

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ

по дисциплине **«Технологии мультимедиа в разработке интерфейсов»**
Направление подготовки 09.03.02 «Информационные системы и технологии»
Направленность (профиль) «Прикладное программирование и
интернет-технологии»
Квалификация выпускника: бакалавр

Ставрополь

2026

Введение

Цель освоения дисциплины Методические указания к выполнению лабораторных работ составлены в соответствии с рабочей программой дисциплины «Технологии мультимедиа в разработке интерфейсов» и предназначены для студентов направления подготовки 09.03.02 «Информационные системы и технологии».

В методических указаниях описаны основные элементы интерфейсов графических редакторов и их возможности, области применения и технология работы в них.

Наряду с этими сведениями, даются такие понятия, как растровая и векторная графика, цветовые модели (принцип получения различных цветов).

Основной задачей является формирование практических компетенций у студентов области, определяемой основной целью курса. Дисциплина «Технологии мультимедиа в разработке интерфейсов»

Ее освоение происходит в 7 семестре. Данная дисциплина необходима при изучении вопросов, связанных с такими понятиями как:

- технологии обработки мультимедиа-контента;
- освоение инструментов UI/UX-дизайна;
- разработка интерфейсов с использованием графики и анимации;
- интеграция мультимедиа в веб- и программные интерфейсы;
- формирование навыков проектирования пользовательского опыта.

1. Перечень планируемых результатов обучения по дисциплинам (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Код, формулировка	Код, индикатора	формулировка	Планируемые обучения	результаты по дисциплине
----------------------	--------------------	--------------	-------------------------	-----------------------------

компетенции		(модулю), характеризующие этапы формирования компетенций, индикаторов
ПК-2 способен использовать современные инструментальные средства и технологии программирования при разработке прикладного программного обеспечения вычислительных средств и систем различного функционального назначения	ПК-2ИД-2-1 Проводит классификацию и осуществляет выбор современных инструментальных средств разработки прикладного программного обеспечения вычислительных средств и систем различного функционального назначения	Результаты демонстрации знаний о правильном проведении классификации и выбор современных инструментальных средств разработки прикладного программного обеспечения вычислительных средств и систем различного функционального назначения. Представленный документ не содержит ошибок. Решены все частные задачи в полном объеме
	ПК-2ИД-2-2 Демонстрирует знания технологий программирования, разработки прикладного программного обеспечения вычислительных средств и систем различного функционального назначения	Результаты демонстрации знаний технологий программирования, разработки прикладного программного обеспечения вычислительных средств и систем различного функционального назначения. Представленный документ не содержит ошибок. Решены все частные задачи в полном объеме
	ПК-2ИД-2-3 Использует современные инструментальные средства разработки прикладного программного обеспечения вычислительных средств и систем различного функционального назначения	Результаты применения современных инструментальных средств разработки прикладного программного обеспечения вычислительных средств и систем различного функционального назначения. Представленный документ не содержит ошибок. Решены все частные задачи в полном объеме
	ПК-2ИД-2-4 Применяет технологии программирования при разработке прикладного программного обеспечения вычислительных средств и систем различного функционального назначения	Результаты применения технологий программирования при разработке прикладного программного обеспечения вычислительных средств и систем различного функционального назначения. Представленный документ не содержит ошибок. Решены все

		частные задачи в полном объеме
ПК-4 способен применять и осваивать новые средства автоматизированного проектирования, разработки, тестирования и сопровождения программного обеспечения	ПК-4ИД-4.1 Демонстрирует знания инструментальных средств для разработки приложений	Результаты демонстрации знаний инструментальных средств для разработки приложений.
	ПК-4ИД-4.2 Осуществляет выбор средств разработки, тестирования и сопровождения программного обеспечения	Результаты выбора средств разработки, тестирования и сопровождения программного обеспечения. Представленный документ не содержит ошибок. Решены все частные задачи в полном объеме
	ПК-4ИД-4.3 Применяет инструментальные средства, новые средства автоматизированного проектирования и разработки программного обеспечения,	Результаты применения инструментальных средств, новых средств автоматизированного проектирования и разработки программного обеспечения. Представленный документ не содержит ошибок. Решены все частные задачи в полном объеме
	ПК-4ИД-4.4 Осваивает новые средства автоматизированного проектирования и разработки программного обеспечения	Результаты освоения новых средств автоматизированного проектирования и разработки программного обеспечения. Представленный документ не содержит ошибок. Решены все частные задачи в полном объеме
	ПК-4ИД-4.5 Осуществляет сопровождение программного обеспечения	Результаты сопровождения программного обеспечения. Представленный документ не содержит ошибок. Решены все частные задачи в полном объеме
	ПК-4ИД-4.6 Разрабатывает и использует методики тестирования	Результаты разработки и использования методики тестирования. Представленный документ не содержит ошибок. Решены все частные задачи в полном объеме

ЛАБОРАТОРНАЯ РАБОТА № 1.

Введение в UI/UX. Понятие интерфейса, UX/UI, роль мультимедиа. Анализ интерфейсов, выявление ошибок

Цель лабораторной работы: формировать у обучающихся представление о пользовательском интерфейсе и пользовательском опыте, освоить базовые принципы UI/UX-проектирования, изучить роль мультимедийных элементов в интерфейсе и приобрести навыки анализа интерфейсов с выявлением типичных ошибок.

Задачи лабораторной работы:

- изучить понятия интерфейса, UI и UX;
- рассмотреть структуру современного цифрового интерфейса;
- понять значение мультимедиа в интерфейсных решениях;
- освоить базовые критерии оценки удобства интерфейса;
- научиться выявлять типичные ошибки в существующих интерфейсах;
- оформить результаты анализа в виде структурированного отчета.

Формируемые компетенции: ПК-2, ПК-4

Теоретическая часть

Интерфейс — это совокупность средств, способов и правил взаимодействия пользователя с программной системой, сайтом, мобильным приложением или цифровым устройством.

С практической точки зрения интерфейс отвечает на три ключевых вопроса:

1. что пользователь видит;
2. как пользователь взаимодействует с системой;
3. как система реагирует на действия пользователя.

Интерфейс является посредником между человеком и функциональностью программного продукта. Даже если система реализует сложные и полезные функции, низкое качество интерфейса может сделать продукт неудобным, непонятным и, как следствие, невостребованным.

Основные функции интерфейса:

- предоставление информации пользователю;
- обеспечение управления системой;
- обратная связь на действия пользователя;
- поддержка логики взаимодействия;
- снижение вероятности ошибок;
- повышение скорости и удобства работы.

UI (User Interface) — это пользовательский интерфейс как визуальная и интерактивная оболочка цифрового продукта.

UI включает: кнопки; поля ввода; меню; иконки; списки; карточки; шрифты; цвета; отступы; сетки; анимации; визуальные состояния элементов.

Иными словами, UI — это то, как интерфейс выглядит и какие визуальные средства используются для взаимодействия.

Задачи UI:

- сделать интерфейс визуально понятным;
- обеспечить логичную структуру элементов;
- создать единый визуальный стиль;
- выделить важные элементы;
- повысить читаемость и предсказуемость взаимодействия.

UX (User Experience) — это пользовательский опыт, то есть совокупность впечатлений, ощущений и результатов, которые получает пользователь при взаимодействии с продуктом.

UX отвечает не только на вопрос «красиво ли выглядит интерфейс», но и на вопрос «удобно ли им пользоваться».

Пример различия UI и UX

Предположим, что мобильное приложение имеет красивый современный дизайн, аккуратные кнопки и приятную цветовую палитру. Это хороший UI. Но если пользователь не может найти кнопку оформления заказа, постоянно ошибается в форме оплаты и не понимает, завершилось ли действие, то UX плохой.

Следовательно:

UI — это визуальное оформление и интерактивные элементы;

UX — это качество пользовательского опыта при работе с интерфейсом.

Мультимедиа в интерфейсе — это использование различных форм представления информации, включая:

1. текст;
2. графику;
3. иконки;
4. иллюстрации;
5. анимацию;
6. аудио;
7. видео;
8. интерактивные визуальные эффекты.

Зачем мультимедиа нужны в интерфейсах

1. Повышение наглядности

Изображения, схемы и иконки позволяют быстрее передать смысл, чем длинные текстовые пояснения.

2. Поддержка восприятия информации

Анимация может показать изменение состояния элемента, последовательность действий или результат операции.

3. Формирование визуальной иерархии

С помощью цвета, размера, движения и контраста можно управлять вниманием пользователя.

4. Эмоциональное воздействие

Качественные мультимедийные элементы создают ощущение современности, надежности и профессионализма.

5. Обратная связь

Звуковые сигналы, микровзаимодействия, анимация кнопок и уведомления помогают пользователю понимать, что система отреагировала на его действие.

Типичные ошибки интерфейсов

Визуальные ошибки

- слишком много цветов;
- низкий контраст текста и фона;
- слишком мелкий шрифт;
- отсутствие визуальной иерархии;
- перегруженность экрана элементами.

Логические ошибки

- непонятная навигация;
- отсутствие явных сценариев действий;
- скрытые важные функции;
- неочевидные иконки;
- сложные многошаговые действия без подсказок.

Ошибки обратной связи

- система не сообщает о результате действия;
- отсутствуют сообщения об ошибках;
- неясно, загрузились ли данные;
- пользователь не понимает, произошло ли сохранение.

Ошибки мультимедиа

- декоративная анимация мешает работе;
- видео запускается автоматически без необходимости;
- звук включается неожиданно;
- изображения слишком тяжелые и замедляют загрузку;
- мультимедиа используются без функционального смысла.

Ошибки доступности

- недостаточный размер активных элементов;
- отсутствие альтернативного текста для изображений;
- использование цвета как единственного средства передачи смысла;
- трудночитаемые шрифты;
- сложное взаимодействие только мышью без клавиатурной навигации.

Этапы анализа интерфейса

Этап 1. Определение объекта анализа

Необходимо выбрать цифровой продукт: сайт, веб-сервис, мобильное приложение, личный кабинет, интернет-магазин, образовательную платформу.

Этап 2. Выделение пользовательской цели

Например:

- зарегистрироваться;
- найти товар;
- оформить заявку;
- пройти тест;
- загрузить файл.

Этап 3. Анализ сценария

Следует пройти путь пользователя шаг за шагом и зафиксировать возникающие трудности.

Этап 4. Выявление ошибок

Ошибки фиксируются по категориям:

- визуальные;
- логические;
- навигационные;
- мультимедийные;
- ошибки обратной связи.

Этап 5. Формулирование рекомендаций

Каждая найденная проблема должна сопровождаться предложением по улучшению.

Методика и порядок выполнения работы

Часть 1. Концептуальный разбор

Письменно ответьте на вопросы в файле:

1. В чем ключевая разница между сломанным UI (кнопка некрасивая) и плохим UX (кнопка ведет не туда, куда ожидалось)?

Приведите 2 примера, где звук в интерфейсе помогает пользователю (а не раздражает).

2. Зачем нужна анимация при переходах между экранами мобильного приложения? (Назовите 2 причины).

Часть 2. Анализ интерфейсов

Задание: Вам выданы скриншоты 3-х интерфейсов (варианты А, Б, В). Либо откройте указанные преподавателем сайты/приложения.

Вариант А: Форма регистрации сложного банка.

Вариант Б: Панель администратора (dashboard) с 50+ элементами.

Вариант В: Интерфейс онлайн-кинотеатра на смартфоне.

Для каждого интерфейса заполните таблицу:

Критерий	Оценка (да/нет/частично)	Комментарий (где именно ошибка)
Ясность цели экрана		
Логичная группировка элементов		
Достаточная контрастность текста		
Понятные иконки (без подписи)		
Наличие быстрой обратной связи		
Адаптив (для веба) / Жесты (для мобильных)		

Выявите не менее 3-х конкретных UI-ошибок и 2-х UX-ошибок для каждого интерфейса.

Пример записи:

Ошибка 1 (UI): Кнопка «Отправить» имеет цвет фона такой же, как у неактивного поля ввода. Пользователь не понимает, кликабельная она или нет.

Ошибка 2 (UX): После нажатия «Купить» нет экрана подтверждения заказа. Товар сразу списывается с карты — это вызывает тревогу.

Часть 3. Роль мультимедиа

Выберите один популярный интерфейс на выбор (RuTube, Spotify, Figma, Duolingo, онлайн-банк).

Опишите, как в нем используются мультимедиа:

Графика/UI-киты: (пример: в Telegram кастомные анимированные стикеры вместо эмодзи).

Анимация: (пример: в Duolingo при правильном ответе персонаж прыгает и сыплются конфетти — эмитация награды).

Звук: (пример: в Spotify при переключении трека — легкий щелчок или fade out/in).

Видео: (пример: в YouTube Shorts автоматическое воспроизведение без звука — для привлечения).

Сделайте вывод: Улучшает ли конкретное мультимедиа в этом продукте удержание пользователя или только визуальный шум? Аргументируйте.

ЛАБОРАТОРНАЯ РАБОТА № 2

Обработка растровых изображений в Photoshop онлайн

Цель лабораторной работы: научиться обрабатывать растровые изображения в Photoshop онлайн.

Задачи лабораторной работы:

- изучить функционал редактора Photoshop онлайн;
- научиться создавать коллажи в редакторе Photoshop онлайн.

Формируемые компетенции: ПК-2, ПК-4

Теоретическая часть

В интернет существуют онлайн редакторы для создания и обработки растровых изображений.

Онлайн редакторы растровых изображений имеют общий алгоритм работы: загрузить/создать изображение, отредактировать в редакторе, скачать обработанный файл.

Рассмотрим несколько примеров обработки растровых изображений в онлайн редакторе.

1. Откройте онлайн редактор «Фотошоп онлайн на русском»
2. Щелкните по ссылке **Photoshop Online** (рис. 1).

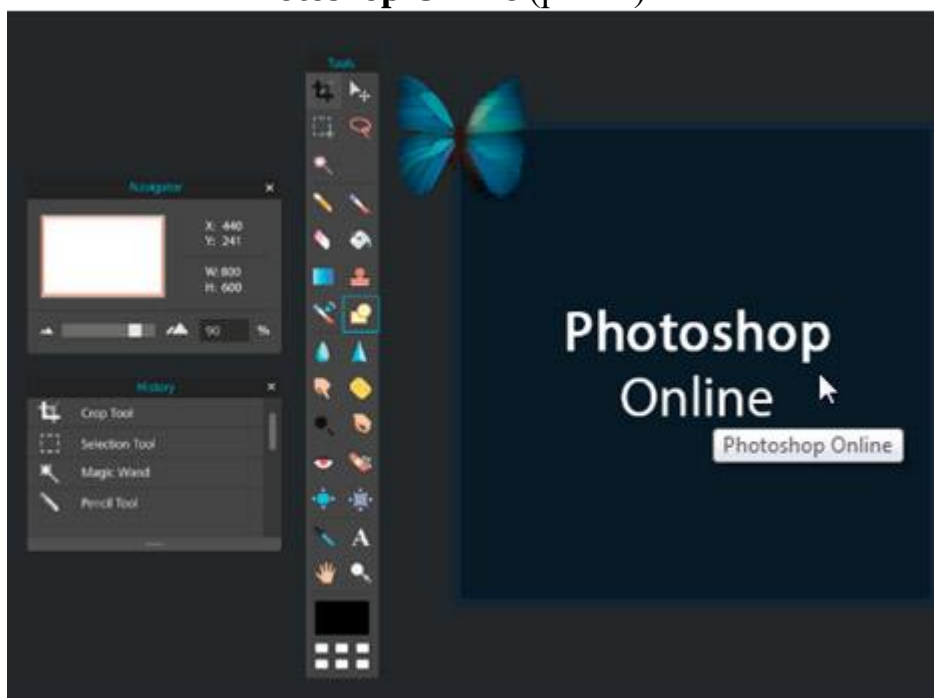


Рисунок 1

Если появилось окно, как на рисунке 2, закройте его.



Рисунок 2

Основы редактирования фотографий

Рассмотрим функции, которые понадобятся Вам для обработки готовых изображений.

Перечислим эти функции:

- изменение размеров изображения;
- поворот изображения;
- кадрирование изображения;
- цветокоррекция изображения;
- сохранение изображения.

Изменение размеров изображения

3. Командой **Файл – Открыть изображение...** откройте в Photoshop Online файл **butterfly.jpg** из папки **photo_dop_1**.



Рисунок 3

4. Выполните команду **Изображение – Размер изображения...**

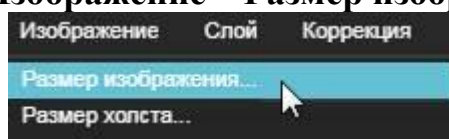


Рисунок 4

Появится окно настроек.

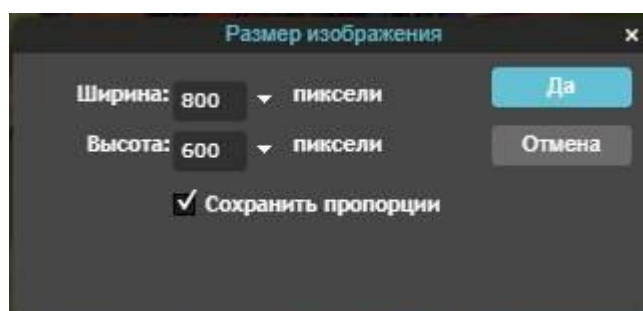


Рисунок 5

5. Щелкните по треугольнику в поле **Ширина**. Откроется бегунок. Перемещая бегунок выберите другой размер ширины в пикселях и нажмите кнопку **Да**.

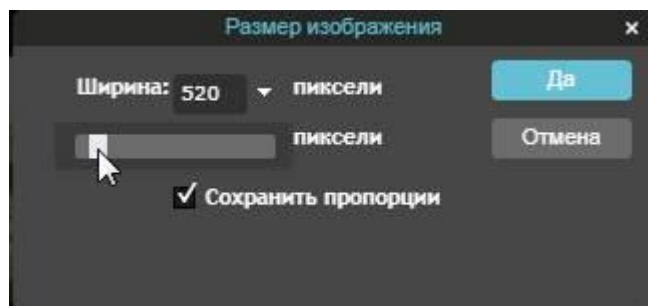


Рисунок 6

Размер изображения уменьшился.

6. Чтобы отменить это действие, найдите палитру **Журнал** в правой части экрана.

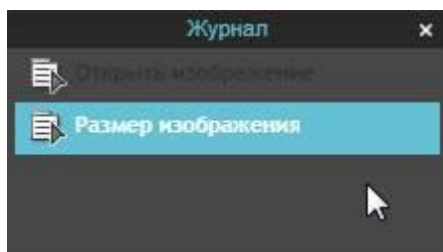


Рисунок 7

7. Щелкните по строке **Открыть изображение**. Таким образом Вы отменили последнее действие. Изображение опять приняло первоначальные размеры.

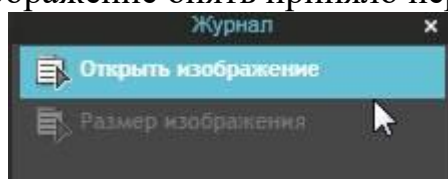


Рисунок 8

8. Вновь щелкните по строке **Размер изображения**. Таким образом Вы вернули отмененное действие. Используйте палитру **Журнал** для отмены и возврата действий.

9. Вернитесь к первоначальным размерам изображения, отменив последнее действие.

Поворот изображения

10. Выполните команду **Изображение – Повернуть на 180°**. Изображение перевернулось.

11. Выполните команду **Изображение – Повернуть на 90° по часовой стрелке**. Выполните команду **Изображение – Отразить по вертикали**.

И так далее. Используйте возможности поворота для Ваших файлов изображений.

12. Поверните изображение с бабочкой так, чтобы оно приняло исходный вид.

Кадрирование изображения

13. В левой панели инструментов выберите инструмент **Обрезка**.

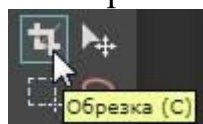


Рисунок 9

14. Обведите этим инструментом на фотографии зону, которая останется после обрезки изображения.



Рисунок 10

После того, как Вы отпустите кнопку мыши, на выделенной зоне появится сетка.

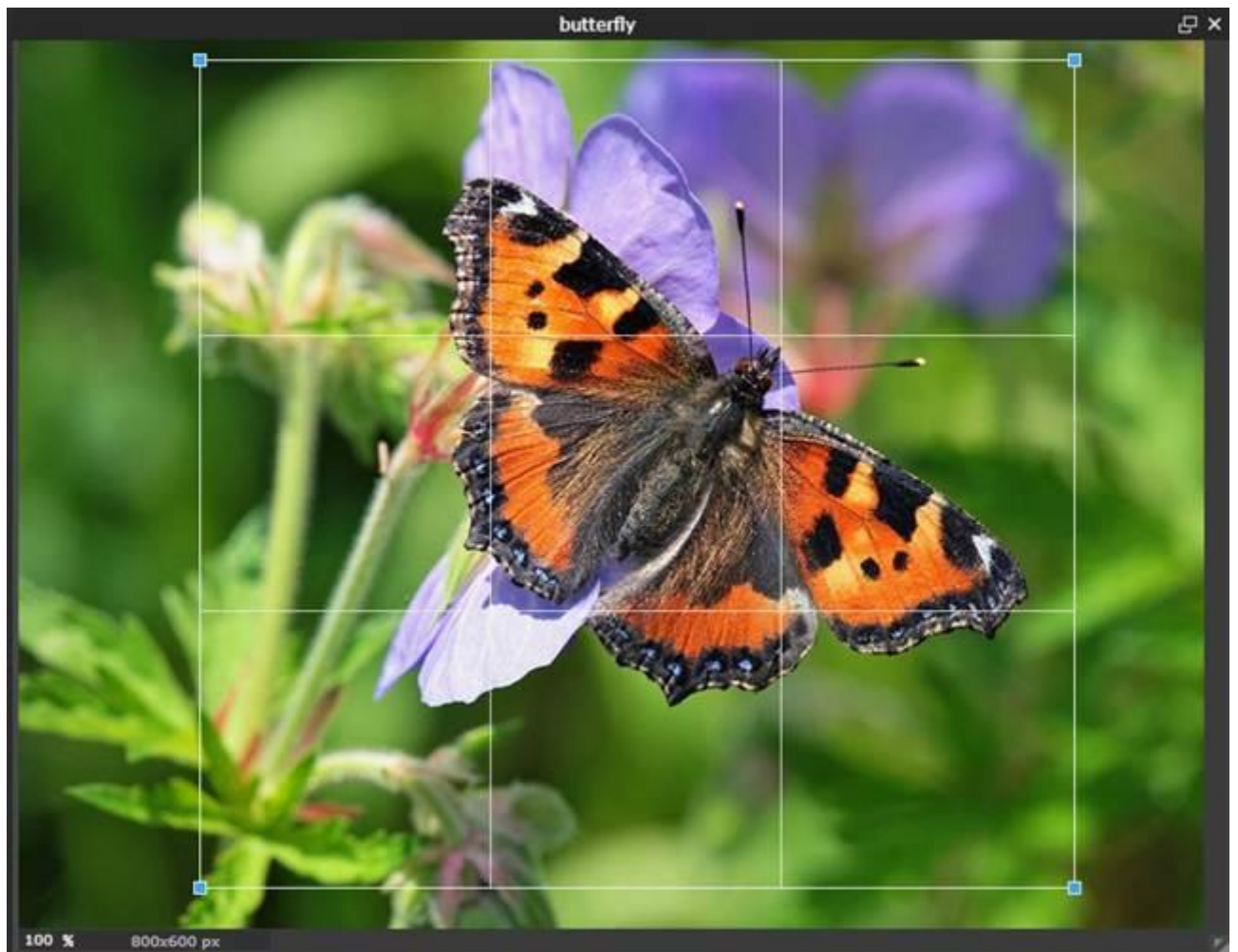


Рисунок 11

Перемещая угловые маркеры, можно отредактировать зону выделения.



Рисунок 12

15. После этого нажмите **Enter**. Изображение обрезаюсь.



Рисунок 13

16. Отмените все действия и верните изображение в исходное состояние.

<

Цветокоррекция изображения

Для цветокоррекции изображения есть группа инструментов в верхнем пункте меню **Коррекция**.

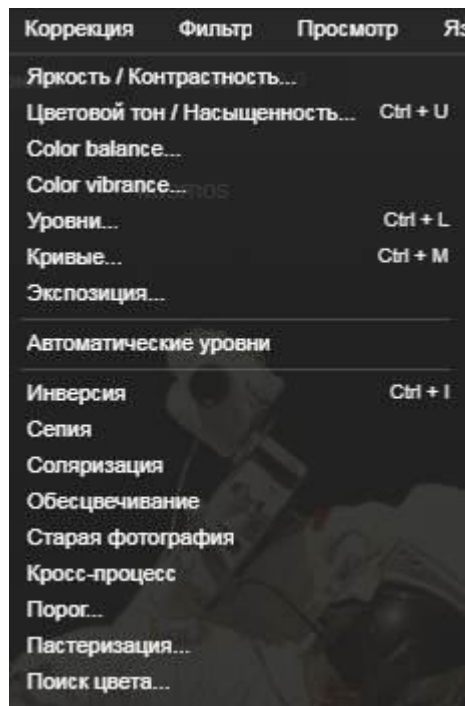


Рисунок 14

17. Попробуйте самостоятельно все эти пункты меню. Мы применили опцию **Старая фотография** дважды.



Рисунок 15

Сохранение изображения

18. Чтобы сохранить отредактированное изображение у себя на компьютере, выполните команду **Файл – Сохранить**.

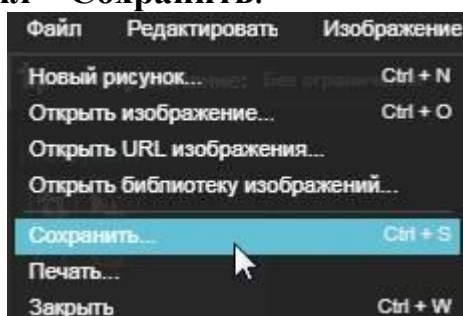


Рисунок 16

Откроется окно с настройками сохранения.

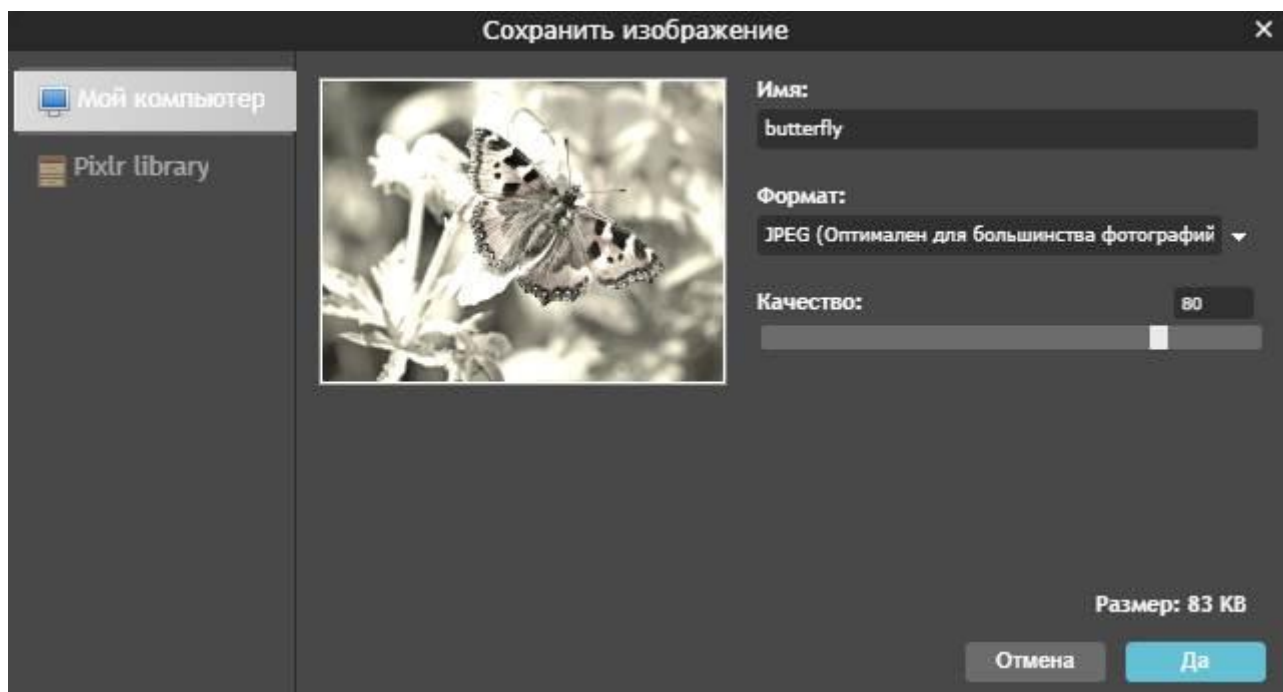


Рисунок 17

В этом окне можно задать имя файла, выбрать формат, и качество сохранения.

Если бегунок **Качество** установить в крайнее правой положение, то будет 100%-е качество и максимальный размер файла в Кб (или Мб). Если выбрать качество пониже, то размер файла в Кб уменьшится, но качество изображения понизится. В правом нижнем углу этого окна можно видеть размер изображения в зависимости от положения бегунка **Качество**.

19. Нажмите кнопку **Да**. Откроется окно с файловой системой Вашего компьютера. Выберите нужную папку и нажмите кнопку **Сохранить**.

Совмещение фотографий

1. Откройте файлы **moln.jpg** и **flower.jpg**. Для этого выберите команду **Файл – Открыть изображение**.

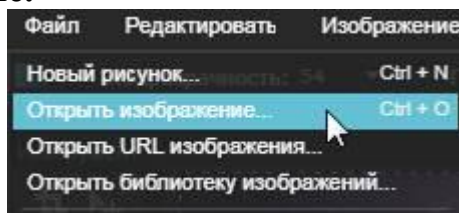


Рисунок 18

2. Найдите эти файлы на своем компьютере. Выделите первый файл и нажмите **Открыть**. Аналогично откройте второй файл. Расположите их рядом.

Примечание: чтобы уменьшить масштаб отображения фотографии, выберите инструмент лупы  в левой панели инструментов и щелкните по изображению с нажатой клавишей Shift.



Рисунок 19

3. Сделайте активным файл **moln.jpg** (щелкните по верхней черной полосе, где написано «moln»).
4. Выполните команду **Редактировать – Выделить все**. Вокруг изображения появилась серая пунктирная линия.



Рисунок 20

5. Выполните команду **Редактировать – Копировать**.
6. Переключитесь на файл **flower.jpg** и выполните команду **Редактировать – Вставить**.
7. В палитре **Слои** переименуйте слои, как показано на рисунке. Для этого щелкните дважды по названию слоя и в режиме редактирования имени слоя введите новое название.

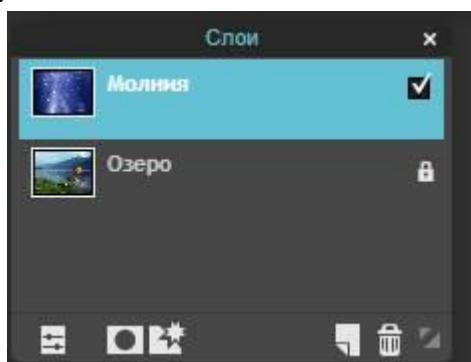


Рисунок 21

8. Сделайте активным слой **Молния**, он подсветится голубым цветом.
9. В палитре слоев внизу нажмите кнопку **Маска слоя**

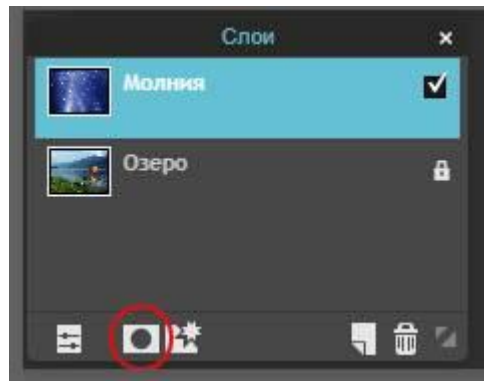


Рисунок 22

Изображение грозы исчезло, но на самом деле слой остался, он просто стал прозрачным. Рядом с пиктограммой грозы в палитре слоев появилась пиктограмма маски слоя с белым фоном и красной рамкой.

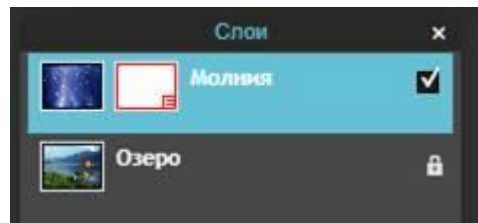


Рисунок 23

10. В левой панели инструментов найдите инструмент градиента , щелкните по нему левой кнопкой мыши.

11. Убедитесь, что настройки градиента вверху экрана совпадают с рисунком.



Рисунок 24

12. Проведите градиентом в центральной части изображения сверху вниз.



Рисунок 25

Примерный результат приведен на рисунке ниже. Слой с грозой стал полупрозрачным. Получился эффект совмещения двух фотографий.



Рисунок 26

13. Щелкните правой кнопкой мыши по слою **Молния** и выберите команду **Применить маску слоя**.

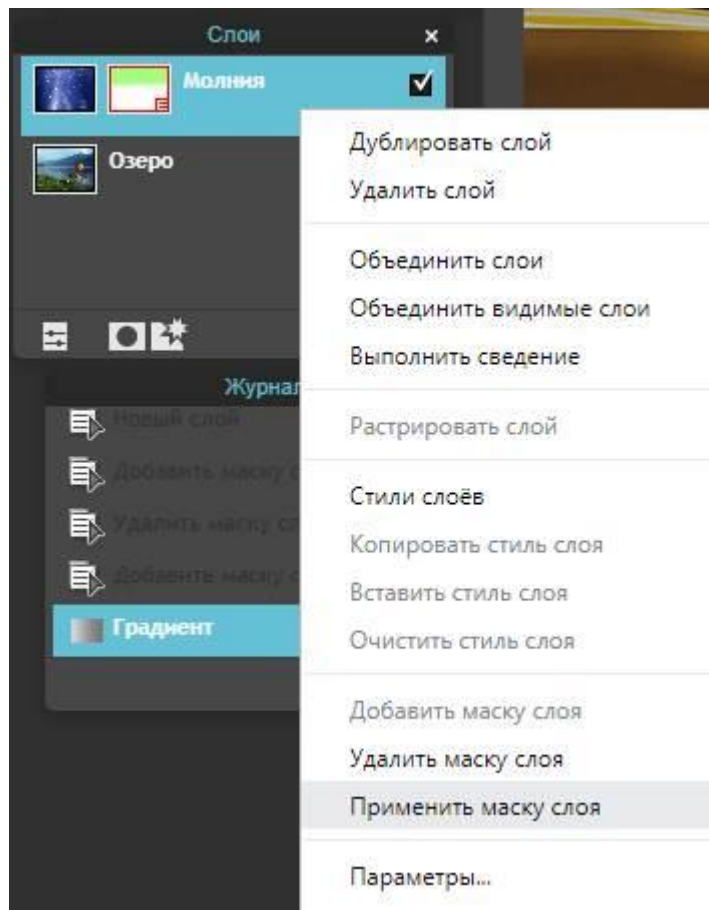


Рисунок 27

14. Выполните команду **Слой – Объединить слои**.

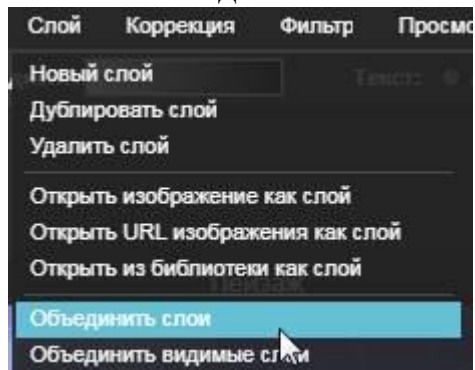


Рисунок 28

15. Выберите команду **Файл – Сохранить**.

В окне сохранения выполните настройки, как показано на рисунке и нажмите кнопку **Да**.



Рисунок 29

16. Выберите папку, в которую сохраняете свои работы. Нажмите кнопку **Сохранить**.

17. Файлы **moln.jpg** и **flower.jpg** закройте без сохранения.

18. Просмотрите вновь созданный файл **Гроза на озере.jpg**.

Фильтры и цветокоррекция

19. Откройте в Photoshop Онлайн файл **Park.jpg**.



Рисунок 30

20. Выполните команду **Фильтр – Волшебное сияние**. Результат на рисунке ниже.



Рисунок 31

21. Сделаем цветокоррекцию фотографии. Для этого выполните команду **Коррекция – Цветовой тон/Насыщенность**.

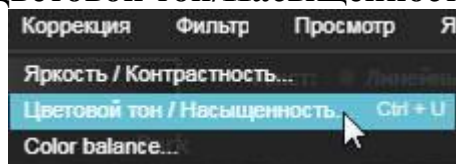


Рисунок 32

22. В появившемся окне **Цветовой тон/Насыщенность** установите флажок **Колоризация**.

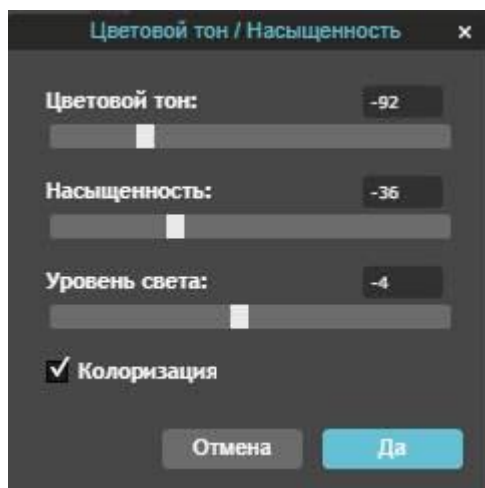


Рисунок 33

23. Перемещая бегунки просмотрите действие этого фильтра. Выберите понравившийся Вам вариант. Нажмите **Да**. У нас получилось вот так.



Рисунок 34

24. Выполните команду **Коррекция – Color balance..** Перемещая бегунки просмотрите действие этого фильтра.

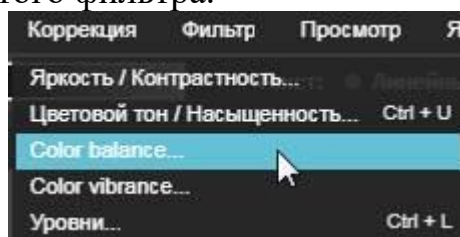


Рисунок 35

25. Выберите понравившийся Вам вариант. Нажмите **Да**.



Рисунок 36

26. Результат ниже



Рисунок 37

27. Сохраните файл на своем компьютере.

Методика и порядок выполнения работы

Из нескольких изображений создадим коллаж, как показано на рисунке 1. Для этого найдите и скачайте изображения подобной тематики и выполните все этапы описанные ниже.



Рисунок 1. Фотографии взяты с сайта quist.pro из книги
«АВСТРАЛИЯ. Путешествие в страну OZ»

1. Откройте онлайн редактор «Фотошоп онлайн на русском» по адресу <https://photoshop-online.biz/>.
2. Щелкните по ссылке **Photoshop Online** (рис. 2).

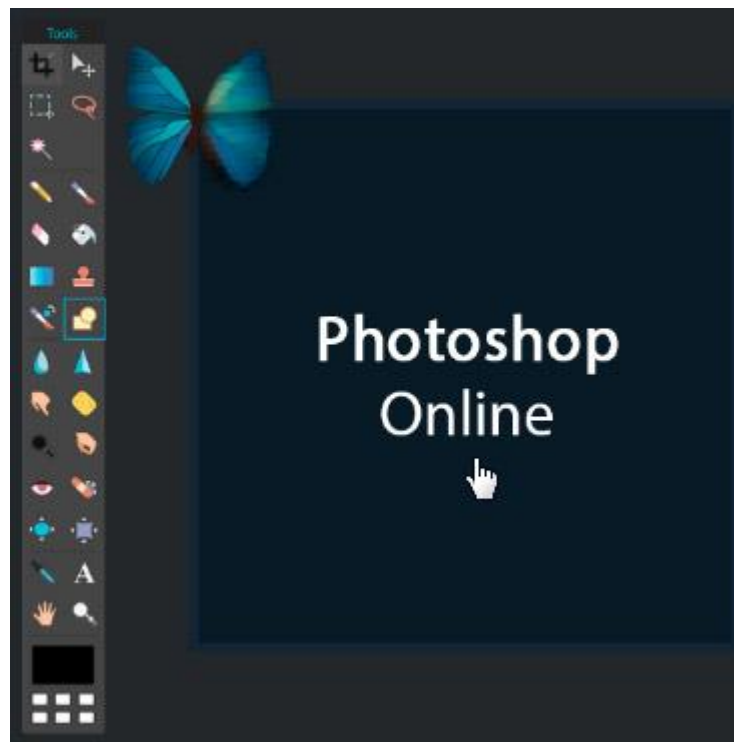


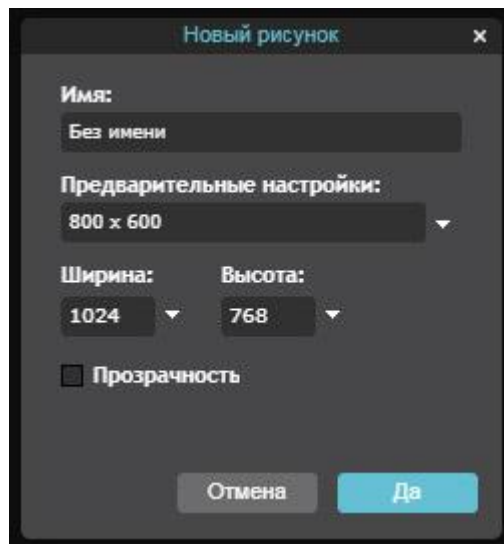
Рисунок 2

3. Если появилось окно, как на рисунке 3, закройте его.

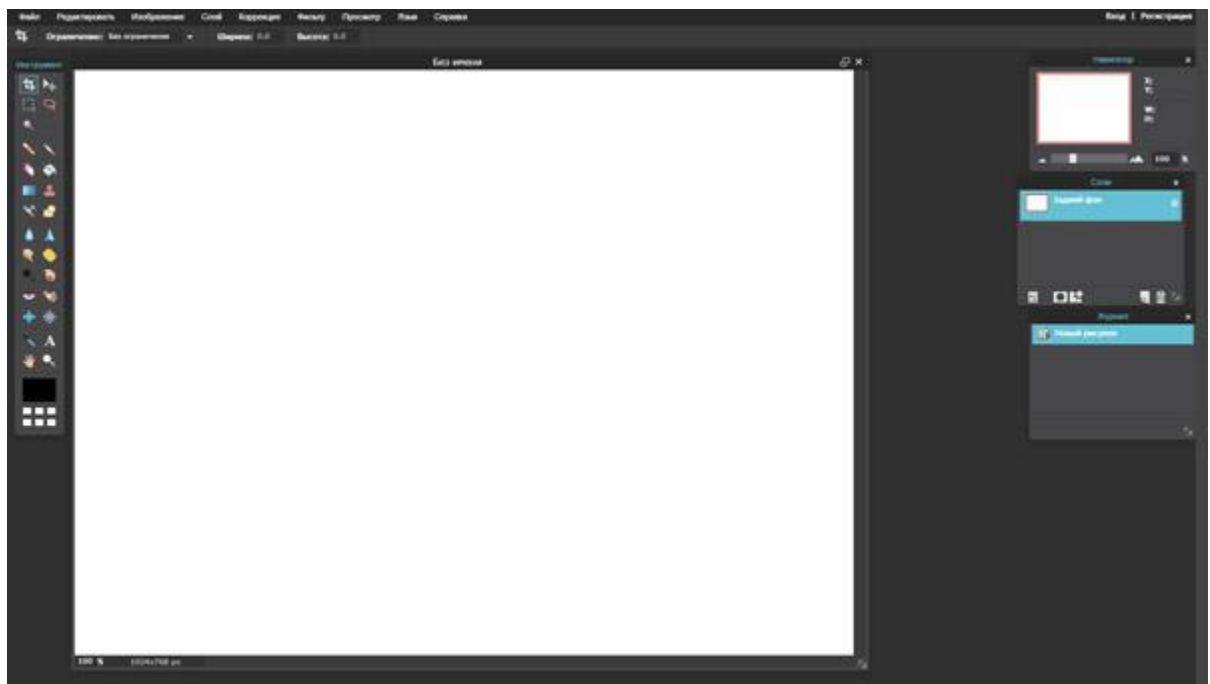


Рисунок 3

4. Создайте новый файл командой **Файл – Новый рисунок...** В окне сделайте настройки, как на рисунке ниже. Потом нажмите **Да**.



5. В результате должен появиться чистый холст.



6. Командой **Файл – Открыть изображение...** откройте изображение **Оперный театр.jpg**. Расположите два окна рядом. Ссылка для скачивания изображений для коллажа расположена в начале урока.

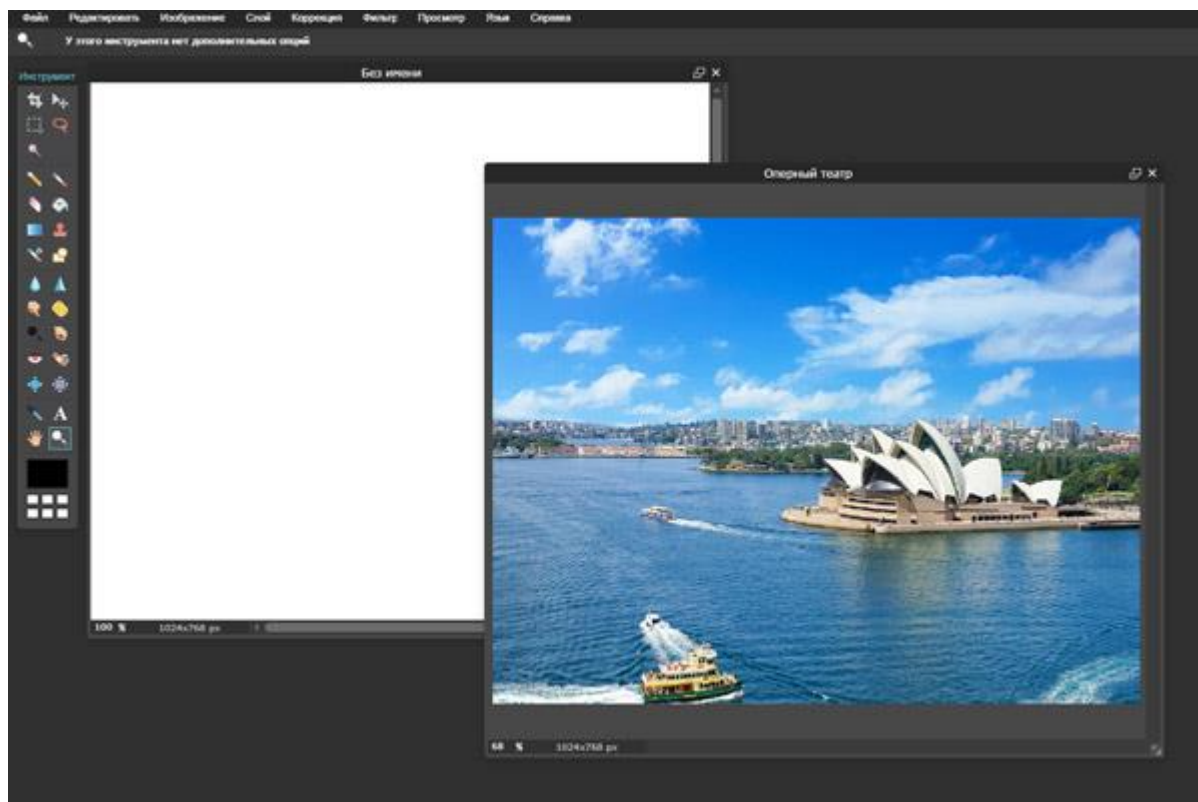


Рисунок 6

7. Убедитесь, что изображение с оперным театром активно (щелкните по «шапке» этого окна). Выполните команду **Редактировать – Выделить все**, вокруг изображения появилась пунктирная линия выделения. Затем выполните команду **Редактировать – Копировать**.
8. Переключитесь на чистый лист (щелкните по «шапке» документа) и выполните команду **Редактировать – Вставить**.



Рисунок 7

9. Закройте файл **Оперный театр.jpg** без сохранения.

10. После того, как Вы переместили изображение театра на чистый холст будущего коллажа, в этом файле образовалось два слоя. Их видно в палитре **Слой**.

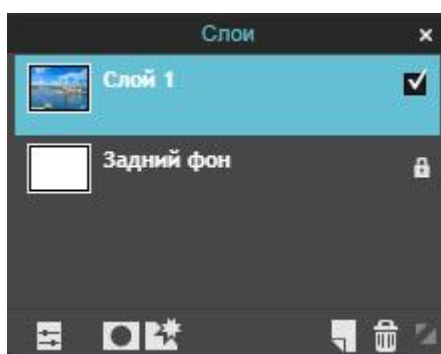


Рисунок 8

11. Если у Вас в окне программы нет этой палитры, выполните команду **Просмотр – Слой**.

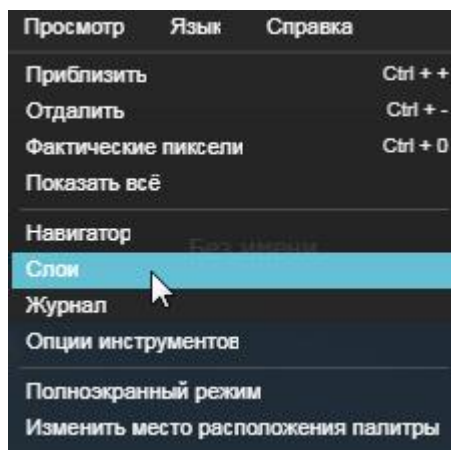


Рисунок 9

12. В палитре **Слой** дважды щелкните по слову **Слой 1** и напишите **Оперный театр**. Таким образом, Вы задали слою новое имя.

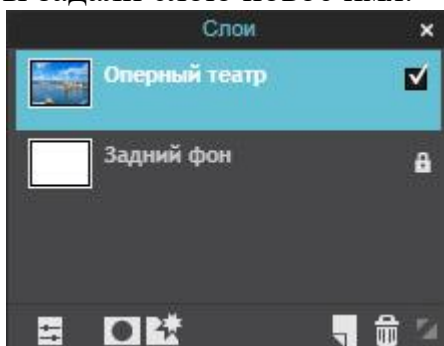


Рисунок 10

13. Выберите инструмент **Лупа** , он находится в нижней части левой панели инструментов.

Лупа имеет два режима: увеличение  и уменьшение . По умолчанию включен режим увеличения, чтобы перейти в режим уменьшения надо держать нажатой клавишу **Shift**.

14. В режиме увеличения  щелкните левой кнопкой мыши по изображению, оно увеличилось.

15. Держите нажатой клавишу **Shift** – значок лупы изменился на режим уменьшения , щелкните по изображению, оно уменьшилось.

16. Найдите в окне программы палитру **Журнал**. Если ее нет, выполните команду **Просмотр – Журнал**.

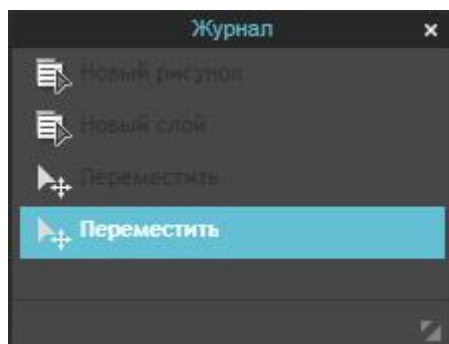


Рисунок 11

17. В этой палитре можно увидеть все Ваши действия с файлом. Она может выглядеть по-разному, в зависимости от того, какие действия Вы выполняли. В этой панели можно отменять действия и возвращать их. Последнее действие выделено голубым цветом. Щелкните на одну строку выше. Таким образом, Вы отменили последнее действие. Если щелкнуть на несколько строк выше, то Вы отмените несколько последних действий.

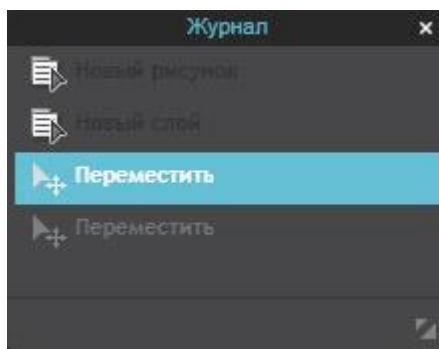


Рисунок 12

18. Снова щелкните по последней строке, таким образом, Вы вернули отмененное действие. Используйте эту панель для отмены и возврата действий.

19. Откройте файл **Сидней.jpg**.

20. Уменьшим размер изображения, сделаем его таким же по ширине, как и коллаж. Выполните команду **Изображение – Размер изображения...** и в открывшемся окне сделайте настройки, как показано на рисунке. Затем нажмите **Да**.

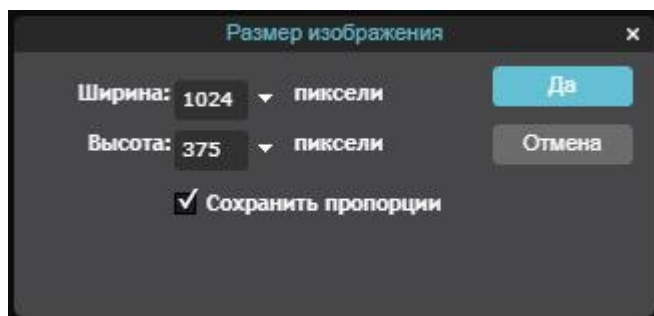


Рисунок 13

21. Теперь нам необходимо скопировать изображение Сиднея в наш коллаж. Для этого выполните команды **Редактировать – Выделить все**, а затем **Редактировать – Копировать**.

22. Переключитесь на наш коллаж и выполните команду **Редактировать – Вставить**. Результат на рисунке ниже.

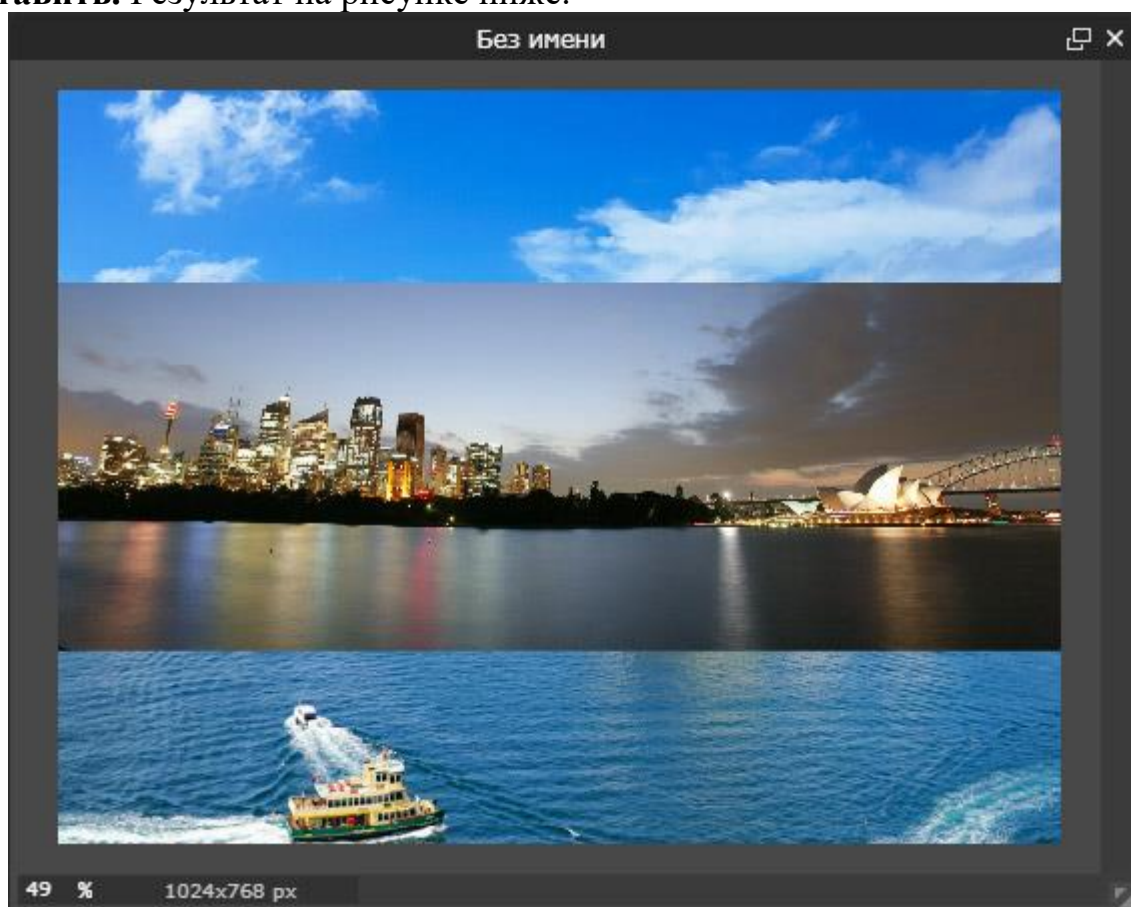


Рисунок 14

23. Переместите изображение Сиднея в верхнюю часть коллажа. Для этого на панели инструментов слева выберите инструмент перемещения , и с нажатой левой клавишей мыши переместите изображение города, как показано на рисунке.



Рисунок 15

24. В палитре слоев переименуйте вновь образовавшийся слой в **Сидней**.

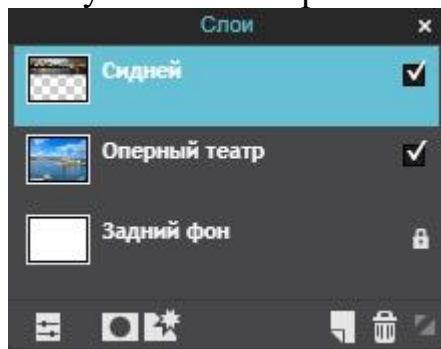


Рисунок 16

25. Видимость слоев можно включать и отключать. Для отключения видимости слоя надо сбросить флажок в правой части строки с названием слоя . Отключите слой **Сидней**, щелкнув левой кнопкой мыши по флажку. Изображение города исчезло, но в файле осталось. Этот слой просто теперь не видим.

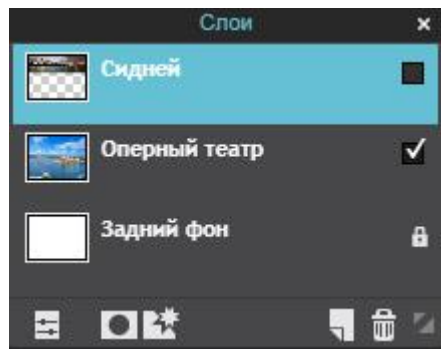


Рисунок 17

26. Включите обратно слой **Сидней**, чтобы он был видим.

27. Обрежем часть изображения города, чтобы она так много не перекрывала небо над оперным театром. Для этого выберите на левой панели инструмент прямоугольного выделения  и обведите прямоугольную область, выделив нижнюю половину воды на панораме города.



Рисунок 18

28. Нажмите клавишу **Delete**. Результат ниже.



Рисунок 19

29. Закройте файл **Сидней.jpg** без сохранения.

30. Теперь поработаем с нижней частью коллажа (см. рис. 1). Откройте изображения **Рельеф.jpg**, **Кинг-каньон.jpg** и **Дюны.jpg**.

31. Установите высоту каждому из этих изображений равной **220 px**. Как Вы помните из предыдущих действий, сделать это можно командой **Изображение – Размер изображения...** В появившемся окне в поле **Высота** установить цифру **220**. Ширина пересчитается автоматически и для каждого из изображений будет разной. После этого нажать **Да**.

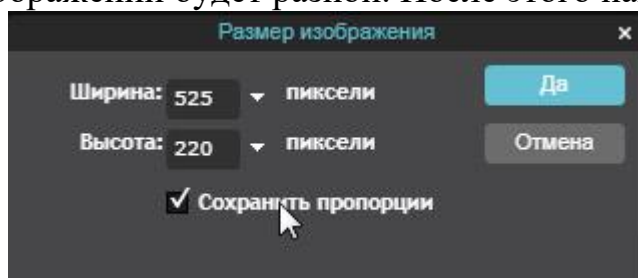


Рисунок 20

32. Скопируйте изображение из файла **Рельеф.jpg** в наш коллаж. Как Вы помните из предыдущих действий для этого надо выполнить команду **Редактировать – Выделить все**, затем **Редактировать – Копировать**. После этого надо переключиться на коллаж и выполнить команду **Редактировать – Вставить**.

33. После вставки инструментом перемещения  разместите изображение рельефа в левом нижнем углу коллажа.



Рисунок 21

34. Новый слой переименуйте в **Рельеф**.

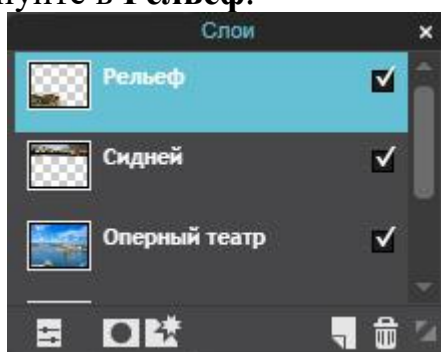


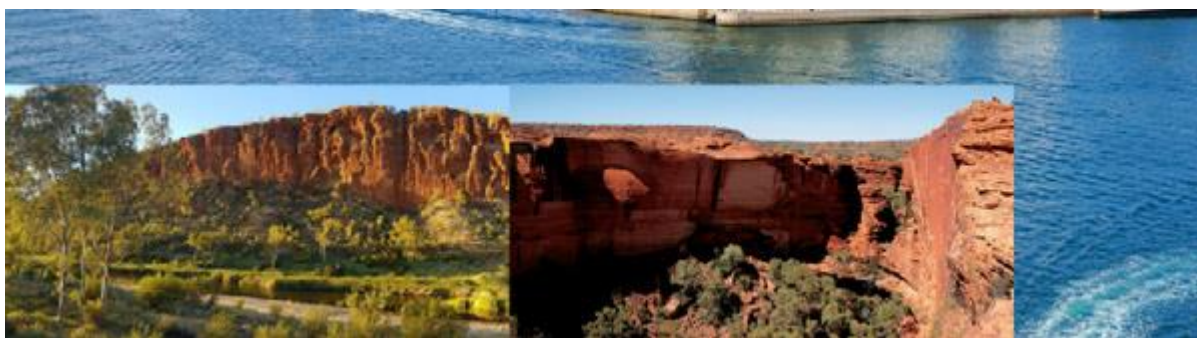
Рисунок 22

35. Закройте файл **Рельеф.jpg** без сохранения.

36. Аналогичным образом в коллаж поместите изображение с каньоном из файла **Кинг-каньон.jpg**. Слой переименуйте в **Каньон**. Разместите изображения с перекрытием друг друга, как показано на рисунке а, затем сровняйте верхние края изображений (рис. б) инструментом перемещения или клавишами со стрелками (инструмент перемещения при этом должен быть включен).



а



б

Рисунок 23

37. Сделаем плавный переход изображения с рельефом в изображение с каньоном. Выделите слой **Каньон** в палитре **Слои**. Он должен окраситься в голубой цвет.

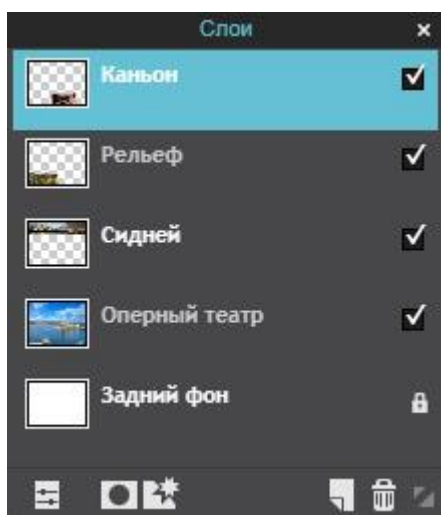


Рисунок 24

38. В нижней части палитры **Слои** нажмите кнопку **Добавить маску слоя** . На слое появилась белая пиктограмма маски слоя, а сам слой в коллаже исчез. На самом деле его просто не видно.

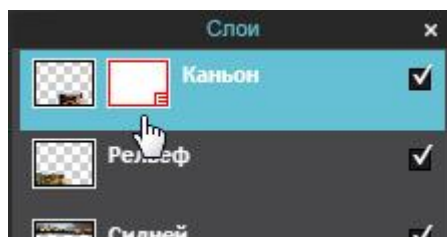


Рисунок 25

39. В левой панели инструментов выберите инструмент **Градиент** . Установите настройки, как показано на рисунке ниже.



Рисунок 26

40. Проведите инструментом градиента слева направо в зоне перекрытия слоев **Рельеф** и **Каньон** (см. рисунок ниже).

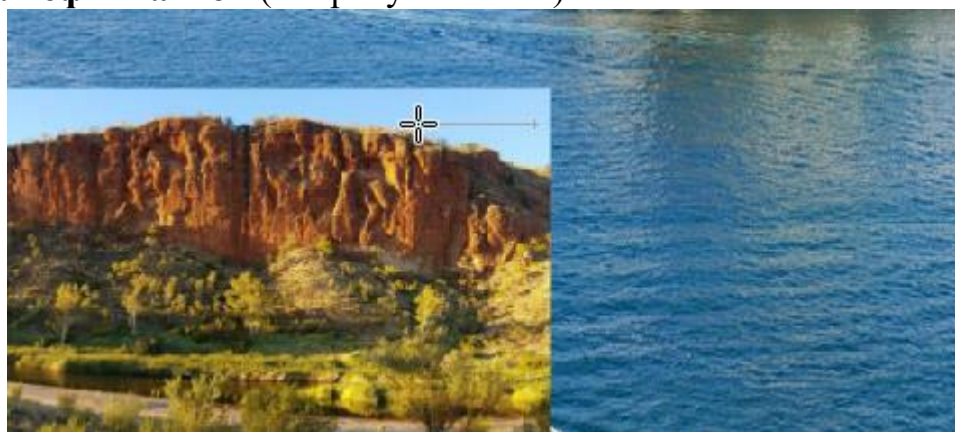


Рисунок 27

41. В результате получился плавный переход одного слоя в другой.



Рисунок 28

42. Щелкните правой кнопкой мыши по слою **Каньон** и выберите команду **Применить маску слоя**.

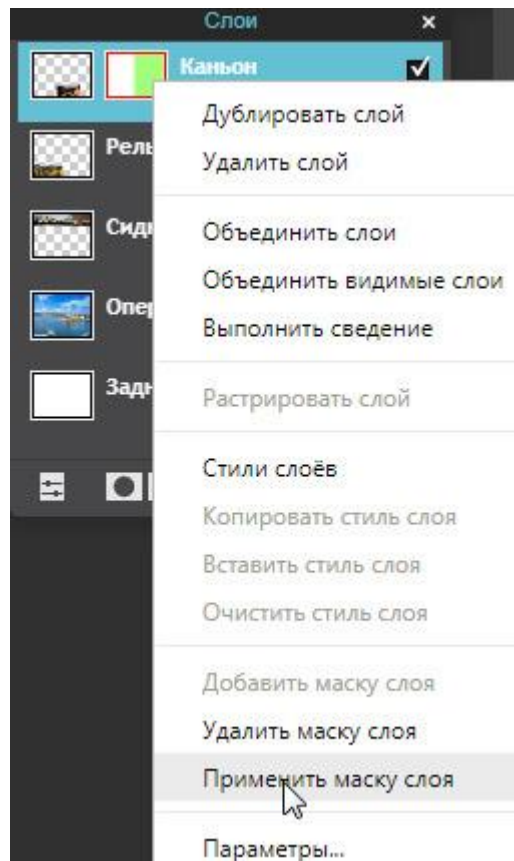


Рисунок 29

43. Разместите в правом нижнем углу коллажа изображение из файла **Дюны.jpg** с перекрытием слоя **Каньон**. Новый слой назовите **Дюны**.



Рисунок 30

44. Аналогично, как Вы делали плавный переход слоя **Каньон** в слой **Рельеф**, сделайте плавный переход слоя **Дюны** в слой **Каньон**. У нас получилось вот так.



Рисунок 31

45. Выполните команду **Применить маску слоя** для слоя **Дюны**.

46. Откройте файл **Великая Океанская дорога.jpg**. Измените высоту изображения до **300px**. Разместите это изображение в коллаже. Слой назовите **Дорога**. Результат на рисунке ниже.



Рисунок 32

47. Изображение Великой Океанской дороги получилось немного большим и перекрывает оперный театр. Уменьшим размеры слоя **Дорога**. Для этого выполните команду **Редактировать – Free transform...** Появится рамка трансформации (контуры рамки могут не совпадать с границами изображения и находиться немного в стороне). Перемещая угловые маркеры, уменьшите размеры изображения дороги, держите при этом нажатой клавишу **Shift**,

чтобы не исказить пропорции. После того, как уменьшили размеры, нажмите **Enter**.



Рисунок 33

48. Разместите слой **Дорога** примерно так, как показано на рисунке ниже.



Рисунок 34

49. Смягчим края слоя **Дорога**. Для этого выберите инструмент **Ластик** . На верхней панели параметров выберите режим ластика **Кисть** с диаметром **100**.

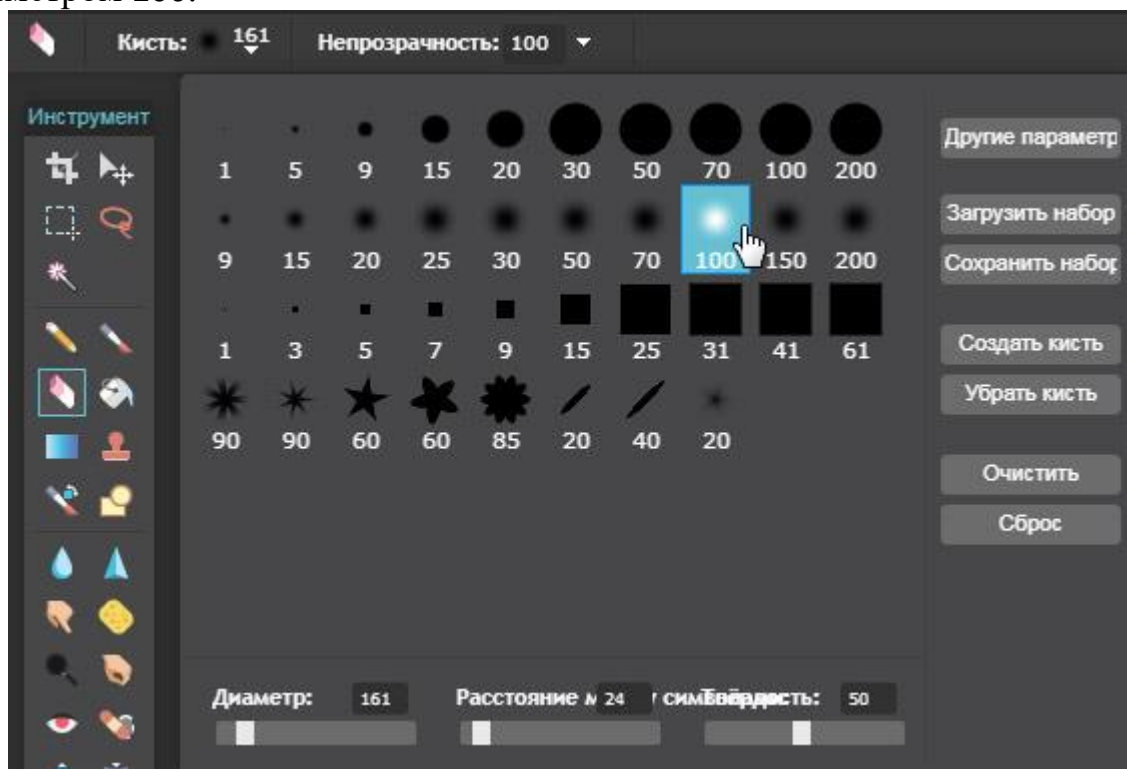


Рисунок 35

50. Аккуратно сотрите резкий край изображения.



Рисунок 36

51. Результат на рисунке ниже.



Рисунок 37

52. Откройте файл **Город.jpg**. Уменьшите высоту изображения до **180 px**. Выберите инструмент выделения . На верхней панели настроек этого инструмента выберите инструмент овального выделения .

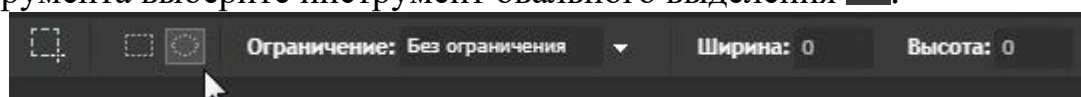


Рисунок 38

53. Сделайте круглое выделение на изображении **Город.jpg**. Затем выполните команду **Редактировать – Копировать**.

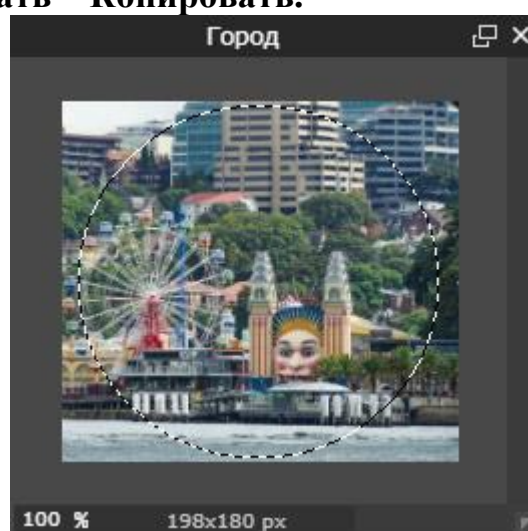


Рисунок 39

54. Скопированный фрагмент разместите в коллаже командой **Редактировать – Вставить**, слой назовите **Город**.



Рисунок 40

55. Сделаем белую рамку вокруг нового круглого изображения. Для этого создайте новый слой командой **Слой – Новый слой**. В палитре **Слои** он разместился выше всех слоев. Назовите его **Белый круг**.

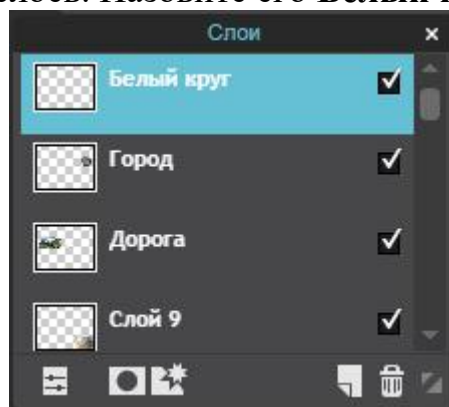


Рисунок 41

56. Сделайте на новом слое круглое выделение чуть больших размеров, чем на слое **Город**.



Рисунок 42

57. На палитре инструментов внизу установите белый цвет.



Рисунок 43

58. Выберите инструмент **Заливка**  и залейте круглую зону выделения белым цветом, щелкнув по зоне выделения.



Рисунок 44

59. Переместите слой **Белый круг** ниже слоя **Город**. Для этого нажмите на слое **Белый круг** левой кнопкой мыши и, не отпуская кнопку, переместите слой ниже слоя **Город**.



Рисунок 45

60. Чтобы круглую обводку вокруг изображения на слое **Город** сделать потоньше, можно применить инструмент **Редактировать – Free Transform...** Примерный результат на рисунке ниже.



Рисунок 46

61. У нас остался тонкий белый контур от выделения. Его можно стереть ластиком.



Рисунок 47

62. Сейчас добавим в наш коллаж немного текста и графики.

63. Создайте новый слой командой **Слой – Новый слой**. Назовите его **Графика**.

64. Выберите инструмент **Кисть** . На верхней панели настроек откройте параметры кисти.



Рисунок 48

65. Методом перебора параметров, изменения цвета, формы кисти, прозрачности, поэкспериментируйте и создайте несколько графических элементов на слое **Графика**. У нас получилось вот так.



Рисунок 49

66. Осталось сделать надписи. Выберите инструмент **Текст** . Щелкните курсором по изображению, появятся настройки инструмента текста. Введите текст в соответствующее окно, установите гарнитуру шрифта, высоту, измените цвет шрифта по своему усмотрению. После этого нажмите **Да**.

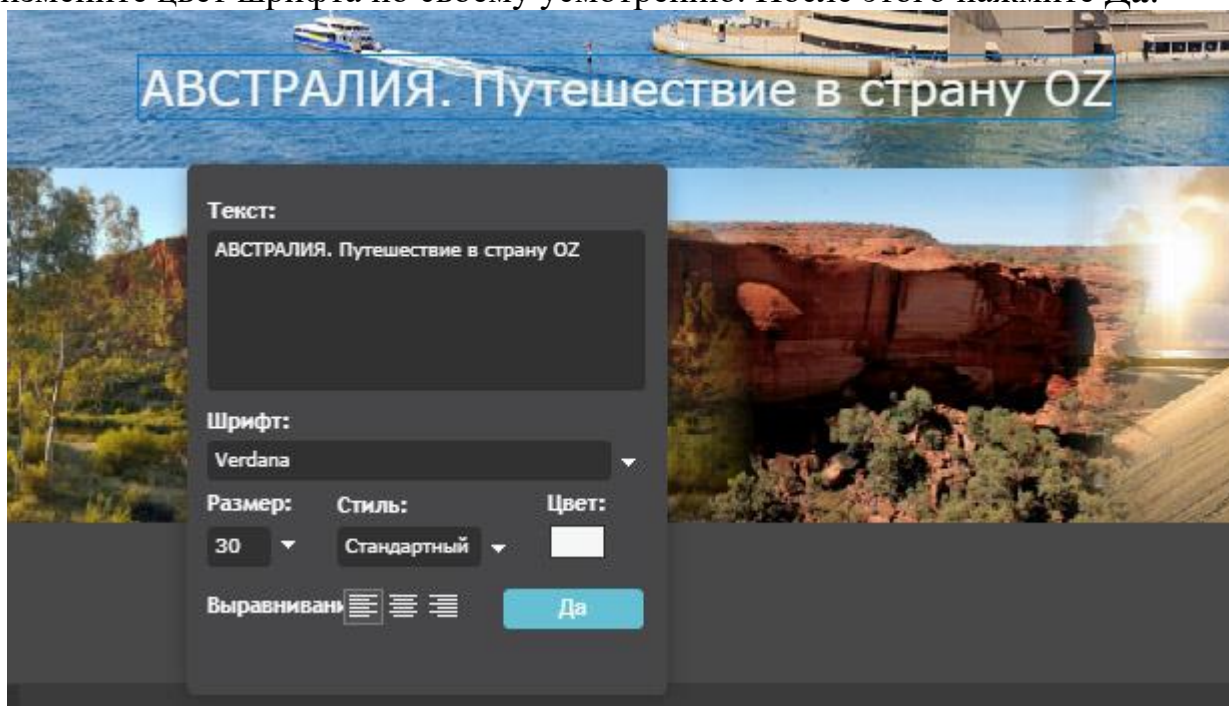


Рисунок 50

67. После подписи изображений наш коллаж выглядит вот так.



Рисунок 51

68. Сохраните файл. Для этого выберите команду **Файл – Сохранить...** Появится окно настроек сохранения. Для того, чтобы сохранить слои для последующего редактирования, необходимо выбрать формат **PXD**. Тогда Вы снова сможете открыть свой файл в Photoshop Online и продолжить работу с ним.

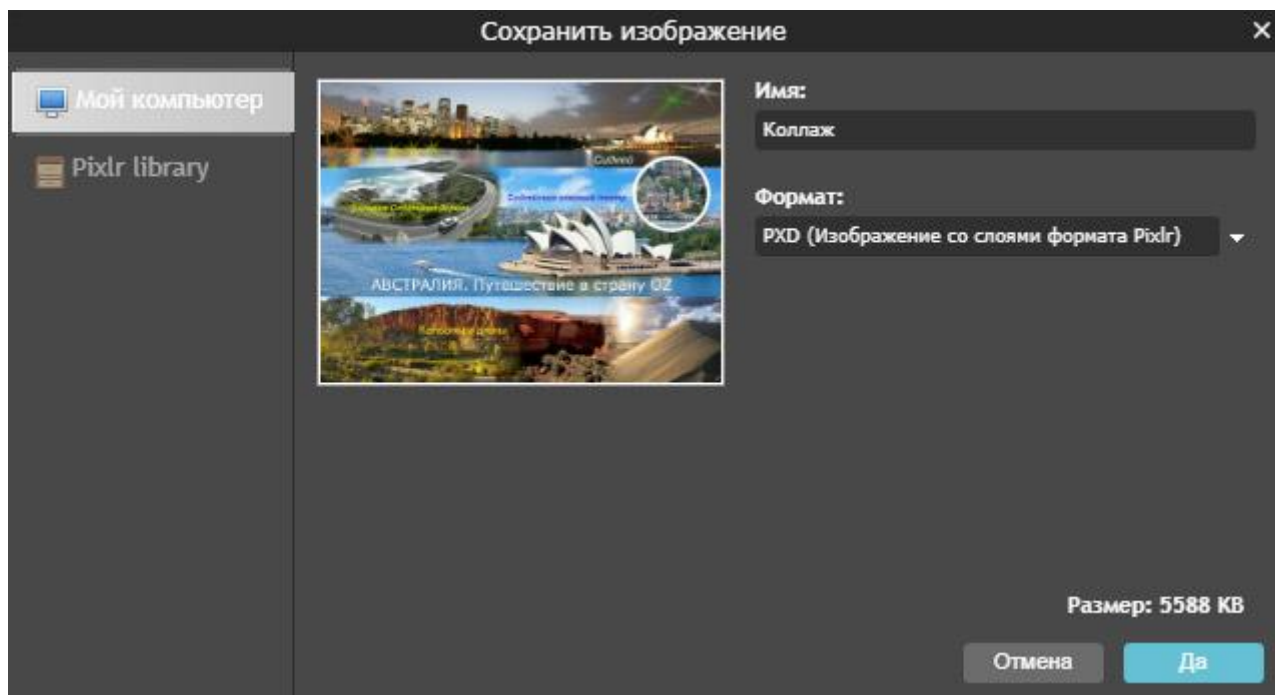


Рисунок 52

69. Если Вы больше не планируете вносить исправлений в изображение, можно сохранить в формате JPEG. Но в этом случае слои будут потеряны и продолжить редактирование изображения по слоям будет невозможно.

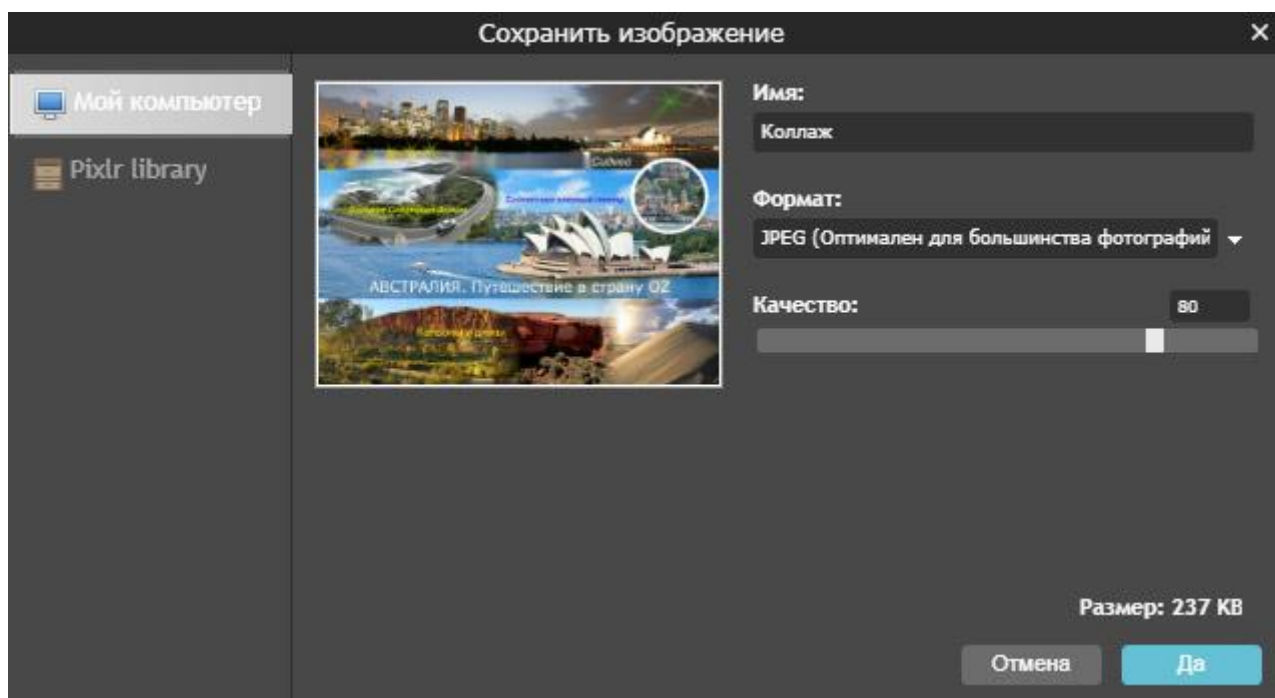


Рисунок 53

ЛАБОРАТОРНАЯ РАБОТА № 3

УСТАНОВКА GIMP. РАСТРОВАЯ ГРАФИКА.

Цель лабораторной работы: научиться работать в программе растровой графике GIMP.

Задачи лабораторной работы:

- научиться устанавливать программу растровой графики GIMP;
- производить подготовку изображений в программе растровой графики GIMP;
- научиться создавать коллажи в программе растровой графики GIMP.

Формируемые компетенции: ПК-2, ПК-4

Теоретическая часть

GIMP — растровый редактор, свободно распространяемый, с открытым исходным кодом.

Это многоплатформенное программное обеспечение для работы над изображениями. GIMP является акронимом, означающим GNU Image Manipulation Program. Редактор GIMP пригоден для решения множества задач по изменению изображений, включая ретушь фотографий, объединение и создание изображений.

Программа GIMP многофункциональна. Её можно использовать как простой графический редактор, как профессиональное приложение для ретуши фотографий, как сетевую систему пакетной обработки изображений, как программу для воспроизводства изображений, как преобразователь форматов изображений и т.д.

GIMP спроектирован расширяемым при помощи дополнений, реализующих любые возможные функции. Передовой интерфейс для программирования позволяет легко автоматизировать выполнение любых задач любого уровня.

Одной из сильных сторон GIMP является его доступность из многих источников для многих операционных систем. GIMP входит в состав большинства дистрибутивов GNU/Linux. GIMP также доступен и для других операционных систем вроде Microsoft Windows™ или Mac OS X™ от Apple (Darwin). GIMP — свободное программное обеспечение, выпускаемое под лицензией GPL(General Public License). GPL предоставляет пользователям право доступа к исходному коду программ и право изменять его.

История появления и развития.

История возникновения и развития GIMP начинается с 1995 года. Инициаторами и первыми создателями были два студента из Беркли – Спенсер Кимбелл и Питер Маттис. Их целью было написать аналог программы Photoshop, который обладал бы ее возможностями, но был бы доступным

бесплатно. В 1996 году был выпущен первый релиз. Первая же версия стала свободной и распространялась под лицензией GPL. Уже тогда GIMP поддерживал плагины для того, чтобы постоянно не переписывать внутренний код фоторедактора. Первая версия имела лишь основные инструменты для работы с изображениями, но её отличительной особенностью стала возможность работы с цветовыми каналами. Несмотря на недостатки программы, она имела успех – поддержка сообществ, создавались уроки и обучающие материалы, писалась документация.

В 1997 году был выпущен GIMP 0.99. GTK и GDK были значительно доработаны и объединены, а то, что получилось в итоге, получило название Gtk+. Написание и выпуск GTK стал очень важным шагом не только в развитии GIMP, но и других программ с открытым исходным кодом. Последующие релизы этой серии публиковались быстро, между ними не было больших задержек. Спенсеру и Питеру удалось выпустить GIMP 0.99.10 и новый Gtk+ к 9 июня 1997 года. Это был их последний релиз. Все последующие версии разрабатывались и поддерживались другими командами разработчиков.

Возможности и функции.

1. Полный набор инструментов, включая кисти, карандаш, распылитель, штамп и т.д.
2. Разумное использование памяти, при котором размер изображения ограничен только свободным местом на жёстком диске.
3. Субпиксельная дискретизация для всех инструментов рисования, дающая высококачественное сглаживание.
4. Полная поддержка альфа-канала для работы с прозрачностью.
5. Слои и каналы.
6. Процедурная база данных для вызова внутренних функций GIMP из внешних приложений, таких как «Script-Fu»
7. Продвинутое возможности по созданию сценариев.
8. Многократные отмена и повтор действий, ограниченные лишь свободным пространством на жёстком диске.
9. Инструменты преобразования, включая вращение, масштабирование, искривление и отражение.
10. Поддерживаемые форматы файлов включают GIF, JPEG, PNG, XPM, TIFF, TGA, MPEG, PS, PDF, PCX, BMP и многие другие.
11. Инструменты выделения, включая прямоугольное, эллиптическое и свободное выделение, «волшебную палочку», кривые Безье и «умное» выделение
12. Дополнения, позволяющие легко добавлять поддержку новых форматов и фильтров.

GIMP — программа, разработанная программистами для Linux. GIMP имеет полностью настраиваемый интерфейс – все кнопки и окошки можно компоновать по желанию пользователя, можно настраивать горячие клавиши.

GIMP поддерживает более 30 форматов изображений, работу со слоями, масками, фильтрами, режимами смешивания. Предоставляется большой набор инструментов для создания и редактирования изображений любой сложности. Благодаря наличию хорошей документации и огромному количеству доступных уроков освоить редактор доступно каждому.

Структура GIMP представляет собой набор модулей, взаимосвязанных друг с другом. Модули можно добавлять и изменять.

Каждый модуль в программе может отвечать за свои действия, реализуют одну или несколько собственных небольших функций (идеология UNIX).

Сами по себе, модули ничего не могут обработать. Ядро программы — графическая библиотека GEGL. Она содержит функции, реализующие обработку изображений. В основе библиотеки — алгоритмы и вся математика. Расширяемость программы реализована также внутри ядра.

GUI (Graphical user interface) в GIMP реализован с помощью библиотеки Gtk+. Она отвечает за то, как программная часть редактора взаимодействует с пользователем. Именно Gtk+ задает внешний вид всех окон, кнопок и других элементов интерфейса. Gtk+ поддерживает различные темы оформления.

GIMP – бесплатный графический редактор для обработки изображений: фотографий, сканированных снимков и др.

Скачать последнюю версию программы можно на сайте производителя по адресу <http://www.gimp.org/downloads/>.

Будем использовать версию **GIMP 2.10.4**.

Методика и порядок выполнения работы

Установка программы GIMP

1. Двойным щелчком запустите файл **gimp-2.10.4-setup-1.exe**, после чего начнется установка.
2. В окне **Select Setup Language** выберите английский язык, нажмите **ОК** (рисунок 1). Несмотря на то, что выбран английский язык, после установки программа будет на русском языке.

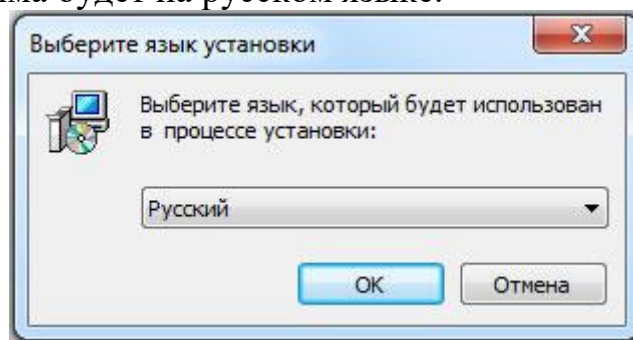


Рисунок 1

3. В следующем окне **Установка GIMP** нажмите кнопку **Установить** (рисунок 2).



Рисунок 2

Установка может занять продолжительное время (рисунок 3).

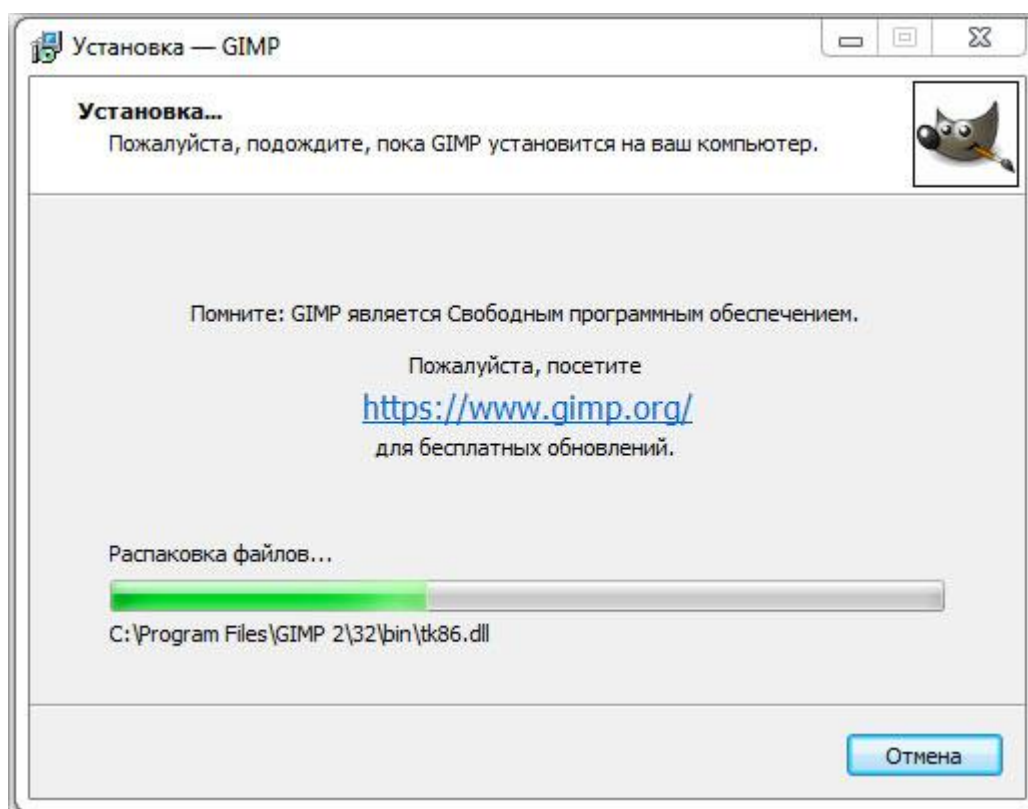


Рисунок 3

4. Дождитесь окончания установки. После этого нажмите кнопку **Завершить** (рисунок 4).

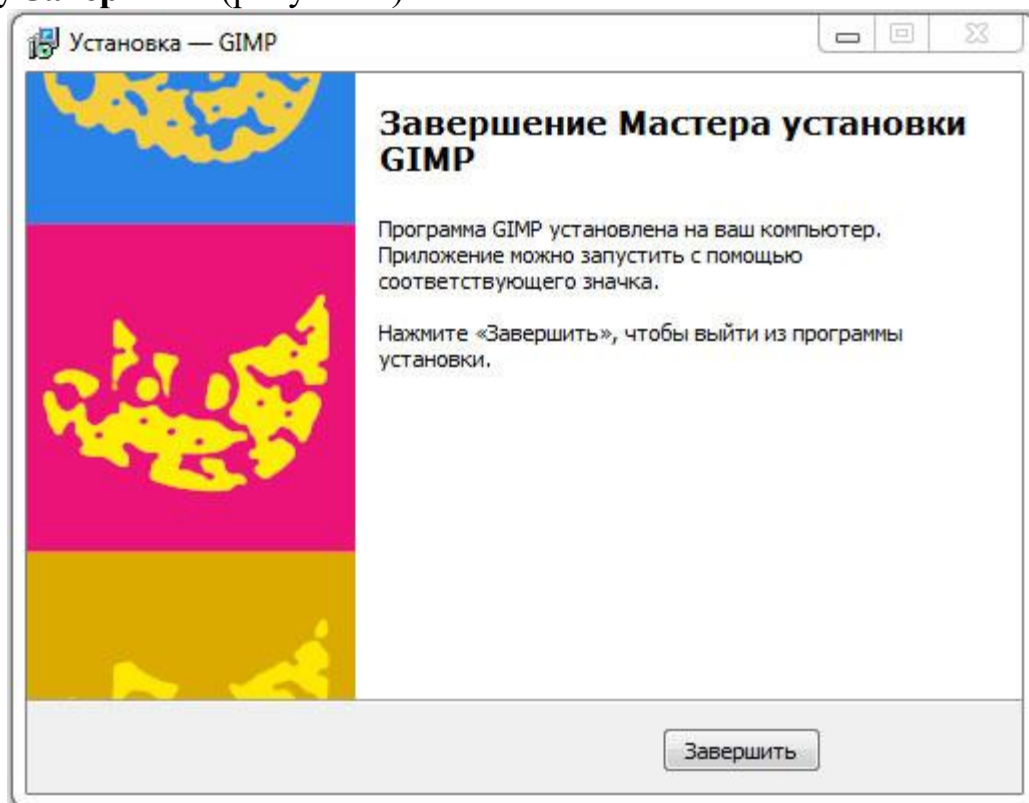


Рисунок 4

5. После установки запустите программу. Запуск также может занять несколько минут (рисунок 5).



Рисунок 5

6. Раскройте окно программы во весь экран, нажав в левом верхнем углу среднюю кнопку **Развернуть** с изображением прямоугольника (рисунок 6).



Рисунок 6

7. Перемещая панели инструментов за «шапку», настройте интерфейс, как показано на рисунке 7.

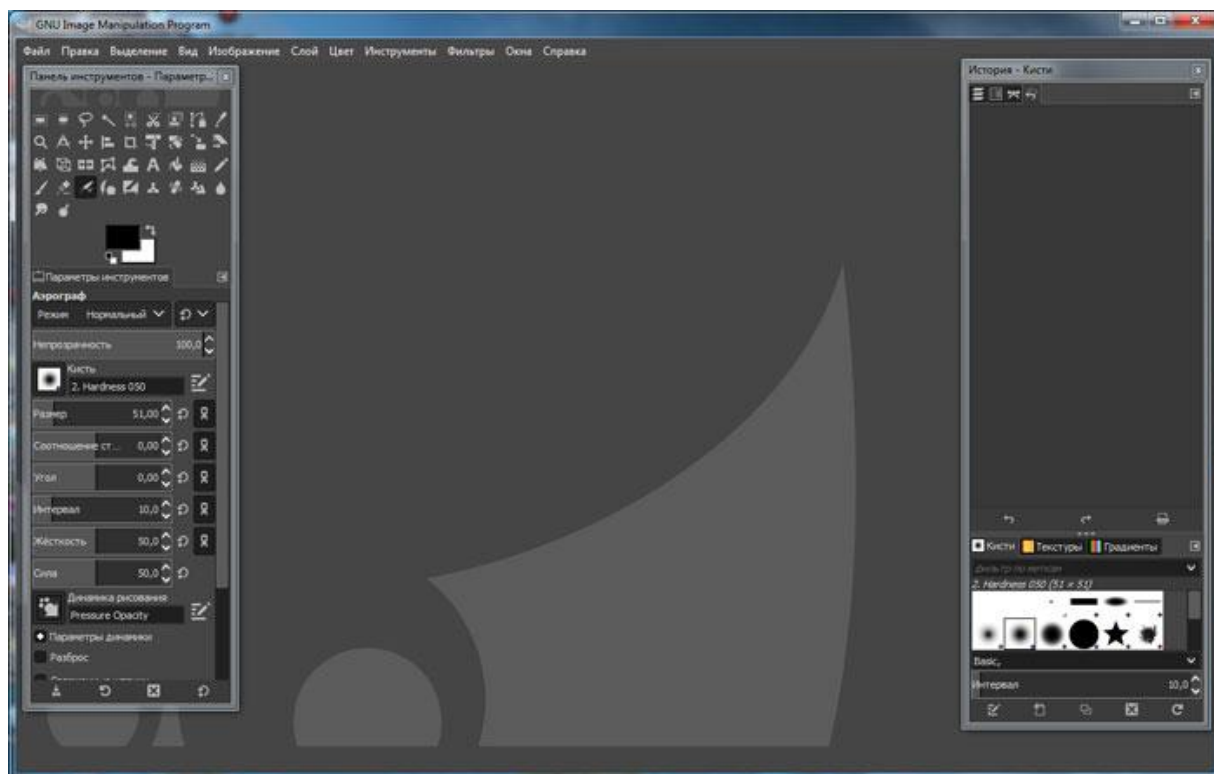


Рисунок 7

Подготовка изображений в программе растровой графики GIMP

Рассмотрим функции, которые понадобятся Вам для обработки готовых изображений.

Перечислим эти функции:

- изменение размеров изображения;
- поворот изображения;
- кадрирование изображения;
- цветокоррекция изображения;
- сохранение изображения.

Изменение размера изображения в программе Gimp

1. Откройте в программе Gimp файл **лес1.jpg**. Для этого выполните команду **Файл – Открыть** (рисунок 8).

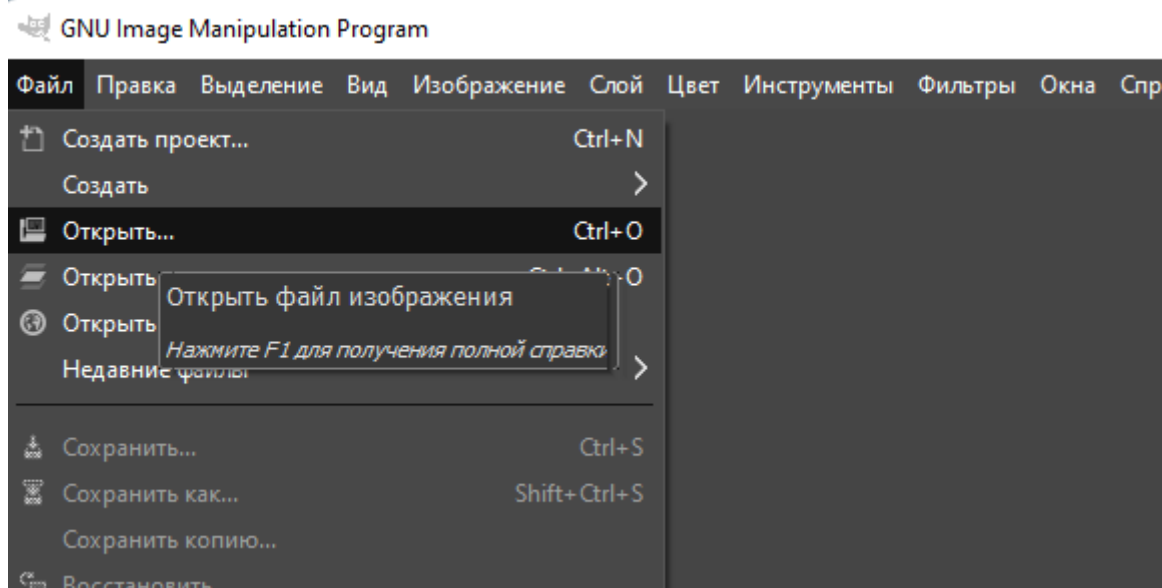


Рисунок 8 –

2. Выделите изображение лес1.jpg, нажмите кнопку **Открыть** (рисунок 9).

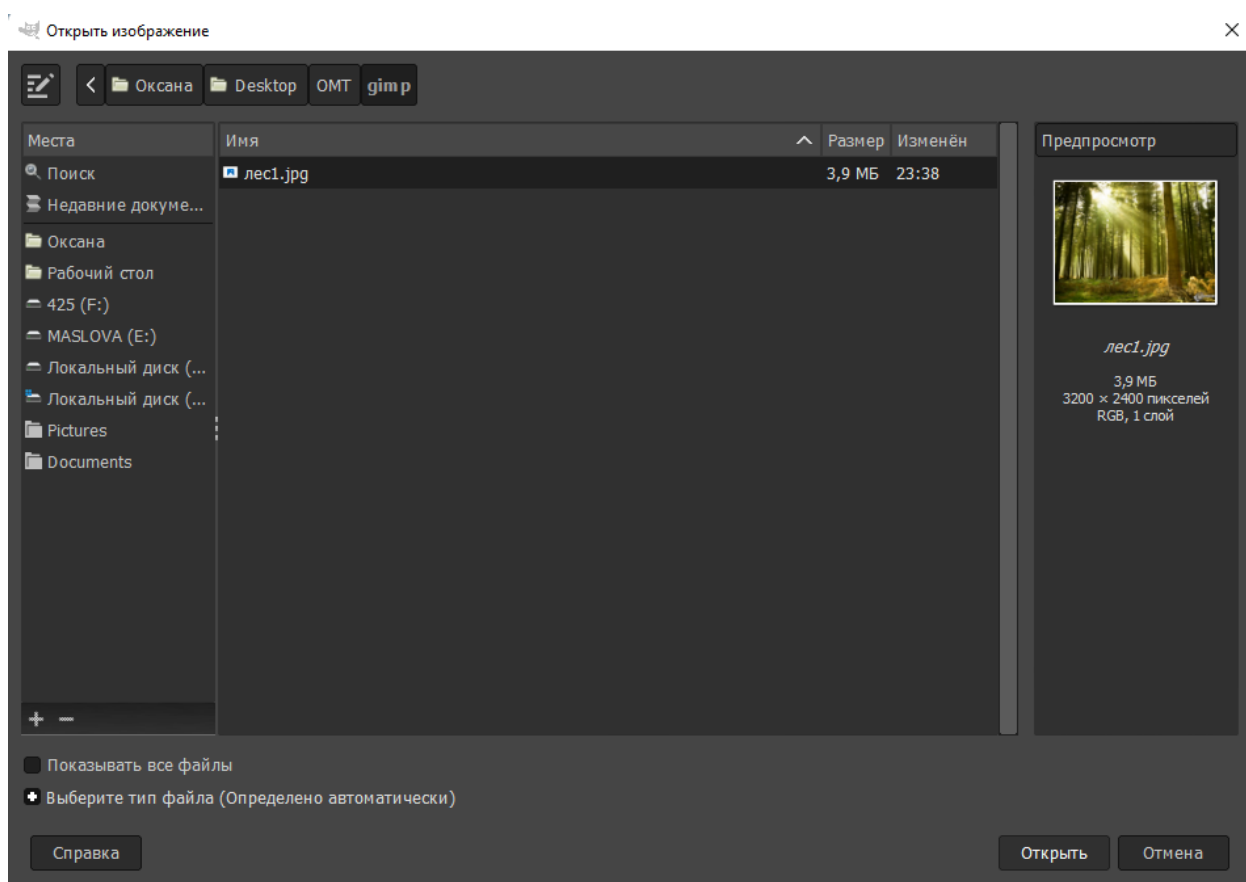


Рисунок 9

Результат на рисунке 10.

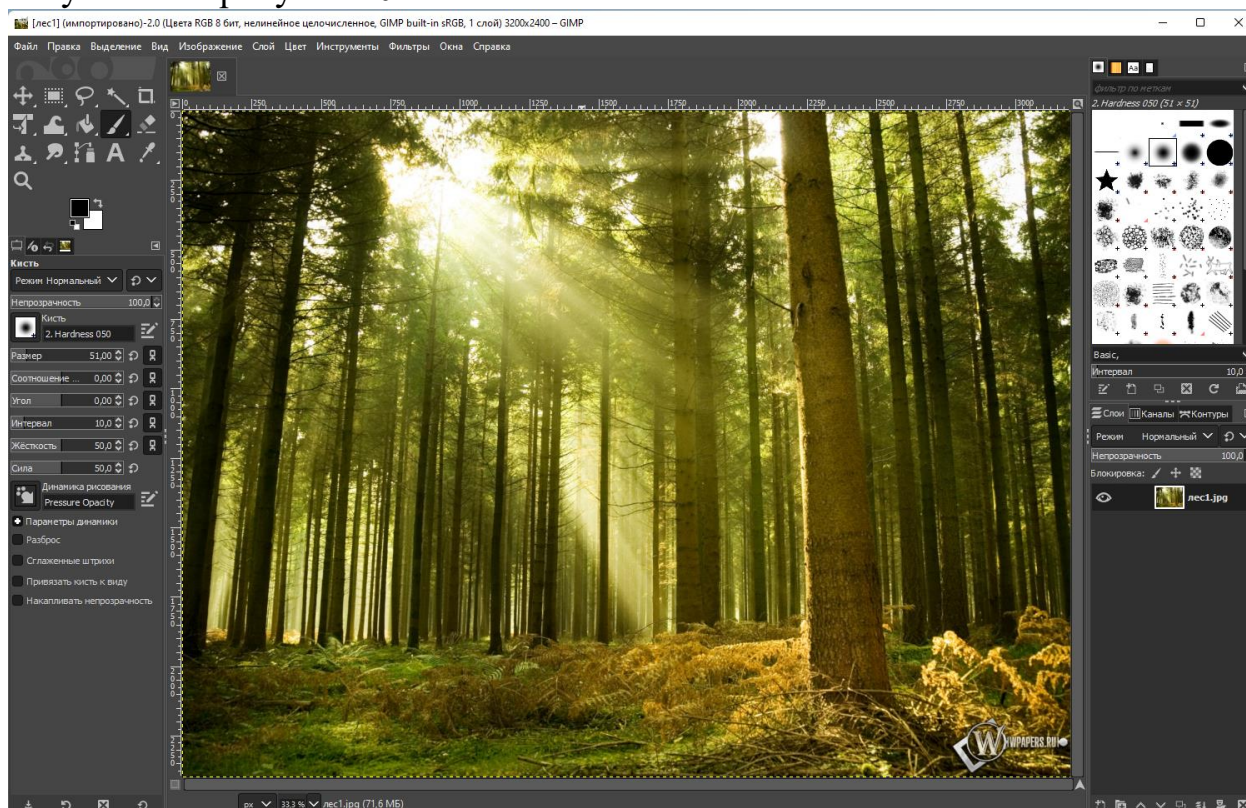


Рисунок 10

3. Сначала определим, какие размеры имеет это изображение. Для этого выполните команду **Изображение – Размер изображения** (рисунок 11).

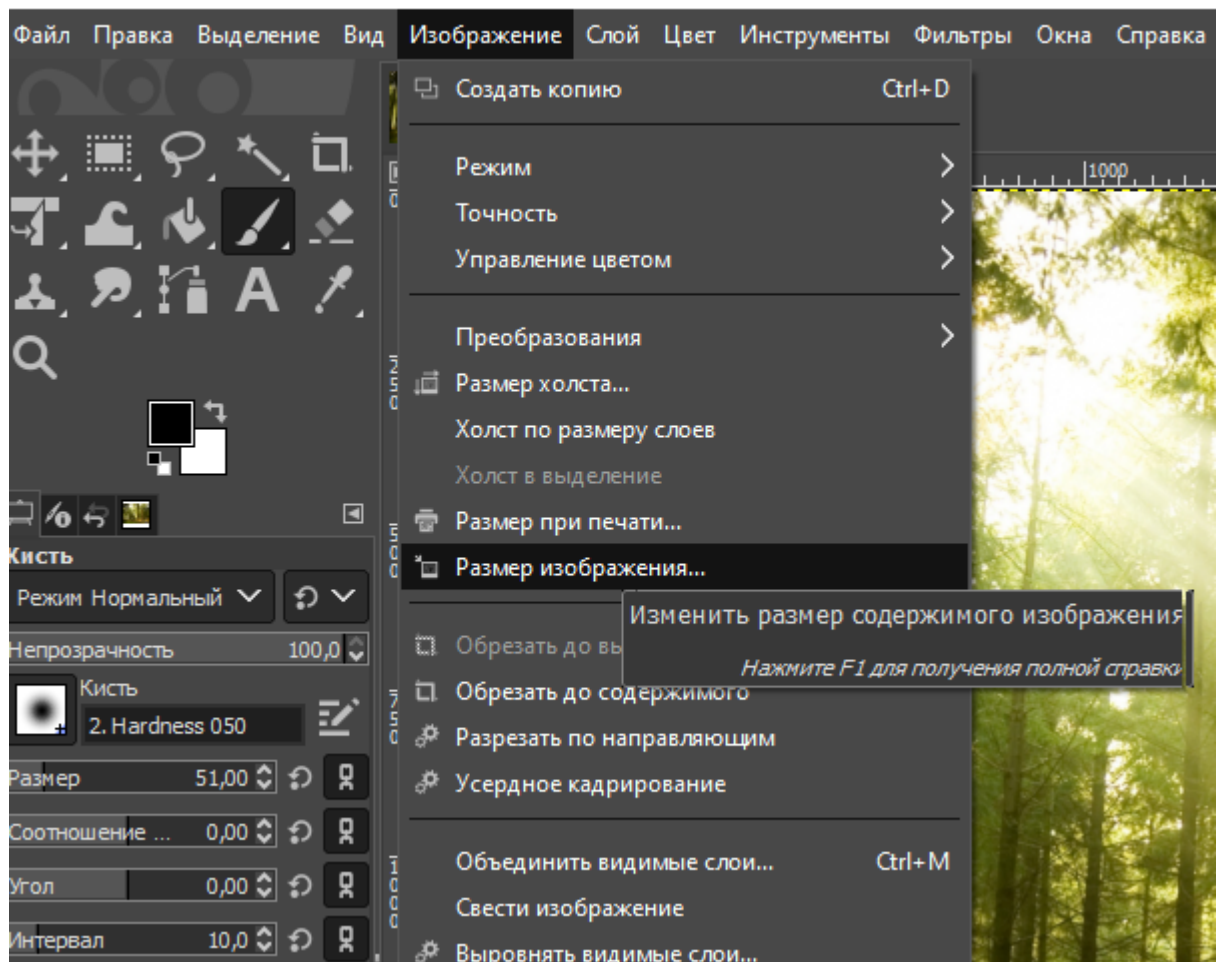


Рисунок 11

Появится окно **Изменить размер изображения** (рисунок 12).

4. Измените единицы измерения на сантиметры.

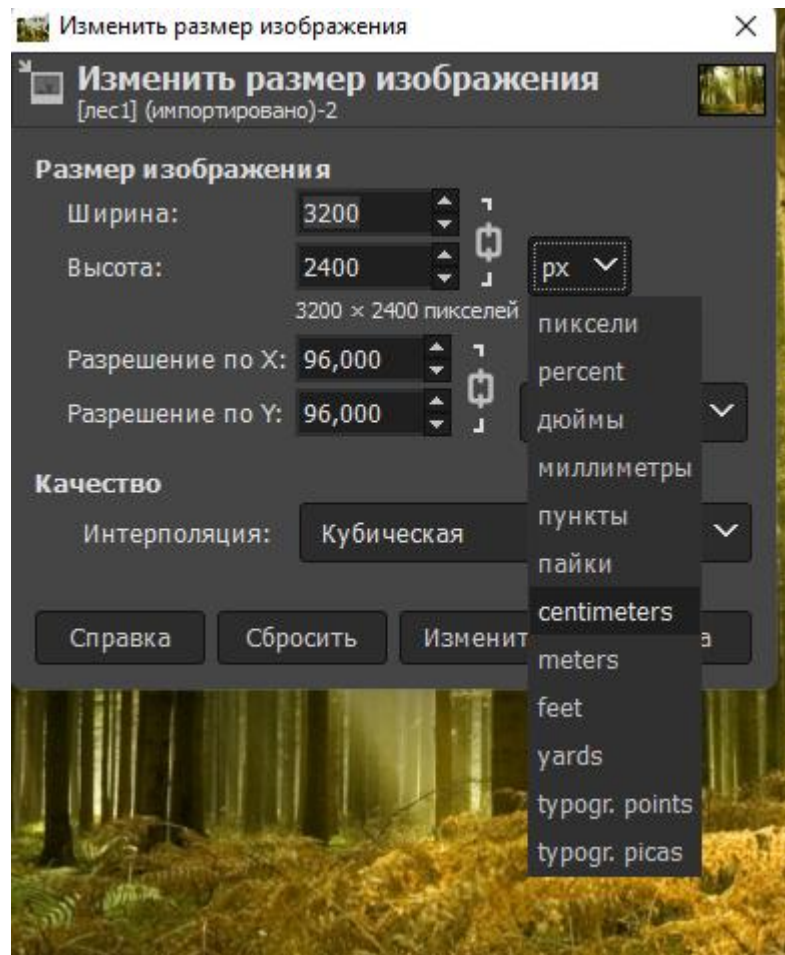


Рисунок 12

Результат на рисунке 12.

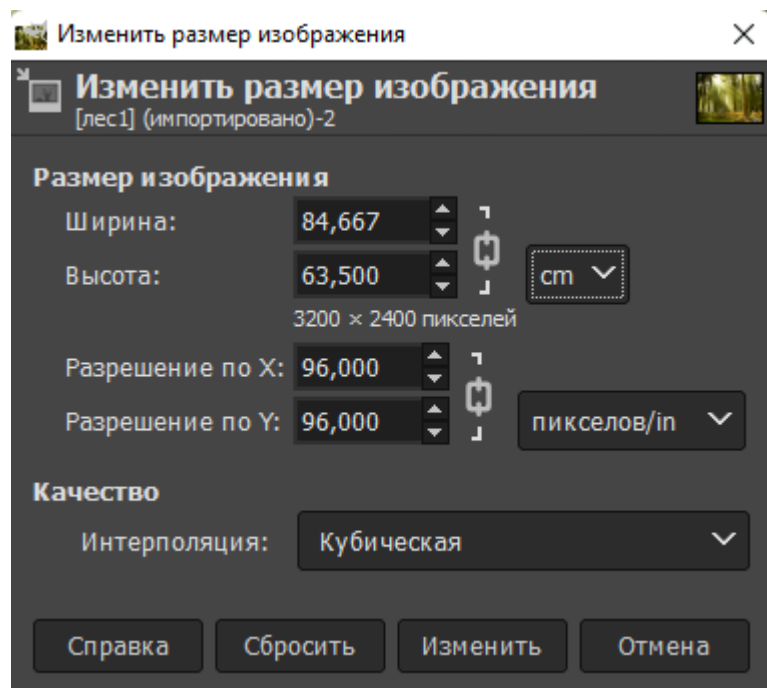


Рисунок 13

Чтобы уменьшить размер изображения в целом, измените величину ширины, затем щелкните по размеру высоты. Программа автоматически пересчитает высоту изображения.

Установите размеры приблизительно 10х6,5 см., как показано на рисунке 8 и нажмите кнопку **Изменить**.

6. Изображение слишком уменьшилось на экране. Возьмите инструмент лупы  на левой панели инструментов и, щелкая по изображению, настройте масштаб отображения фотографии удобным для работы.

7. Чтобы наоборот уменьшить изображение, нажмите клавишу **Ctrl** и, удерживая ее, щелкайте инструментом лупы по фотографии или воспользуйтесь настройками лупы в левой панели параметров инструмента (рисунок 14).

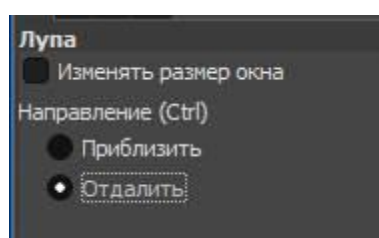


Рисунок 14

8. Для изменения масштаба изображения также можно воспользоваться настройками в нижнем левом углу экрана программы (рисунок 15).

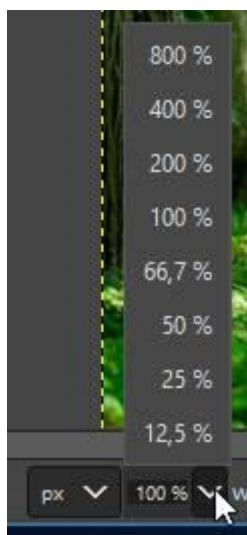


Рисунок 15

Отменить действие можно командой **Правка-Отменить действие...**

Сохранение изображения в программе Gimp

1. Сохраним изображение под новым именем. Для этого выполните команду **Файл – Экспортировать как...** (рисунок 16).

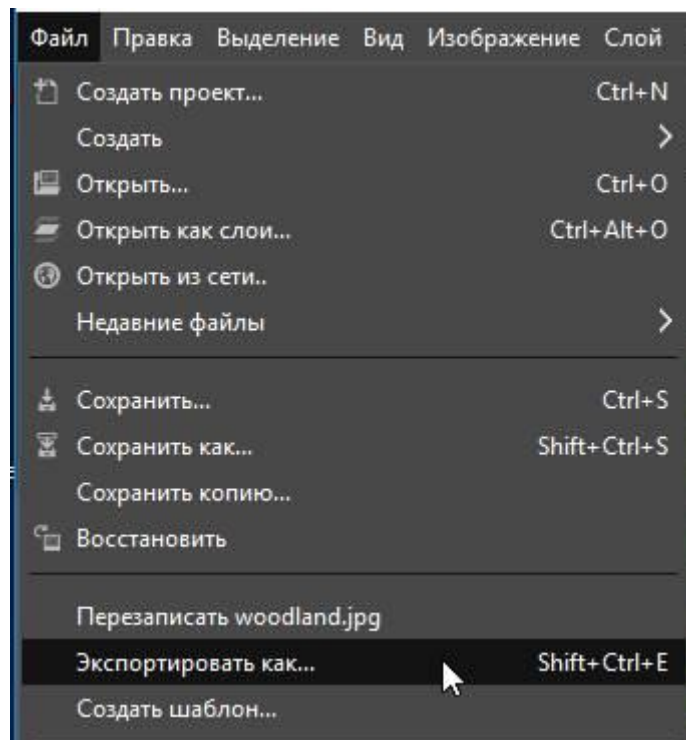


Рисунок 16

2. В появившемся окне **Экспорт изображения** в поле **Имя** введите новое имя файла (**лес2.jpg**) и нажмите кнопку **Экспортировать** (рисунок 17).

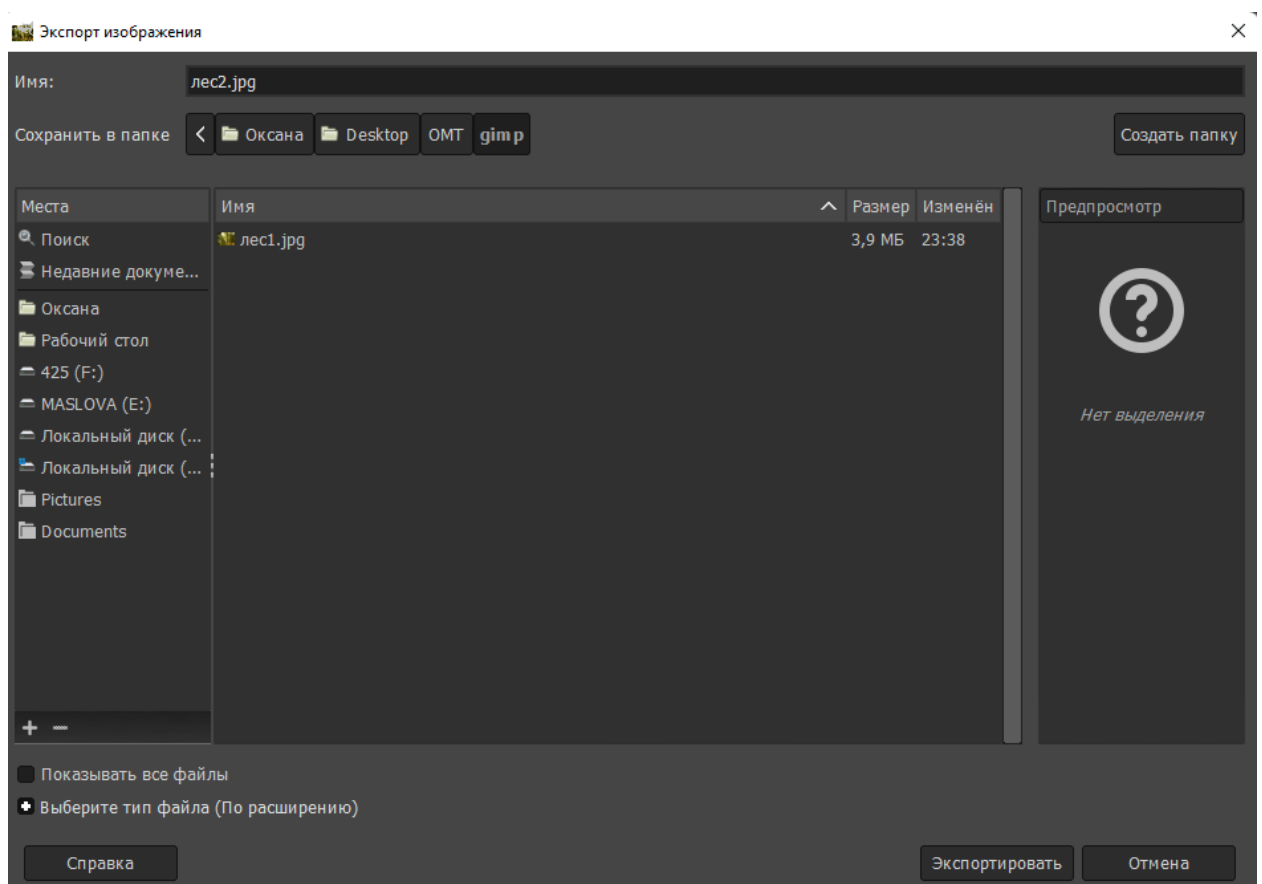


Рисунок 17

3. После этого появится окно **Экспортировать изображение как JPEG**. Установите в этом окне величину **Качество**, равную **70**, после чего нажмите кнопку **Экспорт** (рисунок 18).

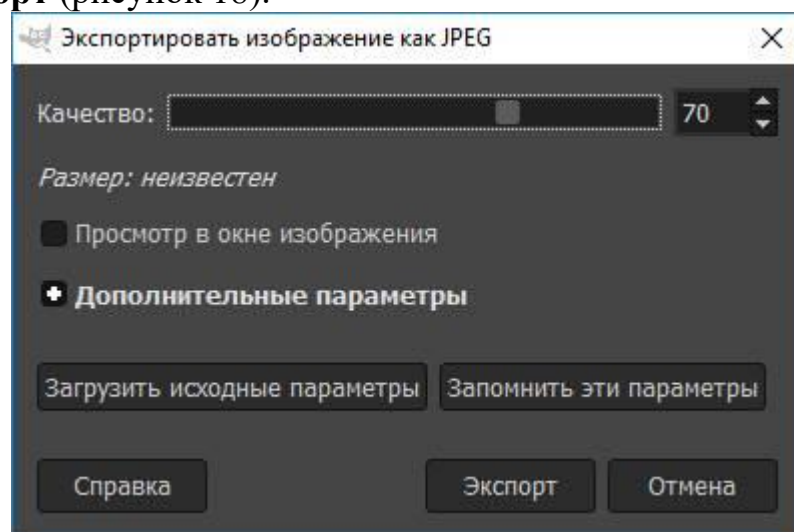


Рисунок 18

Поворот изображения в программе Gimp

Для того, чтобы повернуть изображение в Gimp на 90 градусов по часовой или против часовой стрелки, а также зеркально отобразить изображение по горизонтали или вертикали, используйте функцию **Изображение – Преобразования** (рисунок 19).

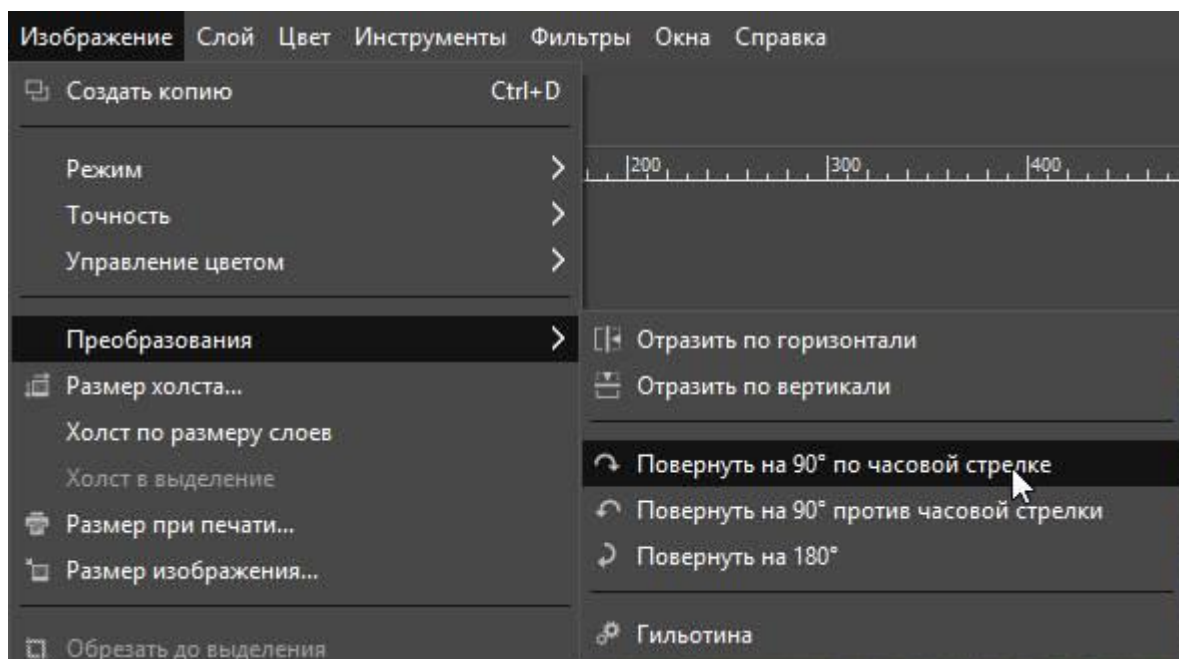


Рисунок 19

Кадрирование изображения в программе Gimp

1. Снова откройте файл **лес1.jpg**. Файл откроется в новом окне программы.

2. На левой панели инструментов выберите инструмент прямоугольного выделения .

3. Щелкните левой кнопкой мыши в левом верхнем углу изображения и, не отпуская мышь, растягивайте рамку выделения примерно так, как показано на рисунке 20.

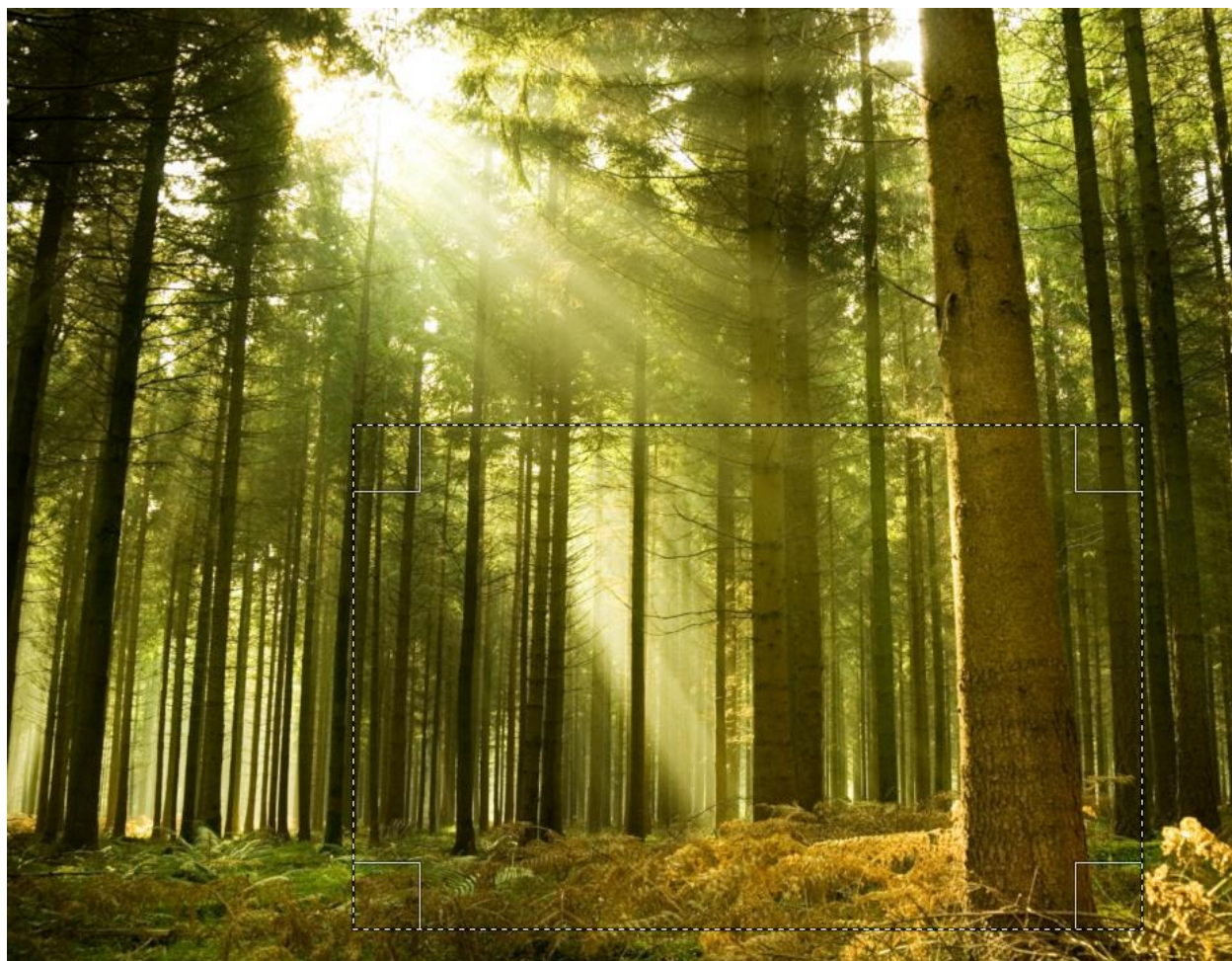


Рисунок 20

4. После этого выполните команду **Изображение – Обрезать до выделения** (рисунок 21).

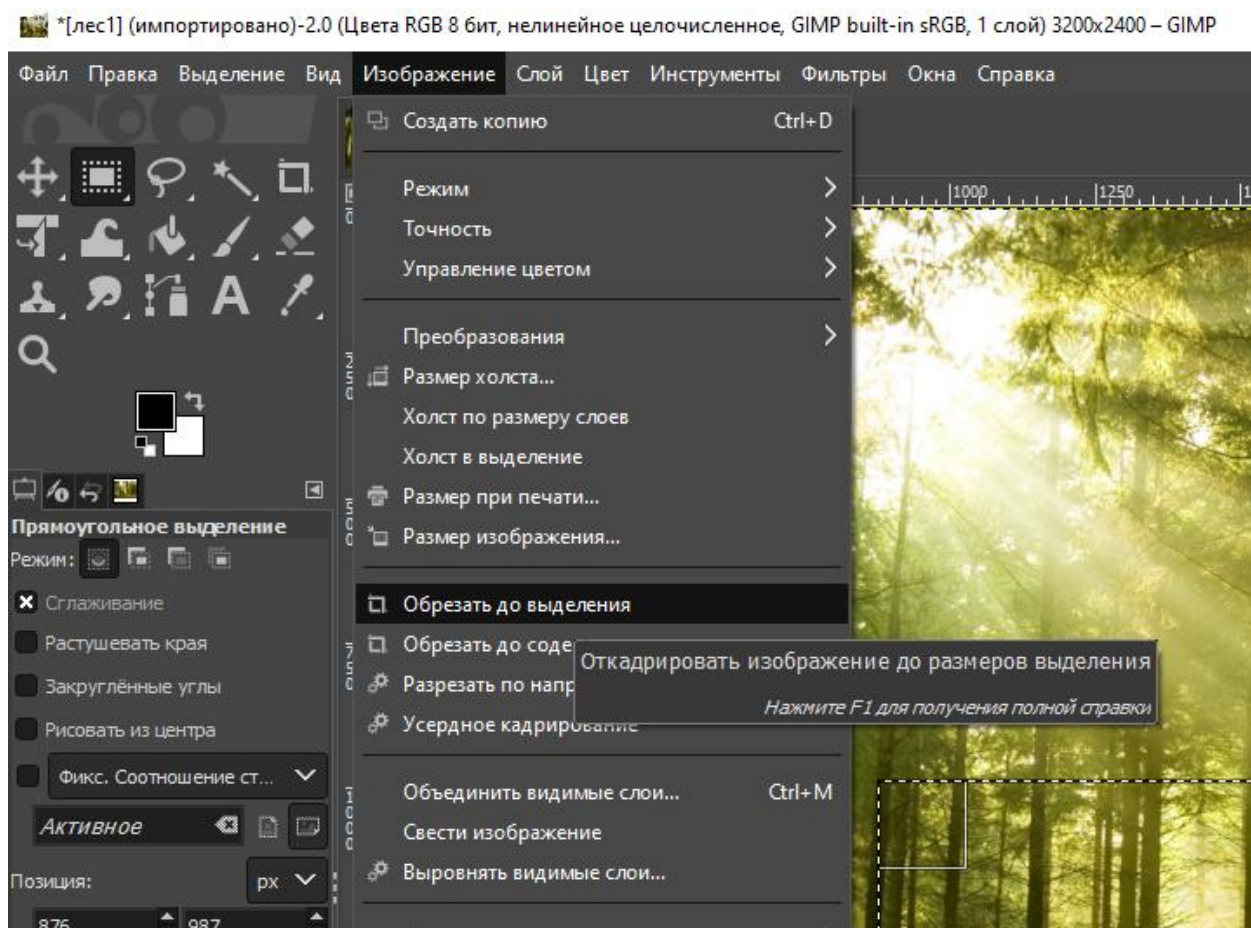


Рисунок 21

5. Сохраните обрезанное изображение в новый файл, используя команду **Файл – Экспортировать**.

Цветокоррекция изображений

1. Откройте файл **лес1.jpg**.

Инструменты цветокоррекции изображений в программе Gimp расположены в меню **Цвет**.

2. Выполните команду **Цвет – Яркость-Контраст** (рисунок 22).

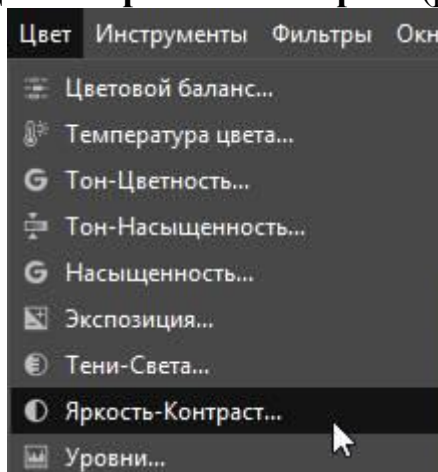


Рисунок 22

3. Появится окно настроек инструмента. Установите флажки **Предпросмотр** и **Сравнение до/после**, если он не установлен, чтобы сразу видеть изменения в файле.
4. Поработайте инструментом, передвигая ползунки в разные стороны, посмотрите, как изменяется изображение. Подберите понравившийся Вам вариант, после чего нажмите **ОК** (рисунок 23).

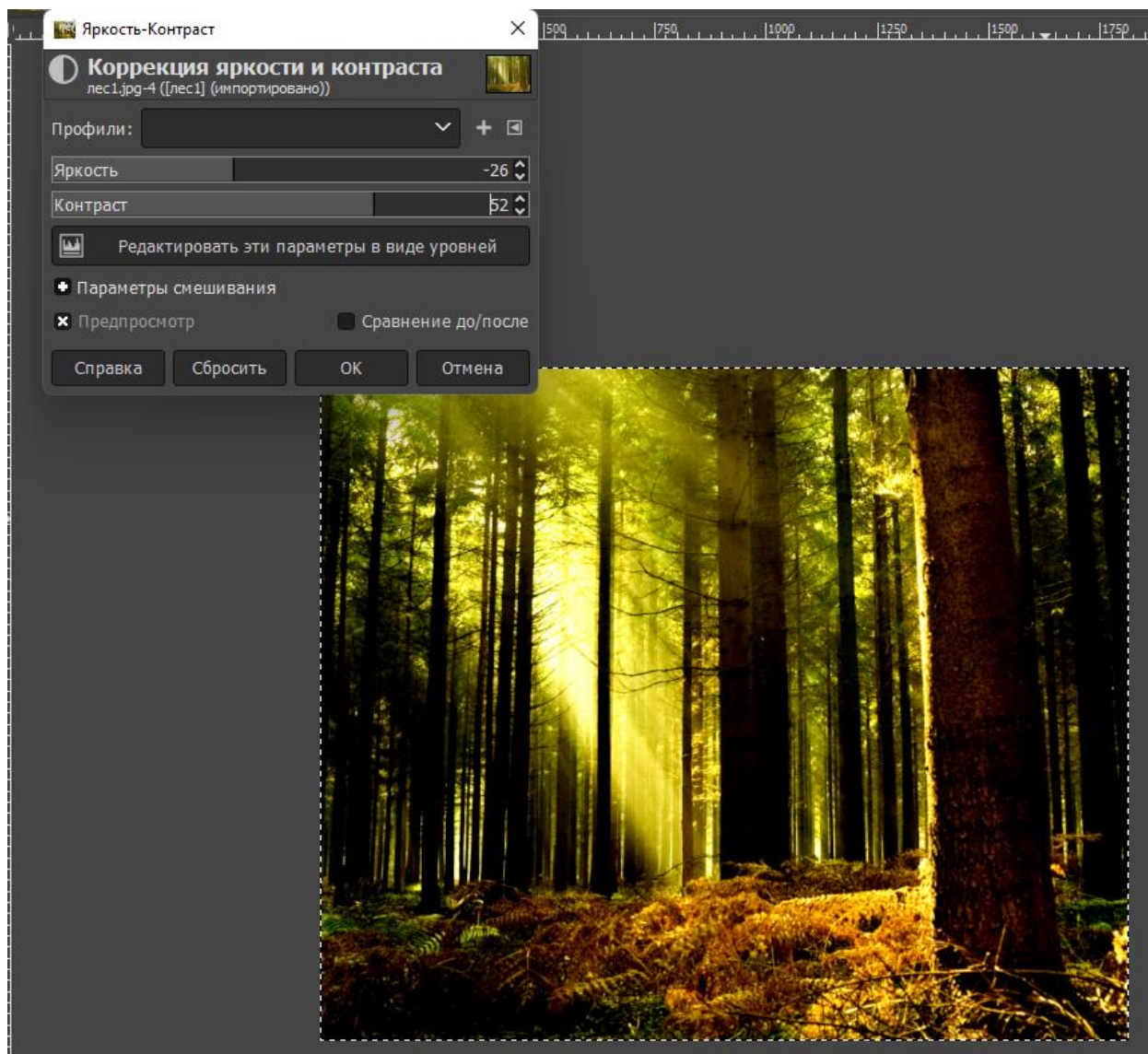


Рисунок 23

5. Подберите какое-нибудь эффектное цветовое изменение изображения и сохраните в новый файл, используя команду **Файл – Экспортировать**.
6. Снова откройте файл **лес1.jpg**.
7. Выполните команду **Цвет – Тон-Насыщенность**.
8. Появится окно настроек инструмента. Установите флажки **Предпросмотр** и **Сравнение до/после**. Передвигая ползунки, посмотрите, как изменяются цветовые оттенки изображения (рисунок 24).

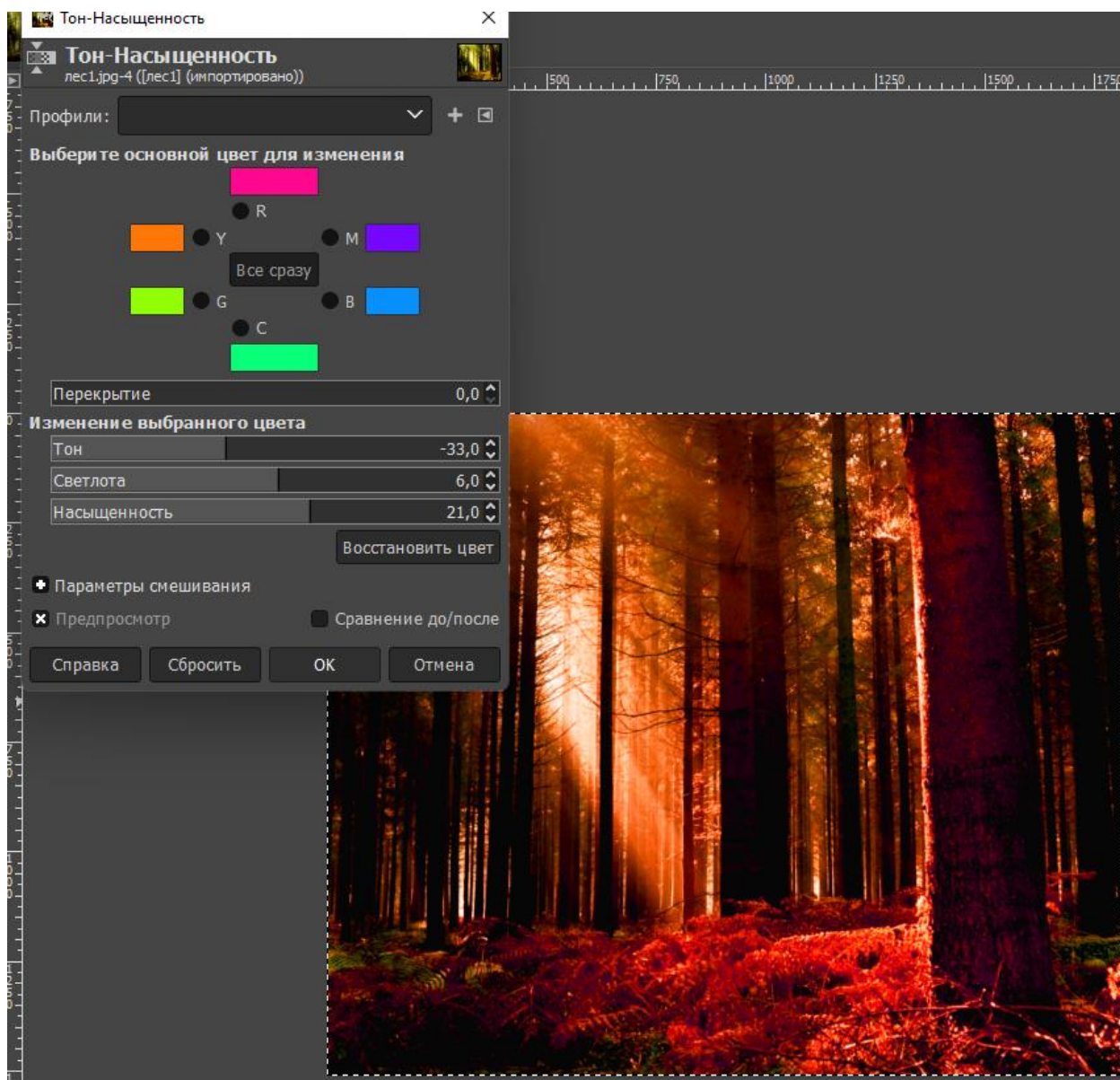


Рисунок 24

9. Подберите какое-нибудь эффектное цветовое изменение изображения и сохраните в новый файл, используя команду **Файл – Экспортировать**.
Еще один популярный инструмент для цветокоррекции изображений – Кривые.
10. Снова откройте файл **лес1.jpg**.
11. Выполните команду **Цвет – Кривые** (рисунок 25).

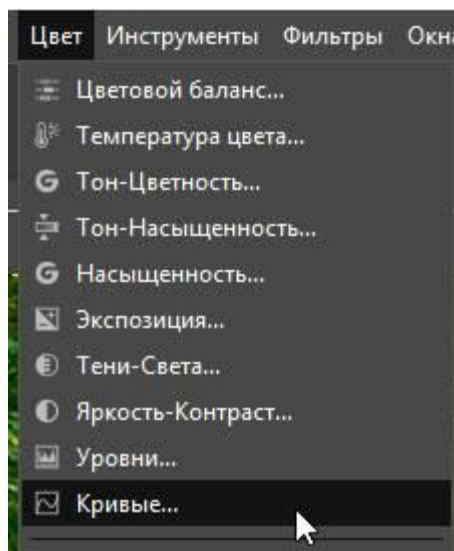


Рисунок 25

Появится окно **Кривые**.

12. Убедитесь, что установлены флажки **Предпросмотр** и **Сравнение до/после**.

13. Щелчками добавьте точки на кривой и, перемещая маркеры, настройте форму кривой, как показано на рисунок 26. На изображении сразу видны изменения.

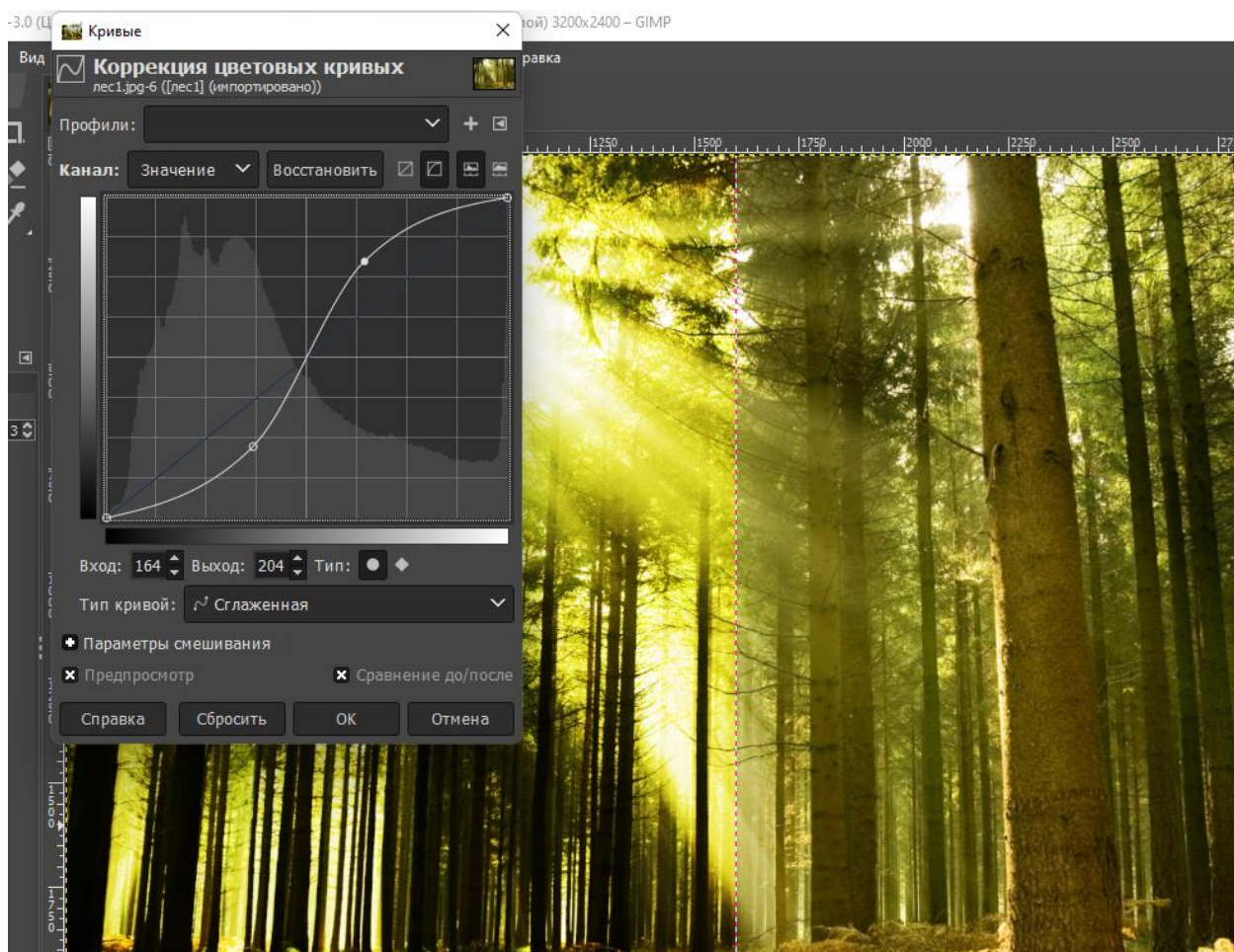


Рисунок 26

14. Поэкспериментируйте с формой кривой, добавляя/перемещая точки.
15. Подберите какое-нибудь эффектное цветовое изменение изображения и сохраните в новый файл, используя команду **Файл – Экспортировать**.

Контрольное задание

Выполнить задание, сохраняя промежуточные результаты, и оформить отчет к лабораторной работе:

Титульный лист;

Цель, задачи работы;

Этапы выполнения работы.

Средствами программы растровой графики GIMP создать коллаж о вашей дисциплине.

Требования к коллажу:

- название темы (дисциплины);
- фоновое изображение;
- подписи к фотографиям/изображениям;

- применение таких эффектов как растушевка края изображения, плавный переход одного изображения в другое;
- применение различных кистей.

В целом коллаж должен выглядеть насыщенно и декоративно, стиль минимализма в дизайне не применять.

Либо можно создать коллаж на свободную тему с соблюдением всех требований:

- название;
- фоновое изображение;
- подписи к фотографиям/изображениям;
- применение таких эффектов как растушевка края изображения, плавный переход одного изображения в другое;
- применение различных кистей.

Контрольное задание

Выполнить все представленные выше задания, сохраняя промежуточные результаты, можно использовать свои изображения, либо применять другие цвета при обработке, чтобы работа была уникальной. Оформить отчет к лабораторной работе:

Титульный лист;

Цель, задачи работы;

Этапы выполнения работы.

Ответы на контрольные вопросы.

ЛАБОРАТОРНАЯ РАБОТА № 4

Графика в интерфейсах. Растровая и векторная графика, SVG. Создание иконок и UI-компонентов в Figma.

Цель лабораторной работы: научиться различать растровую и векторную графику и понимать области их применения в UI.

Задачи лабораторной работы:

- Изучить формат SVG и его преимущества для интерфейсов.
- Освоить базовые инструменты Figma для создания векторных иконок и UI-компонентов.

Формируемые компетенции: ПК-2, ПК-4

Теоретическая часть

В современной веб- и мобильной разработке графическое оформление интерфейса перестало быть второстепенным элементом. Пользователь принимает решение о доверии к продукту в течение первых 50 миллисекунд (исследования Google), причем 94% этого решения связано с визуальным дизайном. Графика в интерфейсе выполняет не только эстетическую, но и функциональную роль:

- **Навигационную** (иконки меню, стрелки назад).
- **Коммуникативную** (индикаторы загрузки, уведомления).
- **Поведенческую** (анимация кнопок, обратная связь).

При этом ключевая проблема, с которой сталкиваются разработчики — **выбор правильного типа графики** (растр или вектор) и **инструментов для её создания**. Ошибка на этом этапе приводит к:

- Медленной загрузке страниц (тяжелые PNG/JPG).
- Размытым иконкам на экранах с высоким DPI (Retina).
- Трудностям с поддержкой и масштабированием дизайна.

Растровая графика — это матрица пикселей (точек), каждая из которых имеет свой цвет. Характеристики:

Разрешение (ширина × высота в px).

Глубина цвета (бит на пиксель).

Ключевое ограничение: при масштабировании программа вынуждена интерполировать (дорисовывать) пиксели, что ведёт к пикселизации или размытию.

Примеры форматов: PNG (lossless, прозрачность), JPEG (lossy, без прозрачности), GIF (анимация, 256 цветов), WebP (современный гибрид).

Векторная графика — это описание изображения математическими формулами: точки, линии, кривые Безье, дуги, заливки.

- **Масштабирование:** всегда остаётся чёткой, так как пересчитываются координаты кривых.

- **Вес файла:** зависит от сложности контура, а не от физического размера.
- **Примеры форматов:** SVG (веб), AI (Adobe Illustrator), EPS (полиграфия).

Что такое SVG?

SVG (Scalable Vector Graphics) — это открытый стандарт W3C с 1999 года. Фактически, это XML-документ, описывающий векторную графику.

viewBox="minX minY width height" — **самый важный атрибут**. Он определяет систему координат внутри SVG. Без него масштабирование ломается.

preserveAspectRatio — как вписывать графику в контейнер (например, xMidYMid meet — по центру, сохраняя пропорции).

fill, stroke, stroke-width, stroke-linecap (как рисовать концы линий), stroke-linejoin (как рисовать углы).

Методика и порядок выполнения работы:

Этап 1. Настройка проекта

1. Создайте новый файл Figma.
2. Создайте фрейм (Frame) размером 1440×1024 (десктоп).
3. Внутри создайте 3 области:
 - Area 1: Сетка иконок (сетка 4×4, ячейка 100×100px)
 - Area 2: UI-компоненты (кнопки)
 - Area 3: SVG-код

Этап 2. Создание 4 базовых иконок по образцу (или своему дизайну)

Задание 3.1: Используя инструменты Pen (P), Shape tools, Boolean, создайте следующие иконки в едином стиле (толщина линии 2px, размер 24×24px):

№	Иконка	Инструменты в Figma	Критерии качества
1	Дом (Home)	Прямоугольник + Треугольник (Polygon)	Крыша не касается стен, дверь по центру
2	Сердце (Like)	Pen tool (кривые Безье) или два круга + повернутый квадрат	Симметричная выемка сверху
3	Корзина (Trash)	Прямоугольники + Линии	Ручка корзины, зубцы внизу
4	Шестеренка (Settings)	Круг + Многоугольник (Boolean: Subtract)	Зубцы равномерно распределены

Результат этапа 2: Скриншот ваших 4 иконок на сетке 24×24.

Этап 3. Создание UI-компонента «Кнопка»

Задание 3.2: Создайте компонент кнопки с 3 состояниями.

1. Нарисуйте прямоугольник 120×40, скругление 8px.
2. Добавьте текст «Нажми».
3. Выделите всё → клик правой кнопкой → **Create component** (или Ctrl+Alt+K).
4. Назовите компонент Button/Primary.
5. Создайте **Variants** (варианты):
 - *State*: Default → Hover → Pressed → Disabled.
6. Настройте цвета для каждого состояния (свойство **Fill** и **Text**) и тень.

Результат этапа 3: Скриншот компонента с панелью Variants (4 варианта кнопки).

Этап 4. Экспорт в SVG и анализ кода

Задание 3.3: Экпортируйте одну из иконок (например, «Сердце») как SVG.

1. Выделите иконку.
2. В правой панели → раздел **Export** → нажмите + → выберите **SVG**.
3. Нажмите **Export**, сохраните файл.
4. Откройте SVG-файл в Блокноте или VS Code.

Задание: Скопируйте сюда первые 10 строк SVG-кода и подпишите, за что отвечает каждый тег/атрибут:

Контрольные вопросы (для защиты)

1. Что будет, если растянуть PNG иконку в 10 раз в вебе? А SVG?
2. Зачем нужна группа «Vector Networks» в Figma?
3. В чем разница между Union, Subtract, Intersect в Boolean операциях?
4. Как сделать компонент с иконкой внутри кнопки, чтобы иконка менялась сама?

ЛАБОРАТОРНАЯ РАБОТА № 5

Проектирование прототипа информационной системы

Цель лабораторной работы: понять структуру пользовательского интерфейса;

Задачи лабораторной работы: Освоить навыки перехода от логической модели (UML) к визуальному представлению системы.

Формируемые компетенции: ПК-2, ПК-4

Теоретическое обоснование

Прототипирование — это процесс создания упрощенной модели информационной системы, отражающей:

- структуру интерфейса;
- основные элементы управления;
- последовательность действий пользователя.

Прототип не содержит программной реализации и используется на этапе проектирования.

Прототип является промежуточным этапом между: логической моделью (UML) и реальной программной реализацией

Он позволяет:

- проверить удобство использования системы;
- выявить ошибки в логике взаимодействия;
- уточнить требования к интерфейсу;
- визуализировать проект.

Ранее разработанные диаграммы:

Диаграммы вариантов использования — определяют функции;

Диаграммы последовательности — описывают сценарии;

Диаграммы деятельности — показывают процессы;

Основные элементы пользовательского интерфейса

При проектировании прототипа используются базовые элементы:

Кнопка (Button) Используется для выполнения действия.

Пример:

«Войти» «Отправить» «Добавить»

Поле ввода (Input Field) Предназначено для ввода данных пользователем.

Пример:

имя email пароль

Список (List)

Отображает набор элементов.

Пример:

список товаров список врачей

Форма (Form)

Совокупность полей ввода и кнопок.

Пользовательский сценарий — это последовательность действий пользователя при работе с системой.

Пример:

1. пользователь открывает систему
2. выбирает услугу
3. вводит данные
4. подтверждает действие
5. получает результат

Прототип должен полностью отражать этот сценарий.

Минимальный прототип включает:

1. Главный экран
2. Экран выбора
3. Экран ввода данных
4. Экран подтверждения
5. Экран результата

Методика и порядок выполнения работы

На основе ранее разработанного проекта разработать **мини-прототип системы.**

1. Описать пользовательский сценарий (не менее 5 шагов)
2. Определить основные экраны
3. Спроектировать интерфейс
4. Показать переходы между экранами
5. Добавить обработку ошибки

В работе должны присутствовать:

- не менее **5 экранов**
- форма ввода данных
- кнопки управления
- экран результата
- минимум **1 ошибка системы**

Методика выполнения:

Выбрать один основной сценарий:

- заказ
- регистрация
- запись
- поиск

Каждый шаг → отдельный экран

Нарисовать:

- элементы интерфейса
- кнопки
- поля

Показать переходы:

Экран 1 → Экран 2 → Экран 3

Обязательно предусмотреть неправильное действие пользователя, описать реакцию системы

Инструменты:

- Figma

- Draw.io
- PowerPoint

Требования к отчету:

1. Название системы
2. Описание сценария
3. Перечень экранов
4. Прототип (изображения)
5. Схема переходов
6. Описание ошибки
7. Вывод

Контрольные вопросы:

1. Что такое прототип?
2. Чем он отличается от программы?
3. Что такое пользовательский сценарий?
4. Какие элементы интерфейса существуют?
5. Зачем нужна обработка ошибок?

ЛАБОРАТОРНАЯ РАБОТА № 6

CSS верстка. Flexbox, Grid, структура страницы. Верстка макета из Figma.

Цель лабораторной работы: освоить современные технологии CSS-верстки: Flexbox и Grid.

Задачи лабораторной работы:

- - Получить навык верстки адаптивного веб-интерфейса по готовому дизайн-макету.
- Интегрировать созданные ранее иконки (лабораторная работа №2) в реальный код.

Формируемые компетенции: ПК-2, ПК-4

Теоретическая часть

В современной веб-разработке умение сверстать макет из Figma является базовым профессиональным навыком для frontend-разработчика, UI/UX-дизайнера и fullstack-разработчика. По данным опроса Stack Overflow 2024, CSS и HTML входят в топ-3 наиболее востребованных технологий, причем 67% разработчиков считают знание Flexbox и Grid обязательным для трудоустройства.

Ключевой вывод: Flexbox и Grid не конкурируют, а дополняют друг друга. Flexbox — для внутренней структуры компонентов (карточка, меню, кнопка), Grid — для глобальной структуры страницы.

Рекомендуется изучить следующие литературные источники для получения теоретических основ:

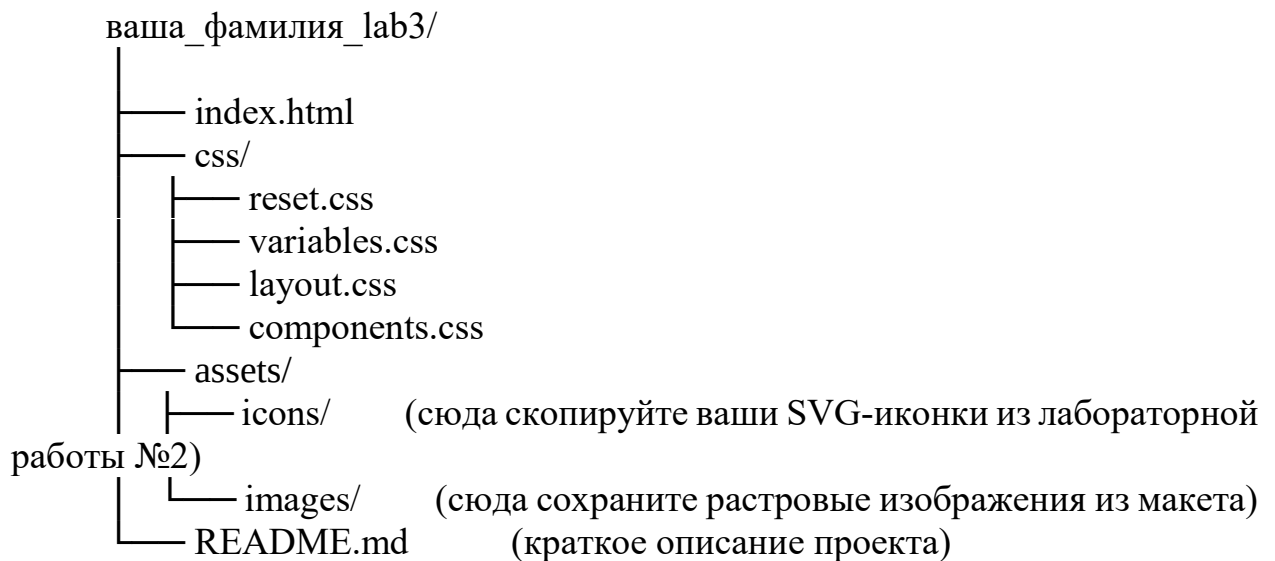
1. Дакетт, Д. HTML и CSS: разработка и дизайн веб-сайтов / Д. Дакетт ; перевод с английского А. В. Снастина. — Москва : Эксмо, 2020. — 480 с. — (Мировой компьютерный бестселлер). — ISBN 978-5-04-102388-9.
2. Эндрю, Р. CSS Grid. Полное руководство / Р. Эндрю ; перевод с английского А. В. Сергеева. — Санкт-Петербург : Питер, 2021. — 288 с. — (Для профессионалов). — ISBN 978-5-4461-1687-9.
3. Шадид, А. CSS-макеты: от Float до Flexbox и Grid / А. Шадид ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2020. — 312 с. — ISBN 978-5-97060-876-4.

Методика и порядок выполнения работы

Подготовка структуры проекта

Создайте файловую структуру

Создайте на компьютере папку проекта со следующей структурой (имена файлов и папок строго соблюдайте):



Подготовьте HTML-файл

Создайте файл index.html и напишите в нем базовую структуру HTML-документа с указанием:

Правильного DOCTYPE.

Русской локализации (lang="ru").

Мета-тега для адаптивности (viewport).

Подключения всех четырех CSS-файлов (reset, variables, layout, components) в правильном порядке.

Пустых контейнеров для будущих блоков (header, sidebar, main, footer) с соответствующими классами.

Написание CSS (базовые стили)

4.1. Сброс стилей (reset.css)

В файле reset.css напишите правила, которые:

Обнуляют отступы (margin и padding) у всех элементов.

Устанавливают box-sizing: border-box для всех элементов (включая псевдоэлементы ::before и ::after).

Убирают подчеркивание у ссылок и наследуют цвет текста.

Убирают стандартную рамку у кнопок и добавляют курсор pointer.

Делают изображения адаптивными (max-width: 100%, height: auto).

4.2. CSS-переменные (variables.css)

В файле variables.css в корневом селекторе :root объявите CSS-переменные для:

Цвета: основной цвет бренда, цвет фона, цвет текста, цвет акцентов, цвет рамок.

Шрифтов: семейство шрифтов, базовый размер, размеры для заголовков (минимум 3 уровня).

Отступов: переменные для маленького, среднего, большого и глобального отступов (не менее 5 значений).

Радиусов скругления: для маленьких (иконки), средних (кнопки), больших (карточки) элементов.

4.3. Глобальные стили

В файле `variables.css` (или отдельном файле) задайте:

Базовый цвет текста для всей страницы.

Базовый цвет фона для всей страницы.

Базовое семейство шрифтов для всей страницы.

Стили для заголовков (размеры, начертание, отступы).

Задание 5. Построение макета страницы с помощью Grid

5.1. Создайте глобальную сетку

В файле `layout.css` создайте класс для основного контейнера страницы (например, `.app` или `.page`) и задайте ему:

`display: grid.`

Определение областей (`grid-template-areas`) с использованием именованных областей согласно схеме, которую вы нарисовали в задании 2.3.

Определение колонок (`grid-template-columns`): укажите ширину сайдбара (из анализа макета) и долю для основного контента (1fr).

Определение строк (`grid-template-rows`): шапка, основной контент, подвал.

5.2. Назначьте области блокам

Присвойте каждому блоку (`header`, `sidebar`, `main`, `footer`) свойство `grid-area` с соответствующим именем из `grid-template-areas`.

5.3. Установите минимальную высоту

Сделайте так, чтобы страница занимала всю высоту окна браузера (минимум 100vh).

5.4. Добавьте отступы

Задайте глобальные отступы (`padding`) для основного контейнера, чтобы контент не прилипал к краям экрана. Используйте переменные из `variables.css`.

Задание 6. Верстка шапки и навигации (Flexbox)

6.1. Структура шапки

Внутри блока `<header class="header">` создайте:

Блок с логотипом (текст или изображение).

Навигационное меню (`<nav>`) со списком ссылок (не менее 4 пунктов).

Блок с иконками пользовательских действий (например, поиск, профиль, корзина). Используйте ваши SVG-иконки из лабораторной работы №2.

6.2. Стилизация шапки

С помощью Flexbox:

Расположите все элементы шапки в одну строку.

Выводите их по вертикали по центру.

Сделайте так, чтобы меню занимало всё свободное пространство между логотипом и блоком иконок (используйте `margin-left: auto` или `justify-content: space-between`).

Добавьте отступы между элементами меню и между иконками.

Задайте нижнюю границу (border-bottom) для шапки.

6.3. Стилизация ссылок и иконок

Оформите ссылки меню: цвет, размер шрифта, отступы.

Добавьте эффект при наведении на ссылки (изменение цвета или подчеркивание).

Для иконок задайте размеры (24×24 или согласно макету) и цвет.

Задание 7. Верстка сайдбара

7.1. Содержимое сайдбара

В зависимости от вашего макета, в блоке `<aside class="sidebar">` создайте один из вариантов:

Вариант А: Вертикальное меню навигации (список ссылок с иконками).

Вариант Б: Блок фильтров (чекбоксы, радиокнопки, диапазон цен).

Вариант В: Профиль пользователя (аватар, имя, статистика).

7.2. Стилизация сайдбара

Используйте Flexbox с направлением column для вертикального расположения элементов.

Задайте фон сайдбара (если он отличается от основного).

Добавьте внутренние отступы.

При необходимости добавьте разделители между элементами.

Задание 8. Верстка основного контента

8.1. Заголовок секции

В блоке `<main class="main">` создайте заголовок секции (например, «Популярные товары» или «Новости»). Оформите его согласно макету.

8.2. Сетка карточек (Grid)

Создайте контейнер для карточек (например, `.cards-grid`) и задайте ему: `display: grid`.

Адаптивную сетку колонок: используйте `repeat(auto-fill, minmax(минимальная_ширина_карточки, 1fr))`. Минимальную ширину возьмите из анализа макета.

Расстояние между карточками (gap) согласно макету.

8.3. Карточка товара (Flexbox)

Создайте карточку товара, которая содержит:

Изображение товара (заглушка или реальное изображение).

Название товара.

Описание (краткое, 1–2 строки).

Цену.

Кнопку «Купить» или «Подробнее».

8.4. Стилизация карточки

С помощью Flexbox (направление column):

Организуйте вертикальное расположение элементов внутри карточки.

Сделайте так, чтобы кнопка всегда прижималась к нижнему краю карточки (используйте `margin-top: auto`).

Добавьте скругление углов, тень при наведении, плавный переход (transition).

Ограничьте высоту изображения (например, через aspect-ratio: 4/3).

8.5. Создание нескольких карточек

Создайте не менее 4–6 карточек (можно дублировать содержимое для демонстрации сетки).

Задание 9. Верстка подвала (footer)

9.1. Содержимое подвала

В блоке `<footer class="footer">` создайте:

Несколько колонок (от 2 до 4) с информацией: контакты, ссылки на соцсети, полезные ссылки, копирайт.

Иконки соцсетей (можно использовать ваши SVG-иконки или новые).

9.2. Стилизация подвала

Используйте Grid или Flexbox для распределения колонок по горизонтали.

Задайте фон подвала (обычно темнее основного).

Измените цвет текста на контрастный.

Добавьте внутренние отступы.

Разместите копирайт по центру или в отдельной строке.

Задание 10. Адаптивная верстка

10.1. Мобильная версия (до 768px)

Напишите медиа-запрос для экранов шириной до 768 пикселей. В нем:

Измените глобальную сетку страницы: уберите сайдбар (или переместите его под основной контент).

Сделайте шапку гибкой: логотип сверху, меню — на следующей строке, или используйте бургер-меню (на ваше усмотрение).

Измените сетку карточек на одну колонку (1 fr).

Уменьшите отступы (padding, gap) в 1.5–2 раза.

Уменьшите размеры шрифтов для заголовков.

10.2. Планшетная версия (768px – 1024px)

Напишите медиа-запрос для экранов шириной от 768 до 1024 пикселей:

Сетка карточек: 2 колонки.

Сайдбар может оставаться, но меньшей ширины.

Отступы — промежуточные между мобильной и десктопной версией.

10.3. Десктопная версия (1024px и более)

Убедитесь, что стили вне медиа-запросов (базовые) соответствуют десктопному макету.

Задание 11. Интеграция иконок из предыдущей работы

11.1. Добавьте иконки в навигацию

Вставьте ваши SVG-иконки (из лабораторной работы №2) в следующие места:

Иконка «Дом» (Home) — рядом с пунктом меню «Главная».

Иконка «Сердце» (Heart) — в блок избранного или лайков.

Иконка «Корзина» (Trash) — в корзину товаров (или в настройки).

Иконка «Шестеренка» (Settings) — в профиль или настройки пользователя.

11.2. Обеспечьте стилизацию иконок

Сделайте так, чтобы иконки:

Имели одинаковый размер (24×24 пикселя).

Меняли цвет при наведении (если это контурные иконки).

Были выровнены по вертикали с текстом (если стоят рядом).

11.3. Проверьте отображение

Убедитесь, что все иконки корректно отображаются на всех размерах экрана.

Выполнить все представленные выше задания, сохраняя промежуточные результаты.

Использовать свои изображения, либо применять другие цвета при обработке, чтобы работа была уникальной. Оформить отчет к лабораторной работе:

Титульный лист;

Цель, задачи работы;

Этапы выполнения работы.

Ответы на контрольные вопросы.

Контрольные вопросы:

Дайте развернутые письменные ответы (по 3–5 предложений на каждый):

1. В чем заключается принципиальная разница между Flexbox и CSS Grid? Приведите примеры ситуаций, когда предпочтительнее использовать Flexbox, а когда — Grid.
2. Почему в современной веб-разработке рекомендуется использовать семантические теги (<header>, <main>, <article>, <section>) вместо универсальных <div>? Назовите не менее трех причин (SEO, доступность, поддерживаемость).
3. Что делает CSS-свойство box-sizing: border-box и почему оно стало негласным стандартом? Опишите, как меняется расчет ширины элемента по сравнению со значением content-box.
4. Объясните разницу между подходами «Mobile First» и «Desktop First» к адаптивной верстке. В каких случаях какой подход предпочтительнее?

ЛАБОРАТОРНАЯ РАБОТА № 7

Анимация интерфейсов. CSS animation, transition. Реализация анимаций элементов.

Цель лабораторной работы: Освоить базовые механизмы CSS-анимации: transition и @keyframes. Научиться применять анимацию к UI-элементам (кнопки, карточки, меню, модальные окна). Понять принципы микро-анимации и её влияние на пользовательский опыт (UX).

Задачи лабораторной работы:

Реализовать несколько типов анимаций на ранее сверстанной странице

Формируемые компетенции: ПК-2, ПК-4

Теоретическая часть

Исторически веб-интерфейсы были статичными: пользователь нажимал на кнопку и видел мгновенное изменение (или, что хуже, ничего не видел). С развитием технологий и повышением ожиданий пользователей анимация стала неотъемлемой частью качественного интерфейса.

Ключевая идея: Анимация — это не украшение, а функциональный инструмент взаимодействия.

Принципы хорошей анимации

1. **Скорость.** Анимация должна быть достаточно быстрой, чтобы не заставлять пользователя ждать. Оптимальный диапазон: 100–300 миллисекунд. Анимация длительностью более 500 миллисекунд начинает восприниматься как торможение.
2. **Естественность.** Движения элементов должны подчиняться физическим законам: ускорение, замедление, инерция. Резкие, линейные движения выглядят неестественно и «роботизированно».
3. **Целенаправленность.** Каждая анимация должна отвечать на вопрос: «Какую проблему пользователя она решает?» Анимация ради анимации — это визуальный шум.
4. **Ненавязчивость.** Анимация не должна отвлекать от основного действия. Если пользователь не заметил анимацию — это хорошо. Если она его раздражает — это плохо.
5. **Доступность.** Некоторые пользователи (с вестибулярными расстройствами, мигренью, эпилепсией) испытывают физический дискомфорт от движущихся элементов. Интерфейс должен уважать их выбор.

В CSS существуют два принципиально разных механизма для создания анимации. Их понимание — ключ к правильному выбору инструмента.

2.1. Transition (переход)

Что это: Механизм, который позволяет свойству CSS плавно изменяться при изменении его значения.

Когда срабатывает: Только при наличии **триггера** — события, которое меняет значение свойства. Чаще всего это:

- Наведение курсора (:hover)
- Фокус на элементе (:focus)
- Добавление или удаление CSS-класса через JavaScript
- Изменение состояния элемента (например, :checked для чекбокса)

Как работает: Браузер вычисляет начальное и конечное значение свойства и «дорисовывает» все промежуточные состояния за указанное время.

Характеристики:

- **Плавность:** Да (за счет интерполяции)
- **Количество этапов:** Только 2 (начало и конец)
- **Цикличность:** Нет (однократный переход)
- **Управление из JS:** Требуется для смены состояния

Типичные применения:

- Изменение цвета кнопки при наведении
- Появление тени у карточки
- Плавное открытие выпадающего списка
- Анимация иконки при наведении (поворот, изменение цвета)

Что можно анимировать через transition:

- opacity — прозрачность
- transform — масштаб, поворот, сдвиг
- color, background-color — цвета
- box-shadow, border-color, outline
- width, height, margin, padding (но с осторожностью)

Что нельзя анимировать через transition:

- display (нет промежуточных значений)
- visibility (только дискретно: видно/не видно)
- font-family (невозможно интерполировать)
- background-image (кроме градиентов)

2.2. Keyframes (ключевые кадры)

Что это: Механизм, который позволяет задать произвольное количество промежуточных состояний анимации.

Когда срабатывает: Может запускаться:

- Автоматически при загрузке страницы
- При добавлении класса через JavaScript
- Бесконечно (в цикле)
- При наведении (если привязать к :hover)

Как работает: Разработчик описывает ключевые точки анимации (обычно 0%, 50%, 100%), а браузер интерполирует состояния между ними.

Характеристики:

- **Плавность:** Да
- **Количество этапов:** Любое (2 и более)
- **Цикличность:** Да (можно повторять бесконечно)
- **Управление из JS:** Требуется для запуска/остановки

Типичные применения:

- Анимация появления элемента при загрузке страницы (fade in, slide in)
- Бесконечные анимации (спиннер, пульсация, мерцание)
- Сложные многоэтапные эффекты (пружинка, bounce, shake)
- Каскадные анимации (элементы появляются по очереди)

Ключевое отличие от transition: Keyframes не требуют триггера — анимация может начаться сразу после загрузки страницы или продолжаться бесконечно.

Правило выбора:

- Нужна анимация при наведении на кнопку? → **transition**
- Нужно, чтобы элементы плавно появились при загрузке? → **keyframes**
- Нужна бесконечная анимация (спиннер)? → **keyframes**
- Нужна анимация ошибки (встряска формы)? → **keyframes**

Свойство will-change сообщает браузеру, какое свойство будет анимироваться. Браузер заранее создает отдельный слой для элемента, что повышает производительность.

Когда использовать:

- Для элементов, которые анимируются часто или долго
- Для сложных анимаций (параллакс, drag-and-drop)

Когда НЕ использовать:

- Для элементов, которые анимируются редко
- Для всех элементов подряд (создание лишних слоев consumes memory)

Современные операционные системы (iOS, Android, Windows, macOS) позволяют пользователю включить настройку «Уменьшить движение» (Reduce Motion). CSS может это обнаружить.

Как это работает:

- Браузер получает сигнал от ОС о предпочтениях пользователя.
- CSS-медиа-запрос prefers-reduced-motion позволяет применить альтернативные стили.

Что нужно сделать разработчику:

- Для всех анимаций (особенно движущихся, вращающихся, параллаксных) предусмотреть вариант без движения или с минимальным движением.
- Отключать анимацию или заменять её на fade (прозрачность) — он не вызывает дискомфорта

Методика и порядок выполнения работы

Используйте сверстанную вами страницу (макет из Figma, который вы верстали в ЛР №3). Убедитесь, что страница корректно открывается в браузере и все элементы отображаются правильно.

Если по какой-то причине у вас нет готового проекта из ЛР №3, используйте предоставленный преподавателем шаблон или создайте простую страницу-визитку с минимальным набором элементов: шапка, навигация, 4–6 карточек, кнопки, подвал.

3.2. Создайте новый CSS-файл для анимаций

В папке `css/` вашего проекта создайте новый файл с именем `animations.css`. Подключите этот файл в `index.html` **после** всех остальных CSS-файлов (`reset`, `variables`, `layout`, `components`), чтобы стили анимаций имели приоритет и не переопределялись другими правилами.

3.3. Составьте план анимаций

Перед тем как приступить к коду, составьте письменный план. Перечислите все элементы вашей страницы, которые вы будете анимировать, и для каждого укажите:

- Какой механизм будете использовать (`transition` или `keyframes`)
- Какое свойство будете анимировать
- Какова будет длительность
- Какова будет функция ускорения

Реализуйте анимацию, при которой вся страница (или её основной контейнер) плавно появляется при загрузке:

Начальное состояние (0%): элемент полностью прозрачен.

Конечное состояние (100%): элемент полностью видим.

Длительность анимации — 0.5–0.7 секунды.

Функция ускорения — плавное появление (`ease-out`).

Анимация должна запускаться автоматически при загрузке страницы.

Создайте модальное окно (всплывающее окно), которое появляется при нажатии на кнопку «Подробнее» или другую подходящую кнопку на вашей странице:

При открытии окно должно появляться с анимацией: плавное увеличение от маленького размера (например, 0.8 от нормального) до нормального, одновременно с появлением из прозрачности.

При закрытии окна анимация должна быть обратной: уменьшение до 0.8 и исчезновение прозрачности.

Фоновое затемнение (`overlay`) должно также появляться и исчезать с анимацией (появление из прозрачности).

Для открытия и закрытия используйте добавление и удаление CSS-классов через JavaScript.

Добавьте кнопку закрытия (крестик) в углу модального окна.

Контрольные вопросы:

1. Создайте модальное окно (всплывающее окно), которое появляется при нажатии на кнопку «Подробнее» или другую подходящую кнопку на вашей странице:
2. При открытии окно должно появляться с анимацией: плавное увеличение от маленького размера (например, 0.8 от нормального) до нормального, одновременно с появлением из прозрачности.

3. При закрытии окна анимация должна быть обратной: уменьшение до 0.8 и исчезновение прозрачности.
4. Фоновое затемнение (overlay) должно также появляться и исчезать с анимацией (появление из прозрачности).
5. Для открытия и закрытия используйте добавление и удаление CSS-классов через JavaScript.
6. Добавьте кнопку закрытия (крестик) в углу модального окна.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное
автономное образовательное
учреждение высшего
образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по организации самостоятельной работы обучающихся
по дисциплине **«Технологии мультимедиа в разработке интерфейсов»**
Направление подготовки 09.03.02 «Информационные системы и
технологии»
Направленность (профиль) «Прикладное программирование и
интернет-технологии»
Квалификация выпускника: бакалавр

Ставрополь

2026

Введение

1. Самостоятельная работа студентов на лабораторных занятиях
- 1.1. Самостоятельная работа студентов на аудиторных занятиях 1.2. Самостоятельная работа студентов во внеаудиторное время.
2. Методические указания студентам по подготовке к собеседованию и материалов по темам самостоятельной работы
3. Критерий оценки самостоятельной работы студента
4. Заключение Приложения

Введение

Подготовка бакалавра – сложный, комплексный процесс, в котором органически сочетается решение всех основных задач – образовательных, воспитательных и развивающих. Деятельность преподавателя в учебном процессе – ведущая. Однако, как бы ни были совершенны применяемые преподавателем в учебном процессе форм и методов обучения, без активной познавательной деятельности самих обучаемых они не могут давать желаемых результатов. Поэтому особое значение приобретает проблема сознательного отношения самих студентов к приобретению знаний, умений и навыков в результате самостоятельной работы.

Самостоятельная работа - это познавательная деятельность, выполняемая студентами самостоятельно, при частичном руководстве преподавателя. Самостоятельная работа предусмотрена учебной программой. Самостоятельная работа является управляемым процессом. Благодаря этому познавательно-практическая деятельность студентов становится целенаправленной, творчески осмысленной, содержательной.

Процесс управления самостоятельной работы служит главным целям обучения:

- усвоение, закрепление, совершенствование знаний в объеме вузовских программ;
- приобретение умений и навыков, составляющих содержание подготовки выпускника высшей школы;
- максимальная активность каждого обучающегося в овладении профессиональных компетенций (ПК-2, 4), формируемых в результате освоения дисциплины.

1. Самостоятельная работа студентов на лабораторных и практических занятиях

1.1. Аудиторные занятия

Задачи:

- • расширение, углубление и детализация знаний, полученных на лекциях;
- • повышение уровня усвоения учебного материала (от уровня информации,
- полученной на лекциях, до уровня знаний за счет привития умений и навыков);
- • развитие научного мышления и речи студентов;
- • проверка и учет знаний;
- • развитие познавательной активности и привитие навыков самостоятельной
- работы, особенно с дополнительной и специальной литературой;
- • привитие навыков ведения коллективной беседы, участия в творческой
- дискуссии, умения аргументированно отстаивать свои взгляды.

Основное требование к подготовке и участию студентов в занятиях заключается в том, что студент должен самостоятельно готовиться к занятиям и творчески в них участвовать.

Этапы подготовки к занятиям включают:

- повторение уже имеющихся знаний по конспекту, а затем по учебнику;
- углубление знаний по теме с использованием рекомендованной литературы;
- выполнение конкретного задания (решение задач, составление отчетов и т.п.).

Особое место занимают лабораторные работы. Главная цель которых – быть связующим звеном теории изучаемой дисциплины с практической реализацией, что позволяет углублять и закреплять теоретические положения, получаемые студентами на лекции, проверять их применение в практике экспериментальным путем, знакомить студентов с оборудованием, приборами, вычислительной техникой, изучать на практике методы научных исследований.

Студенты обеспечиваются методическими указаниями к лабораторным работам, содержащими теоретическую информацию и конкретные практические задания.

Для студентов младших курсов указания даются подробно и детализировано. Для студентов старших курсов предусматривается большая самостоятельность при выполнении заданий. Т.е. они должны уже демонстрировать переход от уровня умений к уровню навыков.

Оформление лабораторных работ должно быть максимально приближено к уровню, на котором ведется экспериментальная научно-исследовательская работа в конкретной предметной области.

1.2. Самостоятельная работа студентов во внеаудиторное время.

Кроме наиболее распространенных в вузах форм самостоятельной работы во внеаудиторное время (написание рефератов, контрольных работ, курсовых работ/проектов и т.д.) учебные планы включают такие виды и соответственно такой компонент фонда оценочных средств, как собеседование (по конкретным вопросам) и выполнение предлагаемых тем для самостоятельной учебно-исследовательской работы.

2. Указания студентам по подготовке к собеседованию и материалов по темам самостоятельной работы

2.1. Указания представлены в виде рекомендаций студентам в ходе изучения вопросов, выносимых на собеседование и материалов тем самостоятельной работы.

Своевременное и качественное выполнение самостоятельной работы базируется на соблюдении настоящих рекомендаций и изучении рекомендованной литературы. Студент может дополнить список использованной литературы современными источниками, не представленными в списке рекомендованной литературы, и в дальнейшем использовать собственные подготовленные учебные материалы при написании рефератов, курсовых и выпускных квалификационных работ.

2.2. Рекомендации

2.2.1. Прежде чем приступить к изучению вопросов для собеседования и предложенной темы самостоятельной работы, необходимо.

Во-первых:

- уяснить суть вопроса (темы) на самостоятельную работу;
- провести подбор рекомендованной литературы;
- составить план работы, в котором определяются основные пункты предстоящей подготовки.

Составление плана дисциплинирует и повышает организованность в работе.

Во-вторых:

- начинать надо с изучения рекомендованной литературы.

Необходимо помнить, что на лекции обычно рассматривается не весь материал, а только его часть. Остальная его часть восполняется в процессе самостоятельной работы. В связи с этим работа с рекомендованной литературой обязательна (см. Приложение 1). Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов.

В процессе этой работы студент должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале.

Заканчивать подготовку следует составлением плана (оглавления) по изучаемому материалу. Это позволит дать целостный, сжатый ответ при собеседовании и представить целостную картину по предложенной тематике.

В процессе самостоятельной работы рекомендуется взаимное обсуждение материала (между студентами), во время которого закрепляются знания, а также приобретается практика в изложении и разъяснении полученных знаний, развивается речь.

При необходимости студент может обращаться за консультацией к преподавателю. Идя на консультацию, студенту необходимо хорошо продумать вопросы, которые требуют разъяснения.

При подготовке к собеседованию и выполнению самостоятельной работы по теме студент должен вести записи. В результате у него создается свой индивидуальный фонд дополнительных материалов для быстрого повторения прочитанного, для накопления знаний. Особенно важны и полезны записи тогда, когда в них находят отражение мысли, возникшие при самостоятельной работе.

Основные формы записи: план (простой и развернутый), выписки, тезисы.

Результаты этих записей могут быть представлены в различных формах.

План – это схема прочитанного материала, краткий (или подробный) перечень вопросов, отражающих структуру и последовательность материала:

краткий

- воспроизведение наиболее важных положений и фактов источника;
- *подробный (развернутый)*
- детализированный план, в котором достаточно подробные записи приводятся по тем пунктам плана, которые нуждаются в пояснении;
- это четко и кратко изложенные (сформулированные) основные положения в результате глубокого осмысливания материала;
- в нем могут присутствовать выписки, цитаты, тезисы.

Подробно составленный план позволяет эффективно использовать материал и получить желаемый результат.

2.2.2. При отчете студента о проделанной самостоятельной работе по предложенной теме, преподаватель может предложить студенту алгоритм действий, рекомендовать еще раз внимательно прочитать весь материал и все записи по теме самостоятельной работы, тщательно продумать свое устное выступление.

2.2.3. Устный отчет должен строиться свободно, убедительно и аргументированно.

2.2.4. Преподаватель следит, чтобы отчет не сводился к простому воспроизведению текста, не допускался и простое чтение. Необходимо, чтобы студент проявлял собственное отношение к тому, о чем он говорит, высказывал свое личное мнение, понимание, обосновывал его и мог сделать правильные выводы из сказанного. При этом студент может обращаться к записям собранного материала, использовать знания факты и наблюдения современной жизни и т. д.

2.2.5. В заключение преподаватель оценивает результаты работы студента по данной теме.

3. Критерий оценки самостоятельной работы студента

Оценка «зачтено» при ответах на вопросы при собеседовании и по итогам устной защиты выполненной самостоятельной работы выставляется студенту, если студент показал знание учебного материала. При этом возможно допустил неточные формулировки основных понятий и терминов или ряд незначительных неточностей, не исказивших существенно суть ответа.

Оценка «не зачтено» при ответах на вопросы при собеседовании и по итогам устной защиты выполненной самостоятельной работ выставляется студенту, если студент не знает значительной части учебного материала, допускает неправильную формулировку основных понятий и терминов, существенно исказившие суть вопроса.

Оценка «не зачтено» выставляется также, если студент не отвечает на вопросы при собеседовании, отсутствует план выполненной работы, нет объяснений основных понятий и терминов.

4. Заключение

Успешное освоение курса предполагает активное, творческое участие студента путем планомерной, повседневной работы как на лекциях, лабораторных (практических) занятиях, так и к подготовке ответов на вопросы к собеседованию и выполнению самостоятельной работы по предлагаемым темам.

Приложение 1

Рекомендации студентам по изучению рекомендованной литературы

Изучение дисциплины следует начинать с проработки рабочей программы по изучаемой дисциплине, особое внимание, уделяя целям и задачам, структуре и содержанию курса.

Студентам рекомендуется получить учебную литературу по дисциплине, необходимую для эффективной работы на всех видах аудиторных занятий, а также для самостоятельной работы по изучению дисциплины.

Приложение 2

Примерная тематика заданий для самостоятельной работы студентов

Студенты выполняют самостоятельную работу в соответствии с индивидуальным заданием, которое определяется примерными темами.

Примерные темы самостоятельной работы

1. Что такое мультимедиа технологии и как они используются в современном мире?

2. Какие виды мультимедиа контента существуют и как они отличаются друг от друга?

3. Какие программные и аппаратные средства используются для создания и обработки мультимедиа контента?
4. Какие принципы лежат в основе дизайна мультимедиа проектов и какие инструменты используются для их создания?
5. Какие технологии используются для передачи мультимедиа контента через сеть интернет?
6. Какие проблемы могут возникнуть при создании и использовании мультимедиа контента и как их можно решить?
7. Какие тенденции развития мультимедиа технологий можно наблюдать в настоящее время и как они повлияют на будущее этой отрасли?
8. Какие профессии связаны с мультимедиа технологиями и какие навыки нужны для работы в этой сфере?
9. Какие примеры успешных мультимедиа проектов можно назвать и какие уроки можно извлечь из их опыта?
10. Какие перспективы развития мультимедиа технологий можно предсказать на ближайшие 5-10 лет?