

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Емельянов Сергей Геннадьевич
Должность: ректор
Дата подписания: 16.12.2021 20:54:36
Уникальный программный ключ:
9ba7d3e34c012eba476ffd2d064c12761995ba730a2b74d163c0ce538076

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра биомедицинской инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Доктионова
« 1 » _____ 2018 г.


МЕТА-АНАЛИЗ В МЕДИЦИНСКИХ И ЭКОЛОГИЧЕСКИХ СИСТЕМАХ

Методические рекомендации по организации и выполнению практических работ для аспирантов направления подготовки 09.06.01 и 12.06.01

УДК 004.93:61

Составитель: С.А. Филист.

Рецензент

Доктор технических наук, профессор А.Ф. Рыбочкин

Мета-анализ в медицинских и экологических системах:
методические рекомендации по организации и выполнению
практических работ / Юго-Зап. гос. ун-т; сост.: С.А. Филист. - Курск,
2018. - 55 с.

Методические указания по структуре, содержанию и стилю изложения материала соответствуют методическим и научным требованиям, предъявляемым к учебным и методическим пособиям.

Предназначены для аспирантов направления подготовки 09.06.01 «Информатика и вычислительная техника (Системный анализ, управление и обработка информации (технические и медицинские системы)) и 12.06.01 «Фотоника, приборостроение, оптические и биотехнические системы и технологии (Приборы, системы и изделия медицинского назначения)»

Текст печатается в авторской редакции

Подписано в печать 1.03.18. Формат 60x84 1/16.

Усл.печ.л. 3,3. Уч.-изд.л. 3,1 Тираж 100 экз. Заказ: 1428. Бесплатно.

Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Практическая работа №1

«Базовые принципы мета-анализа в методологии доказательной медицины»

1.1 Основные методы мета-анализа

Выбор метода анализа определяется типом анализируемых данных (бинарные или непрерывные) и типом модели (фиксированных эффектов, случайных эффектов).

Бинарные данные обычно анализируются путем вычисления отношения шансов (ОШ), относительного риска (ОР) или разности рисков в сопоставляемых выборках. Все перечисленные показатели характеризуют эффект вмешательств. Представление бинарных данных в виде ОШ удобно использовать при статистическом анализе, но этот показатель достаточно трудно интерпретировать клинически. Непрерывными данными обычно являются диапазоны значений изучаемых признаков или нестандартизованная разница взвешенных средних в группах сравнения, если исходы оценивались во всех исследованиях одинаковым образом. Если же исходы оценивались по-разному (например, по разным шкалам), то используется стандартизованная разница средних (так называемая величина эффекта) в сравниваемых группах [78].

Одним из первых этапов мета-анализа является оценка гетерогенности (статистической неоднородности) результатов эффекта вмешательства в разных исследованиях [67].

Для оценки гетерогенности часто используют критерии χ^2 с нулевой гипотезой о равном эффекте во всех исследованиях и с уровнем значимости 0,1 для повышения статистической мощности (чувствительности) теста [81].

Источниками гетерогенности результатов разных исследований принято считать дисперсию внутри исследований (обусловленную случайными отклонениями результатов разных исследований от единого истинного

фиксированного значения эффекта), а также дисперсию между исследованиями (обусловленную различиями между изучаемыми выборками по характеристикам больных, заболеваний, вмешательств, приводящими к несколько разным значениям эффекта — случайными эффектами).

Если предполагается, что дисперсия между исследованиями близка к нулю, то каждому из исследований приписывается вес, величина которого обратно пропорциональна дисперсии результата данного исследования.

Дисперсия внутри исследований в свою очередь определяется как:

$$\sigma = \frac{\sum (X - \mu)^2}{N}, \quad (1.1)$$

где μ - среднее внутри исследований [80]. При нулевой дисперсии между исследованиями можно использовать модель фиксированных (постоянных) эффектов. В этом случае предполагается, что изучаемое вмешательство во всех исследованиях имеет одну и ту же эффективность, а выявляемые различия между исследованиями обусловлены только дисперсией внутри исследований. В этой модели пользуются методом Мантела-Ханзела.

Технология стандартного мета-анализа предполагает проведение мета-анализа предполагает выполнение следующей последовательности действий.

1. Вычисляются доли проявления эффекта в экспериментальной ($PE1i$) и контрольной ($PE2i$) группах – отношения числа случаев, в которых эффект наблюдался, к объему групп (число экспериментов в группах):

$$PE_{1i} = a_i / (a_i + b_i) 100\%, \quad PE_{2i} = c_i / (c_i + d_i) 100\%, \quad (1.2)$$

где $i = \{1..k\}$ – номер исследования; k – число объединяемых исследований.

2. Вычисляется разность долей эффектов в экспериментальной и контрольной группах:

$$\Delta PE_i = PE_{1i} - PE_{2i}. \quad (1.3)$$

3. Вычисляется стандартное отклонение разности эффектов:

$$SD_i = \sqrt{(PE_{1i}(100 - PE_{1i})/n_{i3}) + (PE_{2i}(100 - PE_{2i})/n_{ik})}, \quad (1.4)$$

где n_{i3} – объем экспериментальной группы; n_{ik} – объем контрольной группы.

4. Вычисляются границы 95% доверительного интервала (CI_i) [69]:

$$CI_i = \Delta PE_i \pm 1,96SD_i, \quad (1.5)$$

где «-» соответствует левой (нижней), а «+» – правой (верхней) границам доверительного интервала.

5. Вычисляется достоверность проявления эффекта по одному исследованию (достоверность исследования) по критерию хи-квадрат (χ^2). Формула для вычисления величины критерия выбирается в зависимости от исходных данных [93, 94, 102]. Если объем анализируемой выборки меньше 20 или объем анализируемой выборки находится в интервале между 20 и 40, а наименьшее из значений (a_i , b_i , c_i , d_i) не превосходит 5, то для определения уровня значимости нулевой гипотезы следует использовать формулу Фишера [98, 99]:

$$p_i = ((a_i + b_i)!(c_i + d_i)!(a_i + c_i)!(b_i + d_i)!)/(N_i!a_i!b_i!c_i!d_i!), \quad (1.6)$$

где N_i – объем объединенной (экспериментальной и контрольной) группы в i -м исследовании, ! – знак факториала.

В других случаях следует вычислять значение критерия χ^2 [94, 98]

$$\chi_i^2 = ((a_i \cdot d_i - b_i \cdot c_i)^2 \cdot N_i) / ((a_i + b_i) \cdot (c_i + d_i) \cdot (a_i + c_i) \cdot (b_i + d_i)) \quad (1.7)$$

и по полученному значению критерия при числе степеней свободы $df = 1$ с помощью таблиц с 2-распределения определять уровень значимости нулевой гипотезы – p_i .

6. Вычисляется достоверность проявления эффекта по всем исследованиям (средняя достоверность). Исследования объединяются по величинам Z -критериев. По значению p_i , используя таблицы нормального распределения, необходимо определить значение Z -критерия для каждого исследования (Z_i). Вычислить обобщенное значение Z -критерия $Z_{overall}$:

$$Z_{overall} = \sum_{i=1}^k Z_i / \sqrt{k}. \quad (1.8)$$

По полученному значению $Z_{overall}$, используя статистические таблицы нормального распределения, определить значение средней достоверности проявления эффекта ($p_{overall}$).

7. Определяется однородность исследований – уровень значимости гипотезы о согласованности данных по критерию хи-квадрат, которая позволяет сделать заключение о возможности (корректности) статистического обобщения результатов исследований. Для этого выполняются следующие шаги. Вычисляются среднее значение Z -критерия по всем исследованиям и значение критерия согласия (однородности) хи-квадрат:

$$\bar{Z} = \sum_{i=1}^k Z_i / k, \quad X^2 = \sum_{i=1}^k (Z_i - \bar{Z})^2. \quad (1.9)$$

По полученному значению χ^2 при числе степеней свободы $df = k-1$, используя статистические таблицы, следует определить уровень значимости p и проверить гипотезу об однородности результатов, полученных в экспериментальной и контрольной группах, сравнив полученное значение p с априорно выбранным критическим значением (как правило, 0,05 или 0,01).

8. Вычисляется эффективный размер выборки (r) – оценка несоответствия результатов в экспериментальной и контрольной группах:

$$r = Z_{overall} / \sqrt{\sum_{i=1}^k N_i}. \quad (1.10)$$

9. Вычисляется объем ложной выборки (Nfs) – потребное число исследований, дополнительное включение которых в обобщенную выборку приводило бы к превышению полученного в результате объединения уровня значимости гипотезы об отсутствии эффекта:

$$Nfs = \left(\sum_{i=1}^k Z_i / 1,645 \right)^2 - k. \quad (1.11)$$

Перейти к способу описания данных, заданному таблицей кросс-табуляции, можно, если исследователь привел такие данные, которые позволяют получить величины ai , bi , ci , di решением системы линейных алгебраических уравнений.

Если данные, указанные исследователем, не позволяют определить величины ai , bi , ci , di , то мета-анализ может быть проведен частично [69, 97]. Например, если для исследования задано только значение уровня значимости различий эффектов в экспериментальной и контрольной группах – pi (вероятность отвержения нулевой гипотезы об отсутствии различий эффектов, если она верна), то мета-анализ может быть проведен с шага 6.

Описанная процедура определяет порядок проведения *стандартного мета-анализа*, целью которого является мета-анализ каждого отдельного исследования и исследования, полученного путем их объединения. Для определения степени влияния каждого отдельного исследования на окончательный результат выполняется мета-анализ со скользящим протоколом и кумулятивный мета-анализ.

Технология мета-анализа со скользящим протоколом предполагает последовательное поочередное исключение каждого отдельного исследования из анализа с последующим его возвращением в выборку. В начале процедуры анализируется объединенное исследование. На первом шаге анализа исключается первое исследование, на втором шаге первое исследование «возвращается» в выборку, второе исключается. Процедура завершается после возвращения в выборку последнего исследования (рисунок 1.1).

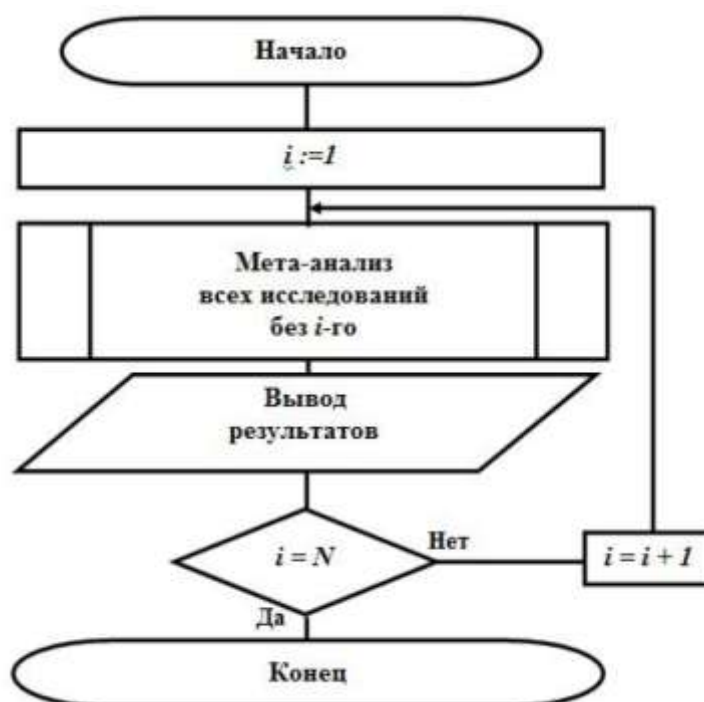


Рисунок 1.1 - Алгоритм мета-анализа со скользящим протоколом

Процедура, предложенная Мантелом и Ханзелом, позволяет оценивать предположительно типичное значение отношения шансов и проверять значимость общей степени связи.

В таблице 1.1 представлены пропорции пациентов в двух городах промышленного кластера, которым был поставлен диагноз шизофрения.

Оценкой Мантела-Ханзела отношения шансов является:

$$\overline{O_{MN}} = \frac{\sum_{i=1}^g \frac{n_{i1}n_{i2}}{n_i} p_{i1}(1-p_{i2})}{\sum_{i=1}^g \frac{n_{i1}n_{i2}}{n_i} p_{i2}(1-p_{i1})}, \quad (1.12)$$

где $\overline{O_{MN}}$ – взвешенное среднее отдельных отношений шансов по группам. Критерий хи-квадрат Мантела-Ханзела проверки значимости общей меры связи основан на взвешенном среднем g разностей между пропорциями.

Таблица 1.1 – Пропорции пациентов больных шизофренией в двух городах промышленного кластера

Исследование	1-й город		2-й город	
	n_{i1}	p_{i1}	n_{i2}	p_{i2}
i=1(возраст 20-34 лет)	105	0,771	105	0,324
i=2(возраст 20-59 лет)	192	0,615	174	0,397
i=3(возраст 35-59 лет)	145	0,566	145	0,359

$$\bar{d} = \frac{\sum_{i=1}^g \frac{n_{i1}n_{i2}}{n_i} (p_{i1} - p_{i2})}{\sum_{i=1}^g \frac{n_{i1}n_{i2}}{n_i}}. \quad (1.13)$$

Статистика хи-квадрат Мантела-Ханзела задается выражением:

$$X^2_{MN} = \frac{(|\sum_{i=1}^g \frac{n_{i1}n_{i2}}{n_i} (p_{i1} - p_{i2})| - 0,5)^2}{\sum_{i=1}^g \frac{n_{i1}n_{i2}}{n_i - 1} \overline{p_i q_i}} \quad (1.14)$$

с 1 степенью свободы.

Для того, чтобы статистика имела распределение хи-квадрат с 1 степенью свободы, каждая из четырех сумм ожидаемых частот:

$$\sum_{i=1}^g n_{i1} \bar{p}_i, \quad \sum_{i=1}^g n_{i2} \bar{p}_i, \quad \sum_{i=1}^g n_{i1} \bar{q}_i, \quad \sum_{i=1}^g n_{i2} \bar{q}_i \quad (1.15)$$

должна отличаться не менее чем на 5 как от своего минимума, так и от своего максимума.

Значит, чтобы с уверенностью пользоваться для статистики распределением хи-квадрат с 1 степенью свободы, вовсе не обязательно иметь большие маргинальные частоты. Число наблюдений в таблице может быть даже равно двум, как в случае связанных пар. Единственное, что нужно при этом – достаточно большое число таблиц, чтобы каждая сумма ожидаемых частот была велика.

Модель случайных эффектов предполагает, что эффективность изучаемого вмешательства в разных исследованиях может быть разной.

Данная модель учитывает дисперсию не только внутри одного исследования, но и между разными исследованиями. В этом случае суммируются дисперсии внутри исследований и дисперсия между исследованиями. Целью мета-анализа непрерывных данных обычно является представление точечных и интервальных (95% ДИ) оценок обобщенного эффекта вмешательства [70].

Существует также ряд других подходов к выполнению мета-анализа: байесовский мета-анализ, кумулятивный мета-анализ, многофакторный мета-анализ, мета-анализ выживаемости.

Байесовский мета-анализ позволяет рассчитать априорные вероятности эффективности вмешательства с учетом косвенных данных. Такой подход особенно эффективен при малом числе анализируемых исследований. Он обеспечивает более точную оценку эффективности вмешательства в модели случайных эффектов за счет объяснения дисперсии между разными исследованиями [72].

Кумулятивный мета-анализ — частный случай байесовского мета-анализа — пошаговая процедура включения результатов исследований в мета-анализ по

одному в соответствии с каким-либо принципом (в хронологической последовательности, по мере убывания методологического качества исследования и т.д.). Он позволяет рассчитывать априорные и апостериорные вероятности в итерационном режиме по мере включения исследований в анализ [90].

Кумулятивный мета-анализ (рисунок 1.2) заключается в последовательном добавлении отдельных исследований в анализируемую группу исследований (выборку).

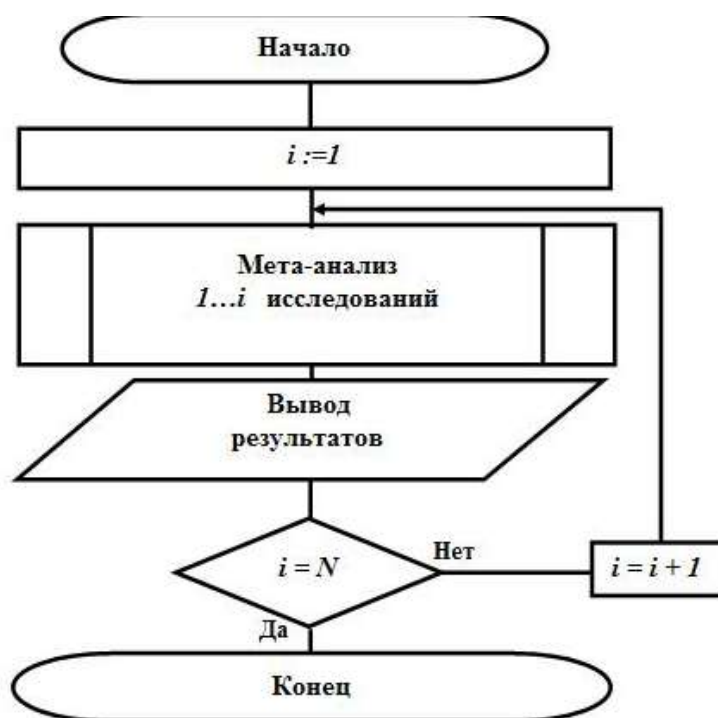


Рисунок 1.2. - Алгоритм кумулятивного мета-анализа

На первом шаге анализируется одно исследование (результаты совпадают с первым шагом стандартного мета-анализа), затем анализируются первое и второе исследования, затем первое, второе и третье исследования и т.д. – до тех пор, пока не будут объединены все исследования.

Итоговый результат (характеристики, полученные для объединенного исследования) во всех видах мета-анализа должен совпадать. На основании промежуточных результатов анализа делаются выводы о достоверности проявления эффекта в отдельных исследованиях. Отсутствие существенных отклонений частного результата (без учета каждого из отдельных исследований)

от конечного результата (по объединенному исследованию) является подтверждением достоверности и надежности полученных результатов.

Регрессионный мета-анализ (логистическая регрессия, регрессия взвешенных наименьших квадратов, модель Кокса и др.) используется при существенной гетерогенности результатов исследований. Он позволяет учесть влияние нескольких характеристик исследования (например, размера выборки, дозы препарата, способа его введения, характеристик больных и др.) на результаты испытаний вмешательства. Результаты регрессионного мета-анализа обычно представляют в виде коэффициента наклона с указанием ДИ [104].

Следует заметить, что мета-анализ может выполняться для обобщения результатов не только контролируемых испытаний медицинских вмешательств, но и когортных исследований (например, исследований факторов риска). Однако при этом следует учитывать высокую вероятность возникновения систематических ошибок [104].

Особый вид мета-анализа – обобщение оценок информативности диагностических методов, полученных в разных исследованиях. Цель такого мета-анализа — построение характеристической кривой взаимной зависимости чувствительности и специфичности тестов (ROC-кривой) с использованием взвешенной линейной регрессии [61].

1.1.3 Анализ чувствительности

После получения обобщенной оценки величины эффекта возникает необходимость определить ее устойчивость. Для этого выполняется так называемый анализ чувствительности [96].

В зависимости от конкретной ситуации его можно проводить на основе нескольких различных методов, например:

- включение и исключение из мета-анализа исследований, выполненных на низком методологическом уровне;

- изменение параметров данных, отбираемых из каждого анализируемого исследования, например, если в каких-либо исследованиях сообщается о клинических исходах в первые 2 нед. заболевания, а в других исследованиях — о клинических исходах в первые 3—4 нед. заболевания, то допустимо сравнение клинических исходов не только для каждого из этих периодов наблюдения, но и для суммарного периода наблюдения длительностью до 4 нед.;

- исключение из мета-анализа наиболее крупных исследований. Если величина эффекта того или иного анализируемого вмешательства при анализе чувствительности существенно не изменяется, то имеются основания полагать, что выводы первичного мета-анализа достаточно обоснованы.

Для качественной оценки наличия такой систематической ошибки мета-анализа обычно прибегают к построению воронкообразной диаграммы рассеяния результатов отдельных исследований в координатах (величина эффекта, размер выборки). При полном выявлении исследований эта диаграмма должна быть симметричной. Вместе с тем существуют и формальные методы оценки существующей асимметрии [82].

1.1.4 Представление результатов мета-анализа

Результаты мета-анализа обычно представляются графически (точечные и интервальные оценки величин эффектов каждого из включенных в мета-анализ исследований; пример на рисунке 1.3) и в виде таблиц с соответствующими статистиками.

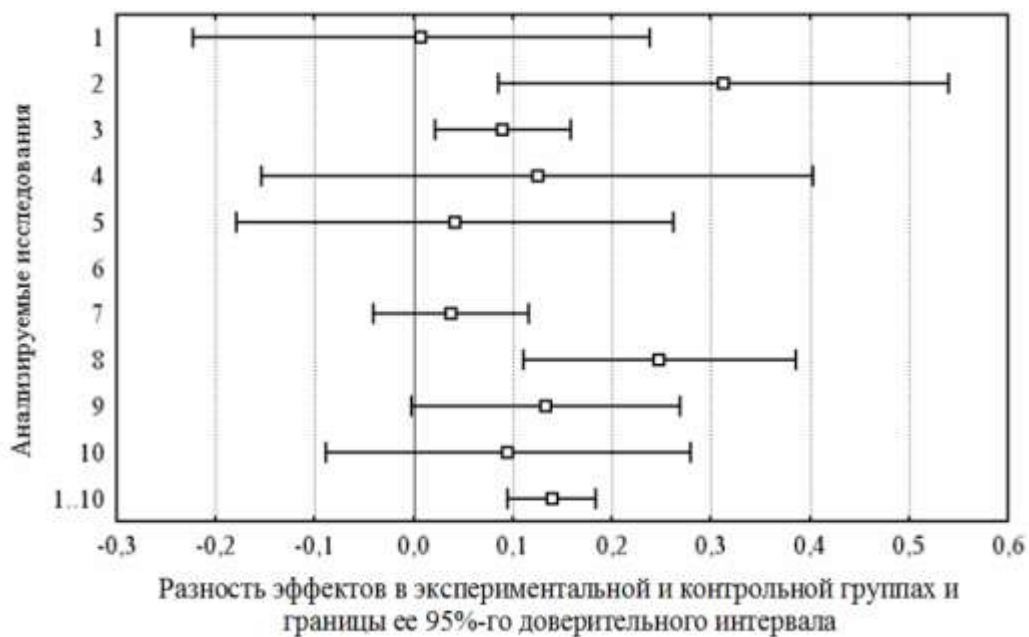


Рисунок 1.3 - Визуальное представление результатов стандартного мета-анализа

По оси ординат располагается порядковый номер анализируемого исследования (или исключенного исследования, или объединяемых исследований соответственно виду мета-анализа), а по оси абсцисс откладывается разность эффектов в экспериментальной и контрольной группах, отмечается левая и правая граница ее 95% доверительного интервала. Отрезки, соединяющие три указанные точки для каждого исследования, характеризуют выраженность эффекта и ее достоверность. Чем правее относительно оси нулевой абсциссы расположен полученный отрезок, тем выше проявление эффекта и ее достоверность. При этом считается, что эффект воздействия изучаемого фактора проявился, если 95% доверительный интервал располагается строго справа от оси нулевой абсциссы.

Контрольные вопросы

1. Чем отличается дискретный сигнал от цифрового?
2. Дайте определение Найквистовской частоты дискретизации.
3. Как изменится спектр сигнала, если он дискретизирован с частотой, меньшей, чем Найквистовская?

4. С какой целью перед дискретизацией аналоговый сигнал подвергают низкочастотной фильтрации? Как выбирается частота среза этого фильтра?

5. Нарисуйте структурную схему дискретизатора. Как в ней реализуется соотношение (1.1)?

6. Какие искажения имеют место при переходе от цифрового сигнала к непрерывному? Как реализуется этот переход?

7. Какие искажения дискретного сигнала вызывает отличие дискретизирующего импульса от δ -импульса Дирака?

8. Нарисуйте частотную характеристику усилительного тракта электрокардиосигнала.

9. С чем связаны искажения сигнала при его квантовании? Как изменится спектр функции в результате квантования?

10. С чем связаны искажения сигнала при его дискретизации? Как изменится спектр функции в результате дискретизации?

11. Объясните, почему в кино колесо отправляющего поезда сначала медленно вращается вперед, потом останавливается, а затем начинается вращаться назад?

12. Пусть мы имеем сигнал вида $\cos(8\pi/3t - \pi/3)$. Какова самая низкая искажаемая дискретизацией частота, если шаг дискретизации равен единице?

13. Как изменится спектр функции в результате квантования? Покажите, что это преобразование нелинейное.

14. Представьте алгоритм равномерного квантования функции $f(t)$.

15. Пусть мы дискретизируем функцию $\cos(13\pi/3t + \pi/3)$ с шагом дискретизации единица. Трансформируется ли при этом частота, если да, то в какую?.

16. К каким сигналам: дискретным или непрерывным может быть применено поэлементное квантование?

17. Представьте алгоритм равномерного квантования функции $f(t)$ в логарифмическом масштабе.

18. Чем определяются погрешности квантования?

19. Пусть мы имеем сигнал вида $\sum_{n=-\infty}^{n=\infty} \cos(2\pi n x / 9 + \pi / 3)$. Какова самая

низкая искажаемая дискретизацией частота, если дискретизация ведется в точках, соответствующих целым значениям x ?

20. Чем принципиально отличается спектр непрерывной и спектр дискретной функции?

21. Колесо велосипеда вращается с частотой 100 Гц. Какова кажущаяся частота вращения колеса, если стробоскоп дает вспышки с частотой 99 вспышек в 1 с?

22. Используя простые тригонометрические соотношения показать, что в точках дискретизации любая синусоида произвольной частоты f равнозначна синусоиде, лежащей в интервале $[0, 1/(2f)]$. Дискретизация ведется с найквистовской частотой в целые моменты t .

23. Приведите случаи, когда доказательство теоремы отсчетов будет некорректно.

24. Перечислите случаи, когда необходимо уменьшать шаг дискретизации по сравнению с расчетным. Почему рекомендуется это делать всегда.

25. Как изменится спектр сигнала при его дискретизации? Покажите, что это преобразование линейное.

Порядок защиты работы

Работа может быть зачтена, если студент представил отчет согласно п.1.5, исследуемые в работе сигналы соответствуют индивидуальному варианту, электронная форма соответствует представленному тексту, и студент дал исчерпывающие ответы на 10 произвольных вопросов из п.1.6.

Список использованных источников

1. Сергиенко А.Б. Цифровая обработка сигналов. - СПб. Питер. 2002.-608 с.
2. Зубов В.С. Программирование на языке TURBO PASCAL (версии 6.0 и 7.0)-М.: Информационно-издательский дом "Филинь", 1997.-304с.
3. Циммерман Франклин. Клиническая электрокардиография: -М.: «Издательство БИНОМ», 1997.-448с.

Практическая работа №2

«Веб-сервис для мета-анализа лекарственных назначений»

Программное обеспечение прототипа веб-сервиса, обеспечивающего мета-анализ микроскопических изображений мазков периферической крови и хранение информации о виде лекарственных воздействий и форменных элементов крови, вовлеченных в развитие патологических состояний, включает модули управления обработкой данных, модуля для идентификации, анализа и классификации форменных элементов в мазках периферической крови и модуль для анализа динамики межклеточных соотношений в результате лекарственных назначений. Отсутствует описание работ по разработке алгоритмов и программных модулей пп. 3.1, 3.2.3, 3.3 отчета, приведено только описание алгоритмов и работы модулей.

Отсутствует использование медицинских данных (кроме одного эпизода) при разработке программы.

1 Программное обеспечение модуля удаленного взаимодействия с пользователем

Модуль удаленного взаимодействия с пользователем представляет набор скриптов на языке Java и требует для работы установленного веб сервера, интерпретатора Java и СУБД MySQL. Сервер СУБД и веб-сервер могут быть установлены на разных компьютерах.

Для создания продукта применялся фреймворк Spring [101], язык Java для программирования основной функциональной части. Для доступа к данным используется драйвер mysql-connector-java, используемый в пуле c3p0. Для создания шаблонов страниц использовался шаблонизатор thymeleaf [105]. Работы, по обоснованию выбора программной платформы для веб-сервер были выполнены на Этапе 1.

Программный модуль предназначен для обеспечения сетевыми функциями системы обработки микроскопических изображений мазков периферической крови и хранение информации о виде лекарственных воздействий и форменных элементов крови, вовлеченных в развитие патологических состояний. Сетевые функции предоставляются пользователю по схеме запрос-ответ. Основные запросы клиента могут быть следующих видов:

- обработка изображений мазков периферической крови (загрузка, хранение, получение и др.), полученных в ходе клинической практики пользователя;
- ведение списка пациентов, у которых получают изображения мазков периферической крови;
- обработка информации о лекарственных средствах, которые могут быть назначены пациентам;
- ведение списка лекарственных назначений пациентам.

Основные ответы от сервера клиенту прототипа веб-сервиса морфологического анализа микроскопических изображений мазков периферической крови могут быть следующих видов:

- получение информации об изображениях мазков периферической крови, полученных в ходе клинической практики пользователя;
- ведение списка пациентов;
- ведение списка лекарственных средств, которые могут быть назначены пациентам;
- ведение списка лекарственных назначений пациентам.

На основе вышеперечисленных требований составим блочную структуру программного обеспечения. Блочная структура должна отражать логическую структуру программного обеспечения и связи между модулями программного обеспечения. Ее можно представить в виде схемы, показанной на рисунке 3.1.

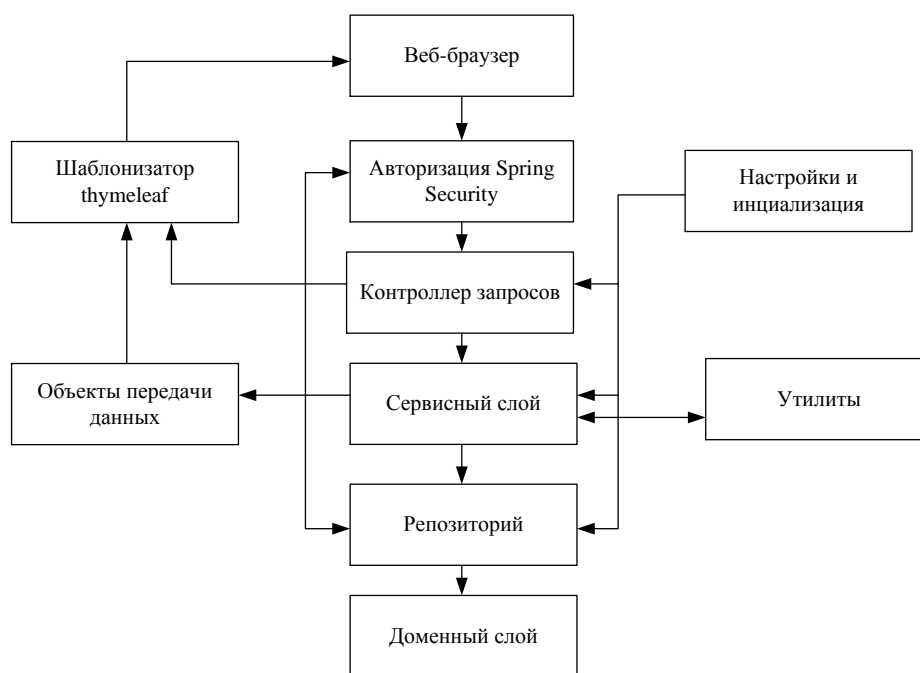


Рисунок 3.1 - Структура программного обеспечения веб-сервиса

Программное обеспечение состоит из следующих логических блоков:

- контроллер запросов;
- сервисный слой;
- репозиторий;
- доменный слой;
- утилиты;
- объекты передачи данных.

Все запросы, поступающие из внешней среды, перенаправляются в блок авторизации Security, который является частью фреймворка Spring. Работа в системе неавторизованных пользователей запрещена. Пользователям будет предложено зарегистрироваться, или войти в систему используя свой логин и пароль.

Контроллер запросов, обрабатывающий http-запросы пользователя, перенаправляет эти запросы на уровень бизнес-логики и формирует ответы пользователю в виде параметров модели представлений. Ответы формируются

шаблонизатором на основе библиотеки thymeleaf [105]. Контроллер запросов изменяет параметры представлений или напрямую, или с помощью объектов передачи данных из модуля объектов передачи данных. Блок описания объектов передачи данных, выполняет функции сериализации данных в/из слоя представлений, необходим для маршалинга данных.

Сервисный слой необходим для реализации бизнес-логики, обработки и трансляции передаваемых данных с уровня представлений на уровень данных.

Пакет описания репозитория служит для хранения сущностей, необходим для формирования интерфейсов доступа к данным предметной области посредством стандартных JPA-интерфейсов с формированием CRUD-модели хранения данных.

Блок доменного уровня сущностей, необходим для представления сущностей предметной области (для прототипа Веб-сервиса, обеспечивающего мета-анализ микроскопических изображений биомолекул и хранение информации о свойствах биомолекул, вовлеченных в развитие патологических состояний) в таблицах реляционной базы данных с указанием взаимосвязей между сущностями.

Блок утилит содержит процедуры маппинга (проецирования) данных из сервисного слоя в слой репозитория и обратно, классов задания ролей в базе данных, а также некоторых вспомогательных классов имплементирующих различные сервисные процедуры.

Блок настройки и инициализации необходим для настройки параметров приложения. Содержит следующие процедуры:

1. Процедура инициализации контекста приложения - необходима для настройки параметров многоязыковой поддержки и загрузки файлов, подключения основных компонент и сборки каркаса приложения, а также определения параметров окружения и места хранения статических ресурсов.

2. Процедура настройки параметров доступа к базе данных - необходима для конфигурации драйверов подключения к базе данных, настройки пула

драйверов подключения, настройки менеджера персистентности, запуска менеджера транзакций.

3. Процедура инициализации секьюрити-менеджера - необходима для настройки уровней доступа, списка защищаемых контроллеров, определения точки входа в приложение и страниц авторизации.

4. Процедура настройки параметров шаблонизатора - необходима для задания путей шаблонов представлений, указания типа продукта шаблонизации и параметров кэша.

Согласно пункту 4.3.6 технического задания прототип Веб-сервиса должен включать следующие элементы:

- персональный компьютер;
- базу данных мазков периферической крови;
- программный модуль для идентификации, анализа и классификации форменных элементов в мазках периферической крови;
- программный модуль для анализа динамики межклеточных соотношений в результате лекарственных назначений.

Проведя работу по оценки необходимой мощности вычислительных средств для решения поставленных задач, были предложены рекомендации по выбору персонального компьютера. Персональный компьютер рекомендуется использовать со следующими параметрами:

–от 3 Гб оперативной памяти, например память 8 шт. Kingston ValueRAM <KVR16R11D8/8> DDR-III DIMM 8Gb <PC3-12800> ECC Registered with Parity CL11;

–процессор семейства x86-64 с частотой не ниже 1 ГГц., например 2 шт. CPU Intel Xeon E5-2630 V2 2.6 GHz/6core/1.5+15Mb/80W/7.2 GT/s LGA2011;

–жесткий диск объемом не менее 30 Гб, например 2 шт. HDD 2 Tb SATA 6Gb/s Western Digital RE <WD2000FYYZ> 3.5" 7200rpm 64Mb.

Рекомендуемая программная конфигурация приведена ниже:

- виртуальная машина Java v7;

–СУБД MySQL v5;

–программная платформа MATLAB R2014a;

–веб-браузеры Mozilla Firefox, Google Chrome, Internet Explorer как программы-клиенты пользователей.

После выбора конфигурации программной платформы приступаем к выбору операционной системы.

Выбор программной платформы обуславливает выбор операционной системы. Отметим, что из-за кроссплатформенности частей выбранная программная платформа может быть запущена как на операционной системе из ряда Windows, так и на операционной системе из ряда Linux. Причем как на серверных операционных системах, так и на пользовательских операционных системах. Главное требование – это поддержка протокола TCP/IP и возможность запуска приведенной программной конфигурации, чему удовлетворяют все вышеприведенные семейства операционных систем.

Для администрирования СУБД рекомендуется установить на сервер phpMyAdmin.

Для корректной работы программного обеспечения должны быть выполнены следующие условия: доступность сервера через сеть Internet, а также развернута СУБД и создана база данных. Если в базе данных первоначально не было необходимых таблиц, то JPA-менеджер создаст необходимые таблицы и связи между ними.

Сформируем входные данные программного обеспечения.

К входным данным программного обеспечения относят:

–базовый URL-адрес сервера;

–адрес сервера баз данных;

–имя базы данных;

–имя пользователя базы данных и его пароль;

–максимально допустимый размер загружаемых файлов.

Данные, поступающие от пользователей:

–ФИО;

- логин;
- пароль;
- данные об используемых лекарствах;
- данные о пациентах;
- данные о лекарственных назначениях;
- данные о полученных изображениях мазков периферической крови и сами изображения с привязкой к пациенту и лекарственному назначению.

Все запросы от пользователей отправляются по протоколу http. Форматы запросов приведены далее.

Адрес и поля запроса «Вход в систему» представлены на рисунке 3.2.

Для выхода из системы необходимо направить get-запрос по адресу /logout.

Адрес - /login

Имя пользователя

ContentTypeHeader - text/plain

ContentDispositionHeader - form-data; name=\" j_username\"

Пароль

ContentTypeHeader - text/plain

ContentDispositionHeader - form-data; name=\" j_password\"

Рисунок 3.2 - Адреса и поля запроса «Вход в систему»

На рисунке 3.3 представлены поля и адрес запроса «Регистрация в системе».

Адрес - /signup

Имя

ContentTypeHeader - text/plain

ContentDispositionHeader - form-data; name=\"


```
firstName\"

    Фамилия
    ContentTypeHeader - text/plain
    ContentDispositionHeader - form-data; name=\" lastName\"

    Логин
    ContentTypeHeader - text/plain
    ContentDispositionHeader - form-data; name=\" username\"

    Пароль
    ContentTypeHeader - text/plain
    ContentDispositionHeader - form-data; name=\" password\"

    Пароль еще раз
    ContentTypeHeader - text/plain
    ContentDispositionHeader - form-data; name=\"
repassword\"
```

Рисунок 3.3 - Поля и адрес запроса «Регистрация в системе»

Для получения списка используемых лекарств get-запрос на адрес `/medications/get`.

Для создания описания нового лекарства используется POST-запрос на адрес `/medications/create`. Адрес и поля этого запроса приведены на рисунке 3.4.

Адрес - /medications/create
Наименование
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\"name \"
Описание
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\"description \"
Тип
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\"type \"

Рисунок 3.4 - Адрес и поля запроса для создания описания нового лекарства

Для редактирования описания нового лекарства используется GET-запрос на адрес /medications/ edit. Адрес и поля этого запроса приведены на рисунке 3.5.

Адрес - /medications/ edit
Идентификатор
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\"id \"

Рисунок 3.5 - Адрес и поля запроса «редактирование существующего описания нового лекарства»

Для удаления описания нового лекарства используется GET-запрос на адрес /medications/ delete. Адрес и поля этого запроса приведены на рисунке 3.6.

Адрес - /medications/ delete
Идентификатор
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\"id \"

Рисунок 3.6 - Адрес и поля запроса «удаление существующего
лекарственного назначения»

Для получения списка введенных назначений используется get-запрос на адрес / prescriptions /get.

Для создания нового лекарственного назначения используется POST-запрос на адрес / prescriptions /create.

Адрес и поля этого запроса приведены на рисунке 3.7.

Адрес - / prescriptions /create
Доза
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\" doze \"
Описание
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\"description \"
пациент
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\"patient _id\"
Лекарство
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\" medication _id\"

Рисунок 3.7 - Адрес и поля запроса «создание нового лекарственного
назначения»

Для редактирования лекарственного назначения используется GET-запрос на адрес / prescriptions / edit. Адрес и поля этого запроса приведены на рисунке 3.8.

Адрес - / prescriptions / edit
Идентификатор
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\"id \"

Рисунок 3.8 - Адрес и поля запроса «редактирование существующего лекарственного назначения»

Для удаления лекарственного назначения используется GET-запрос на адрес / prescriptions / delete. Адрес и поля этого запроса приведены на рисунке 3.9.

Адрес - / prescriptions / delete
Идентификатор
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\"id \"

Рисунок 3.9 - Адрес и поля запроса «удаление существующего лекарственного назначения»

Для получения списка описания пациентов используется get-запрос на адрес / patients /get.

Для создания нового описания пациента используется POST-запрос на адрес / patients /create.

Адрес и поля этого запроса приведены на рисунке 3.10.

Адрес - / patients /create
Имя
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\" firstname \"
Фамилия
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\" lastname \"
Отчество
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\" middlename \"
Дата рождения
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\" birth \"

Пол
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\" male \"

Рисунок 3.10 - Адрес и поля запроса «создание нового описания пациента»

Для редактирования описания пациента используется GET-запрос на адрес / patients / edit. Адрес и поля этого запроса приведены на рисунке 3.11.

Адрес - / patients / edit
Идентификатор
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\" id \"

Рисунок 3.11 - Адрес и поля запроса «редактирование существующего описания пациента»

Для удаления описания пациента используется GET-запрос на адрес / patients / delete. Адрес и поля этого запроса приведены на рисунке 3.12.

Адрес - / patients / delete
Идентификатор
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\"id \"

Рисунок 3.12 - Адрес и поля запроса «удаление существующего описания пациента»

Для получения списка изображений используется get-запрос на адрес / pictures / get.

Для создания нового описания изображения используется POST-запрос на адрес / pictures / create. Для загрузки файла изображения на сервер используется POST-запрос на адрес / pictures / upload с параметром MultipartFile. Адрес и поля запроса для создания нового описания изображения приведены на рисунке 3.13.

Адрес - / pictures / create
Имя
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\" name \"
Локаатор ресурса
ContentTypeHeader - text/plain
ContentDispositionHeader - form-data; name=\" url \"
Путь к файлу
ContentTypeHeader - text/plain

```
ContentDispositionHeader - form-data; name=\" path \"  
  
Пациент  
ContentTypeHeader - text/plain  
ContentDispositionHeader - form-data; name=\" patient_id \"
```

Рисунок 3.13 - Адрес и поля запроса «создание нового описания изображения»

Для редактирования описания изображения используется GET-запрос на адрес / pictures / edit. Адрес и поля этого запроса приведены на рисунке 3.14.

```
Адрес - / pictures / edit  
  
Идентификатор  
ContentTypeHeader - text/plain  
ContentDispositionHeader - form-data; name=\"id \"
```

Рисунок 3.14 - Адрес и поля запроса «редактирование существующего описания изображения»

Для удаления описания изображения используется GET-запрос на адрес / pictures / delete. Адрес и поля этого запроса приведены на рисунке 3.15.

```
Адрес - / pictures / delete  
  
Идентификатор  
ContentTypeHeader - text/plain  
ContentDispositionHeader - form-data; name=\"id \"
```

Рисунок 3.15 - Адрес и поля запроса «удаление существующего описания изображения»

Сформулируем требования к выходным данным программного обеспечения.

К выходным данным программы можно отнести следующее:

- html-страницы, содержащие информацию по запросу пользователя;
- файлы изображений мазков периферической крови;
- ответы системы в случае возникновения ошибок.

В случае возникновения ошибок возвращаются следующие сообщения с соответствующими кодами HTTP-ответа (отличным от 200):

Ошибка запрета доступа `error.access.denied`. Сообщение: «Доступ не разрешен. У вас недостаточно прав для доступа!».

Ошибка неверного запроса пользователя `error.BadRequest`. Сообщение: «Неправильный запрос. Проверьте данные.».

Внутренняя ошибка сервера `error.InternalServerError`. Сообщение: «Внутренняя ошибка сервера.».

Недопустимые данные `error.WrongData`. Сообщение: «Проверьте вводимые данные.».

Совпадение логина `status.signup.invalid.username.duplicate`. Сообщение: «Такой логин уже занят!».

Несоответствие паролей на форме ввода пользователя `status.signup.invalid.password.notmatching`. Сообщение: «Пароли не соответствуют друг другу!».

Контрольные вопросы

1. Что представляет собой модуль удаленного взаимодействия с пользователем?
2. Для чего предназначен программный модуль?
3. Перечислите виды основных запросов клиента.
4. Из каких логических блоков состоит программное обеспечение?
5. Какие процедуры содержит блок настройки и инициализации?

Практическая работа №3

«Программное обеспечение для построения интерьера»

Целью практического занятия является описание процесса запуска и наладки прототипа Веб-сервиса. Прототип Веб-сервиса может поставляться или в виде проекта некомпилрованных файлов-скриптов, или в виде скомпилированной библиотеки с файлами запуска для различных операционных систем (Windows, Linux). Для наладки удобно использовать некомпилрованный проект, содержащий серверные исполняемые скрипты. Для запуска некомпилрованного проекта в среде Eclipse необходимо проимпортировать gradle-проект, построить его модель и над файлом Application.java в контекстном меню выбрать пункт Run As\Java Application (рисунок 3.1).

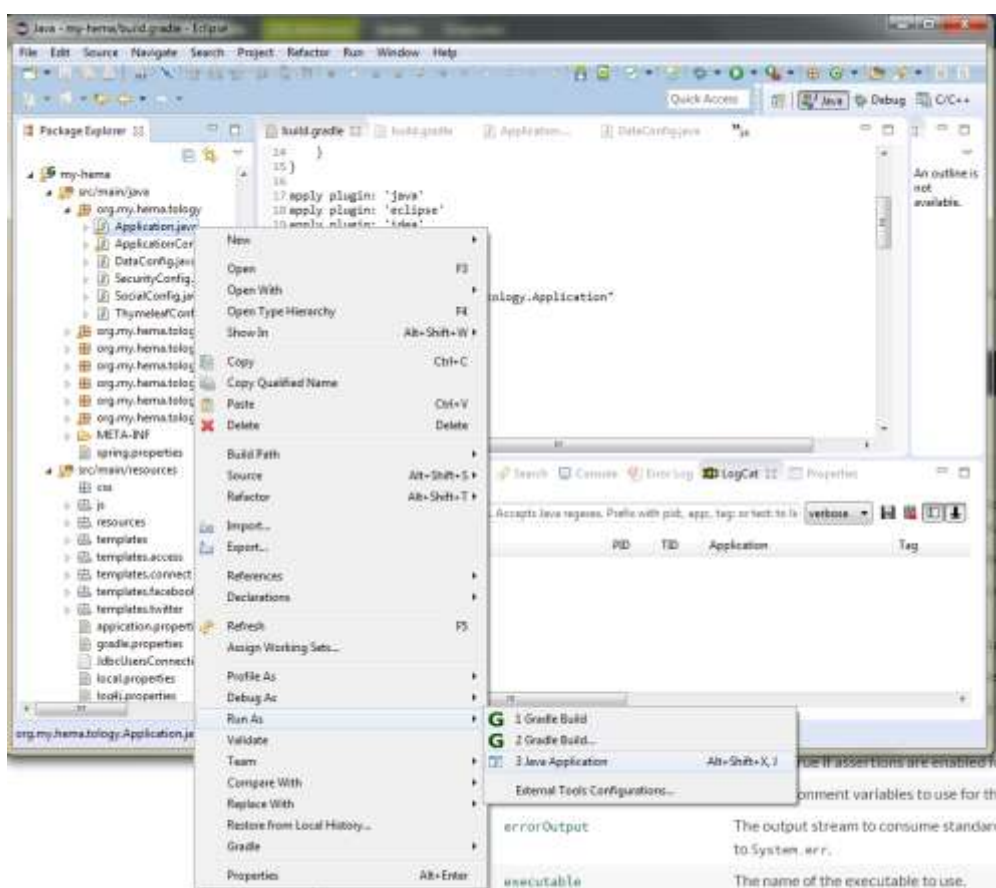


Рисунок 3.1 – Запуск некомпилрованного проекта в среде Eclipse

2015-06-30 10:17:36.414 INFO 6420 --- [main]
o.s.s.core.SpringSecurityCoreVersion : You are running with Spring Security Core
3.2.1.RELEASE

2015-06-30 10:17:36.415 INFO 6420 --- [main]
o.s.s.config.SecurityNamespaceHandler : Spring Security 'config' module version is
3.2.1.RELEASE

2015-06-30 10:17:36.440 INFO 6420 --- [main]
erInvocationSecurityMetadataSourceParser : Creating access control expression
attribute 'hasRole('ROLE_USER')' for /

2015-06-30 10:17:36.440 INFO 6420 --- [main]
erInvocationSecurityMetadataSourceParser : Creating access control expression
attribute 'permitAll' for /login

2015-06-30 10:17:36.500 INFO 6420 --- [main]
s.s.c.h.HttpSecurityBeanDefinitionParser : Checking sorted filter chain: [Root bean:
class [org.springframework.security.web.context.SecurityContextPersistenceFilter];
scope=; abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;
autowireCandidate=true; primary=false; factoryBeanName=null;
factoryMethodName=null; initMethodName=null; destroyMethodName=null, order =
200, Root bean: class
[org.springframework.security.web.context.request.async.WebAsyncManagerIntegratio
nFilter]; scope=; abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;
autowireCandidate=true; primary=false; factoryBeanName=null;
factoryMethodName=null; initMethodName=null; destroyMethodName=null, order =
400, Root bean: class
[org.springframework.security.web.authentication.logout.LogoutFilter]; scope=;
abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;
autowireCandidate=true; primary=false; factoryBeanName=null;
factoryMethodName=null; initMethodName=null; destroyMethodName=null, order =
700,
<org.springframework.security.web.authentication.UsernamePasswordAuthenticationFi

```
lter#0>,          order      =      1100,          Root      bean:      class
[org.springframework.security.web.authentication.www.BasicAuthenticationFilter];
scope=; abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;
autowireCandidate=true;          primary=false;          factoryBeanName=null;
factoryMethodName=null; initMethodName=null; destroyMethodName=null, order =
1500,          Root          bean:          class
[org.springframework.security.web.savedrequest.RequestCacheAwareFilter]; scope=;
abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;
autowireCandidate=true;          primary=false;          factoryBeanName=null;
factoryMethodName=null; initMethodName=null; destroyMethodName=null, order =
1600,          Root          bean:          class
[org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter];
scope=; abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;
autowireCandidate=true;          primary=false;          factoryBeanName=null;
factoryMethodName=null; initMethodName=null; destroyMethodName=null, order =
1700,          Root          bean:          class
[org.springframework.security.web.authentication.AnonymousAuthenticationFilter];
scope=; abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;
autowireCandidate=true;          primary=false;          factoryBeanName=null;
factoryMethodName=null; initMethodName=null; destroyMethodName=null, order =
2000,          Root          bean:          class
[org.springframework.security.web.session.SessionManagementFilter]; scope=;
abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;
autowireCandidate=true;          primary=false;          factoryBeanName=null;
factoryMethodName=null; initMethodName=null; destroyMethodName=null, order =
2100,          Root          bean:          class
[org.springframework.security.web.access.ExceptionTranslationFilter]; scope=;
abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;
autowireCandidate=true;          primary=false;          factoryBeanName=null;
factoryMethodName=null; initMethodName=null; destroyMethodName=null, order =
```

2200, <org.springframework.security.web.access.intercept.FilterSecurityInterceptor#0>, order = 2300]

2015-06-30 10:17:37.194 INFO 6420 --- [main] trationDelegate\$BeanPostProcessorChecker : Bean 'transactionAttributeSource' of type [class org.springframework.transaction.annotation.AnnotationTransactionAttributeSource] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)

2015-06-30 10:17:37.200 INFO 6420 --- [main] trationDelegate\$BeanPostProcessorChecker : Bean 'transactionInterceptor' of type [class org.springframework.transaction.interceptor.TransactionInterceptor] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)

2015-06-30 10:17:37.205 INFO 6420 --- [main] trationDelegate\$BeanPostProcessorChecker : Bean 'org.springframework.transaction.config.internalTransactionAdvisor' of type [class org.springframework.transaction.interceptor.BeanFactoryTransactionAttributeSourceAdvisor] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)

2015-06-30 10:17:37.431 INFO 6420 --- [main] e.j.JettyEmbeddedServletContainerFactory : Server initialized with port: 8080

2015-06-30 10:17:37.433 INFO 6420 --- [main] org.eclipse.jetty.server.Server : jetty-8.1.14.v20131031

2015-06-30 10:17:37.457 INFO 6420 --- [main] / : Initializing Spring embedded WebApplicationContext

2015-06-30 10:17:37.457 INFO 6420 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 1441 ms

2015-06-30 10:17:37.527 INFO 6420 --- [main]
o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: Ant
[pattern='/resources'], []

2015-06-30 10:17:37.650 INFO 6420 --- [main]
com.mchange.v2.log.MLog : MLog clients using log4j logging.

2015-06-30 10:17:37.664 INFO 6420 --- [main]
com.mchange.v2.c3p0.C3P0Registry : Initializing c3p0-0.9.0.4 [built 23-January-
2006 22:20:29 -0500; debug? true; trace: 10]

2015-06-30 10:17:37.688 INFO 6420 --- [main]
j.LocalContainerEntityManagerFactoryBean : Building JPA container
EntityManagerFactory for persistence unit 'hibernatePersistenceUnit'

2015-06-30 10:17:37.695 INFO 6420 --- [main]
o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [
name: hibernatePersistenceUnit
...]

2015-06-30 10:17:37.749 INFO 6420 --- [main] org.hibernate.Version
: HHH000412: Hibernate Core {4.3.1.Final}

2015-06-30 10:17:37.750 INFO 6420 --- [main]
org.hibernate.cfg.Environment : HHH000206: hibernate.properties not found

2015-06-30 10:17:37.751 INFO 6420 --- [main]
org.hibernate.cfg.Environment : HHH000021: Bytecode provider name :
javassist

2015-06-30 10:17:37.911 INFO 6420 --- [main]
o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons
Annotations {4.0.4.Final}

2015-06-30 10:17:37.998 INFO 6420 --- [main]
c.mchange.v2.c3p0.PoolBackedDataSource : Initializing c3p0 pool...
com.mchange.v2.c3p0.ComboPooledDataSource@18474a2[acquireIncrement -> 5,
acquireRetryAttempts -> 30, acquireRetryDelay -> 1000, autoCommitOnClose -> false,
automaticTestTable -> null, breakAfterAcquireFailure -> false, checkoutTimeout -> 0,

connectionTesterClassName -> com.mchange.v2.c3p0.impl.DefaultConnectionTester,
description -> null, driverClass -> com.mysql.jdbc.Driver, factoryClassLocation -> null,
forceIgnoreUnresolvedTransactions -> false, identityToken -> 18474a2,
idleConnectionTestPeriod -> 25, initialPoolSize -> 3, jdbcUrl ->
jdbc:mysql://localhost/test2, loginTimeout -> 0, maxIdleTime -> 0, maxPoolSize ->
100, maxStatements -> 50, maxStatementsPerConnection -> 0, minPoolSize -> 10,
numHelperThreads -> 3, preferredTestQuery -> null, properties -> {user=*****,
password=*****}, propertyCycle -> 300, testConnectionOnCheckin -> false,
testConnectionOnCheckout -> false, usesTraditionalReflectiveProxies -> false]

2015-06-30 10:17:38.171 INFO 6420 --- [main]
org.hibernate.dialect.Dialect : HHH000400: Using dialect:
org.hibernate.dialect.MySQL5InnoDBDialect

2015-06-30 10:17:38.211 INFO 6420 --- [main]
o.h.e.jdbc.internal.LobCreatorBuilder : HHH000424: Disabling contextual LOB
creation as createClob() method threw error :
java.lang.reflect.InvocationTargetException

2015-06-30 10:17:38.324 INFO 6420 --- [main]
o.h.h.i.ast.ASTQueryTranslatorFactory : HHH000397: Using
ASTQueryTranslatorFactory

main] org.hibernate.tool.hbm2ddl.SchemaUpdate : HHH000232: Schema
update complete

2015-06-30 10:17:38.837 INFO 6420 --- [main]
o.s.s.web.DefaultSecurityFilterChain : Creating filter chain:
org.springframework.security.web.util.matcher.AnyRequestMatcher@1,
[org.springframework.security.web.context.SecurityContextPersistenceFilter@1d26ad2,
org.springframework.security.web.context.request.async.WebAsyncManagerIntegration
Filter@1b25753,
org.springframework.security.web.authentication.logout.LogoutFilter@e21292,
org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilt
er@c06358,

org.springframework.security.web.authentication.www.BasicAuthenticationFilter@e21568,
org.springframework.security.web.savedrequest.RequestCacheAwareFilter@173ce28,
org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter@11b35d6,
org.springframework.security.web.authentication.AnonymousAuthenticationFilter@82400e, org.springframework.security.web.session.SessionManagementFilter@3d5863,
org.springframework.security.web.access.ExceptionTranslationFilter@1dece42,
org.springframework.security.web.access.intercept.FilterSecurityInterceptor@1d3f8aa

2015-06-30 10:17:38.845 INFO 6420 --- [main]
o.s.s.c.h.DefaultFilterChainValidator : Checking whether login URL '/login' is accessible with your configuration

2015-06-30 10:17:39.068 INFO 6420 --- [main]
o.s.b.c.e.ServletRegistrationBean : Mapping servlet: 'dispatcherServlet' to [/]

2015-06-30 10:17:39.070 INFO 6420 --- [main]
o.s.b.c.embedded.FilterRegistrationBean : Mapping filter: 'metricFilter' to: [/*]

2015-06-30 10:17:39.072 INFO 6420 --- [main]
o.s.b.c.embedded.FilterRegistrationBean : Mapping filter: 'org.springframework.security.filterChainProxy' to: [/*]

2015-06-30 10:17:39.072 INFO 6420 --- [main]
o.s.b.c.embedded.FilterRegistrationBean : Mapping filter: 'org.springframework.security.web.access.intercept.FilterSecurityInterceptor#0' to: [/*]

2015-06-30 10:17:40.082 INFO 6420 --- [main] / :
Initializing Spring FrameworkServlet 'dispatcherServlet'

2015-06-30 10:17:40.082 INFO 6420 --- [main]
o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization started

2015-06-30 10:17:40.095 INFO 6420 --- [main]
o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization completed in 13 ms


```

2015-06-30 10:17:40.130 INFO 6420 --- [ main]
o.e.jetty.server.AbstractConnector : Started SelectChannelConnector@0.0.0.0:8080
2015-06-30 10:17:40.133 INFO 6420 --- [ main]
.s.b.c.e.j.JettyEmbeddedServletContainer : Jetty started on port: 8080
2015-06-30 10:17:40.134 INFO 6420 --- [ main]
org.my.hema.tology.Application : Started Application in 4.373 seconds (JVM
running for 4.694)

```

Для запуска скомпилированной версии Веб-сервиса необходимо запустить прилагаемый файл `my-hema.bat` для операционной системы Windows или `my-hema` для семейств ОС Linux. В этом случае запустится стандартная консоль операционной системы и лог запуска приложения будет выводиться туда (рисунок 6.2).

```

C:\Windows\system32\cmd.exe - c:\my-hema\bin\my-hema.bat
2015-06-30 13:32:41.020 INFO 3692 --- [ main] o.springframework.core.
SpringProperties : Found 'spring.properties' file in local classpath
:: Spring Boot :: (v1.0.2.RELEASE)
2015-06-30 13:32:41.141 INFO 3692 --- [ main] org.my.hema.tology.Appl
ication : Starting Application on newPC with PID 3692 (C:\my-hema\lib\
hematology-0.1.0.jar started by M in c:\my-hema\bin)
2015-06-30 13:32:41.162 INFO 3692 --- [ main] ationConfigEmbeddedWebA
pplicationContext : Refreshing org.springframework.boot.context.embedded.Annotat
ionConfigEmbeddedWebApplicationContext@77e9807f: startup date [Tue Jun 30 13:32:
41 MSK 2015]; root of context hierarchy

```

Рисунок 6.2 – Запуск Веб-сервиса из консоли

После окончания процесса запуска можно проверить доступность сервиса, набрав в адресной строке персонального компьютера, на котором был запущен Веб-сервис, адрес `http://localhost:8080/`. Если тестирование происходит на компьютере, который отличен от Веб-сервера, но включен в ту же локальную сеть, то вместо части адреса обозначенного “localhost” можно ввести локальный IP-адрес Веб сервера, например, для IP-адреса 192.168.0.1 адресная строка в

браузере должна принимать вид `http:// 192.168.0.1:8080/`. Для тестирования видимости из глобального Интернета вместо “localhost” необходимо указывать зарегистрированный URL сервера, или его глобальный (WAN) IP-адрес. При этом в коммуникационном оборудовании, с помощью которого сервер подключен к глобальной сети, необходимо обеспечивать проброс портов 8080 (port forwarding). Успешный запуск Веб-сервиса в браузере приведен на рисунке 6.3. На этом рисунке можно видеть, что адрес в браузере несколько отличается от введенного – это результат работы блока Spring Security, который перенаправляет любые запросы от неавторизованных пользователей на страницу /login.

Возможные ошибки при запуске Веб-сервиса связаны с невозможностью подключения к необходимой СУБД. Так, если указаны неправильные параметры подключения в файле `Spring.properties`, в частности неверно указан адрес сервера СУБД, или неверно указаны параметры учетной записи пользователя СУБД (логин, пароль), то хотя приложение и стартует, но связи с базой данных не происходит. Это видно по логам запуска Веб-сервиса. Пример лога запуска Веб-сервиса с неправильно определенными параметрами подключения к СУБД приведен в листинге 6.2.

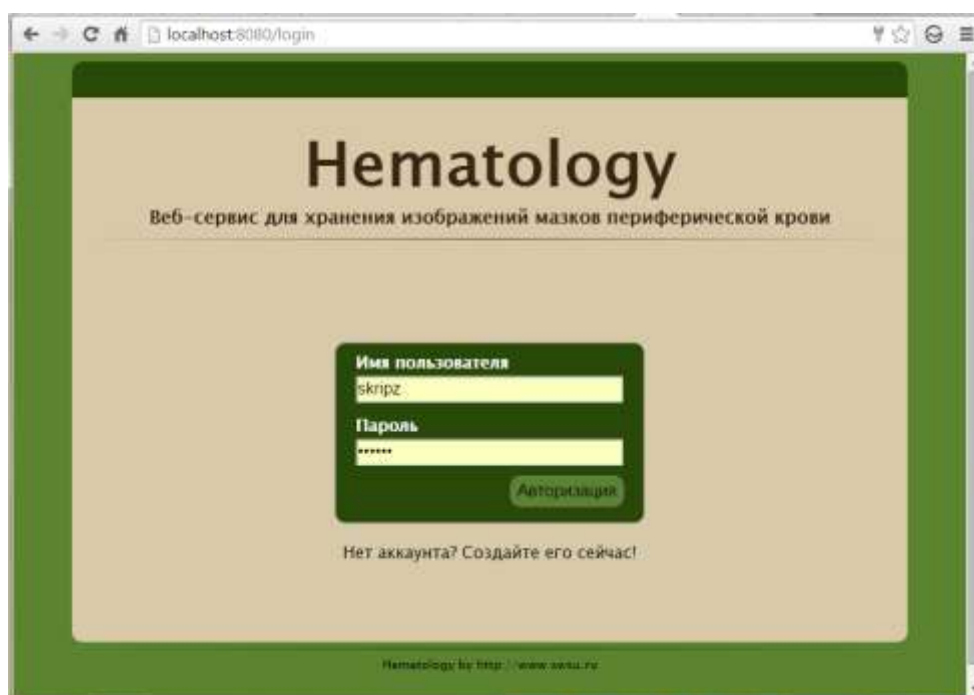


Рисунок 6.3 – Внешний вид успешно запущенного веб-сервиса

Листинг 6.2. Фрагмент лога приложения, которому не удалось подключиться к СУБД

```
2015-06-30 10:24:28.496 INFO 7936 --- [main]
o.springframework.core.SpringProperties : Found 'spring.properties' file in local
classpath
```

```

. ____ _ _ _ _
^\\/_'_ _ _ _(_)_ _ _ _ \\ \\ \\
(( )\\_ |'_ |'_ |'_ |'_ \\_ `| \\ \\ \\ \\
\\_ _ _ )| | )| | | | | | | ( | | ) ) ) )
' | _ _ | . _ _ | | _ _ | _ _ , | / / / /
=====|_|=====|_|/=/////
:: Spring Boot :: (v1.0.2.RELEASE)
```

```
2015-06-30 10:24:28.617 INFO 7936 --- [main]
org.my.hema.tology.Application : Starting Application on newPC with PID 7936
(D:\w3\my-hema\bin started by M in D:\w3\my-hema)
```

```
2015-06-30 10:24:28.638 INFO 7936 --- [main]
ationConfigEmbeddedWebApplicationContext : Refreshing
org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicat
ionContext@8e4bb0: startup date [Tue Jun 30 10:24:28 MSK 2015]; root of context
hierarchy
```

```
2015-06-30 10:24:28.958 INFO 7936 --- [main]
o.s.b.f.xml.XmlBeanDefinitionReader : Loading XML bean definitions from class
path resource
```

```
e.j.JettyEmbeddedServletContainerFactory : Server initialized with port: 8080
```

```
2015-06-30 10:24:30.040 INFO 7936 --- [main]
org.eclipse.jetty.server.Server : jetty-8.1.14.v20131031
```

2015-06-30 10:24:30.068 INFO 7936 --- [main] / :
Initializing Spring embedded WebApplicationContext
2015-06-30 10:24:30.068 INFO 7936 --- [main]
o.s.web.context.ContextLoader : Root WebApplicationContext: initialization
completed in 1432 ms
2015-06-30 10:24:30.159 INFO 7936 --- [main]
o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: Ant
[pattern='/resources'], []
2015-06-30 10:24:30.282 INFO 7936 --- [main]
com.mchange.v2.log.MLog : MLog clients using log4j logging.
2015-06-30 10:24:30.291 INFO 7936 --- [main]
com.mchange.v2.c3p0.C3P0Registry : Initializing c3p0-0.9.0.4 [built 23-January-
2006 22:20:29 -0500; debug? true; trace: 10]
2015-06-30 10:24:30.318 INFO 7936 --- [main]
j.LocalContainerEntityManagerFactoryBean : Building JPA container
EntityManagerFactory for persistence unit 'hibernatePersistenceUnit'
2015-06-30 10:24:30.324 INFO 7936 --- [main]
o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [
name: hibernatePersistenceUnit
...]
2015-06-30 10:24:30.379 INFO 7936 --- [main] org.hibernate.Version
: HHH000412: Hibernate Core {4.3.1.Final}
2015-06-30 10:24:30.380 INFO 7936 --- [main]
org.hibernate.cfg.Environment : HHH000206: hibernate.properties not found
2015-06-30 10:24:30.382 INFO 7936 --- [main]
org.hibernate.cfg.Environment : HHH000021: Bytecode provider name :
javassist
2015-06-30 10:24:30.546 INFO 7936 --- [main]
o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons
Annotations {4.0.4.Final}

2015-06-30 10:24:30.634 INFO 7936 --- [main]
c.mchange.v2.c3p0.PoolBackedDataSource : Initializing c3p0 pool...
com.mchange.v2.c3p0.ComboPooledDataSource@18474a2[acquireIncrement -> 5,
acquireRetryAttempts -> 30, acquireRetryDelay -> 1000, autoCommitOnClose -> false,
automaticTestTable -> null, breakAfterAcquireFailure -> false, checkoutTimeout -> 0,
connectionTesterClassName -> com.mchange.v2.c3p0.impl.DefaultConnectionTester,
description -> null, driverClass -> com.mysql.jdbc.Driver, factoryClassLocation -> null,
forceIgnoreUnresolvedTransactions -> false, identityToken -> 18474a2,
idleConnectionTestPeriod -> 25, initialPoolSize -> 3, jdbcUrl ->
jdbc:mysql://localhost2/test2, loginTimeout -> 0, maxIdleTime -> 0, maxPoolSize ->
100, maxStatements -> 50, maxStatementsPerConnection -> 0, minPoolSize -> 10,
numHelperThreads -> 3, preferredTestQuery -> null, properties -> {user=*****,
password=*****}, propertyCycle -> 300, testConnectionOnCheckin -> false,
testConnectionOnCheckout -> false, usesTraditionalReflectiveProxies -> false]

2015-06-30 10:25:07.433 WARN 7936 --- [r\$PoolThread-#0]
c.m.v2.resourcepool.BasicResourcePool :
com.mchange.v2.resourcepool.BasicResourcePool\$AcquireTask@593ed5 --
Acquisition Attempt Failed!!! Clearing pending acquires. While trying to acquire a
needed new resource, we failed to succeed more than the maximum number of allowed
acquisition attempts (30).

2015-06-30 10:25:07.436 WARN 7936 --- [r\$PoolThread-#1]
c.m.v2.resourcepool.BasicResourcePool :
com.mchange.v2.resourcepool.BasicResourcePool\$AcquireTask@1254422 --
Acquisition Attempt Failed!!! Clearing pending acquires. While trying to acquire a
needed new resource, we failed to succeed more than the maximum number of allowed
acquisition attempts (30).

2015-06-30 10:25:07.440 WARN 7936 --- [main]
o.h.e.jdbc.internal.JdbcServicesImpl : HHH000342: Could not obtain connection to
query metadata : Connections could not be acquired from the underlying database!

2015-06-30 10:25:07.461 INFO 7936 --- [main]
org.hibernate.dialect.Dialect : HHH000400: Using dialect:
org.hibernate.dialect.MySQL5InnoDBDialect

2015-06-30 10:25:07.476 INFO 7936 --- [main]
o.h.e.jdbc.internal.LobCreatorBuilder : HHH000422: Disabling contextual LOB
creation as connection was null

2015-06-30 10:25:07.600 INFO 7936 --- [main]
o.h.h.i.ast.ASTQueryTranslatorFactory : HHH000397: Using
ASTQueryTranslatorFactory

2015-06-30 10:25:07.860 INFO 7936 --- [main]
org.hibernate.tool.hbm2ddl.SchemaUpdate : HHH000228: Running hbm2ddl schema
update

2015-06-30 10:25:07.861 INFO 7936 --- [main]
org.hibernate.tool.hbm2ddl.SchemaUpdate : HHH000102: Fetching database metadata

2015-06-30 10:25:44.561 WARN 7936 --- [r\$PoolThread-#0]
c.m.v2.resourcepool.BasicResourcePool :
com.mchange.v2.resourcepool.BasicResourcePool\$AcquireTask@1d22ed --
Acquisition Attempt Failed!!! Clearing pending acquires. While trying to acquire a
needed new resource, we failed to succeed more than the maximum number of allowed
acquisition attempts (30).

2015-06-30 10:25:44.586 ERROR 7936 --- [main]
org.hibernate.tool.hbm2ddl.SchemaUpdate : HHH000319: Could not get database
metadata

java.sql.SQLException: Connections could not be acquired from the underlying
database!

at com.mchange.v2.sql.SqlUtils.toSQLException(SqlUtils.java:104)

at

com.mchange.v2.c3p0.impl.C3P0PooledConnectionPool.checkoutPooledConnection(C
3P0PooledConnectionPool.java:264)

at
com.mchange.v2.c3p0.PoolBackedDataSource.getConnection(PoolBackedDataSource.java:94)
at
com.mchange.v2.c3p0.ComboPooledDataSource.getConnection(ComboPooledDataSource.java:521)
at
org.hibernate.engine.jdbc.connections.internal.DatasourceConnectionProviderImpl.getConnection(DatasourceConnectionProviderImpl.java:139)
at
org.hibernate.tool.hbm2ddl.SuppliedConnectionProviderConnectionHelper.prepare(SuppliedConnectionProviderConnectionHelper.java:51)
at org.hibernate.tool.hbm2ddl.SchemaUpdate.execute(SchemaUpdate.java:194)
at org.hibernate.tool.hbm2ddl.SchemaUpdate.execute(SchemaUpdate.java:178)
at org.hibernate.internal.SessionFactoryImpl.<init>(SessionFactoryImpl.java:521)
at org.hibernate.cfg.Configuration.buildSessionFactory(Configuration.java:1857)
at
org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl\$4.perform(EntityManagerFactoryBuilderImpl.java:850)

2015-06-30 10:25:46.204 INFO 7936 --- [main] / :
Initializing Spring FrameworkServlet 'dispatcherServlet'
2015-06-30 10:25:46.205 INFO 7936 --- [main] :
o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet':
initialization started
2015-06-30 10:25:46.217 INFO 7936 --- [main] :
o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet':
initialization completed in 12 ms
2015-06-30 10:25:46.254 INFO 7936 --- [main] :
o.e.jetty.server.AbstractConnector : Started SelectChannelConnector@0.0.0.0:8080

```
2015-06-30 10:25:46.258 INFO 7936 --- [main]
.s.b.c.e.j.JettyEmbeddedServletContainer : Jetty started on port: 8080
```

```
2015-06-30 10:25:46.259 INFO 7936 --- [main]
org.my.hema.tology.Application : Started Application in 77.867 seconds (JVM
running for 78.178)
```

Ошибка в листинге 6.2 говорит, что «ННН000319: Could not get database metadata» - «не могу получить информацию о базе данных». И хотя контейнер сервлетов запускается, Веб сервис без СУБД полноценно работать не будет. Любые операции с СУБД (в частности, попытка сравнить введенную информацию о пользователе – логин, пароль) работать не будут. Результат обзора Веб-сервиса в таком случае представлен на рисунке 6.4.

При этом ошибки подключения к базе данных будут выходить в лог (6.3). В данном случае об этом говорит сообщение «Could not open JPA EntityManager for transaction;» - не могу открыть менеджер персистентности JPA (связи с базами данных) для транзакции.

Пути наладки Веб-сервиса в данном случае – это проверка данных подключения и правильная настройка следующих параметров файла Spring.properties:

- app.jdbc.url в виде «jdbc\:mysql\://» + «URL сервера» + «/»+ «Имя БД»;
- app.jdbc.username в виде «Логин пользователя БД»;
- app.jdbc.password в виде «Пароль пользователя БД».

Листинг 6.3 - Фрагмент лога приложения при ошибках подключения к базе данных

```
2015-06-30 11:43:53.873 INFO 8984 --- [ qtp26492970-22]
o.s.b.a.audit.listener.AuditListener : AuditEvent [timestamp=Tue Jun 30 11:43:53
MSK 2015, principal=anonymousUser, type=AUTHORIZATION_FAILURE,
```



```
data={type=org.springframework.security.access.AccessDeniedException,
message=Access is denied}}
```

```
2015-06-30 11:44:38.353 WARN 8984 --- [r$PoolThread-#2]
c.m.v2.resourcepool.BasicResourcePool :
com.mchange.v2.resourcepool.BasicResourcePool$AcquireTask@1a0dac7 --
Acquisition Attempt Failed!!! Clearing pending acquires. While trying to acquire a
needed new resource, we failed to succeed more than the maximum number of allowed
acquisition attempts (30).
```

```
2015-06-30 11:44:38.354 WARN 8984 --- [ qtp26492970-17]
o.h.engine.jdbc.spi.SqlExceptionHelper : SQL Error: 0, SQLState: null
```

```
2015-06-30 11:44:38.355 ERROR 8984 --- [ qtp26492970-17]
o.h.engine.jdbc.spi.SqlExceptionHelper : Connections could not be acquired from the
underlying database!
```

```
2015-06-30 11:44:38.363 INFO 8984 --- [ qtp26492970-17]
o.s.b.a.audit.listener.AuditListener : AuditEvent [timestamp=Tue Jun 30 11:44:38
MSK 2015, principal=skripz, type=AUTHENTICATION_FAILURE,
data={type=org.springframework.security.authentication.AuthenticationServiceExcepti
on, message=Could not open JPA EntityManager for transaction; nested exception is
javax.persistence.PersistenceException:
org.hibernate.exception.GenericJDBCException: Could not open connection}]
```


:: Spring Boot :: (v1.0.2.RELEASE)

2015-06-30 10:50:53.637 INFO 5956 --- [main]
org.my.hema.tology.Application : Starting Application on newPC with PID 5956
(D:\w3\my-hema\bin started by M in D:\w3\my-hema)

2015-06-30 10:50:53.658 INFO 5956 --- [main]
ationConfigEmbeddedWebApplicationContext : Refreshing
org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicat
ionContext@8e4bb0: startup date [Tue Jun 30 10:50:53 MSK 2015]; root of context
hierarchy

2015-06-30 10:50:53.982 INFO 5956 --- [main]
o.s.b.f.xml.XmlBeanDefinitionReader : Loading XML bean definitions from class
path resource [spring-security.xml]

.....

2.1/org.eclipse.jetty.orbit/javax.el/2.2.0.v201108011116/ec8944c11833d84b0283a
5afbad0fafb264f86a9/javax.el-2.2.0.v201108011116.jar,
file:/C:/Users/M/.gradle/caches/modules-2/files-
2.1/org.eclipse.jetty.orbit/com.sun.el/2.2.0.v201108011116/15f7774c3fa514835a371f47
c152317704ea411a/com.sun.el-2.2.0.v201108011116.jar,
file:/C:/Users/M/.gradle/caches/modules-2/files-
2.1/org.eclipse.jetty.orbit/org.eclipse.jdt.core/3.7.1/5b79bfee0852ca685e33cab74496fa3
400271b5b/org.eclipse.jdt.core-3.7.1.jar, file:/C:/Users/M/.gradle/caches/modules-
2/files-2.1/org.eclipse.jetty/jetty-
jsp/8.1.14.v20131031/87dc1723a32113d74be2420d940f6825e3b6af42/jetty-jsp-
8.1.14.v20131031.jar]

Exception in thread "main"
org.springframework.context.ApplicationContextException: Unable to start embedded
container; nested exception is
org.springframework.boot.context.embedded.EmbeddedServletContainerException:
Unable to start embedded Jetty servlet container

at
org.springframework.boot.context.embedded.EmbeddedWebApplicationContext.onRefresh(EmbeddedWebApplicationContext.java:135)
at
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:476)
at
org.springframework.boot.context.embedded.EmbeddedWebApplicationContext.refresh(EmbeddedWebApplicationContext.java:120)
at
org.springframework.boot.SpringApplication.refresh(SpringApplication.java:648)
at org.springframework.boot.SpringApplication.run(SpringApplication.java:311)
at org.springframework.boot.SpringApplication.run(SpringApplication.java:909)
at org.springframework.boot.SpringApplication.run(SpringApplication.java:898)
at org.my.hema.tology.Application.main(Application.java:23)

Caused by:

org.springframework.boot.context.embedded.EmbeddedServletContainerException:
Unable to start embedded Jetty servlet container
at
org.springframework.boot.context.embedded.jetty.JettyEmbeddedServletContainer.initialize(JettyEmbeddedServletContainer.java:78)
at
org.springframework.boot.context.embedded.jetty.JettyEmbeddedServletContainer.<init>(JettyEmbeddedServletContainer.java:63)
at
org.springframework.boot.context.embedded.jetty.JettyEmbeddedServletContainerFactory.getJettyEmbeddedServletContainer(JettyEmbeddedServletContainerFactory.java:256)

at

org.springframework.boot.context.embedded.jetty.JettyEmbeddedServletContainerFactory.getEmbeddedServletContainer(JettyEmbeddedServletContainerFactory.java:133)

at

org.springframework.boot.context.embedded.EmbeddedWebApplicationContext.createEmbeddedServletContainer(EmbeddedWebApplicationContext.java:159)

at

org.springframework.boot.context.embedded.EmbeddedWebApplicationContext.onRefresh(EmbeddedWebApplicationContext.java:132)

... 7 more

Caused by: java.net.BindException: Address already in use: bind

at sun.nio.ch.Net.bind0(Native Method)

at sun.nio.ch.Net.bind(Unknown Source)

at sun.nio.ch.Net.bind(Unknown Source)

at sun.nio.ch.ServerSocketChannelImpl.bind(Unknown Source)

at sun.nio.ch.ServerSocketAdaptor.bind(Unknown Source)

at

org.eclipse.jetty.server.nio.SelectChannelConnector.open(SelectChannelConnector.java:187)

at org.eclipse.jetty.server.AbstractConnector.doStart(AbstractConnector.java:316)

at

org.eclipse.jetty.server.nio.SelectChannelConnector.doStart(SelectChannelConnector.java:265)

at

org.eclipse.jetty.util.component.AbstractLifeCycle.start(AbstractLifeCycle.java:64)

at org.eclipse.jetty.server.Server.doStart(Server.java:293)

at

org.eclipse.jetty.util.component.AbstractLifeCycle.start(AbstractLifeCycle.java:64)

at

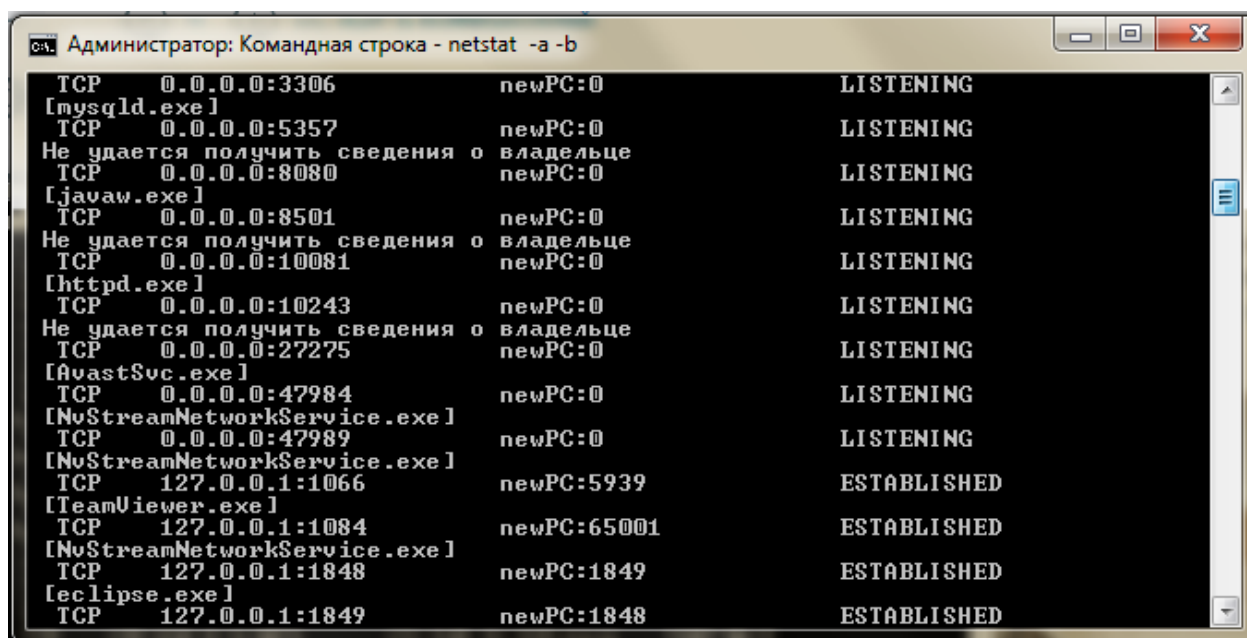
org.springframework.boot.context.embedded.jetty.JettyEmbeddedServletContainer.initialize(JettyEmbeddedServletContainer.java:68)

... 12 more

В логе 6.4 информативным является сообщение «Unable to start embedded Jetty servlet container», которое сообщает, что не может запуститься встроенный контейнер сервлетов.

Возможные пути наладки Веб-сервиса в данном случае – это поиск приложений, которые захватывают порт 8080, например, с помощью системной команды netstat -a -b.

На рисунке 6.5 показан результат работы этой команды, из которого видно, что процесс, который захватил порт 8080 называется Javaw.exe. Для исправления ситуации рекомендуется закрыть этот процесс или собственными средствами, или средствами операционной системы.



```
Администратор: Командная строка - netstat -a -b
TCP    0.0.0.0:3306                newPC:0             LISTENING
[mysqld.exe]
TCP    0.0.0.0:5357                newPC:0             LISTENING
Не удается получить сведения о владельце
TCP    0.0.0.0:8080                newPC:0             LISTENING
[javaw.exe]
TCP    0.0.0.0:8501                newPC:0             LISTENING
Не удается получить сведения о владельце
TCP    0.0.0.0:10081               newPC:0             LISTENING
[httpd.exe]
TCP    0.0.0.0:10243               newPC:0             LISTENING
Не удается получить сведения о владельце
TCP    0.0.0.0:27275               newPC:0             LISTENING
[AvastSvc.exe]
TCP    0.0.0.0:47984                newPC:0             LISTENING
[NoStreamNetworkService.exe]
TCP    0.0.0.0:47989                newPC:0             LISTENING
[NoStreamNetworkService.exe]
TCP    127.0.0.1:1066               newPC:5939          ESTABLISHED
[TeamViewer.exe]
TCP    127.0.0.1:1084               newPC:65001         ESTABLISHED
[NoStreamNetworkService.exe]
TCP    127.0.0.1:1848               newPC:1849          ESTABLISHED
[eclipse.exe]
TCP    127.0.0.1:1849               newPC:1848          ESTABLISHED
```

Рисунок 6.5 – Результат работы команды netstat -a -b.

Таким образом, представлено описание процесса запуска и наладки прототипа Веб-сервиса. Прототип Веб-сервиса может поставляться или в виде

проекта некомпилерованных файлов-скриптов, или в виде компилированной библиотеки с файлами запуска для различных операционных систем (Windows, Linux). Для наладки удобно использовать некомпилерованный проект, содержащий серверные исполняемые скрипты. Для запуска некомпилерованного проекта в среде Eclipse необходимо проимпортировать gradle-проект, построить его модель и над файлом Application.java в контекстном меню выбрать пункт Run As\Java Application.

Контрольные вопросы

1. Что включает в себя понятие "программное обеспечение"?
2. Охарактеризуйте основные категории ПО.
3. В чем отличие прикладных программ от системных и инструментальных?
4. В чем состоит назначение операционной системы?
5. Что такое файл и как организована файловая система?
6. Приведите пример иерархической файловой структуры.
7. Что такое базовая система ввода-вывода (BIOS)?