

государственное бюджетное образовательное учреждение
высшего образования Московской области
«Университет «Дубна»
(государственный университет «Дубна»)

Институт системного анализа и управления
Кафедра системного анализа и управления



Рабочая программа дисциплины (модуля)

Разработка приложений на JAVA

наименование дисциплины (модуля)

Направление подготовки (специальность)

01.03.02 Прикладная математика и информатика

Уровень высшего образования

Бакалавриат

Направленность (профиль) программы (специализация)

Математическое моделирование

Форма обучения

очная

Дубна, 2021


Преподаватель:
старший преподаватель Филозова И. А.
кафедра Системного анализа и управления


подпись


Рабочая программа разработана в соответствии с требованиями ФГОС ВО по направлению подготовки высшего образования **01.03.02 Прикладная математика и информатика**


Программа рассмотрена на заседании кафедры системного анализа и управления

Протокол заседания № 12 от «11» июня 2021 г.

Заведующий кафедрой д. т. н. профессор /  / Черемисина Е. Н.

СОГЛАСОВАНО

Заведующий кафедрой РИВС д. т. н. профессор /  / Кореньков В. В.

И.о. директора Института САУ /  / Черемисина Е.Н.

Эксперт Айрян Э.А., к.ф.-м.н., старший научный сотрудник, начальник сектора ЛИТ ОИЯИ

(Ф.И.О., ученая степень, ученое звание, место работы, должность; подпись, заверенная по месту работы)

Подпись Айряна Э.А. заверяю:

к.ф.-м.н., ученый секретарь ЛИТ ОИЯИ




Подгайный Д.В.

1. Цели и задачи освоения дисциплины (модуля)

Целью освоения дисциплины является освоение технологии создания программного обеспечения на платформе Java EE. Основными задачами изучения дисциплины являются обеспечение студентов знаниями о методах и приемах разработки программного обеспечения на платформе Java EE, освоение на практике технологии разработки, тестирования и развертывания программного обеспечения на платформе Java EE.

2. Объекты профессиональной деятельности при изучении дисциплины (модуля)

Объектами профессиональной деятельности в рамках изучаемой дисциплины являются: информатика и управление; теория, алгоритмы, приложения; математическое и информационное обеспечение экономической деятельности; математические методы и программное обеспечение защиты информации; математическое и программное обеспечение компьютерных сетей; информационные системы и их исследование методами математического прогнозирования и системного анализа; интеллектуальные системы; программная инженерия; системы управления предприятием; сетевые технологии.

3. Место дисциплины (модуля) в структуре ОПОП

Дисциплина входит в вариативную обязательную часть профессионального цикла Б1.В.ОД.8, изучается во 7-м семестре, форма промежуточной аттестации – зачет с оценкой.

Дисциплины, которые предшествуют текущей дисциплине:

Информатика;

Структуры и алгоритмы обработки данных;

Программные технологии ИНТЕРНЕТ.

Дисциплина является предшествующей для изучения следующих дисциплин:

Обобщенное программирование.

4. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников)

Формируемые компетенции (код компетенции, уровень (этап) освоения) (последний – при наличии в карте компетенции)	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций
ОПК-2 – способностью приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии	З2(ОПК-2) Знать: теорию баз данных, языки запросов и языки программирования для работы с базами данных У3(ОПК-2) Уметь: разрабатывать структуру баз данных, выполнять реализацию хранения и логики обработки данных
ОПК-3 – способностью к разработке ал-	З1(ОПК-3) Знать: основные парадигмы и

<p>горитмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям</p>	<p>языки программирования</p> <p>37 (ОПК-3) Знать: архитектуру и стандарты технологий платформы JavaEE</p> <p>У2(ОПК-3) Уметь: выполнять проектирование и конструирование компонентов программного обеспечения в рамках поставленного задания *)</p> <p>У3(ОПК-3) Уметь: выбирать алгоритмы и выполнять их программную реализацию для решения типовых задач предметной области</p> <p>У5(ОПК-3) Уметь: устанавливать, испытывать и использовать программные средства</p> <p>В1(ОПК-3) Владеть: современными интегрированными средами разработки программного обеспечения</p> <p>В2(ОПК-3) Владеть: навыками тестирования, отладки и верификации программ *)</p>
<p>ПК-1 – способностью собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям</p>	<p>34(ПК-1) Знать: Базовые знания о структуре, основных принципах и технологиях компьютерных сетей, необходимые для решения практических задач в области информационных систем и технологий</p>
<p>ПК-15 – способностью проводить сбор и анализ научно-технической информации в сети Интернет и из других источников</p>	<p>В2(ПК-15): Владеть: навыками разработки информационных запросов к базам данным, информационно-поисковым и решающим системам в различных областях знания *)</p>
<p>ПК-16 – способностью к разработке эффективных программных решений и пониманию социальных и этических аспектов внедрения этих решений.</p>	<p>У3(ПК-16) Уметь: оценивать временную и ресурсную трудоемкость программной реализации алгоритма</p>
<p>ПК-17 – способностью к разработке и применению алгоритмических и программных решений, математических методов обработки данных в области профессиональной деятельности</p>	<p>31(ПК-17) Знать: формальные методы и инструменты разработки программного продукта</p>

*) Результат обучения сформулирован на основании требований профессионального стандарта: 06.001 «Программист», для выполнения обобщенной трудовой функции D: «Разработка требований и проектирование программного обеспечения»

**) Результат обучения сформулирован на основании требований профессионального стандарта: 06.015 «Специалист по информационным системам» для выполнения обобщенной трудовой функции С: «Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы»

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

Объем дисциплины (модуля) составляет 4 зачетных единиц, всего 144 часов, из которых:

34 часов составляет контактная работа обучающегося с преподавателем¹:

34 часа – практические занятия;

часов мероприятия промежуточной аттестации (зачёт с оценкой в 7-м семестре);

110 часов составляет самостоятельная работа обучающегося.

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

¹ Перечень видов учебных занятий уточняется в соответствии с учебным планом.

Наименование и краткое содержание разделов и тем дисциплины (модуля) Форма промежуточной аттестации по дисциплине (модулю)	Всего (часы)	В том числе:								
		Контактная работа (работа во взаимодействии с преподавателем), часы из них ²						Самостоятельная работа обучающегося, часы, из них		
		Лекционные занятия	Семинарские занятия	Практические занятия	Лабораторные занятия	Учебные занятия, направленные на проведение текущего контроля успеваемости (коллоквиумы, практические контрольные занятия и др.)*	Всего	Выполнение домашних заданий	Подготовка рефератов и т.п.	Всего
Технология Java и ее применение. Базовые сведения о Java. Характеристика. Особенности. Виртуальная машина Java. Платформа Java. Версии Java: Jave SE, Java EE, Java ME, Android SKD. Обзор сред разработки. Синтаксис. Типы данных. Объ- ектно-ориентированное программирование на Java.	52			14		ПР-2.1	14	38		38
Java SE 8. Принципы объектно-ориентированного дизайна прило- жений. Обработка исключений. Коллекции. Базовый ввод-вывод. Модель обработки событий в JAVA. Графи- ческий интерфейс пользователя. Персистентность данных в объектно-ориентированных приложениях. Организация взаимодействия с базами данных. Многопоточное про- граммирование в Java. Сетевое программирование. Сред- ства пакета java.net. Лямбда функции и функциональное программирование на Java.	19			9		ПР-2.2 - ПР-2.9	9	10		10
Разработка веб-приложений и веб-сервисов на платформе Java EE. Спецификация Java EE. Организация сборки проекта на Java. Архитектура приложения уровня предприятия. Servlet API. Разработка пользовательского интерфейса при помощи JSF. Технология ORM. Обзор EJB. Работа с Web Services.	73			11		ПР-2.9 - ПР-2.17	11	62		62
Промежуточная аттестация (зачет с оценкой)**										

² Перечень видов учебных занятий уточняется в соответствии с учебным планом.

Итого	144	0		34			34	110		110
--------------	-----	---	--	----	--	--	----	-----	--	-----

* Опрос (ПР-1), Практические работы (ПР-2). Текущий контроль проводится за счет времени, отведенного на аудиторные занятия.

** Промежуточная аттестация может проходить как в традиционных форма (зачет, экзамен), так и в иных формах: балльно-рейтинговая система, защита порт-фолио, комплексный экзамен, включающий выполнение практических заданий (возможно наряду с традиционными ответами на вопросы по программе дисциплины (модуля)).

7. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю) и методические указания для обучающихся по освоению дисциплины (модулю)

Дано описание рекомендуемого режима и характера учебной работы, в том числе в части выполнения самостоятельной работы, методические указания по проведению конкретных видов учебных занятий, а также методические материалы к используемым в учебном процессе техническим средствам, информационно-коммуникационным и образовательным технологиям.

Методические указания для студентов

Практические (семинарские) занятия

Практические занятия по курсу «Разработка приложений на JAVA» имеют цель ознакомить студентов с одним из самых востребованных на рынке корпоративных приложений языков программирования Java, выработать и развить навыки разработки приложений на Java.

Прохождение всего цикла практических занятий является условием допуска студента к зачету.

Студент должен вести активную познавательную работу, которая заключается в применении полученных теоретических знаний, типовых программных продуктов для решения прикладных задач.

Тематика практических работ.

Объектно-ориентированное программирование на Java.

Средства пакета java.util.

Обработка исключений.

Сетевые средства.

Многопоточные приложения.

Организация работы с базами данных (JDBC API).

Функциональное программирование на Java.

Разработка приложений на платформе JavaFX.

Сервлеты.

Серверные страницы Java (JSP). Взаимодействие JSP и сервлета.

Сессии, обработчики web событий.

Технология EJB.

Доступ к базам данных с использованием ORM фреймворков. Java Persistence API, Hibernate.

Web-сервисы.

Методические рекомендации для преподавателя

При реализации дисциплины «Разработка приложений на Java» проводятся практические занятия, предусмотрено выполнение практических заданий, самостоятельная работа студентов по углубленному рассмотрению отдельных разделов дисциплины.

Содержание дисциплины отвечает следующим дидактическим требованиям:

- изложение материала от простого к сложному, от известного к неизвестному;
- логичность, четкость и ясность в изложении материала;
- возможность проблемного изложения, дискуссии, диалога с целью активизации деятельности студента;
- тесная связь теоретических положений и выводов с практикой и будущей профессиональной деятельностью студентов.

Практические занятия курса проводятся по ключевым и наиболее важным темам, разделам учебной программы. Формы организации обучения предполагают также выполнение практических работ по индивидуальному заданию в малых группах по 2-3 человека. Практические работы студенты начинают выполнять на семинаре (установочное занятие). Часть работы выполняется самостоятельно.

При изучении дисциплины предусматривается использование интерактивных форм проведения занятий. Проводятся устные опросы по рассматриваемым темам. Студенты участвуют в дискуссии, задают друг другу вопросы.

Методическое обеспечение инновационных форм учебных занятий

В рамках данной дисциплины предусмотрено несколько видов интерактивных образовательных технологий – тренинг, работа в малых группах (команде), проблемное обучение, проектный метод, опережающая самостоятельная работа.

1. **Тренинг** – форма активного обучения, целью которого является передача знаний, развитие умений и навыков разработки web-приложений.
2. **Работа в малых группах (команде)** – совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности.
3. **Проблемное обучение** – стимулирование студентов к самостоятельному приобретению знаний, необходимых для решения конкретной проблемы.

Задачи проблемного обучения:

1. Развитие мышления и способностей студентов, развитие творческих умений.

2. Усвоение студентами знаний, добытых в ходе активного поиска и самостоятельного решения проблем. В результате эти знания, умения более прочные, чем при традиционном обучении.
3. Воспитание активной, творческой личности студента, умеющего видеть и решать нестандартные профессиональные проблемы.

Три основных метода проблемного обучения:

1. Проблемное изложение. Преподаватель ставит проблемы и сам их решает.
2. Частично-поисковая деятельность. Постепенное приобщение к решению проблем.
3. Самостоятельная исследовательская деятельность. Студенты самостоятельно формулируют проблему и решают её под контролем преподавателя.

Основные условия успешного проблемного обучения:

1. необходимо вызвать интерес к содержанию проблемы;
 2. обеспечить посильность работы с возникающими проблемами;
 3. получаемая при решении информация должна быть значимой,
 4. важной в учебно-профессиональном плане;
 5. проблемное обучение реализуется успешно лишь при определенном стиле общения между преподавателем и обучаемыми, когда возможна свобода выражения своих мыслей и взглядов учениками при пристальном доброжелательном внимании преподавателя к мыслительному процессу учащегося. В результате, такое общение в виде диалога направлено на поддержание познавательной, мыслительной активности студентов.
4. **Опережающая самостоятельная работа** - изучение студентами нового материала до его изучения в ходе аудиторных занятий.
 5. **Проектный метод обучения** – это совокупность таких приемов и способов обучения, при которых студенты с помощью коллективной или индивидуальной деятельности по отбору, распределению и систематизации материала по определенной теме, составляют проект.

Проект – это самостоятельная, оригинальная работа, выполняемая студентами в соответствии с избранной ими темой-проблемой и включающая в себя отбор, распределение и информатизацию материала.

Результатом этой деятельности всегда должен быть какой-либо продукт: программа, комплект технической документации, брошюра, альбом, сценарий и т.д.

Задача преподавателя – с учетом возрастных и индивидуальных особенностей студентов, их интересов, потребностей, планов на будущее – создать такую рабочую атмо-

сферу, которая бы стимулировала их мыслительную, коммуникативную и творческую деятельность, в частности:

- на подготовительном этапе инициировать идеи проекта или создать условия для появления идеи проекта, а также оказать помощь в первоначальном планировании;
- на этапе реализации проекта – роль помощника, консультанта по отдельным вопросам, источника дополнительной информации. Существенная роль отводится координации действий между отдельными микрогруппами и участниками проекта.
- на заключительном этапе – подведение итогов работы в качестве независимого эксперта (контрольно-оценочная функция).

Студенты:

- самостоятельно и с желанием получают знания из разных источников;
- учатся пользоваться этими знаниями для решения новых познавательных и практических задач;
- приобретают коммуникативные умения, работая в разных группах;
- развивают свои исследовательские умения (выявление проблемы, сбор информации из литературы, документов и т.д., наблюдение, эксперимент, анализ, построение гипотез, обобщение);
- развивают аналитическое мышление.

Методические указания для самостоятельной работы обучающихся

Самостоятельная работа студентов, предусмотренная учебным планом, заключается в выполнении домашних заданий. Отдельные темы теоретических разделов курса прорабатываются студентами самостоятельно в соответствии с планом самостоятельной работы и конкретными заданиями преподавателя с учетом индивидуальных особенностей студентов.

8. Применяемые образовательные технологии для различных видов учебных занятий и для контроля освоения обучающимися запланированных результатов обучения

Перечень обязательных видов работы студента включает:

- посещение практических занятий;
- выполнение и защита практических работ;
- выполнение домашних работ.

Интерактивные образовательные технологии, используемые в аудиторных занятиях.

Семестр	Вид за- нятия (Л, ПЗ)	Используемые интерактивные образова- тельные технологии	Количество часов
7	ПЗ	Тренинги	14
		Проблемное обучение	
		Работа по индивидуальному заданию, про- ектный метод	
Итого интерактивных занятий, часов:			14

В начале практического занятия студенты получают теоретическую справку, содержащую сведения, необходимые для успешного выполнения задания. Справочный материал излагается в виде презентации и/или электронного документа. Некоторые разделы теоретического курса изучаются с использованием опережающей самостоятельной работы: студенты получают задания на ознакомление с новым материалом до его изучения на практических занятиях.

При прохождении практических занятий студентам предлагается работать в малых группах: учебная группа разбивается на несколько небольших групп – по 2-3 человека.

Каждая группа выполняет задание по индивидуальному варианту. Процесс выполнения заданий осуществляется на основе обмена мнениями и выбора способа решения. После этого каждый студент приступает к выполнению практического задания на компьютере и оформлению проделанной работы.

Методы обучения включают использование средств мультимедийного представления информации (презентации, ролики, схемы, модели, диаграммы).

9. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю)

9.1 Перечень компетенций выпускников образовательной программы с указанием результатов обучения (знаний, умений, владений), характеризующих этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Полный перечень компетенций выпускников образовательной программы **01.03.02 Прикладная математика и информатика** с указанием результатов обучения (знаний, умений, владений), характеризующих этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования приведен в картах компетенций образовательной программы.

Перечень компетенций выпускников образовательной программы **01.03.02 Прикладная математика и информатика**, в формировании которых участвует данная дисциплина представлен в разделе 4 рабочей программы дисциплины. Указание результатов обучения (знаний, умений, владений), характеризующих этапы формирования компетенций, описание показателей и критериев оценивания компетенций на различных этапах их формирования представлено ниже.

9.2. Описание шкал оценивания

Критерии оценивания ответов студентов на зачете с оценкой:

Оценка «отлично»	Студент показывает высокий уровень теоретических и практических знаний по изучаемой дисциплине, видит междисциплинарные связи. Умеет анализировать практические ситуации. Ответ построен логично. Материал излагается четко, ясно, аргументировано. Уместно используется информационный и иллюстративный материал.
Оценка «хорошо»	Студент показывает достаточный уровень теоретических и практических знаний, свободно оперирует категориальным аппаратом. Умеет анализировать практические ситуации, но допускает некоторые погрешности. Ответ построен логично, материал излагается грамотно.
Оценка «удовлетворительно»	Студент показывает знание основного лекционного и практического материала. В ответе не всегда присутствует логика изложения. Студент испытывает затруднения при приведении практических примеров.
Оценка «неудовлетворительно»	Студент показывает слабый уровень теоретических знаний, не может привести примеры из реальной практики. Неуверенно и логически непоследовательно излагает материал. Неправильно отвечает на дополнительные вопросы или затрудняется с ответом на них.

Описание шкал оценивания для различных заданий, выполняемых в рамках текущего контроля, представлено в методических материалах, определяющих процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

9.3. Критерии и процедуры оценивания результатов обучения по дисциплине, характеризующих этапы формирования компетенций

Компетенция ОПК-2 – способностью приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии

РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (модулю)	Уровень освоения компетенции	КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю) ШКАЛА оценивания					ПРОЦЕДУРЫ ОЦЕНИВАНИЯ
		1	2	3	4	5	
З2(ОПК-2) Знать: теорию баз данных, языки запросов и языки программирования для работы с базами данных	базовый	Не знает (не ориентируется)	Демонстрирует фрагментарные знания, допускает грубые ошибки	Демонстрирует общие, но не структурированные знания	Демонстрирует сформированные, но содержащие отдельные пробелы знания	Демонстрирует высокий уровень знаний	<i>Опросы, зачет с оценкой</i>
У3(ОПК-2) Уметь: разрабатывать структуру баз данных, выполнять реализацию хранения и логики обработки данных	базовый	Не умеет	Демонстрирует отдельные умения, допускает грубые ошибки	Демонстрирует общие умения, допускает существенные ошибки	Демонстрирует сформированные умения, допускает несущественные ошибки	Демонстрирует высокий уровень умений	<i>Выполнение практических заданий в течение семестра</i>

Компетенция ОПК-3 – способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей,

образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям

РЕЗУЛЬТАТ ОБУ- ЧЕНИЯ по дисциплине (мо- дулю)	Уровень освоения компетенции	КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю) ШКАЛА оценивания					ПРОЦЕДУРЫ ОЦЕНИВАНИЯ
		1	2	3	4	5	
31(ОПК-3) Знать: основные пара- дигмы и языки программирования	базовый	Не знает (не ориентируется)	Демонстрирует фрагментарные знания, допус- кает грубые ошибки	Демонстрирует об- щие, но не структу- рированные знания	Демонстрирует сформированные, но содержащие отдельные пробелы знания	Демонстрирует высокий уро- вень знаний	<i>Опросы, зачет с оценкой</i>
37 (ОПК-3) Знать: архитектуру и стандарты техно- логий платформы JavaEE	базовый	Не знает (не ориентируется)	Демонстрирует фрагментарные знания, допус- кает грубые ошибки	Демонстрирует об- щие, но не структу- рированные знания	Демонстрирует сформированные, но содержащие отдельные пробелы знания	Демонстрирует высокий уро- вень знаний	<i>Опросы, зачет с оценкой</i>
У2(ОПК-3) Уметь: выполнять проек- тирование и кон- струирование компонентов программного обеспечения в рамках поставлен- ного задания *)	базовый	Не умеет	Демонстрирует отдельные уме- ния, допускает грубые ошибки	Демонстрирует об- щие умения, допус- кает существенные ошибки	Демонстрирует сформированные умения, допускает несущественные ошибки	Демонстрирует высокий уро- вень умений	<i>Выполнение практических заданий в тече- ние семестра</i>
У3(ОПК-3) Уметь: выбирать алгорит- мы и выполнять их программную реа- лизацию для ре-	базовый	Не умеет	Демонстрирует отдельные уме- ния, допускает грубые ошибки	Демонстрирует об- щие умения, допус- кает существенные ошибки	Демонстрирует сформированные умения, допускает несущественные ошибки	Демонстрирует высокий уро- вень умений	<i>Выполнение практических заданий в тече- ние семестра</i>

РЕЗУЛЬТАТ ОБУ- ЧЕНИЯ по дисциплине (мо- дулю)	Уровень освоения компетенции	КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю) ШКАЛА оценивания					ПРОЦЕДУРЫ ОЦЕНИВАНИЯ
шения типовых задач предметной области							
У5(ОПК-3) Уметь: инсталлировать, испытывать и ис- пользовать про- граммные средства	базовый	Не умеет	Демонстрирует отдельные уме- ния, допускает грубые ошибки	Демонстрирует об- щие умения, допус- кает существенные ошибки	Демонстрирует сформированные умения, допускает несущественные ошибки	Демонстрирует высокий уро- вень умений	<i>Выполнение практических заданий в тече- ние семестра</i>
В1(ОПК-3) Вла- деть: современны- ми интегрирован- ными средами раз- работки про- граммного обеспе- чения	базовый	Не владеет навыками	Демонстрирует отдельные навыки, допус- кает грубые ошибки	Демонстрирует об- щие навыки, допус- кает существенные ошибки	Демонстрирует сформированные навыки, допускает несущественные ошибки	Демонстрирует высокий уро- вень владения навыками	<i>Выполнение практических заданий в тече- ние семестра</i>
В2(ОПК-3) Вла- деть: навыками тестирования, от- ладки и верифика- ции программ *)	базовый	Не владеет навыками	Демонстрирует отдельные навыки, допус- кает грубые ошибки	Демонстрирует об- щие навыки, допус- кает существенные ошибки	Демонстрирует сформированные навыки, допускает несущественные ошибки	Демонстрирует высокий уро- вень владения навыками	<i>Выполнение практических заданий в тече- ние семестра</i>

Компетенция ПК-1 – способностью собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям

РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (модулю)	Уровень освоения компетенции	КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю) ШКАЛА оценивания					ПРОЦЕДУРЫ ОЦЕНИВАНИЯ
		1	2	3	4	5	
34(ПК-1) Знать: Базовые знания о структуре, основных принципах и технологиях компьютерных сетей, необходимые для решения практических задач в области информационных систем и технологий	базовый	Не знает (не ориентируется)	Демонстрирует фрагментарные знания, допускает грубые ошибки	Демонстрирует общие, но не структурированные знания	Демонстрирует сформированные, но содержащие отдельные пробелы знания	Демонстрирует высокий уровень знаний	<i>Опросы, зачет с оценкой</i>

Компетенция ПК-15 – способностью проводить сбор и анализ научно-технической информации в сети Интернет и из других источников

РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (модулю)	Уровень освоения компетенции	КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю) ШКАЛА оценивания					ПРОЦЕДУРЫ ОЦЕНИВАНИЯ
		1	2	3	4	5	
В2(ПК-15): Владеть: навыками разработки информационных запросов к базам данным, инфор-	базовый	Не владеет навыками	Демонстрирует отдельные навыки, допускает грубые ошибки	Демонстрирует общие навыки, допускает существенные ошибки	Демонстрирует сформированные навыки, допускает несущественные ошибки	Демонстрирует высокий уровень владения навыками	<i>Выполнение практических заданий в течение семестра</i>

РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (мо- дулю)	Уровень освоения компетенции	КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю) ШКАЛА оценивания					ПРОЦЕДУРЫ ОЦЕНИВАНИЯ
мационно- поисковым и ре- шающим систе- мам в различных областях знания *)							

Компетенция ПК-16 – способностью к разработке эффективных программных решений и пониманию социальных и этических аспектов внедрения этих решений

РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (модулю)	Уровень освоения компетенции	КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю) ШКАЛА оценивания					ПРОЦЕДУРЫ ОЦЕНИВАНИЯ
		1	2	3	4	5	
УЗ(ПК-16) Уметь: оцени- вать временную и ресурсную трудоемкость программной реализации ал- горитма	базовый	Не умеет	Демонстрирует отдельные уме- ния, допускает грубые ошибки	Демонстрирует об- щие умения, допус- кает существенные ошибки	Демонстрирует с- формированные умения, допускает несущественные ошибки	Демонстрирует высокий уро- вень умений	<i>Выполнение практических заданий в тече- ние семестра</i>

Компетенция ПК-17 – способностью к разработке и применению алгоритмических и программных решений, математических методов обработки данных в области профессиональной деятельности

РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (модулю)	Уровень освоения компетенции	КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю) ШКАЛА оценивания					ПРОЦЕДУРЫ ОЦЕНИВАНИЯ
		1	2	3	4	5	
З1(ПК-17) Знать: формальные ме- тоды и инстру- менты разработ- ки программного продукта	базовый	Не знает (не ориентируется)	Демонстрирует фрагментарные знания, допуска- ет грубые ошиб- ки	Демонстрирует об- щие, но не структу- рированные знания	Демонстрирует сформированные, но содержащие отдельные пробелы знания	Демонстрирует высокий уро- вень знаний	<i>Опросы, зачет с оценкой</i>

9.4. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения, характеризующих этапы формирования компетенций и (или) для итогового контроля сформированности компетенции

Перечень вопросов к зачету с оценкой

1. Язык программирования Java. Характеристика. Особенности. История развития
2. Виртуальная машина Java. Платформа Java. Версии Java.
3. Реализация принципов ООП в JAVA.
4. Принципы объектно-ориентированного дизайна приложений.
5. Обработка исключений.
6. Коллекции. Средства пакета java.util.
7. Базовый ввод-вывод.
8. Модель обработки событий в JAVA.
9. Графический интерфейс пользователя.
10. Персистентность данных в объектно-ориентированных приложениях. Организация взаимодействия с базами данных.
11. Многопоточное программирование в Java.
12. Сетевое программирование. Средства пакета java.net.
13. Лямбда функции и функциональное программирование на Java.
14. Архитектура веб-приложений.
15. Архитектура приложения уровня предприятия.
16. Сервлеты. Servlet API.
17. Технология Java Server Pages (JSP).
18. Паттерны проектирования.
19. Паттерны «Фабрика», «Одиночка».
20. Паттерны «Модель-представление-контроллер».
21. Технология JavaServer Faces. Особенности, отличия от сервлетов и JSP. Преимущества и недостатки. Структура JSF-приложения. Использование JSP-страниц в JSF-приложениях.
22. Технология ORM.
23. Технология EJB (Enterprise JavaBeans).
24. Web Services.
25. Java для встроенных устройств. Мидлеты.

Материалы для текущего контроля

Формы текущего контроля: проверка выполнения заданий, контроль посещаемости, экспресс-опросы.

Вопросы к опросам:

1. Что такое виртуальная Java-машина?
2. Что такое платформа Java?
3. Что такое байт-код?
4. Какие результаты даст компиляция и запуск программы?
**class My{
public static void Main(String args[]){
System.out.println("Hello!");
}**
5. Что такое простой тип в JAVA?
6. Что такое явное и неявное приведение типов?
7. Назовите основные принципы ООП. Как они реализуются в Java?
8. Перечислите и поясните варианты использования ключевого слова **final**; **static**.
9. Каким образом реализуется наследование в JAVA?
Какие определения метода **sum** являются правильными?
1) **abstract class My{public int sum(int x,int y){return(x+y);}**
2) **class My{public final int sum(int x,int y){return(x+y);}**
3) **class My{public int sum(int x,int y);}**
4) **abstract class My{abstract public int sum(int x,int y);}**
5) **abstract class My{abstract public final int sum(int x,int y);}**
10. Какое описание области доступа в C# аналогично **protected** в JAVA?
11. Чем отличается абстрактный класс от интерфейса?
12. Является ли обязательным размещение кода **y=x/a** в блоке **try ... catch**? Почему?
13. В чем сходство и различие механизма обработки исключений в Java и C#?
14. Можно ли создать подклассы от **String** и **StringBuffer**?
15. Поясните фрагмент кода: **String s1, s2; if (s1==s2)) ...**
16. Какие варианты создания объектов допустимы:
a) **Byte b = new Byte(100);** b) **Short s = new Short (257);**
c) **Double d = new Double("abcd");** d) **Byte b = new Byte(128);**
e) **Integer i = new Integer(1500);** f) **Double d = new Double("2.115E-3");**
17. В чем различия коллекций и массивов?
18. Реализуется ли интерфейс **Comparable** в классе **Boolean**?
19. От чего зависит скорость доступа к произвольному элементу коллекции на основе класса **LinkedList**?
20. Как в java реализуется встроенная синхронизация потоков?
21. Какие возможности параллелизма реализованы в пакете **java.util.concurrent**?
22. В чем различия библиотек **AWT** и **Swing**?
23. Что такое менеджер компоновки?
24. В чем заключается модель делегирования событий?
25. Что такое класс адаптера?
26. Какие существуют виды потоков ввода/вывода?
27. Назовите основные родительские классы потоков ввода/вывода.
28. Что общего и чем отличаются следующие потоки: **InputStream**, **OutputStream**, **Reader**, **Writer**?
29. Какие есть режимы доступа к файлу?
30. Какой класс-надстройка позволяет читать данные из входного байтового потока в формате примитивных типов данных?
31. Какой класс-надстройка позволяет ускорить чтение/запись за счет использования буфера?

32. Расскажите об интерфейсе FilenameFilter?
33. Что такое сериализация?
34. Какой уровень эталонной модели ISO OSI обеспечивает
- передачу данных между любыми узлами сети с требуемым уровнем надежности?
 - доставку данных между любыми узлами в сети с произвольной топологией?
 - передачу кадра данных между любыми узлами сети?
 - средства управления диалогом между узлами и синхронизации в рамках процедуры обмена сообщениями?
 - набор сетевых сервисов, предоставляемых конечным пользователям и приложениям?
 - передачу битов в среде(коаксиальный кабель, оптоволокно и т.п.)?
35. Какой класс предназначен для открытия серверного сокета?
36. Какие классы реализуют протокол TCP?
37. Какой класс предназначен для открытия клиентского TCP-сокета?
38. Какие классы реализуют протокол UDP?
39. В каком классе реализованы методы для отправки и приема дейтаграмм?
40. В каком классе реализованы методы для упаковки и извлечения данных в/из дейтаграмм(у/ы)?
41. Что такое персистентность данных? Как она реализуется в объектных приложениях?
42. Объектно-реляционное отображение. Охарактеризуйте технологию ORM.
43. Технология сервлетов. Архитектура сервлета. Servlet API.
44. Технология JSP.
45. Основы технологии Enterprise JavaBeans.
46. Паттерны проектирования в Java.
47. Паттерны «Фабрика», «Одиночка».
48. Паттерны «Модель-представление-контроллер».
49. Web-фреймворки для Java.
50. Концепция веб-сервисов. Реализация в Java-технологиях.

Практические работы

№ раздела дисциплины	Наименование практических занятий
ПР-2.1	Объектно-ориентированное программирование на Java
ПР-2.2	Важнейшие объектные типы. Обработка исключений. Коллекции.
ПР-2.3	Персистентность данных в объектно-ориентированных приложениях. Организация взаимодействия с базами данных
ПР-2.4	Модель обработки событий в JAVA. Графический интерфейс пользователя.
ПР-2.5	Сетевое программирование. Средства пакета java.net.
ПР-2.6	Многопоточное программирование на JAVA
ПР-2.7	Технология сервлетов
ПР-2.8	Технология Java Server Pages.
ПР-2.9	Компонентное программирование. JavaBeans
ПР-2.10	Основы Java EE.
ПР-2.11	Технология EJB (Enterprise JavaBeans)
ПР-2.12	Использование JMS (Java Messaging System)
ПР-2.13	Паттерны проектирования. Паттерны «Фабрика», «Одиночка».
ПР-2.14	Паттерны «Модель-представление-контроллер».

№ раздела дисциплины	Наименование практических занятий
ПР-2.15	Аспектно-ориентированное программирование. Обзор фреймворка Spring.
ПР-2.16	Программирование встроенных устройств. Мидлеты.
ПР-2.17	Технология Web-сервисов

9.5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Для контроля знаний студентов по данной дисциплине необходимо проводить текущий и промежуточный контроль. Текущий контроль выполняется в виде приема допусков и защит практических работ, устного опроса. Промежуточный контроль проводится в виде зачета, на котором обсуждаются теоретические вопросы курса. Практическая часть зачитывается по результатам работ, выполненным в семестре, на основе балльно-рейтинговой системы. При рейтинговой системе все знания, умения и навыки, приобретаемые студентами в результате изучения дисциплины, оцениваются в баллах.

Оценка качества работы в рейтинговой системе является накопительной и используется для оценивания системной работы студентов в течение всего периода обучения. Максимально возможная сумма баллов по всем видам заданий приравнивается 100 %. При этом также учитывается посещаемость занятий – 0,5 балла за посещение одного занятия продолжительностью 1 академический час, при этом максимальный балл за посещаемость равен 18. Каждое практическое задание оценивается в зависимости от сложности от 2 до 30 баллов. Максимальный балл за каждый вид задания определяется преподавателем в зависимости от уровня сложности задания в начале семестра, проставляется в журнале успеваемости и доводится до сведения студентов.

Чтобы быть допущенным к зачету, студент обязан проделать и сдать все практические работы. Работы, пропущенные без уважительной причины, а также работы, не защищенные студентом в течение 2-х недель после проведения практической части, оцениваются преподавателем с понижающим коэффициентом 0,5.

Преподаватель может использовать «штрафы» в виде уменьшения набранных баллов за пропуск лекционных занятий, за нарушение сроков выполнения учебной работы, за систематический отказ отвечать на семинарских занятиях и т.д.

10 Ресурсное обеспечение

10.1. Перечень основной и дополнительной учебной литературы

10.1.1. Основная литература

1. Кубенский, А. А. Функциональное программирование : учебник и практикум для академического бакалавриата / А. А. Кубенский. — М. : Издательство Юрайт, 2017. — 348 с. — (Серия : Бакалавр. Академический курс). — ISBN 978-5-9916-9242-7.

<https://biblio-online.ru/book/658E3C89-AAD5-498B-8B34-A29E1750D810>

2. Основы языка программирования 1С 8.3: учеб. пособие / Э.Г. Дадян. — М.: Вузовский учебник: ИНФРА-М, 2017. — 132 с.

<http://znanium.com/catalog.php?bookinfo=750728>

3. Риз Р. Обработка естественного языка на Java: Исследование разных подходов к организации и извлечению полезной текстовой информации из неструктурированных данных с использованием Java / Риз Ричард; Пер.с англ. А.В.Снастина. - М.: ДМК Пресс, 2016. - 264с.: ил. - Предм.указ.:с.258. - ISBN 978-5-97060-331-

http://lib.uni-dubna.ru/biblweb/search/bibl.asp?doc_id=159986&full=yes

і. Дополнительная литература

1. **Свиштунов А.Н.** Построение распределенных систем на Java: Учебное пособие / Свиштунов Алексей Николаевич. - М.: БИНОМ. Лаборатория знаний: ИНТУИТ.РУ, 2015. - 279с.: ил. - (Основы информационных технологий). - ISBN 978-5-9963-0444-

http://lib.uni-dubna.ru/biblweb/search/bibl.asp?doc_id=160336&full=yes

2. **Йенер М.** Java EE. Паттерны проектирования для профессионалов / Йенер Мурат, Фидом Алекс. - СПб.: Питер, 2016. - 240с.: ил. - ISBN 9785496019453.

http://lib.uni-dubna.ru/biblweb/search/bibl.asp?doc_id=160484&full=yes

в. Периодические издания

1. Сетевое научное издание «Системный анализ в науке и образовании» (свидетельство о регистрации Эл № ФС77-51141 от 14 сентября 2012 г.).
2. Вестник Международного университета природы, общества и человека "Дубна" / гл. ред. Е.Н. Черемисина. — Дубна: Международный университет природы, общества и человека "Дубна".- (Системный анализ в современном обществе). — Журнал.
3. Программные продукты и системы: научно-практическое издание. / гл. ред. С.В. Емельянов. — Тверь: МНИИПУ. — Журнал. — Международное научно-практическое приложение к журналу "Проблемы теории и практики управления".
4. Теоретический и прикладной научно-технический журнал "Программная инженерия" /гл. ред. В.А. Васенин. — Москва: Новые технологии. — Журнал.- ISSN 2220-3397

с. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

10.3.1. Электронно-библиотечные системы и базы данных научного содержания

1. Электронно-библиотечная система «Университетская библиотека онлайн»: www.bibloclub.ru
2. Электронно-библиотечная система «ЮРАЙТ» <https://biblio-online.ru>
3. Научная электронная библиотека (НЭБ): <http://elibrary.ru/defaultx.asp>
4. Электронно-библиотечная система «Znanium» <http://znanium.com/>
5. БД российских журналов East View : <http://dlib.eastview.com>
6. Базы данных компании EBSCO Publishing: <http://search.ebscohost.com/>
7. БД российских научных журналов на Elibrary.ru (РУНЭБ): <http://elibrary.ru/defaultx.asp>.
8. <http://www.scopus.com/home.url>
9. Web of Science webofknowledge.com
10. БД российских научных журналов на Elibrary.ru (РУНЭБ): <http://elibrary.ru/defaultx.asp>.

10.3.2. Профессиональные ресурсы сети «Интернет»

1. Материалы портала сообщества разработчиков приложений, администраторов и архитекторов баз данных, использующих стандартизированные технологии и продукты Oracle. *Oracle Technology Network*. - <http://www.oracle.com/technetwork/ru/index.html>.
2. Материалы IT-портала Центра информационных технологий <http://www.citforum.ru>.
3. Материалы INTUIT.RU
4. Сообщество аналитиков: <http://www.uml2.ru/>.

1. Материалы IT-портала Центра информационных технологий <http://www.citforum.ru>

10.2. Перечень информационных технологий, используемых при осуществлении образовательного процесса, включая программное обеспечение, информационные справочные системы (при необходимости)

Используется лицензионное программное обеспечение:

- Интегрированная среда разработки NetBeans IDE 8.1.
- Пакет Java EE 7 SDK.

11. Описание материально-технической базы

Лекционные занятия

Специальные помещения: Учебная аудитория для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля

Аудитория укомплектована: специальной (учебной мебелью) на 80 посадочных мест, доской, персональным компьютером (без маркировки), проектором.

"Практические занятия/семинары

Специальные помещения: Компьютерный класс (учебная аудитория) для групповых и индивидуальных консультаций, организаций самостоятельной работы, в том числе, научно исследовательской. Аудитория укомплектована: специальной (учебной мебелью) на 12 посадочных мест, доской, компьютерами с мониторами.

Специализированный компьютерный класс подключенный к сети Интернет и к локальной сети университета, обеспечивающей доступ к программному обеспечению и ЛМС системы MOODLE для проведения семинарских занятий.

Обучающиеся из числа инвалидов и лиц с ограниченными возможностями здоровья могут использовать специализированное программное и материально-техническое обеспечение:

- обучающиеся с нарушениями опорно-двигательного аппарата при необходимости могут

использовать адаптивные технические средства: функцию «сенсорная клавиатура», «управление указателем мыши с клавиатуры», специально оборудованные джойстики, увеличенные выносные кнопки, клавиатуры с большими клавишами или накладки «Клавита»;
- обучающиеся с ограничениями по зрению могут прослушать доступный аудиоматериал или прочесть тексты, увеличив шрифт на экране монитора компьютера.

Рекомендуется использовать экранную лупу и другие визуальные вспомогательные средства, чтобы изменить шрифт текста, межстрочный интервал, синхронизацию с речью и т.д., программы экранного доступа (скринридеры для прочтения текстовой информации через синтезированную речь) и/или включить функцию «экранного диктора» на персональном компьютере с операционной системой Windows 7, 8, 10, Vista, XP.

Студенты с полным отсутствием зрения могут использовать тексты, напечатанные шрифтом Брайля, а для набора текста на компьютере – клавиатуры Брайля;

- обучающиеся с ограничениями по слуху могут воспользоваться индивидуальными техническими средствами (аппараты «Глобус», «Монолог», индивидуальными слуховыми аппаратами, компьютерной аудиогарнитурой, наушниками и др.) при прослушивании необходимой информации, а также услугами сурдопереводчика.

При необходимости обучающиеся с ограниченными возможностями здоровья и инвалиды обеспечиваются печатными и (или) электронными образовательными ресурсами (образовательная программа, учебники, учебные пособия и др.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла).

Полное описание аудиторий для реализации образовательной программы по данной дисциплине представлено в Справке материально-технического обеспечения текущего года обучения.

12. Язык преподавания

русский

Приложение №1

Тема: Синтаксис. Типы данных. Массивы.

1. Написать программу для решения квадратного уравнения с заданными коэффициентами.
2. Написать программу, для нахождения площади треугольника по трем сторонам. Длина стороны треугольника – случайное целое число большее 1, меньшее 50 (необходимо проверить, существует ли треугольник с полученными сторонами).
3. Написать программу, определяющую четность или нечетность целого числа (число генерируется случайным образом).
4. Даны два числовых массива X и Y с количеством элементов 5 и 7. Получите третий массив Z из 12 элементов, добавив элементы массива X в конец массива Y. Выведите на консоль массивы X, Y, Z.
5. Заполните массив десятью случайными числами, каждое из которых лежит в пределах от 50 до 200 и выведите значения элементов массива на консоль.
6. Даны два числовых массива X и Y с количеством элементов 10 и 4. Получите третий массив Z из 6 элементов, добавив в начало массива Y первый и последний элемент массива X. Выведите на консоль массивы X, Y, Z.
7. Даны два числовых массива X и Y с количеством элементов 15. Получите третий массив Z из 15 элементов, сложив соответствующие элементы массивов X и Y. Выведите на консоль массивы X, Y, Z.
8. Заполните двумерный массив (5, 5) числами от 1 до 25. Выведите значения элементов массива на консоль в виде матрицы.
9. Определите наименьший элемент в двумерном массиве (7, 7), элементами которого являются случайные числа от 1 до 1000.

Тема: Java SE 8.

1. Написать программу для нахождения площади треугольника по трем сторонам.

Формула Герона a, b, c – стороны треугольника, p — полупериметр треугольника:

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

$$p = \frac{a+b+c}{2}.$$

1.1. Структура программы:

```
class Triangle{
    // переменные:
    // длины сторон треугольника
    // площадь треугольника

    // методы
    // вычисляет и возвращает площадь треугольника
    // проверка существования треугольника
}

class DemoTriangle {
    public static void main(String [] args)
        // создать 3 объекта класса Triangle
        // инициализировать переменные объекта
```

```

// вычислить площадь
// вывести на консоль сообщение: «Площадь треугольника со сторонами ... равна ...».
    }
}

```

2. Разработать классы, имитирующие работу по обслуживанию банковского счета.

Счет создается с указанием владельца и присвоением номера счета.

Для выполнения операций первичного установления суммы, пополнения суммы и выдачи средств со счета разработать соответствующие методы.

Информация о счете должна выдаваться при печати соответствующего объекта класса (перекрыть метод `Object.toString()`).

Метод, который отвечает за выдачу средств со счета, должен уменьшить сумму на счете на величину, запрошенную клиентом. В случае если в результате этой операции остаток средств на счете составит меньше 1000 единиц, то операция не должна быть проведена, и вызывающий метод должно быть выброшено исключение `BalanceException` (создать класс для этой исключительной ситуации).

Протестировать работу разработанных классов.

3. Калькулятор для рациональных чисел

Для создания калькулятора подготовить класс `Rational`, определяющий рациональные числа, т.е. числа вида $\frac{n}{m}$, где n и m – целые числа.

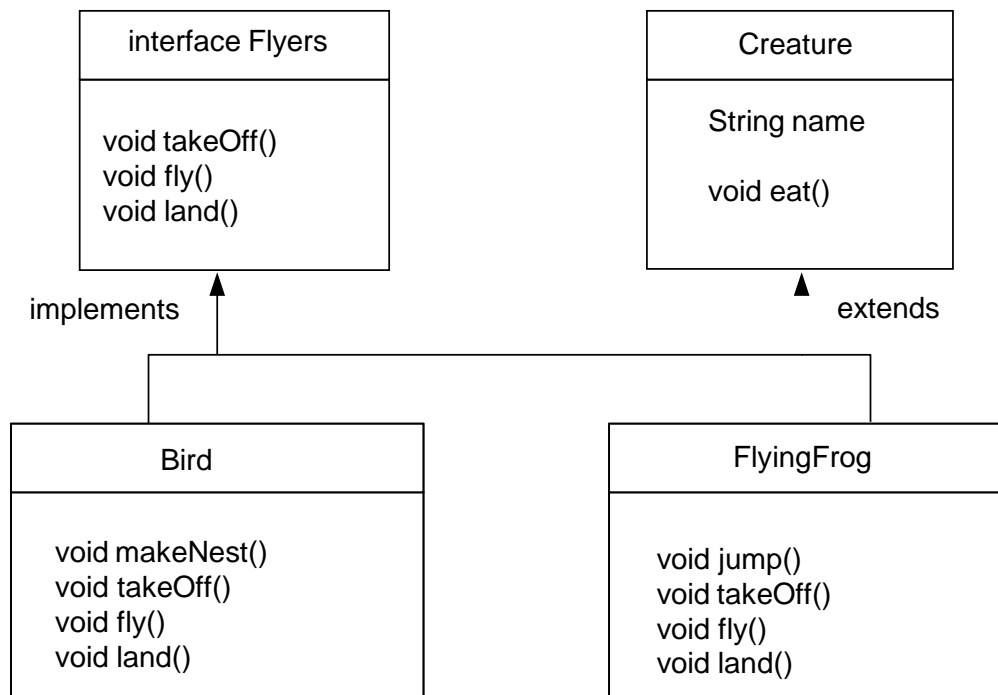
Предусмотреть различные конструкторы: `Rational()`, `Rational (int n)`, `Rational(int n, int m)`. В последнем конструкторе предусмотреть выброс исключительной ситуации `ArithmeticException` с сообщением «Division by zero: object has not been created» в вызывающий метод в случае, если параметр $m=0$.

Обеспечить перекрытие методов `Object` : `toString()`, `equals(Object obj)`.

Реализовать методы сложения, умножения, деления и вычитания для двух аргументов, являющихся рациональными числами. Реализовать эти же методы в случае, когда сложение, умножение, деление и вычитание выполняются с объектом класса и рациональным числом – аргументом методов.

Протестировать работу созданного класса.

4. Создать классы в одном пакете по приведенной схеме:



Методы в классах реализовать в виде печати соответствующих сообщений. Например, для метода `eat()` можно напечатать "I am hungry and will garbage down almost anything".

Создать класс `SelectFlyers` с методом `main`, в котором создать массив из 7 разных созданий (зверей, птиц, земноводных). Каждый должен объявить свое имя, заявить умеет ли он летать, и если умеет, то должен взлететь (`takeOff()`), полетать (`fly()`), приземлиться (`land()`) и его надо накормить (`eat()`).

Проверку *умеет ли создание летать* выполнить двумя способами:

- с помощью оператора `instanceof`
- с помощью обработки исключения `ClassCastException`.

5. Создать и протестировать классы, имитирующие игру «Поле чудес».

6. Разработать классы, для работы с переводом текста на заданный пользователем язык.

Класс соответствующего переводчика должен динамически загружаться по заданному имени во время работы приложения. (Использовать `Class.forName(String className)` и метод `newInstance()` на переменной типа `Class`).

7. Написать код, моделирующий колоду карт и процесс тасования колоды.

8. Разработать и протестировать классы для заполнения телефонного справочника и извлечения персональной информации о телефонном номере. Для хранения справочника подобрать соответствующую коллекцию.

9. Разработать и протестировать классы, имитирующие розыгрыш путевки на Международный студенческий форум среди студентов вашей группы. О студенте требуется информация об участии в других студенческих мероприятиях (конференциях, круглых столах и т.д.).

Право поехать у того, кто участвовал в наибольшем количестве мероприятий. Путевка – од-

на. Если у нескольких человек одинаковые шансы, то среди них организовать лотерею, перетасовав список претендентов случайным образом. Первый в списке получает путевку.

10. Реализовать приложение, которое выводит на консоль содержимое директории.

11. Разработать и протестировать классы, обеспечивающие копирование файлов. Использовать классы символьных и байтовых потоков, указать ограничения в применении.

12. Разработать и протестировать классы, реализующие хранение объектов Java в файле и восстановление объектов из этого файла в приложении.

13. Выдать на консоль содержимое страницы Web ресурса, определяемого некоторым URL .

14. Разработать и протестировать классы, имитирующие работу «эхо-сервера» с несколькими клиентами (ответ сервера повторяет информацию, принятую от клиента).

15. Реализовать упрощенную систему “СтудентИНФО”. Ключевые функциональные требования:

1. хранить и редактировать информацию о студентах.
2. хранить и предоставлять информацию о группах.
3. генерировать отчеты по данным о студентах и группах.

Примерная структура программы:

1) Управляющая Система:

- Список студентов
- Список групп
- Добавить студента
- Удалить студента
- Редактировать данные о студенте
- Перевести всех студентов из одной группы в другую
- Удалить всех студентов из группы
- Отчеты (список группы, список юношей и т.д.)

2) Студент:

- ИД студента
- Фамилия
- Имя
- Отчество
- Дата рождения
- Пол
- ИД Группы (ссылка на группу в которой учится студент)
- Год обучения в группе

3) Группа:

- ИД группы
- Название
- Фамилия куратора
- Название направления

- ✓ Для классов Студент и Группа переопределить метод **String toString()** , возвращающий описание объекта в виде строки.
- ✓ Для сравнения студентов реализовать интерфейс Comparable (метод — `public int compareTo(Object obj)`)

Метод возвращает целое число больше 0, если объект obj меньше, 0 — если они равны, и отрицательное число — если больше.

- ✓ Класс Collator позволяет сравнивать строки с учетом языка (Locale) а также устанавливать насколько точно должно быть сравнение — с учетом регистра или без.

```
Collator c = Collator.getInstance(new Locale("ru"));
System.out.println(c.compare("Ангелина", "Анна"));
System.out.println("Иванов".compareTo("Петров"));
```

✓ Класс DateFormat - класс для преобразования даты
...DateFormat.getDateInstance(DateFormat.SHORT).format(dateOfBirth)...

Для класса *Управляющая Система* рекомендуется воспользоваться шаблоном проектирования Singleton. Этот шаблон показывает, как можно использовать только один экземпляр объекта. Реализация делается следующим образом:

- конструктор объявляется private (теперь напрямую создать объект нельзя)
- описывается одна статическая переменная такого же класса (т.е. для всех объектов класса будет только одна такая переменная)
- создается метод (обычно его называют getInstance), который возвращает ссылку на единственный объект.

```
class ManagementSystem:
```

```
// закрытый конструктор
private ManagementSystem() {
    loadGroups();
    loadStudents();
}

// метод getInstance - проверяет, инициализирована ли статическая
// переменная и возвращает ее
public static synchronized ManagementSystem getInstance() {
    if (instance == null) {
        instance = new ManagementSystem();
    }
    return instance;
}
```

В методе main:

```
// перенаправить стандартный вывод в файл
// если будут проблемы с кодировкой
try {
    System.setOut(new PrintStream("out.txt"));
} catch (FileNotFoundException ex) {
    ex.printStackTrace();
    return;
}
```

```
//работа с календарем
Calendar c = Calendar.getInstance();
c.set(1991, 8, 31);
```

Экземпляры классов Group и Student организовать в коллекции.

16. Реализовать простейший графический интерфейс к системе “СтудентИНФО”, позволяющий отображать:

1. Список групп
2. Список студентов для группы

Для данного оконного приложения используется:

- ✓ класс JFrame пакета java.swing.
- ✓ контейнер: стандартная панель — класс JPanel. Доступ к ней осуществляется через вызов метода getContentPane().
- ✓ интерфейс слушателя ActionListener (метод actionPerformed())
- ✓ менеджеры компоновки **BorderLayout** и **FlowLayout**

Внешний вид формы:

Форма разделена на две части — верхнюю и нижнюю. На верхней части - компонент для ввода года в виде числового слайдера (спинера).

Нижняя делится на две части. На левой будет располагаться список групп, а на — правой список студентов (рис.1.).

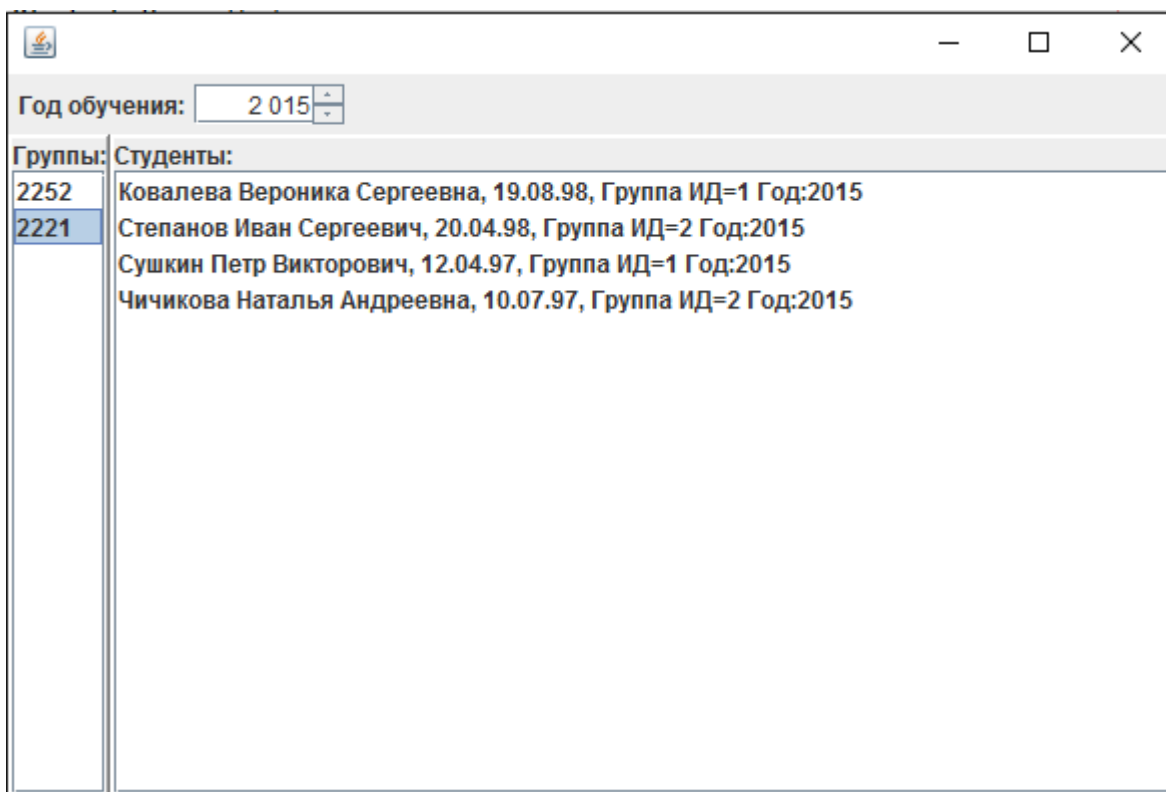


Рис.1. Примерный внешний вид формы

Структура программы:

- ✓ Классы Student и Group остаются без изменений
 - ✓ В классе ManagementSystem удаляется метод main
 - ✓ Добавляется класс формы StudentsFrame
- Запуск формы желательно производить в отдельном потоке:

```
public static void main(String args[]) {  
    SwingUtilities.invokeLater(new Runnable() {  
        public void run() {  
            StudentsFrame sf = new StudentsFrame();  
            sf.setDefaultCloseOperation(EXIT_ON_CLOSE);  
            sf.setVisible(true);  
        }  
    });  
}
```

```

    }
  });
}

```

- ✓ Классы Student, Group, ManagementSystem размещаются в пакете logic, класс StudentsFrame – в пакете frame (рис.2).
!!!Это не обязательно, но важно правильно указывать расположение используемых классов в исходниках проекта.

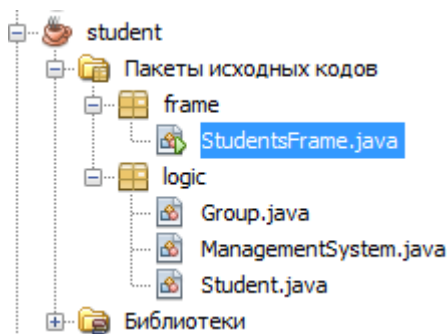
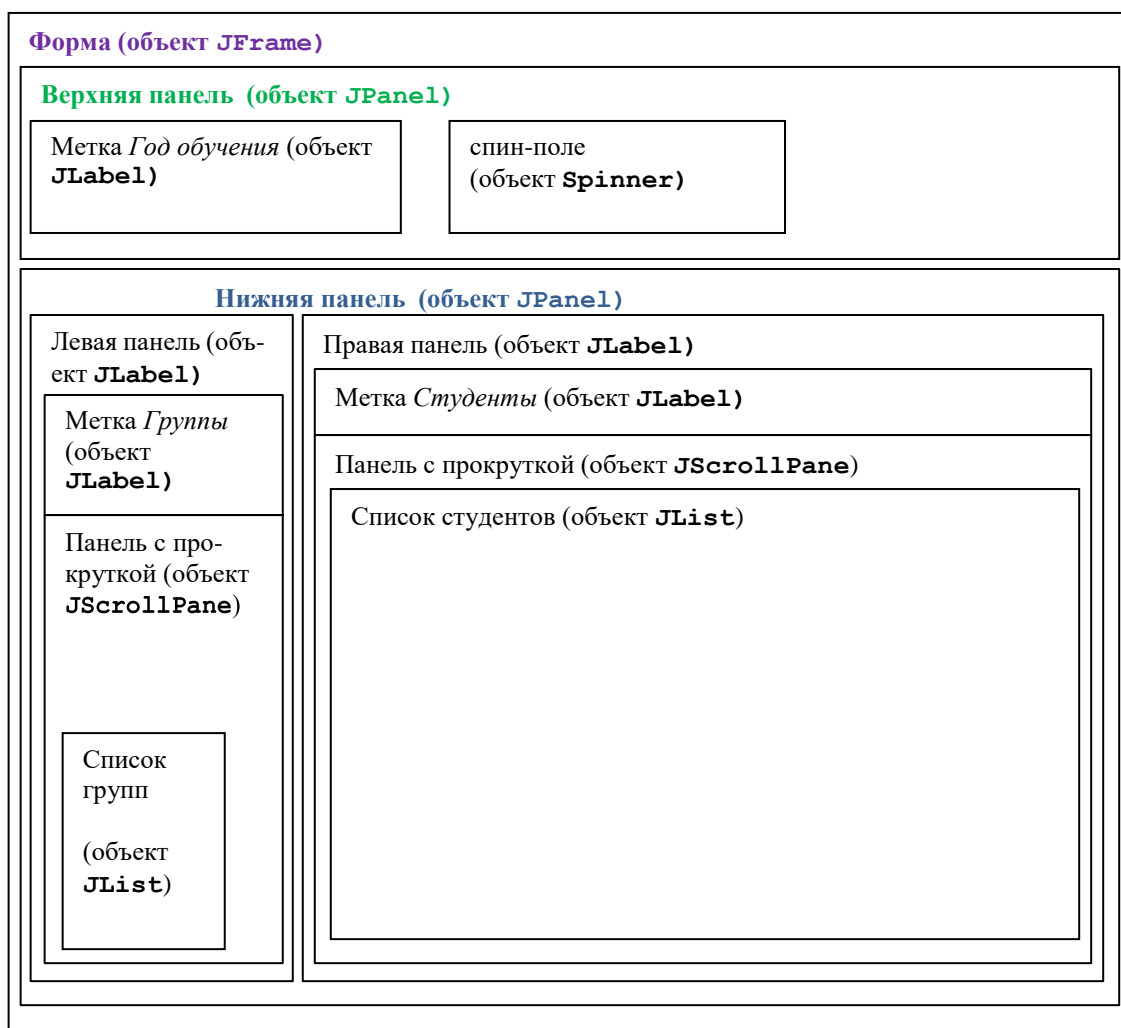


Рис.2. Структура каталога для описанного расположения классов

Структура формы:



18. Реализовать в системе “СтудентИНФО” хранение информации в БД MySQL.

1. Создать под управлением СУБД MySQL таблицы groups и students:

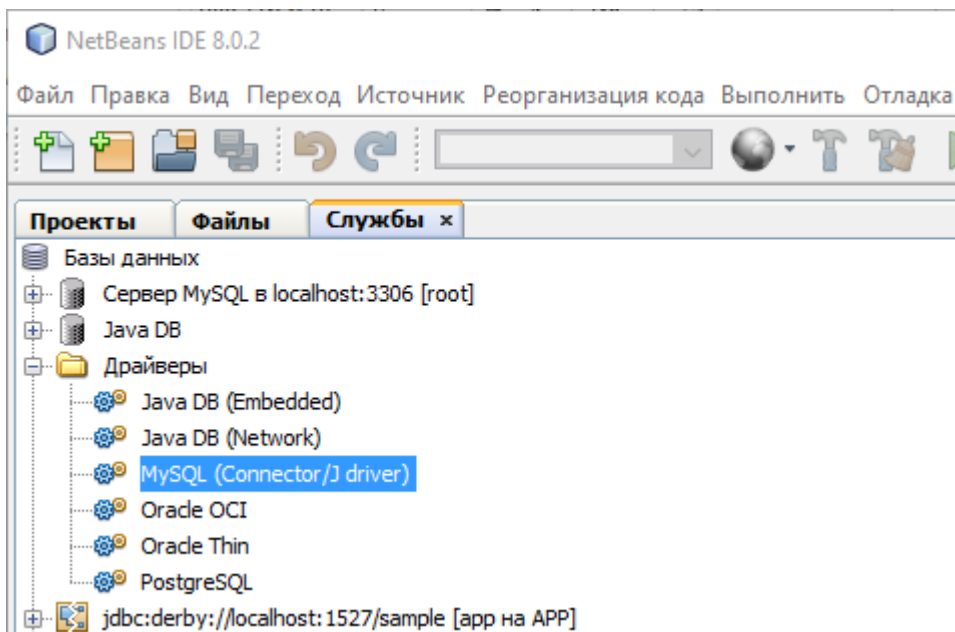
```
groups (group_id int primary key,  
        groupName varchar(255) not null,  
        curator varchar(255) not null,  
        speciality varchar(255) not null);
```

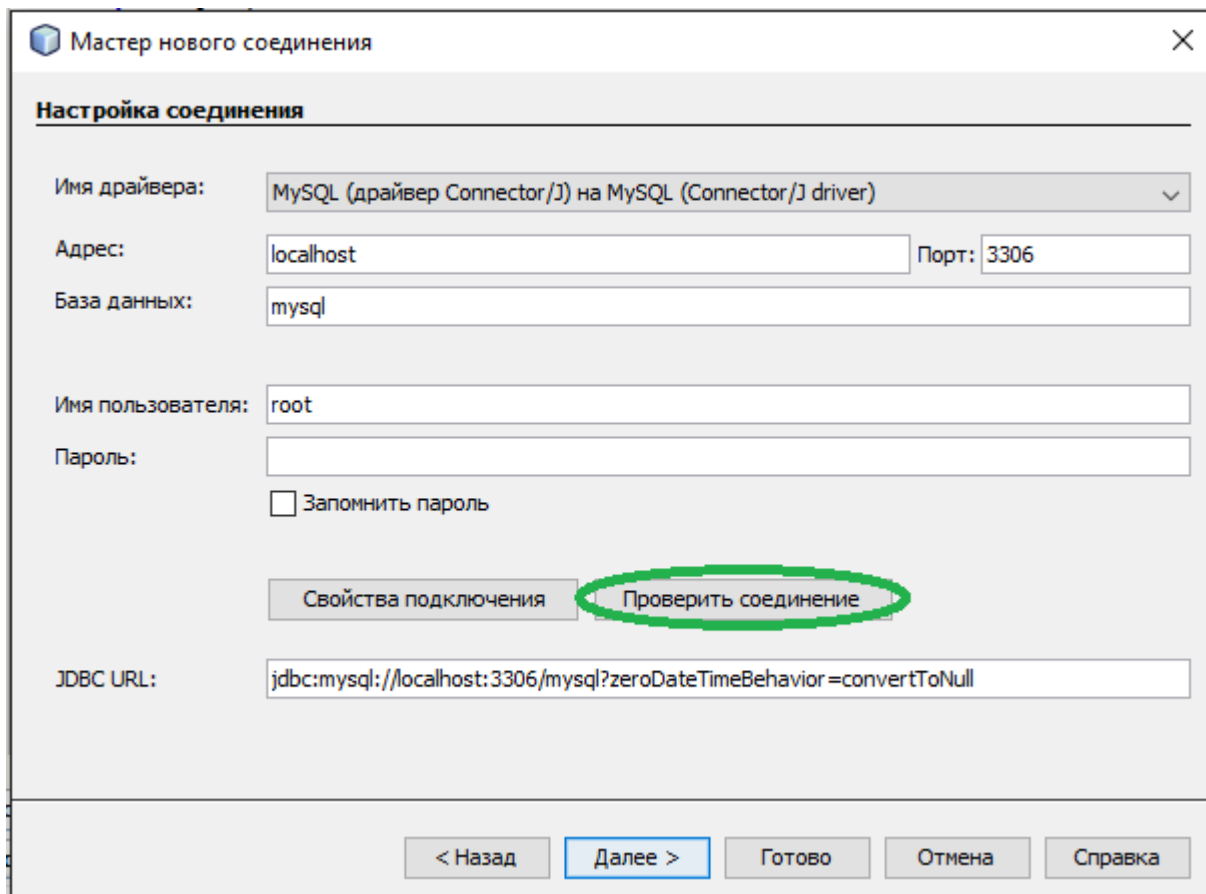
```
students(student_id int primary key,  
          firstName varchar(255) not null,  
          surName varchar(255) not null,  
          patronymic varchar(255) not null,  
          dateOfBirth date not null,  
          sex char(1),  
          group_id int not null,  
          educationYear int not null);
```

Один из возможных вариантов:

Установить локальный сервер – например, пакет хамpplite (находится на сетевом диске Programms/Network.Servers/HTTP)

2. Сделать первоначальное наполнение БД
3. Проверить соединение с БД через вкладку Службы, щелкнув по MySQL (Connector/J driver) правой кнопкой мыши и указав параметры соединения в появившемся окне:





4. Для работы с БД понадобятся:

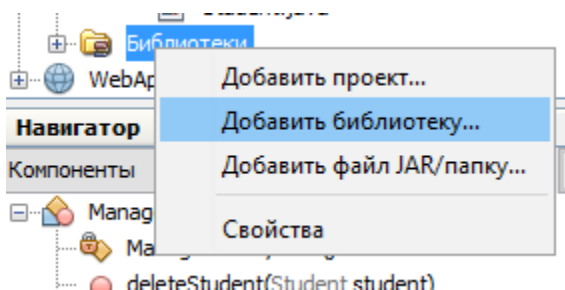
✓ классы из пакета **java.sql**:

```
java.sql.Connection;
java.sql.DriverManager;
java.sql.ResultSet;
java.sql.SQLException;
java.sql.Statement;
```

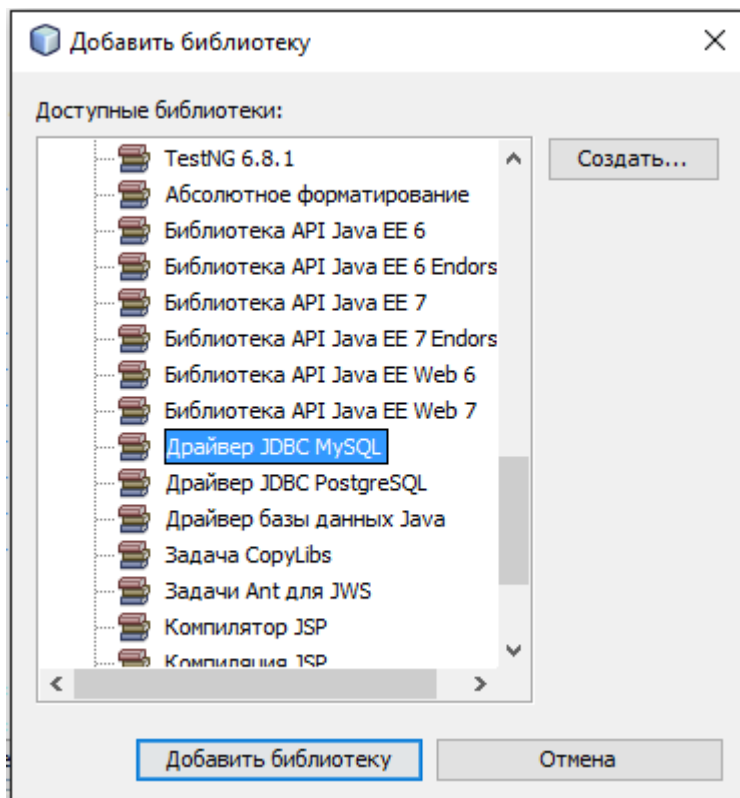
✓ Драйвер для **MySQL JDBC (Java Database Connectivity)**

Технология JDBC позволяет единообразно получать доступ к различным базам данных.

Для того, чтобы класс драйвера был доступен из кода, его нужно подключить в свой проект, щёлкнув правой кнопкой мыши <Библиотеки>, выбрать <Добавить библиотеку>:



а затем <Драйвер JDBC MySQL>:



Пример:

```
....
Connection con = null;
Statement stmt = null;
ResultSet rs = null;
//Операции с БД вызывают исключения
    try {
//загрузка класса драйвера
        Class.forName("com.mysql.jdbc.Driver");
        String url = "jdbc:mysql://localhost:3306/students";
//Установка соединения
        con = DriverManager.getConnection(url, "root", "");
//Создание запроса для отправки на сервер
        stmt = con.createStatement();
//Выполнение запроса и размещение результатов в результирующий набор
        //записей
        rs = stmt.executeQuery("SELECT * FROM students");
//Проход по возвращенному набору и вывод результатов
        while (rs.next()) {
            String str = rs.getString(1) + ":" + rs.getString(2);
            System.out.println(str);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        // освобождение открытых ресурсов
        try {
            if (rs != null) {
                rs.close();
            }
            if (stmt != null) {
                stmt.close();
            }
            if (con != null) {
```

```

        con.close();
    }
} catch (SQLException ex) {
    ex.printStackTrace();
    System.err.println("Error: " + ex.getMessage());
}

```

...

19. Усовершенствовать ранее созданный графический интерфейс к системе “СтудентИНФО” следующим образом:

- ✓ В верхней части экрана — компонент для ввода года в виде числового слайдера (спиннера), реагирующий на обновление списка студентов.
- ✓ Список групп, реагирующий на изменение выделенной группы.
- ✓ Список студентов: 1) отображает студентов из выделенной группы, 2) студенты отображаются в виде таблицы.
- ✓ Редактирование студентов в нижней части таблицы со списком студентов.
- ✓ Редактирование списка групп — для удаления всех студентов из группы и для перевода всех студентов в другую группу.
- ✓ При добавлении нового студента или редактировании существующего будем выводиться диалоговое окно. При добавлении студента диалоговое окно не должно закрываться после добавления студента — пользователь может сразу начать вводить данные другого студента. При этом значения для полей ГРУППА и ГОД должны оставаться такими, как были, а все остальные поля очищаются. Группы выбираются из списка.
- ✓ В меню добавить пункт «Отчеты» — список всех студентов.

Тема: Разработка веб-приложений и веб-сервисов на платформе Java EE.

20. Реализовать традиционный сервлет “Hello, World!!!”.

21. Разработка параметризованного сервлета.

22. Реализовать сервлет, извлекающий данные о студентах из базы данных.

23. Реализовать JSP для отображения афоризма из коллекции, выбираемого случайным образом.

24. Реализовать СтудентИНФО в виде web-приложения.

25. Реализовать “СтудентИНФО” в виде Enterprise приложения.

26. Разработать web-сервис, реализующий вывод курса валют.