов, оформление отчётной документации	ПК-8.1. Знает классические методы решения задач, современные программные комплексы для проведения расчётных исследований. ПК-8.2. Умеет самостоятельно проводить расчётные исследования, выбирать и применять современные программные комплексы, получать, обрабатывать и анализировать результаты исследований. ПК-8.3. Имеет практический опыт применения математического и компьютерного моделирования.

2. Структура дисциплины

2.1. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся:

объем дисциплины составляет 3 з.е., 108 ак. ч., из которых 48 ак. ч. составляет контактная работа обучающихся с преподавателем (24 ак. ч. занятий лекционного типа, 24 ак. ч. лабораторных, практических занятий, семинаров), 60 ак. ч. составляет самостоятельная работа обучающихся.

2.2. Календарный график реализации дисциплины (в случае реализации дисциплины в формате интенсива):

день	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1 пара	лек	лек	лек	лек	лек	лек	лек	лек	лек	лек	лек	лек	лек	лек	пр	пр	пр	сам	ЭКЗ
2 пара	лек	лек	лек	лек	пр	пр	пр	пр	пр	пр	пр	сам	сам	сам	сам	сам	сам	сам	ЭКЗ
3 пара	пр	пр	пр	пр	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	ЭКЗ
4 пара	пр	пр	пр	пр	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	сам	ЭКЗ

В таблице использованы следующие сокращения: лек – лекция, пр – практическое занятие, сам – самостоятельная работа, экз – экзамен.

2.3. Учебный план дисциплины и её содержание, структурированное по темам (разделам):

Наименование разделов и тем дисциплины	Всего (ак. часы)	В том числе						Самостоятельная работа обучающегося, ак. часы	Форма текущего контроля успеваемости и промежуточной аттестации
		Контактная работа (работа во взаимодействии с преподавателем), ак. часы из них							
		Лекции	Лабораторные занятия	Практические занятия	Семинары	Учебные занятия, направленные на проведение текущего контроля успеваемости, промежуточной аттестации	Всего		
Введение. Краткая история дисциплины. Общая постановка задачи оптимизации. Примеры	18	4		4			8	10	
Классификация лекарственных соединений в соответствии с их химическим строением, происхождением, фармакологической группой и назначением.	18	4		4			8	10	

[illegible]

[illegible]

Домашнее задание №1 выдается студентам в одном варианте и состоит из трех задач, которые требуется решить с использованием информации из интернета и дать исчерпывающий ответ. Срок выполнения домашнего задания - 1 неделя. Форма представления обучающимися домашнего задания - представленные в письменном виде ответы.

1. Определите основные функциональные группы следующих лекарств: новокаин, ибупрофен, толбутамид, аспирин, левофлуксацин. К какой группе можно отнести каждое из веществ по фармакологической группе и химическому происхождению?

2. Сравните кислотно-основное титрование с применением индикаторов и потенциометрическое титрование. Какие преимущества и недостатки у этих методов?

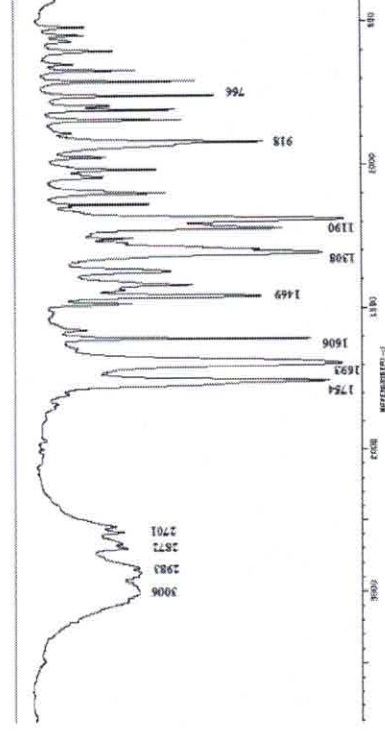
3. Постройте кривую титрования ибупрофена раствором щелочи.

Домашнее задание №2 выдается студентам в одном варианте и состоит из трех задач, которые требуется решить. Срок выполнения домашнего задания - 1 неделя. Форма представления обучающимися домашнего задания - представленные в письменном виде решения.

1. . Рассчитать время удерживания и удерживаемый объем компонента, элюирующегося из колонки, имеющей 200 теоретических тарелок, при скорости движения диаграммной ленты 720 мм/ч, если полуширина хроматографического пика составляет 3 мм. Объемная скорость газа-носителя равна 30 мл/мин.

2. Рассчитайте содержание кортизона ацетата в таблетках, если навеску порошка растертых таблеток массой 0,1157 г растворили в 100 мл этанола, отфильтровали от таблеточной массы. 5,0 мл полученного раствора довели этанолом до 100 мл. Оптическая плотность полученного раствора при 238 нм равна 0,520. Удельный показатель поглощения стандартного образца кортизона ацетата в тех же условиях равен $390,0 \text{ \%}^{-1} \cdot \text{см}^{-1}$. Средняя масса одной таблетки – 0,2140 г.

3. На рисунке приведен ИК-спектр аспирина. Установите соответствие между основными сигналами в спектре и фрагментами молекулы.



Домашнее задание №3 выдается студентам в одном варианте и состоит из ситуационной задачи, которую требуется решить. Срок выполнения домашнего задания - 1 неделя. Форма представления обучающимися домашнего задания - представленный в письменном виде ответ.

1. При выполнении контрольной работы по анализу лекарственной формы, содержащей рибофлавина 0,002 г; калия иодида 0,2 г; глюкозы 0,2 г; воды очищенной до 10 мл, студент предложил определить содержание иодида калия argentометрическим методом Фаянса, содержание глюкозы иодометрическим обратным титрованием, а содержание рибофлавина по имидной группе косвенной алкаиметрией. Преподаватель отметил, что в схеме анализа допущена ошибка. Укажите, какую ошибку допустил студент?

Примерный список экзаменационных вопросов

1. Вольтамперометрия. 2. ИК-спектроскопия.
2. Кислотно-основное титрование. 2. Скрининг лекарственных кандидатов.
3. Электромагнитное излучение и его взаимодействие с молекулами. Спектры. 2. Вискозифективная жидкостная хроматография.
4. Определение растворимости лекарственного препарата. 2. Исследования взаимодействия лекарственных соединений с мишенями.
5. Окислительно-восстановительное титрование. 2. Определение коэффициентов межфазного распределения. Гидрофобность, липофильность.
6. Определение гигроскопичности лекарственного препарата. 2. Флуоресцентная спектроскопия.
7. Методы пробоподготовки. 2. Газовая хроматография.
8. Окислительно-восстановительное титрование. 2. Исследования взаимодействия лекарственных соединений с мишенями.
9. Потенциометрическое титрование. 2. Стандартные тесты фармацевтических материалов.
10. Кислотно-основное титрование. 2. ИК-спектроскопия.
11. Определение гигроскопичности лекарственного препарата. 2. Анализ лекарств в биологических жидкостях.
12. Типы лекарственных форм. 2. Связь результатов *in vitro* с ADMET-свойствами *in vivo*.
13. Определение температуры фазовых переходов лекарственных препаратов. 2. Скрининг лекарственных кандидатов.
14. Определение растворимости лекарственного препарата. 2. Хромато-масс-спектроскопия.
15. Определение температуры фазовых переходов лекарственных препаратов. 2. Скрининг лекарственных кандидатов.
16. Вольтамперометрия. 2. Газовая хроматография.
17. Окислительно-восстановительное титрование. 2. ИК-спектроскопия.
18. Определение температуры фазовых переходов лекарственных препаратов. 2. ИК-спектроскопия.
19. Дифференциальная сканирующая калориметрия. 2. Хромато-масс-спектроскопия.
20. Определение температуры фазовых переходов лекарственных препаратов. 2. Стандартные тесты фармацевтических материалов.
21. Электромагнитное излучение и его взаимодействие с молекулами. Спектры. 2. Анализ лекарств в биологических жидкостях.
22. Типы лекарственных форм. 2. Определение коэффициентов межфазного распределения. Гидрофобность, липофильность.
23. Методы пробоподготовки. 2. Связь результатов *in vitro* с ADMET-свойствами *in vivo*.
24. Потенциометрическое титрование. 2. Флуоресцентная спектроскопия.
25. Кислотно-основное титрование. 2. Связь результатов *in vitro* с ADMET-свойствами *in vivo*.

26. Термогравиметрия. 2. Анализ лекарств в биологических жидкостях.
27. Методы пробоподготовки. 2. Стандартные тесты фармацевтических материалов.
28. Определение гигроскопичности лекарственного препарата. 2. Газовая хроматография.
29. Атомно-абсорбционная спектроскопия. 2. Стандартные тесты готовых лекарственных препаратов.
30. Дифференциальная сканирующая калориметрия. 2. Хромато-масс-спектрометрия.

5. Методические материалы для обучающихся

Конспект и презентации лекций

6. Методические материалы для преподавателей

Теоретическое содержание дисциплины состоит в рассмотрении основных положений и теоретических вопросов по предмету дисциплины в соответствии с целями и задачами, указанными в п. 1 рабочей программы дисциплины (РПД). Содержание лекционных занятий конкретизировано в соответствии с элементами теоретического, практического изучения и применения объектов, образующих предмет изучения дисциплины и включающих:

- основные понятия и их определения;
- особенности строения и функционирования объектов, их основные свойства, характеристики, параметры;
- задачи (проблемы) теоретического и/или практического изучения объектов, их создания и применения;
- методы, средства и способы их теоретического и/или практического изучения и совершенствования.

Глубина изложения теоретических вопросов определяется объемом часов, отведенных для занятий лекционного типа (лекции) и на самостоятельную работу обучающихся в рамках проработки материалов лекций.

7. Ресурсное обеспечение

7.1. Кадровое обеспечение

Руководитель программы: А.В. Омельченко - профессор, доктор физико-математических наук, декан факультета Санкт-Петербургская школа физико-математических и компьютерных наук НИУ ВШЭ – Санкт-Петербург, руководитель направления «Вычислительная биология» АНО ВО «Университет «Сириус».

Авторы программы: И.А. Седов: доцент, доктор химических наук, ведущий научный сотрудник Химического института им. А.М. Бутлерова Казанского федерального университета

7.2. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине:

В рамках самостоятельной работы обучающиеся используют интернет-ресурсы:

1. <https://edu.rsc.org/resources/analysis>
2. <https://collection.asdlib.org/>

7.3. Перечень основной учебной литературы:

1. Handbook of Pharmaceutical Analysis / L. Ohannesian; A. J. Streeter. Marcel Dekker. 2002.
2. Фармацевтическая химия. 1. теоретический материал по темам. Под редакцией академика РАМН, профессора А.П. АРЗАМАСЦЕВА. Москва 2004.

7.4. Перечень дополнительной учебной литературы:

1. Introduction to Pharmaceutical Chemical Analysis / S.Hansen; S. Pedersen-Bjergaard; K.Rasmussen. John Wiley & Sons Ltd.2018.
2. Основные методы исследования лекарственных средств: фармацевтических препаратов, лекарственного растительного сырья и лекарственных форм/ Т. А. Четкина. — М. : Издательство Медицина, 1999. — 127 с. ЭБС ВОЗ.

7.5. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»:

1. <https://www.click2drug.org/>
2. <http://www.rcsb.org/>
3. <http://zinc15.docking.org/>

7.6. Описание материально-технической базы:

7.6.1. Аудиторный фонд и оборудование:

№ п.п.	Вид аудитории	Технические средства и оборудование (кол-во)	Расходные материалы (кол-во)
1.	Учебная аудитория для проведения лекционных занятий	Компьютер с проектором Меловая или маркерная доска, мел или маркер.	Мел или маркер 15 шт.
2.	Учебная аудитория для проведения практических занятий	Компьютер с проектором Меловая или маркерная доска, мел или маркер.	Мел или маркер 10 шт.
3	Химическая лаборатория	Оборудование для проведения физико-химических исследований, химическая посуда и реактивы	

7.6.2. Оборудование для лабораторных и практических занятий:

Оборудование для проведения физико-химических исследований, химическая посуда и реактивы.

7.7. Перечень лицензионного и свободно распространяемого программного обеспечения:

Для проведения всех занятий используется проектор и компьютер для проекции слайдов. Для самостоятельной работы необходимо следующее ПО: пакет библиотек для Python (Anaconda), инструмент для сборки Haskell (Stack), компилятор C++ (clang). Кроме того, для работы будут использованы программы Mathematica, OriginPro, Office, Mnova.