


Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования «Новосибирский национальный исследовательский  
государственный университет» (Новосибирский государственный университет, НГУ)

Экономический факультет

---

Согласовано  
Декан ЭФ  
Богомолова Т.Ю.

  
\_\_\_\_\_

подпись

« 19 » 10 20 20 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

### ПРОГРАММНЫЕ ИНТЕРФЕЙСЫ И ВИЗУАЛИЗАЦИЯ ДАННЫХ

Направление подготовки: 38.03.05 Бизнес-информатика  
Направленность (профиль): Бизнес-информатика

Форма обучения: очная

Разработчики:

Степанюк К.С.,  
Денисенко М.Ю.,  
К.э.н. Костин А.В

Зав. кафедрой применения математических методов  
в экономике и планировании

д.э.н, профессор Мкртчян Г.М.

Новосибирск  
2020

## Содержание

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.....	3
2. Место дисциплины в структуре образовательной программы .....	5
3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося .....	5
4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий.....	5
5. Перечень учебной литературы .....	7
6. Перечень учебно-методических материалов по самостоятельной работе обучающихся..	7
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины .....	8
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине .....	8
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине .....	8
10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине.....	9

## 1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

### Цель дисциплины:

Дисциплина «Программные интерфейсы и визуализация данных» имеет своей целью сформировать у студентов навыки разработки программных интерфейсов и многокомпонентных программных систем

### Основные задачи дисциплины:

- развить практические навыки разработки программ на языке Java;
- научить пользоваться вспомогательными инструментами для упрощения и ускорения написания программ;
- развить навыки ведения командной разработки программных систем.

### Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Результаты освоения образовательной программы (компетенции)	В результате изучения дисциплины обучающиеся должны:		
	знать	уметь	владеть
ОПК-1. Способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.	- принципы ООП и обработки исключительных ситуаций - основы криптографии и шифрования	- писать программы различной сложности; - организовывать наборы инструкций и функций в виде крупных компонентов – модулей; - использовать модели обработки исключительных ситуаций; - использовать инструменты отладки и тестирования	- навыками отладки кода, поиска и исправления ошибок
ОПК-3. Способность работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях	- общую синтаксическую модель языка Java; - общие методы решения алгоритмических задач (полный перебор, рекурсия); - принципы клиент-серверного взаимодействия, на	- писать программы различной сложности; - разрабатывать пользовательские интерфейсы; - использовать динамическую типизацию и ссылки; - использовать способы упаковки программного кода многократного	- навыком написания программного кода; - навыками событийно-управляемого программирования; - навыками создания, использования и перегрузки модулей и пакетов модулей; - объектно-ориентированным

Результаты освоения образовательной программы (компетенции)	В результате изучения дисциплины обучающиеся должны:		
	знать	уметь	владеть
	базе веб-приложений; - инструментарий для командной разработки программ	использования и предотвращения появления избыточного кода; - организовать наборы инструкций и функций в виде крупных компонентов – модулей; - писать объектно-ориентированный программный код; - использовать модели обработки исключительных ситуаций; - использовать инструменты отладки и тестирования; -использовать классические алгоритмы обработки данных и применять абстрактные типы данных.	стилем программирования
ПК-16. Умение разрабатывать контент и ИТ-сервисы предприятия и Интернет-ресурсов	- принципы клиент-серверного взаимодействия, на базе веб-приложений; - инструментарий для командной разработки программ	- писать программы различной сложности; - разрабатывать пользовательские интерфейсы; - организовать наборы инструкций и функций в виде крупных компонентов – модулей; - создавать программное обеспечение, работая в команде	- навыками событийно-управляемого программирования; - навыками создания, использования и перегрузки модулей и пакетов модулей; - объектно-ориентированным стилем программирования; - навыками отладки кода, поиска и исправления ошибок; - навыком командной разработки

## 2. Место дисциплины в структуре образовательной программы

Дисциплина «Программные интерфейсы и визуализация данных» является обязательной (для набора 2017 года является элективной), преподается в 6 семестре (для набора 2016-2019 годов в 5 семестре).

Дисциплины (практики), изучение которых необходимо для освоения дисциплины «Программные интерфейсы и визуализация данных»: Теоретические основы информатики, Программирование, Основы Web-программирования.

Дисциплины (практики), для изучения которых необходимо освоение дисциплины «Программные интерфейсы и визуализация данных»: Управление ИТ-проектами, Основы программирования и конфигурирования в корпоративных информационных системах.

## 3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

Трудоемкость дисциплины – 5 зачетных единиц, 180 часов (для набора 2017 года 4 зачетные единицы, 144 часов).

Форма промежуточной аттестации: дифференцированный зачет.

Вид деятельности	Семестр
	6
<b>Контактная работа, часов, в том числе:</b>	74
лекции	-
практические занятия	64
груп. работа с преподавателем	8
контактная работа при аттестации	2
консультации перед экзаменом	-
<b>Самостоятельная работа, часов, в том числе:</b>	<b>106</b>
самостоятельная работа во время занятий	100
самостоятельная работа во время промежуточной аттестации	6
<b>Всего, часов</b>	<b>180</b>

## 4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий 6 семестр

№ п/п	Раздел дисциплины	Семестр	Виды учебной работы, в часах	
			практические занятия	самостоятельная работа во время занятий
1	Знакомство с языком Java	6	4	6
2	Введение в ООП	6	4	5
3	Коллекции	6	4	7
4	Исключения. Git	6	4	5
5	Файлы. Паттерны	6	4	6
6	Многопоточные приложения	6	4	6
7	Maven	6	4	6

8	Базы данных	6	4	7
9	Веб-сервер. Шифрование	6	4	8
10	Веб-разработка в Java	6	4	8
11	HTTP	6	4	6
12	Командная работа в ИТ	6	20	30
	<b>Всего</b>		64	100

Содержание дисциплины «Программные интерфейсы и визуализация данных»:

Содержание разделов	
1	<b>Знакомство с языком Java.</b> Сравнение языков Java и Python. Общая синтаксическая модель языка Java. Общие методы решения алгоритмических задач.
2	<b>Введение в ООП.</b> JVM, JDK, JRE. Байт-код. ООП. Наследование. Инкапсуляция. Полиморфизм. Правила оформления кода.
3	<b>Коллекции.</b> Иерархия коллекций. Основные интерфейсы коллекций в Java.
4	<b>Исключения. Git.</b> Обработка исключений. Системы контроля версий. Git.
5	<b>Файлы. Паттерны.</b> Поток ввода-вывода. Работа с файлами. Паттерны проектирования. Паттерн Singleton.
6	<b>Многопоточные приложения.</b> Процесс. Поток. Создание потоков. Состояния потока. Синхронизация потоков. Проблемы, которые может создавать многопоточность. Паттерн Builder.
7	<b>Maven.</b> Автоматизация сборки. Функциональные части Maven. Зависимости и плагины. Основные фазы сборки проекта.
8	<b>Базы данных. СУБД. SQL.</b> Реляционная база данных. JDBC. Проектирование БД.
9	<b>Веб-сервер. Шифрование.</b> Локальная сеть. Глобальная сеть. Стек протоколов. TCP/IP. SSL. Симметричное шифрование. Асимметричное шифрование.
10	<b>Веб-разработка в Java.</b> Server-side rendering. Client-side rendering. Java сервлет.
11	<b>HTTP.</b> Методы HTTP запросов. Коды состояния. URI. URN. URL.
12	<b>Командная работа в ИТ.</b> Методологии разработки. Тайм-менеджмент в ИТ. Разработка по ТЗ.

#### Практические занятия (64 ч)

Содержание практического занятия	Объем, час
Занятие на тему «Знакомство с языком Java»	4
Занятие на тему «Введение в ООП»	4
Занятие на тему «Коллекции»	4
Занятие на тему «Исключения. Git»	4
Занятие на тему «Файлы. Паттерны»	4
Занятие на тему «Многопоточные приложения»	4
Занятие на тему «Maven»	4
Занятие на тему «Базы данных»	4
Занятие на тему «Веб-сервер. Шифрование»	4
Занятие на тему «Веб-разработка в Java»	4
Занятие на тему «HTTP»	4
Занятие на тему «Командная работа»	4
Презентации итоговых проектов	16

## Самостоятельная работа студентов (100 ч)

Перечень занятий на СРС	Объем, час
Подготовка к тестированию	24
Выполнение домашних заданий	46
Работа по созданию итогового проекта	30

## 5. Перечень учебной литературы

### 5.1 Основная литература

1. Роганов, Е.А. Основы информатики и программирования: курс / Е.А. Роганов. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2006. – 336 с. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=234651> (дата обращения: 21.10.2020). – Текст : электронный.

2. Гуськова, О.И. Объектно ориентированное программирование в Java : учебное пособие / О.И. Гуськова ; Московский педагогический государственный университет. – Москва : Московский педагогический государственный университет (МПГУ), 2018. – 240 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=500355> (дата обращения: 21.10.2020). – Библиогр. в кн. – ISBN 978-5-4263-0648-6. – Текст : электронный.

### 5.2 Дополнительная литература

3. Крахоткина, Е.В. Технологии разработки Internet-приложений : учебное пособие / Е.В. Крахоткина ; Северо-Кавказский федеральный университет. – Ставрополь : Северо-Кавказский Федеральный университет (СКФУ), 2016. – 124 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=459070> (дата обращения: 21.10.2020). – Библиогр. в кн. – Текст : электронный.

4. Кузнецов, С. Введение в модель данных SQL: курс / С. Кузнецов. – 2-е изд., исправ. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 351 с. : илл. – (Основы информационных технологий). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=429087> (дата обращения: 21.10.2020). – Библиогр. в кн. – ISBN 5-9556-00028-0. – Текст : электронный.

5. Беликова, С.А. Основы HTML и CSS: проектирование и дизайн веб-сайтов: учебное пособие по курсу «Web-разработка» : [16+] / С.А. Беликова, А.Н. Беликов ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2020. – 176 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=598663> (дата обращения: 21.10.2020). – Библиогр. в кн. – ISBN 978-5-9275-3435-7. – Текст : электронный.

## 6. Перечень учебно-методических материалов по самостоятельной работе обучающихся

6. Студенты используют личные аккаунты на GitHub, куда выкладывают свои домашние задания.

## **7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

При освоении дисциплины используются следующие ресурсы информационно-телекоммуникационной сети «Интернет»:

- GitHub (<https://github.com/>);
- Документация Java (<https://docs.oracle.com/javase/8/docs/api/index.html>);
- Статья о Continuous Integration (<https://habr.com/ru/post/352282/>);
- Документация об Apache Maven (<https://maven.apache.org/>)

Взаимодействие обучающегося с преподавателем (синхронное и (или) асинхронное) осуществляется через личный кабинет студента в ЭИОС, электронную почту, социальные сети.

### **7.1 Современные профессиональные базы данных:**

*Современные профессиональные базы данных не используются.*

### **7.2. Информационные справочные системы**

*Не используются*

## **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине**

Перечень программного обеспечения:

Windows, IntelliJ IDEA, Microsoft Office.

## **9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Для реализации дисциплины «Программные интерфейсы и визуализация данных» используются специальные помещения:

1. Учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля, промежуточной и итоговой аттестации;
2. Помещения для самостоятельной работы обучающихся;

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду НГУ.

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».



## 10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине

Перечень результатов обучения по дисциплине «Программные интерфейсы и визуализация данных» и индикаторов их достижения представлен в виде знаний, умений и владений в разделе 1.

### 10.1 Порядок проведения текущего контроля и промежуточной аттестации по дисциплине

#### **Текущий контроль успеваемости:**

После завершения раздела дисциплины студентам предоставляется возможность выполнения домашних лабораторных работ (домашних заданий), а также поводится оценивание знаний студентов на практических занятиях (тестирование). Каждое домашнее задание и каждый вопрос на тестировании позволяет получить определенное количество баллов. В сумме за все вопросы на тестированиях можно получить 33 балла, за все домашние задания – 37 баллов.

#### **Промежуточная аттестация:**

Формой промежуточной аттестации по дисциплине является дифференцированный зачет, который представляет собой защиту итоговой работы. Максимальное количество баллов на зачете – 30. Таким образом, максимальное количество баллов, которое можно набрать по курсу – 100 баллов.

Оценивание результатов обучения по дисциплине «Программные интерфейсы и визуализация данных» осуществляется по бально-рейтинговой системе и включает следующие оценочные средства:

Оценочные средства	Баллы (максимум)
<b>Текущий контроль</b>	
Тестирование на практических занятиях	33
Выполнение домашних заданий	37
<b>Промежуточная аттестация</b>	
Защита итогового проекта	30
<b>Итого</b>	<b>100</b>

### **Описание критериев и шкал оценивания индикаторов достижения результатов обучения по дисциплине «Программные интерфейсы и визуализация данных»**

Таблица 10.1

Код компетенции	Результат обучения по дисциплине	Оценочное средство
ОПК-1	Знать: - принципы ООП и обработки исключительных ситуаций - основы криптографии и шифрования Уметь: - организовывать наборы инструкций и функций в виде крупных компонентов – модулей; - использовать модели обработки исключительных ситуаций; Владеть:	Тестирование Домашнее задание Итоговый проект

	- навыками отладки кода, поиска и исправления ошибок	
ОПК-3	<p>Знать:</p> <ul style="list-style-type: none"> <li>- общую синтаксическую модель языка Java;</li> <li>- общие методы решения алгоритмических задач (полный перебор, рекурсия);</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>- писать программы различной сложности;</li> <li>- использовать динамическую типизацию и ссылки;</li> <li>- использовать способы упаковки программного кода многократного использования и предотвращения появления избыточного кода;</li> <li>- писать объектно-ориентированный программный код;</li> <li>- использовать инструменты отладки и тестирования;</li> <li>- использовать классические алгоритмы обработки данных и применять абстрактные типы данных</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>- навыком написания программного кода;</li> <li>- навыками создания, использования и перегрузки модулей и пакетов модулей;</li> <li>- объектно-ориентированным стилем программирования</li> </ul>	<p>Тестирование</p> <p>Домашнее задание</p> <p>Итоговый проект</p>
ПК-16	<p>Знать:</p> <ul style="list-style-type: none"> <li>- принципы клиент-серверного взаимодействия, на базе веб-приложений;</li> <li>- инструментарий для командной разработки программ</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>- писать программы различной сложности;</li> <li>- разрабатывать пользовательские интерфейсы;</li> <li>- создавать программное обеспечение, работая в команде</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>- навыками событийно-управляемого программирования;</li> <li>- навыком командной разработки</li> </ul>	<p>Тестирование</p> <p>Домашнее задание</p> <p>Итоговый проект</p>

Таблица 10.2

<b>Критерии оценивания результатов обучения</b>	<b>Шкала оценивания</b>
<p><b><u>Качество домашних заданий:</u></b></p> <ul style="list-style-type: none"> <li>- правильная работа написанной программы,</li> <li>- правильное выполнение всех пунктов заданий,</li> <li>- грамотная защита своей программы (с описанием её структуры),</li> <li>- наличие исчерпывающих ответов на дополнительные вопросы.</li> </ul>	<p><i>Отлично</i></p> <p><b>80,1–100</b></p> <p><b>баллов</b></p>

<p>В выполненных заданиях обучающийся мог допустить непринципиальные неточности.</p> <p><b>Тестирование:</b> – отсутствие ошибок при ответе на тестовые вопросы, ответы полные</p> <p><b>Итоговый проект:</b> – полное соответствие написанной программы указанным требованиям, – грамотная защита своей программы – наличие исчерпывающих ответов на дополнительные вопросы.</p> <p>В выполненных заданиях обучающийся мог допустить непринципиальные неточности.</p>	
<p><b>Качество выполнения домашних заданий:</b> - в работе программы есть мелкие недочеты, которые обучающийся не может исправить; - не выполнен малозначимый пункт задания, - наличие затруднений в формулировке собственных суждений при защите программы и ответе на дополнительные вопросы.</p> <p><b>Тестирование:</b> – не менее 80% ответов на тестовые вопросы должны быть полными и правильными.</p> <p><b>Итоговый проект:</b> – полное соответствие написанной программы указанным требованиям, – грамотная защита своей программы с незначительными недочетами – наличие полных ответов на дополнительные вопросы с возможным наличием ошибок.</p>	<p><i>Хорошо</i> <b>60,1–80,0</b> баллов</p>
<p><b>Качество выполнения домашних заданий:</b> - в работе программы есть крупные недочеты, которые обучающийся не может исправить; - не выполнен важный пункт задания, - наличие затруднений в формулировке собственных суждений при защите программы и ответе на дополнительные вопросы.</p> <p><b>Тестирование:</b> – не менее 60% ответов на тестовые вопросы должны быть полными и правильными.</p> <p><b>Итоговый проект:</b> – наличие несоответствий написанной программы указанным требованиям, – неполная защита программы – наличие неполных и / или содержащих существенные ошибки ответов на дополнительные вопросы.</p>	<p><i>Удовлетворительно</i> от <b>40,1</b> до <b>60,0</b> баллов</p>
<p><b>Качество выполнения домашних заданий:</b> - в работе программы есть крупные недочеты, которые обучающийся не может исправить; - не выполнен важный пункт задания, - грубые ошибки при защите программы и ответе на дополнительные вопросы.</p> <p><b>Тестирование:</b> – присутствие многочисленных ошибок (более 60% ответов содержат ошибки)</p> <p><b>Итоговый проект:</b></p>	<p><i>Неудовлетворительно</i> менее <b>40,1</b> баллов</p>

<ul style="list-style-type: none"> <li>– значительное несоответствий написанной программы указанным требованиям,</li> <li>– неполная защита программы,</li> <li>– отсутствие ответов на дополнительные вопросы.</li> </ul>	
--	--

Баллы, набранные за выполнение заданий текущего контроля и промежуточной аттестации, конвертируются в оценку по дисциплине следующим образом:

Итоговая сумма набранных баллов	Оценка
≤ 40	неудовлетворительно
от 40,1 до 60	удовлетворительно
от 60,1 до 80	хорошо
от 80,1 до 100	отлично

### Типовые контрольные задания и иные материалы, необходимые для оценки результатов обучения

#### Примеры вопросов для тестирования:

1. (1 балл)

**Вариант 1:** Что такое и для чего нужны JRE и JVM?

**Вариант 2:** Что такое и для чего нужны JDK и байт-код?

2. (1 балл)

Написать основные принципы ООП и дать им определение.

3. (1 балл)

**Вариант 1:** Рассмотрим продукты в холодильнике. Требуется создать классы:

\* **Product** (продукт), в котором будут поля price (цена) и supplier (производитель).

\* **Cheese** (сыр), который наследуется от класса **Product** и вносит дополнительное поле weight (вес).

\* **Orange** (апельсин), который также будет наследоваться от класса **Product** и добавит поле count (количество).

**Вариант 2:** Рассмотрим сотрудников крупной компании. Требуется создать классы:

\* **Person** (человек), в котором будут поля name (имя) и age (возраст).

\* **Employee** (работник), который наследуется от класса **Person** и вносит дополнительное поле salary (зарплата).

\* **Scientist** (ученый), который наследуется от класса **Employee** и вносит дополнительное поле numberOfArticles (количество публикаций).

## Примеры домашних заданий:

### Задача #1 (2 балла)

Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, вывода информации о фигуре, изменения размеров и вычисления площади, периметра..

### Задача #2 (4 балла)

Требуется создать систему учёта для магазина смартфонов.

Класс **Smartphone** должен иметь поля: **serialNumber** (уникальный серийный номер, тип String), **price** (цена, тип int), **supplier**(производитель, тип String), **phoneModel** (модель, тип String) и метод **public void printInfo(Smartphone)** для вывода подробной информации о смартфоне.

Класс **PhoneShop** должен хранить коллекцию смартфонов (использовать `ArrayList<Smartphone>`) и иметь методы:

```
public void addSmartphone(Smartphone newSmartphone) - добавить новый смартфон
public Smartphone getBySerialNumber(String serialNumber) - получить смартфон по серийному номеру
public void deleteSmartphone(Smartphone smartphone) - удалить смартфон
public List<Smartphone> getAll() - получить все смартфоны, которые есть в наличии
public Map<String, Integer> getCountBySupplier() - получить карту, в которой ключ - производитель, а значение - количество смартфонов этого производителя
public List<Smartphone> getMaxPrice(int price) - получить список смартфонов, цена которых <= заданной
public void printInfo(List<Smartphone> smartphones) - распечатать информацию о списке смартфонов
```

В отдельном классе, в методе **main** необходимо добавить несколько смартфонов методом **addSmartphone** с повторяющимися производителями. Далее вызвать остальные методы в произвольной последовательности с выводом результатов в консоль (для этого использовать методы **printInfo**)

## Пример технического задания для итогового проекта:

### Т3. Социальная сеть “Инстаграм”

#### Бизнес-требование

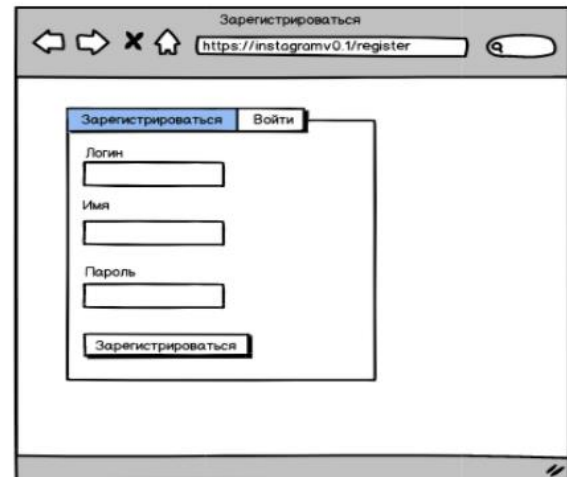
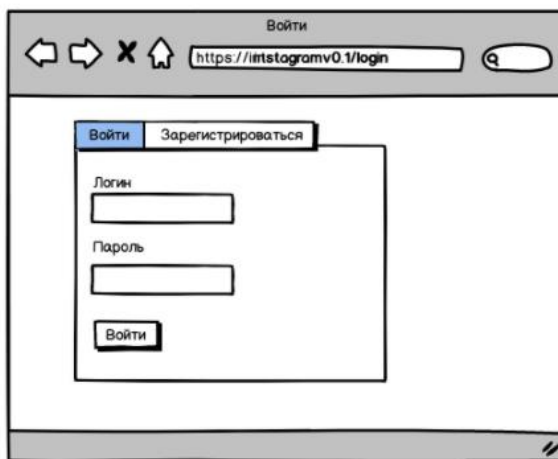
Создать социальную сеть, где пользователи могли бы делиться своими фотографиями и текстовыми заметками, а также просматривать страницы других пользователей приложения.

#### Функциональные требования

##### Страница “Вход/Регистрация”

Для входа в систему пользователю необходимо зарегистрироваться или войти в свой аккаунт (если он уже был ранее зарегистрирован).

## Макет интерфейса



## Страница "Пользовательский интерфейс"

На своей странице пользователь видит:

1. Свое имя
2. Кнопка "Загрузить пост", которая переводит пользователя на страницу "Новый Пост"
3. Кнопка "Выйти" для выхода пользователя из приложения
4. Список постов всех пользователей (текст и изображения)

## Макет интерфейса



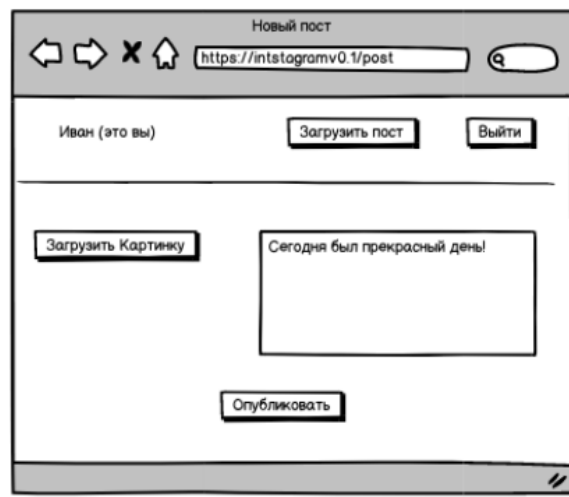
## Страница “Новый пост”

Пользователь имеет возможность опубликовать новый пост, загрузив изображение и/или оставив текстовый комментарий. Пост не может быть пустым.

Макет интерфейса

## Технологии

1. Back-End: Java + JSP
2. Front-End: HTML/CSS/JavaScript
3. СУБД - MySQL



Оценочные материалы по текущему контролю и промежуточной аттестации, предназначенные для проверки соответствия уровня подготовки по дисциплине «Программные интерфейсы и визуализация данных» планируемым результатам освоения образовательной программы (в соответствии с образовательными стандартами), хранятся на кафедре-разработчике РПД в печатном и электронном виде.

**Лист актуализации рабочей программы дисциплины  
«Программные интерфейсы и визуализация данных»**

№	Характеристика внесенных изменений (с указанием пунктов документа)	Дата и № протокола Ученого совета ЭФ	Подпись ответственного