

## Рекомендации по организации изучения учебного предмета «Информатика» на повышенном уровне (VIII, IX классы)

На II ступени общего среднего образования учебный предмет «Информатика» может изучаться на повышенном уровне. Дополнительные учебные часы (1 или 2 часа в неделю) целесообразно использовать для стимулирования учебной деятельности учащихся по овладению знаниями, умениями и навыками, для развития индивидуальных качеств, творческих способностей и формирования на этой основе предметных, метапредметных и личностных компетенций.

### VIII КЛАСС

Общее количество часов:

- на базовом уровне – 35, из них: резерв – 1; контрольная работа – 1;
- на повышенном уровне (добавление 1 часа) – 70, из них: резерв – 1; контрольная работа – 1;
- на повышенном уровне (добавление 2 часов) – 105, из них: резерв – 1; контрольная работа – 1.

Для изучения учебного предмета «Информатика» на повышенном уровне рекомендуется использовать следующую литературу:

Информатика. 8–9 классы. Дидактические и диагностические материалы : пособие для учителей учреждений общ. среднего образования с белорус. и рус. языками обучения / С. И. Зенько [и др.] ; под ред. С. И. Зенько. – Мозырь : Выснова, 2018. – 191 с. – (Компетентностный подход).

### ТЕМА 1. ОСНОВЫ АНИМАЦИИ

Количество часов на изучение темы:

- на базовом уровне – 6
- на повышенном уровне (добавление 1 часа) – 8 (6+2)
- на повышенном уровне (добавление 2 часов) – 10 (6+4)

Содержание работы предполагают разработку плана анимационного ролика, определение параметров форматирования текста, образца движущегося предмета или составного объекта, траектории его движения и поверхности могут быть выработаны совместно с учащимися (при модерации обсуждения учителя) по аналогии с материалами приложения 1; знакомство с примерами *stop motion* (кукольной анимации), моушен-дизайна и др.

Возможные темы проектов: «Анимированные логотипы», «Анимированный мультфильм», «Правила дорожного движения», «Правила поведения в компьютерном классе» и др.

### ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ (ПОВЫШЕННЫЙ УРОВЕНЬ)

Учащиеся

*знают* основные понятия анимации – «анимация», «компьютерная

анимация», «библиотека объектов», «типы символов» («графика», «кнопка», «клип»);

*умеют* разрабатывать план анимационного ролика и реализовывать его средствами редактора для создания анимации;

*владеют* приемами, позволяющими зациклить ролик корректно (прием совпадения первого и последнего кадров; прием исчезновения объекта из поля зрения и др.)

## **ТЕМА 2. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

Количество часов на изучение темы:

- на базовом уровне – 13
- на повышенном уровне (добавление 1 часа) – 39 (13+26)
- на повышенном уровне (добавление 2 часа) – 57 (13+44)

Содержание работы предполагает прохождение ряда этапов: определение исходных данных, определение результатов, составление алгоритма, описание переменных, написание программы, тестирование программы.

Сюжеты простейших компьютерных игр и практико-ориентированных заданий могут быть сформулированы учителем либо совместно с учащимися.

В качестве образца можно использовать примеры из приложения 2.

### **ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ (ПОВЫШЕННЫЙ УРОВЕНЬ)**

Учащиеся

*знают* понятия «алгоритмическая конструкция “Следование”», «алгоритмическая конструкция “Ветвление”», «алгоритмическая конструкция “Повторение”»;

*умеют* реализовывать этапы решения задач на компьютере;

*владеют* приемами разработки алгоритмов для решения прикладных задач (простейших компьютерных игр) на основе использования алгоритмических конструкций «Следование», «Ветвление» и «Повторение».

## **ТЕМА 3. ТЕХНОЛОГИЯ ОБРАБОТКИ ТЕКСТОВЫХ ДОКУМЕНТОВ**

Количество часов на изучение темы:

- на базовом уровне – 9
- на повышенном уровне (добавление 1 часа) – 13 (9+4)
- на повышенном уровне (добавление 2 часа) – 20 (9+11)

Содержание работы включает: редактирование текста; создание и форматирование списков; вставку таблиц, формул и рисунков; иллюстрирование текстового документа; использование стилей.

Тексты могут быть подготовлены учителем или взяты из приложения 3.

## ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ (ПОВЫШЕННЫЙ УРОВЕНЬ)

Учащиеся

*знают* понятия «список», «стиль»; виды списков и стилей;

*умеют*:

осуществлять поиск и замену в тексте;

выполнять автоматическую проверку правописания;

размещать в текстовом документе различные объекты;

*владеют* приемами стилевого форматирования заголовков и генерации оглавления документа.

Контрольная работа по теме 3 (1 час)

## ТЕМА 4. ТЕХНОЛОГИЯ ОБРАБОТКИ АУДИО- И ВИДЕОИНФОРМАЦИИ

Количество часов на изучение темы:

- на базовом уровне – 5
- на повышенном уровне (добавление 1 часа) – 8 (5+3)
- на повышенном уровне (добавление 2 часов) – 16 (5+11)

Содержание этапов работы над аудио- и видеоклипами по темам и технические требования к итоговому продукту могут быть выработаны совместно с учителем по аналогии с примерами, приведенными в приложении 4.

Возможные темы проектов: «Цифровая музыка», «Моя аудиокнига», видеообъяснение «Польза и вред компьютерных игр», «Мои земляки...» и др.

## ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ (ПОВЫШЕННЫЙ УРОВЕНЬ)

Учащиеся

*знают* понятия «аудиоинформация», «видеоинформация», «аудиофайл», «видеофайл», «конвертация аудиофайла», «конвертация видеофайла»;

*умеют*:

конвертировать аудио- и видеозаписи средствами свободного программного обеспечения;

создавать фонограммы к видеофильму (видеоклипу) для случаев, когда имеющаяся музыкальная композиция по длительности больше или меньше соответствующего видеофильма;

*владеют* приемами интегрированного использования прикладных программ по обработке аудио- и видеозаписей.

## IX КЛАСС

Общее количество часов:

- на базовом уровне – 34, из них: резерв – 1; контрольная работа – 1;
- на повышенном уровне (добавление 1 часа) – 68, из них: резерв – 1; контрольная работа – 1;
- на повышенном уровне (добавление 2 часов) – 102, из них: резерв – 1; контрольная работа – 1.

Для изучения учебного предмета «Информатика» на повышенном уровне рекомендуется использовать следующую литературу:

Информатика. 8–9 классы. Дидактические и диагностические материалы : пособие для учителей учреждений общ. среднего образования с белорус. и рус. языками обучения / С. И. Зенько [и др.] ; под ред. С. И. Зенько. – Мозырь : Выснова, 2018. – 191 с. – (Компетентностный подход).

### ТЕМА 1. ИНФОРМАЦИОННЫЕ РЕСУРСЫ СЕТИ ИНТЕРНЕТ

Количество часов на изучение темы:

- на базовом уровне – 5
- на повышенном уровне (добавление 1 часа) – 8 (5+3)
- на повышенном уровне (добавление 2 часов) – 13 (5+8)

Содержание этапов работы предполагает применение электронных словарей сети Интернет, различных поисковых сервисов и облачных технологий для нахождения, анализа и структуризации данных, направленных на подготовку сообщений (докладов и др.), носящих межпредметный практико-ориентированный характер.

Требования к итоговому продукту могут быть выработаны совместно с учителем по аналогии с приведенными в приложении 5.

Возможные темы проектов: «Правила поведения при использовании электронной почты», «Киберпреступность и методы борьбы с ней», «Выдающиеся белорусы», «Лексическое значение слов, обозначающих противоправные действия в интернете» и др.

### ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ (ПОВЫШЕННЫЙ УРОВЕНЬ)

Учащиеся

*знают* понятия «IP-адрес», «доменное имя», «URL-адрес», «информационный образовательный ресурс», «спам», «облачные технологии» и правила сетевого этикета;

*умеют*

создавать безопасные пароли;

использовать облачные технологии для реализации совместной работы над проектом;

*владеют* приемами, направленными на соблюдение правил сетевого этикета и мер безопасного поведения в интернете.

## ТЕМА 2. АЛГОРИТМЫ ОБРАБОТКИ СТРОКОВЫХ ВЕЛИЧИН

Количество часов на изучение темы:

- на базовом уровне – 8
- на повышенном уровне (добавление 1 часа) – 20 (8+12)
- на повышенном уровне (добавление 2 часов) – 34 (8+26)

Содержание работы предполагает прохождение ряда этапов: определение исходных данных, определение результатов, составление алгоритма, описание переменных, написание программы, тестирование программы.

Сюжеты практико-ориентированных заданий могут быть сформулированы учителем либо совместно с учащимися. В качестве образца можно использовать задания, приведенные в приложении 6.

### ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ (ПОВЫШЕННЫЙ УРОВЕНЬ)

Учащиеся

*знают* основные стандартные процедуры и функции для работы со строковыми величинами;

*умеют* реализовывать этапы решения задач по обработке текстовых данных на компьютере.

## ТЕМА 3. ОБРАБОТКА ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

Количество часов на изучение темы:

- на базовом уровне – 10
- на повышенном уровне (добавление 1 часа) – 19 (10+9)
- на повышенном уровне (добавление 2 часов) – 24 (10+14)

Содержание работы включает обработку данных в электронных таблицах при выполнении практических и проектных заданий с использованием таких функций табличного процессора, как условное форматирование, подбор параметра и расширенный фильтр. План составления расчетных таблиц, их оформление и предполагаемые формы представления результатов могут быть разработаны совместно с учащимися — при модерации обсуждения со стороны учителя — на основе аналогий с примерами, приведёнными в приложении 7.

Возможные темы проектов: «Водоснабжение», «Электроэнергия», «Отопление», «Транс-порт», «Автомобиле-строение», «Интернет-трафик». «Работа школьного кафе» и др.

### ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ (ПОВЫШЕННЫЙ УРОВЕНЬ)

Учащиеся

*знают* понятия «формула», «ссылка», «виды ссылок» (относительная, абсолютная, смешанная), «сортировка», «фильтр»; принцип относительной

адресации;

*умеют* редактировать и форматировать таблицы, созданные средствами табличного процессора;

*владеют* приемами, позволяющими создавать и анализировать расчетные таблицы с учетом особенностей предметной области.

Контрольная работа по теме 3 (1 час)

#### **ТЕМА 4. КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ МОДЕЛИ**

Количество часов на изучение темы:

- на базовом уровне – 9
- на повышенном уровне (добавление 1 часа) – 19 (9+10)
- на повышенном уровне (добавление 2 часов) – 29 (9+20)

Содержание работы прохождения ряда этапов: постановка задачи, разработка плана создания модели, выбор средства для создания модели, создание модели, исследование модели и получение решения задачи.

Важно уделять внимание построению как 2D-моделей, так и 3D-моделей. При наличии возможностей модели могут быть распечатаны (в том числе и на 3D-принтере).

Сюжеты практико-ориентированных заданий могут быть сформулированы учителем либо совместно с учащимися. В качестве образца можно использовать материалы, приведенные в приложении 8.

#### **ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ (ПОВЫШЕННЫЙ УРОВЕНЬ)**

Учащиеся

*знают* понятия «информационная модель», «виды информационных моделей» (мыслительная, документальная, компьютерная) и средства создания компьютерных моделей;

*умеют* отбирать необходимые средства для создания компьютерных информационных моделей, конструировать и исследовать с их помощью компьютерные информационные модели.

## ПРИЛОЖЕНИЕ 1

### Основы анимации: методические рекомендации

#### Задание 1. Создание анимации формы

Задание предназначено для выработки навыков создания анимации превращения формы. Необходимо обратить внимание на точность при редактировании контура фигуры, иначе автоматическая анимация будет выглядеть крайне неаккуратно. Удобными инструментами, обеспечивающими возможность точной работы, являются **Сетка** и **Направляющие**.

Также необходимо обратить внимание на цикличность воспроизведения ролика и вследствие этого желательность совпадения первого и последнего ключевого кадра.

#### Задания 2 и 3. Создание анимации движения

Задания предназначены для выработки навыков создания анимации движения символов прямолинейно и по траектории. При обращении внимания на цикличность воспроизведения ролика желательно, кроме приема совпадения первого и последнего ключевых кадров, рассмотреть другие решения, позволяющие зациклить ролик корректно. Например, исчезновение объекта из поля зрения.

*Общее замечание.* Проверьте наличие указанных в заданиях файлов в соответствующих папках из предлагаемого к пособию архива. Имена дисков, папок и файлов в задании могут быть изменены в соответствии с системой хранения и изменения файлов, принятой в конкретном компьютерном классе.

### Основы анимации: методические рекомендации

Задания контрольно-диагностических материалов направлены на проверку форсированности практических умений по созданию флеш-анимации.

Предполагаемое время на выполнение заданий контрольно-диагностических материалов – 40 минут: 10 минут на прочтение условия контрольно-диагностической работы и составление плана создания ролика; 30 минут на работу за компьютером. Для соблюдения норм непрерывного времени работы за компьютером учителю необходимо соответствующим образом организовать работу учащихся. Например, составление плана создания ролика осуществляется за партами.

В готовом ролике должны быть отображены основные виды автоматической анимации: анимация превращения формы (*Shape*) и анимация движения (*Motion*). Термины автоматической анимации, анимации превращения формы, анимации движения вписываются в текст ролика.

Для выполнения задания учащимся нет необходимости обладать навыками художественного рисования, поскольку все объекты создаются при помощи инструментов рисования фигур. Точность рисования правильных фигур определяется не приблизительно, а через указание соответствующих значений, отображаемых на панели свойств фигур.

Результатом работы должны быть два файла – *Flash-N-Фамилия-Имя fla* и готовый ролик *Flash-N-Фамилия-Имя.swf*.

Содержание выполненной работы может быть примерно следующим:

- пункты 1–3 – появляется текст по словам;
- пункты 4–5 – из точки появляется круг, круг превращается в квадрат, затем все в обратном порядке (рисунок 1).

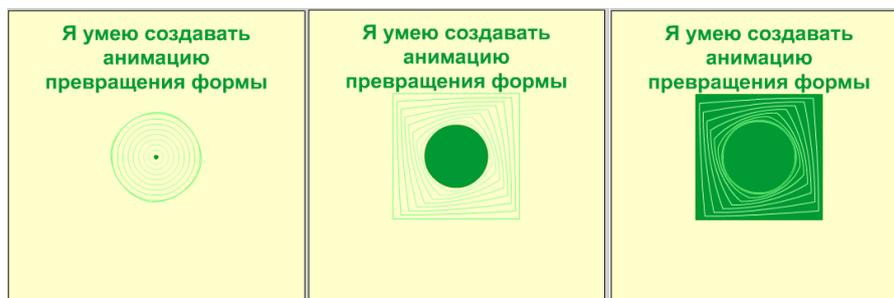


Рисунок 1. – Реализация пунктов 1–5 контрольно-диагностического задания по теме «Основы анимации»

- пункты 6–11 – нарисован статичный мяч, он опускается вниз на поверхность, катается с вращением вправо-влево, затем летает с вращением вправо-влево по дуге (рисунок 2).



Рисунок 2. – Реализации пунктов 9–11 контрольно-диагностического задания по теме «Основы анимации»

### Основы анимации: дидактические материалы

**Задание 1.** Создайте ролик с указанными ниже характеристиками

Создайте новый документ с размером рабочего поля 10 см на 10 см. Установите частоту 12 кадров в секунду. Нарисуйте движение квадратика по прилагаемому образцу в файле *Задание\_1.swf*. Сохраните рабочий файл на диске *D* в папке *VIII класс* под именем *Квадрат fla* и готовый ролик – под именем *Квадрат.swf*.

**Решение.** Выполните следующую последовательность действий:



7. Сохраните файл как *Квадрат.fla* и проверьте ролик – файл *Квадрат.swf*, который сохранится автоматически.

**Задание 2.** *Создайте ролик с указанными ниже характеристиками*

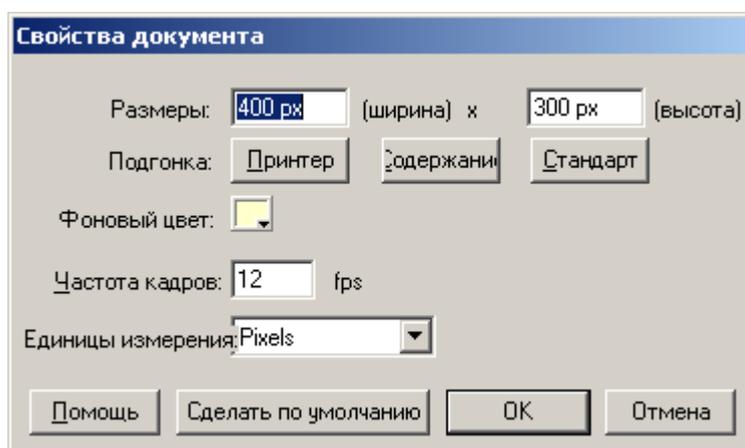
Создайте новый документ с размером рабочего поля 400 px на 300 px. Установите частоту 12 кадров в секунду. Нарисуйте движение шариков по прилагаемому образцу в файле *Задание\_2.swf*.

Сохраните рабочий файл на диске *D* в папке *VIII класс под именем Движение1.fla* и готовый ролик – под именем *Движение1.swf*.

**Решение.** Выполните следующую последовательность действий:

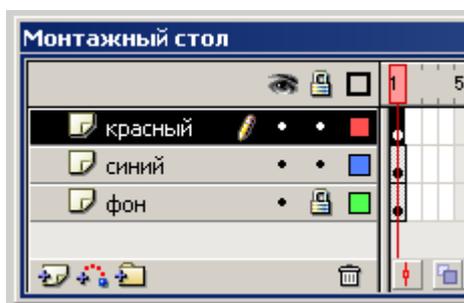
1. Откройте редактор флеш-анимации. Настройте размеры документа 400 px на 300 px, предварительно выбрав единицы измерения – пиксели. Настройте частоту воспроизведения – 12 кадров в секунду и выберите фоновый цвет, отличный от белого (рисунок 5).

Количество кадров в ролике может быть 50.



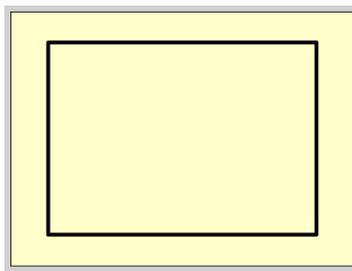
**Рисунок 5.** Установка свойств документа в задании 2 темы «Основы анимации»

2. Для работы над изображением удобно использовать слои. Создайте 3 слоя – *Фон*, *Синий* и *Красный* (рисунок 6).



**Рисунок 6.** Иллюстрация фрагмента монтажного стола с тремя слоями в задании 2 темы «Основы анимации»

3. На слое *Фон* нарисуйте прямоугольник и удалите у него заливку (рисунок 7).

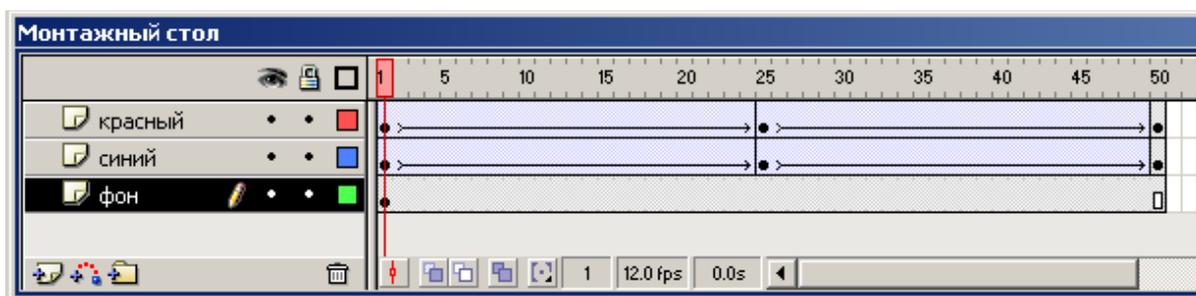


**Рисунок 7.** Изображение прямоугольника без заливки на слое *Фон* в задании 2 темы «Основы анимации»

4. В кадре 1 на слое *Синий* нарисуйте синий шарик и преобразуйте его в символ (используя клавишу *F8*). Шарик разместите в начальном положении – снизу. В кадре 50 вставьте ключевой кадр (используя клавишу *F6*). При этом автоматически будет скопировано содержимое кадра 1.

5. Аналогичным образом на слой *Красный* вставьте красный шарик.

6. На слое *Синий* в кадре 25 вставьте ключевой кадр (используя клавишу *F6*), в котором измените положение синего шарика – переместите его наверх. Пары кадров 1–5 и 25–50 соединяются автоматической анимацией движения – *Motion* (рисунок 8).



**Рисунок 8.** Иллюстрация фрагмента монтажного стола с настройкой автоматической анимации движения в задании 2 темы «Основы анимации»

7. Аналогичным образом в слое *Красный* вставьте в кадре 25 ключевой кадр и измените положение красного шарика.

8. Сохраните файл под названием *Движение1 fla* и проверьте ролик – файл *Движение1.swf*, который создается автоматически.

**Задание 3.** Создайте ролик с указанными ниже характеристиками

Создайте новый документ с размером рабочего поля 700 px на 500 px. Установите частоту 12 кадров в секунду. Нарисуйте движение шарика по прилагаемому образцу в файле *Задание\_3 swf*. Сохраните рабочий файл на диске *D* в папке *VIII класс* под именем *Движение2 fla* и готовый ролик – под именем *Движение2.swf*.

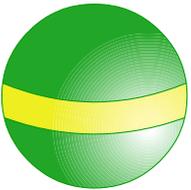
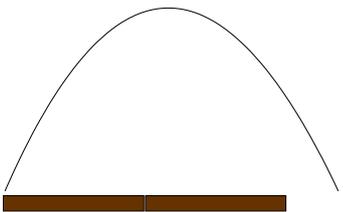
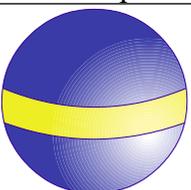
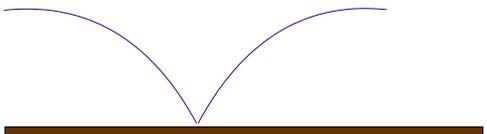
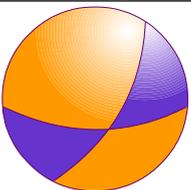
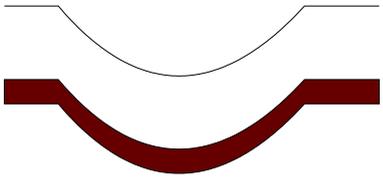
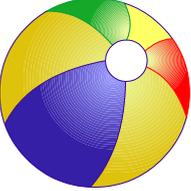
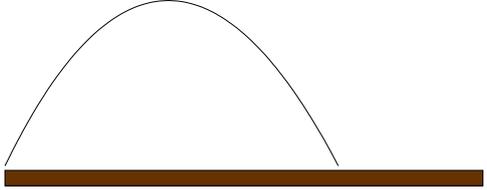
**Описание параметров формирования текста, образцов мяча, траектории движения мяча и поверхности по вариантам:**

**диагностические материалы**

Информация с требованиями к форматированию текста, образцом мяча, траекторией его движения и поверхностью по вариантам представлена в таблице 1.

Таблица 1

### Требования по вариантам к тексту, виду мяча и траектории его движения

Параметры форматирования текста	Образец мяча	Траектория движения мяча и поверхность
Вариант 1.		
<p>шрифт – <i>Arial Cyr</i>,  размер – 32 пт,  цвет темно-зеленый,  стиль начертания – полужирный,  выравнивание – горизонтально по центру</p>		
Вариант 2		
<p>шрифт – <i>Times New Roman Cyr</i>,  размер – 40 пт,  цвет синий,  стиль начертания – полужирный,  выравнивание – горизонтально по центру</p>		
Вариант 3		
<p>шрифт – <i>Courier New Cyr</i>,  размер – 46 пт,  цвет фиолетовый,  стиль начертания – полужирный,  выравнивание – горизонтально по центру</p>		
Вариант 4		
<p>шрифт – <i>Arial Cyr</i>,  размер – 36 пт,  цвет темно-красный,  стиль начертания – полужирный,  выравнивание – горизонтально по центру</p>		

## ПРИЛОЖЕНИЕ 2

### Основы алгоритмизации и программирования: *методические рекомендации*

Предлагаются три задания разных уровней сложности. Выполнение заданий способствует формированию у учащихся умений понимать и анализировать готовый программный код, вносить изменения в программу, самостоятельно составлять алгоритмы и писать программы построения таблиц значений функций с учетом их области определения на указанном промежутке.

#### *Задание 1. Работа с условием задачи и готовым кодом программы*

В задании предлагается использование готового алгоритма решения задачи и готового кода программы, реализующей данный алгоритм. Учащимся предлагается понять, как программа работает, а затем выполнить упражнения из раздела *Тестирование*, а также внести небольшие изменения в код программы.

#### *Задание 2. Восстановление алгоритма и кода программы для задачи, которую необходимо решить*

В задании необходимо заполнить пропуски в описании алгоритма и коде программы решения задачи. Задание аналогично предыдущему, но учащимся предлагается не полное описание алгоритма, а частичное, как и код программы на языке программирования. Также нужно заполнить пропуски и выполнить задания из раздела *Тестирование*.

#### *Задание 3. Поэтапное самостоятельное решение задачи*

Это индивидуальное задание. Учащийся должен самостоятельно разработать алгоритм, написать и протестировать программу. Задание представляет собой практико-ориентированную задачу.

Выполнение заданий позволяет учащимся использовать на практике алгоритмические конструкции *Повторение* и *Ветвление* для построения таблиц значений функций. Результаты работы программ можно использовать для построения графиков функций. Результаты работы программ можно использовать для построения графиков функций.

Рекомендуемые формы выполнения задания – фронтальная (задание 1), индивидуальная или групповая (задание 2), индивидуальная (задание 3).

### Основы алгоритмизации и программирования: *дидактические материалы*

*Задание 1.* Для разработки новейшего лекарства ученым необходимо выращивать новый вид бактерий. Количество бактерий в конце эксперимента зависит от того, какой была температура в начале эксперимента. Удалось установить зависимость количества бактерий от температуры. Она выражается

функцией:  $y = \frac{x + 2}{x^2 + 3}$ . Бактерии живут, если температура находится в промежутке  $[-3; 3]$ . При проведении эксперимента используют ровно  $k$  емкостей с бактериями, устанавливая одинаковую разницу температур между емкостями. Программисту поручили написать программу, которая по

введенному значению  $k$  выведет количество бактерий для каждого значения температуры.

Для решения задачи программисту пришлось пройти ряд этапов:

1. Определение исходных данных.
2. Определение результатов.
3. Составление алгоритма решения задачи.
4. Описание переменных.
5. Написание программы.
6. Тестирование программы.

Описание этапов, выполненных программистом:

*Этап 1. Определение исходных данных*

Переменная  $k$  – количество емкостей.

*Этап 2. Определение результатов.*

$k$  значений температуры в соответствующих им значений количества бактерий.

*Этап 3. Составление алгоритма решения задачи*

1. Ввод значений  $k$ .

2. Вычисление шага, с которым изменяется температура.

Значение шага зависит от количества емкостей, участвующих в эксперименте. Исходный температурный промежуток разбивается на части, длина которых и равна шагу. Количество таких промежутков на 1 меньше количества емкостей. Для вычисления значения шага нужно разделить длину промежутка изменения температуры (она равна 6) на количество промежутков разбиения ( $k-1$ ).

3. Для получения таблицы в цикле нужно последовательно получать и выводить на экран значение аргумента и соответствующие значения функции:

- начальное значение аргумента  $x = -3$ . Для получения очередного значения аргумента нужно к текущему значению прибавить значение шага, с которым изменяется аргумент;

- значение функции вычисляется по формуле: 
$$y = \frac{x + 2}{x^2 + 3}.$$

После вычисления полученные значения выводятся на экран. Для того чтобы значения выводились строго одно под другим, нужно использовать вывод с указанием формата вывода (задать количество позиций для вывода значений);

- для подсчета количества будем использовать переменную  $n$  – переменную цикла с параметром. Значения переменной изменяются от 1 до  $k$ .

*Этап 4. Описание переменных*

Переменные  $k, n$  имеют тип *integer*.

Переменные  $x, y, h$  имеют тип *real*.

*Этап 5. Написание программы*

```

Var k,n: integer;
      x,y,h: real;
begin
  writeln('Укажите количество емкостей');
  readln(k);
  h:=6/(k-1);
  x:=-3;
  for n:=1 to k do
  begin
    y:=(x+2)/(x*x+3);
    writeln(x:7:2,y:10:3);
    x:=x+h;
  end;
end.

```

Однако программист не успел выполнить этап 6 – этап тестирования программы. Помогите ему это сделать.

*Упражнение 1.* Проверьте работу программы в соответствии с данными, представленными в таблице 2.

Таблица 2

Данные для тестирования кода программы

Ввод k	Вывод
5	-3.00-0.0833 -1.500.0952 0.000.6667 1.500.6667 3.000.4167

*Упражнение 2.* Заполните таблицу для тестирования кода программы (таблица 3).

Таблица 3

Шаблон для заполнения данными для тестирования кода программы

Номер теста	Ввод k	Вывод
1	2	
2	3	
3	6	

*Упражнение 3.* Замените в решении задачи цикл *for* на цикл *while*. В качестве условия в цикле *while* можно использовать следующее:  $x \leq 3$ .

*Упражнение 4.* В программу перед циклом *for* были добавлены следующие строки:

```

writeln('-----');
writeln("|  x  | y  |");
writeln('-----');

```

Для чего это сделано?

*Упражнение 5.* \*Внесите изменения в программу так, чтобы и основная часть таблицы печаталась аналогично.

**Решение**

1. Для выполнения упражнения 1 необходимо скопировать программу, сохранить ее, а затем запустить и ввести значение 5, сверить полученный ответ с приведенным в таблице.

2. Результат выполнения упражнения 2 получен подсчетом значений функции пошагово с использованием калькулятора (таблица 4).

Таблица 4

Данные для тестирования кода программы

Номер теста	Ввод $k$	Вывод
1	2	-3.00 - 0.083 3.00 - 0.417
2	3	-3.00 - 0.083 0.00 - 0.667 3.00 - 0.417
3	6	-3.00 - 0.083 -1.80 - 0.032 -0.60 - 0.417 0.60 - 0.774 1.80 - 0.609 3.00 - 0.417

3. Код программы:

```
var k,n:integer;
    x,y,h:real;
begin
  writeln(' Укажите количество емкостей');
  readln(k);
  h:=6/(k-1);
  x:=-3;
  while x<=3 do
  begin
    y:=(x+2)/(x*x+3);
    writeln(x:7:2,y:10:3);
    x:=x+h;
  end;
end.
```

4. Приведенные строки позволяют вывести заголовок для таблицы значений функции.

5. Код программы:

```
var k,n:integer;
    x,y,h:real;
begin
  writeln('Укажите количество емкостей');
  readln(k);
  h:=6/(k-1);
  x:=-3;
  writeln('-----');
  writeln('| x | y |');
  writeln('-----');
  for n:=1 to k do
```

```

begin
y:= (x+2)/(x*x+3);
writeln('x:7:2,'|',y:10:3,'|');
x:=x+h;
end;
writeln('-----');
end.

```

Результат работы программы представлен на рисунке 9.

#### Окно вывода

Укажите количество емкостей

7

x	y
-3.00	-0.083
-2.00	0.000
-1.00	0.250
0.00	0.067
1.00	0.750
2.00	0.571
3.00	0.417

Рисунок 9. Окно вывода выполнения программы

**Задание 2.** Программисту было дано следующее техническое задание: «Напишите программу для решения задачи. Выведите на экран таблицу значений функции  $y = \frac{x+2}{x^2-9}$  на промежутке  $[-5; 5]$ . Количество точек на промежутке задается пользователем».

Им был подготовлен отчет о проделанной части работы. К сожалению, в момент сохранения документа произошел сбой, и восстановленный отчет имеет следующий вид.

Для решения задачи было запланировано выполнить следующие этапы:

1. Определение исходных данных.
2. Определение результатов.
3. Составление алгоритма решения задачи.
4. Описание переменных.
5. Написание программы.
6. Тестирование программы.

Описание выполненных этапов

*Этап 1. Определение исходных данных*

Переменная \_\_\_\_\_ – количество точек.

*Этап 2. Определение результатов*

$k$  значений аргумента и соответствующих им значений функции.

*Этап 3. Составление алгоритма решения задачи*

1. Ввод числа \_\_\_\_\_.

2. Вычисление шага  $h$ , с которым изменяется значение аргумента. Значение шага зависит от количества точек, введенных пользователем. Точки разбивают исходный промежуток на части, длина которых и равна шагу. Количество таких промежутков на \_\_\_\_ меньше количества точек. Для вычисления значения шага нужно разделить длину промежутка (она равна \_\_) на количество промежутков разбиения  $(k-1)$ .

3. Для получения таблицы в цикле нужно последовательно получать и выводить на экран значения аргумента и соответствующие ему значение функции:

- начальное значение аргумента  $x = \underline{\hspace{2cm}}$ . Для получения следующего значения аргумента нужно к текущему значению прибавить значение шага, с которым изменяется аргумент;

- значение функции вычисляется по формуле  $y = \frac{x + 2}{x^2 - 9}$ . После вычисления полученные значения выводятся на экран. Для того чтобы значения выводились строго одно под другим, нужно использовать вывод с указанием формата вывода (задать количество позиций для вывода значения);

- при значениях  $x = \underline{\hspace{2cm}}$  и  $x = \underline{\hspace{2cm}}$  знаменатель обращается в нуль. Для того чтобы в программе не происходило деление на \_\_\_\_ и программа не выводила *Infinity*, нужно вставить проверку;

- для подсчета количества точек будем использовать переменную  $n$  – переменную цикла с параметром. Значения переменной изменяются от 1 до  $k$ .

*Этап 4. Описание переменных*

Переменные  $k, n$  имеют тип \_\_\_\_\_.

Переменные \_\_\_\_\_ имеют тип *real*.

*Этап 5. Написание программы*

```
var k,n: _____;
```

```
    x,y,h: _____;
```

```
begin
```

```
  writeln(' Введите количество точек');
```

```
  readln(_____);
```

```
  h:= _____/(k-1);
```

```
  x:=-5;
```

```
  for n:=1 to k do
```

```
  begin
```

```
    y:= (x+2)/(x*x-9);
```

```
    if (_____) _____(_____) then
```

```
      writeln(x:7:2,y:10:3)
```

```
    else
```

```
      writeln(x:7:2, ' значение функции не определено');
```

```
      x:=x+ _____;
```

```
    end;
```

```
  end.
```

Однако некоторые места в отчете не сохранились после восстановления поврежденного файла. Помогите заполнить пропуски в отчете и реализовать оставшийся этап – этап тестирования программы, выполнив следующие упражнения.

**Упражнение 1.** Проверьте работу программы в соответствии с данными, представленными в таблица 5.

Таблица 5

Данные для тестирования кода программы		
Ввод $k$	Вывод	
11	-5.00	-0.188
	-4.00	-0.286
	-3.00	значение функции не определено
	-2.00	0.000
	-1.00	-0.125
	0.00	-0.222
	1.00	-0.375
	2.00	-0.800
	3.0	начение функции не определено
	4.00	0.857
	5.00	0.438

**Упражнение 2.** Заполните таблицу для тестирования кода программы (таблица 6).

Таблица 6

Шаблон для заполнения данными для тестирования кода программы		
Номер теста	$k$	Вывод
1	3	
2	6	
3	15	

**Упражнение 3.** Замените в решении задачи цикл *for* на цикл *while*.

В качестве условия в цикле *while* можно использовать следующее:  
 $x \leq$  \_\_\_\_\_.

**Упражнение 4.** В программу перед циклом *for* были добавлены следующие строки:

```
writeln('-----');
```

```
writeln('| x | y |');
```

```
writeln('-----');
```

Для чего это сделано?

**Упражнение 5.** \*Внесите изменения в программу так, чтобы и основная часть таблицы печаталась аналогично.

### Решение

Восстановленный отчет программиста:

*Этап 1. Определение исходных данных*

Переменная  $k$  – количество точек.

*Этап 2. Определение результатов*

$k$  значений аргумента и соответствующих им значений функции.

*Этап 3. Составление алгоритма решения задачи*

1. Ввод значения  $k$ .

Вычисление шага  $h$ , с которым изменяется значение аргумента. Значение шага зависит от количества точек, введенных пользователем. Точки разбивают исходный промежуток на части, длина которых и равна шагу. Количество таких промежутков на 1 меньше количества точек. Для вычисления значения шага нужно разделить длину промежутка (она равна 11) на количество промежутков разбиения ( $k-1$ ).

2. Для получения таблицы в цикле нужно последовательно получать и выводить на экран значения аргумента и соответствующие ему значения функции:

- начальное значение аргумента  $x = -5$ . Для получения очередного значения аргумента нужно к текущему значению прибавить значение шага, с которым изменяется аргумент;

- значение функции вычисляется по формуле  $y = \frac{x + 2}{x^2 - 9}$ .

После вычисления полученные значения выводятся на экран. Для того чтобы значения выводились строго одно под другим, нужно использовать вывод с указанием формата вывода (задать количество позиций для вывода значения);

- при значениях  $x = -3$  и  $x = 3$  знаменатель обращается в нуль. Для того чтобы в программе не происходило деление на 0 и программа не выводила *Infinity*, нужно вставить проверку;

- для подсчета количества точек будем использовать переменную  $n$  – переменную цикла с параметром. Значения переменной изменяются от 1 до  $k$ .

*Этап 4. Описание переменных*

Переменные  $k, n$  имеют тип *integer*.

Переменные  $x, y, h$  имеют тип *real*.

*Этап 5. Написание программы*

```
var k,n:integer;
    x,y,h:real;
```

```
begin
```

```
writeln('Введите количество точек');
```

```
readln(k);
```

```
h:=10/(k-1);
```

```
x:=-5;
```

```
for n:=1 to k do
```

```
begin
```

```
y:=(x+2)/(x*x-9);
```

```
if (x<>3) and (x<>-3) then
```

```
writeln(x:7:2,y:10:3)
```

```
else
```

```
writeln(x:7:2, ' значение функции не определено');
```

```
x:=x+h;
```

```
end;
```

```
end.
```

Выполнение упражнений:

1. Для выполнения *Упражнения 1* необходимо скопировать программу, сохранить ее, а затем запустить и ввести значение 11, сверить полученный ответ с приведенным в таблице.

2. Результат выполнения *Упражнения 1* получен впоследствии подсчета значений функции пошагово с использованием калькулятора (таблица 7).

Таблица 7

Данные для тестирования кода программы

Номер теста	Ввод $k$	Вывод	
1	3	-5.00	-0.188
		0.00	-0.222
		5.00	0.438
2	6	-5.00	-0.188
		-3.00	значение функции не определено
		-1.00	-0.125
		1.00	-0.375
		3.00	значение функции не определено
		5.00	0.438
3	15	-5.00	-0.188
		-4.29	0.244
		-3.57	-0.418
		-2.86	1.024
		-2.14	0.032
		1.43	-0.082
		0.71	-0.151
		0.00	-0.222
		0.71	0.320
		1.43	-0.493
		2.14	-0.940
		2.86	-5.805
		3.57	1.484
		4.29	0.671
		5.00	0.438

3. Код программы:

```

var k,n:integer;
    x,y,h:real;
begin
writeln(' Введите количество точек');
readln(k);
h:=10/(k-1);
x:=-5;
while x<=5 do
begin
y:=(x+2)/(x*x-9);
if (x<>3) and (x<>-3) then
writeln(x:7:2,y:10:3)
else
writeln(x:7:2, ' значение функции не определено');
x:=x+h;
end;
end.

```

4. Приведенные строки позволяют вывести заголовок для таблицы значений функции.

5. Код программы:

```

var k,n:integer;
    x,y,h:real;
begin
writeln('Введите количество точек');
readln(k);
h:=10/(k-1);
x:=-5;
writeln('-----');
writeln('|   x   |   y   |');
writeln('-----');
for n:=1 to k do
begin
y:=(x+2)/(x*x-9);
if (x<>3) and (x<>-3) then
writeln('|,x:7:2, '|,y:10:3, '|')
else
writeln('|,x:7:2, '|, ' значение функции не определено');
x:=x+h;
end;
writeln('-----');
end.

```

**Задание 3.** Написать программу для решения задачи. Для функции  $y = \frac{1}{x^2} + x^3$ , получить таблицу значений на отрезке  $[a, b]$ . Количество точек на промежутке задается пользователем. Значения  $a$  и  $b$  вводятся с клавиатуры ( $a < 0$ ).

**Решение**

```

var k,n:integer;
    x,y,h,a,b:real;
begin
writeln('Введите количество точек');
readln(k);
writeln(' Введите границы отрезка');
readln(a,b);
h:=(b-a)/(k-1);
x:=a;
for n:=1 to k do
begin
if x<>0 then
begin
y:=1/(x*x)-x*x*x;
writeln(x:7:2,y:11:3);
end
else
writeln(x:7:2,'не существует');
x:=x+h;
end;
end.

```

### ПРИЛОЖЕНИЕ 3

#### Технология обработки текстовых документов: *методические рекомендации*

Большой дидактический потенциал в заданиях по теме «Технологии обработки текстовых документов» содержится непосредственно в текстах, которые несут смысловую нагрузку и помогают реализации задач компетентностного подхода.

##### *Задание 1. Поиск и замена в тексте проверки правописания*

Задание предназначено для выработки навыков по быстрому поиску и замене элементов текста. Повторяющиеся и лишние пробелы возле знаков препинания – широко распространенные ошибки при наборе текстов. Задание знакомит с простыми способами исправления подобных ошибок.

При замене слов важным является удачный выбор шаблонов поиска и замены. Также следует обратить внимание на то, что слова в начале предложений после замены должны начинаться с прописных букв.

##### *Задание 2. Стилевое форматирование заголовков, генерация оглавления документа*

Задание направлено на отработку умений применять стили при формировании текста, а также на развитие умений использовать рациональные способы конструирования оглавления текстового документа с помощью автособираемого оглавления.

При подготовке содержания текстового документа к этому заданию использовались материалы открытого интернет-ресурса *Музей истории энергетики* (режим доступа: 20.12.2019, адрес: [http://www.kids.myenergy.ru/theory\\_and\\_power](http://www.kids.myenergy.ru/theory_and_power)).

Текст для выполнения задания 2 достаточно объемный для того, чтобы продемонстрировать удобство использования стилей, которые не только позволяют быстро изменять оформление документов, но и определяют структуру, используемую при чтении, поиске и при создании оглавления текстового документа.

*Общее замечание.* Проверьте наличие указанных в заданиях файлов в соответствующих папках. Файлы помещаются в указанные папки из прилагаемого к теме архива. Имена дисков, папок и файлов в задании могут быть изменены в соответствии с системой хранения и именования файлов, принятой в конкретном компьютерном классе.

#### Технология обработки текстовых документов: *дидактические материалы*

##### *Задание 1. Исправить ошибки в тексте*

На диске *D* в папке *VIII класс* откройте файл *Паровоз.docx*. Используя инструменты поиска и замены, исправьте текст. Из текста необходимо удалить все лишние пробелы и исправить знаки препинания. Текст написан про парусник бригантину, поэтому все формы слова «паровоз» замените на слово «бригантина». Сохраните рабочий файл на диске *D* в папке *VIII класс* под именем *Бригантина.docx*.

**Решение**

Для выполнения задания выполните следующие действия.

1. Откройте документ в текстовом редакторе и сохраните его под новым именем, как указано в задании.
2. Включите отображение непечатаемых символов, используя пиктограмму  **Отобразить все знаки**.
3. Замените повторяющиеся пробелы, используя инструмент **Заменить...** Для этого задайте образец для поиска *два пробела*, а для замены *один пробел* – и нажмите кнопку **Заменить все**. Так как в тексте встречается больше двух пробелов, идущих подряд, то кнопку **Заменить все** нажимайте до тех пор, пока в окне с сообщением о количестве выполненных замен не появится значение 0.
4. Удалите пробелы и запятые возле точек, используя инструмент **Заменить...** Для этого задайте образец для поиска *пробел точка запятая*, а для замены – *точка* и нажмите кнопку **Заменить все**.
5. Замените двойные запятые одиночными, используя инструмент **Заменить...** Для этого задайте образец для поиска *запятая запятая*, а для замены – *запятая* и нажмите кнопку **Заменить все**.
6. Замените все формы слова «паровоз», начинающиеся с прописной буквы, используя инструмент **Заменить...** Для этого задайте образец для поиска *Паровоз*, а для замены – *Бригантина*. Установите флажок **Учитывать регистр** и нажмите кнопку **Заменить все**.
7. Замените все формы слова «паровоз», начинающиеся со строчной буквы, используя инструмент **Заменить...** Для этого задайте образец для поиска *паровоз*, а для замены – *бригантина*. Установите флажок **Учитывать регистр** и нажмите кнопку **Заменить все**.
8. Выполните проверку правописания, используя пункт **Правописание** вкладки **Рецензирование**.
9. Прочитайте документ и проверьте, что все замены выполнены правильно. Найденные ошибки, сразу исправьте.
10. Полученный результат сохраните.

**Задание 2.** Выполните форматирование документа с помощью стилей и создайте оглавление документа

На диске *D* в папке *VIII класс* откройте файл *Энергия.docx*. Назначьте заголовкам, имеющимся в тексте, стили **Заголовок 1** и **Заголовок 2**. Добавьте перед текстом автособираемое оглавление. Сохраните рабочий файл на диске *D* в папке *VIII класс* под именем *Теория и практика энергетики.docx*.

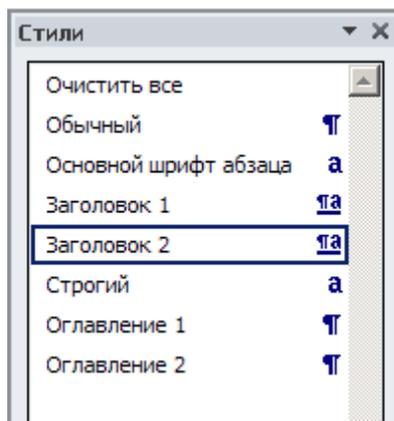
**Решение**

Для выполнения задания выполните следующие действия:

1. Откройте документ в текстовом редакторе и сохраните его под новым именем, как указано в задании.
2. Абзацам, выделенным зеленым цветом – текст «Источники энергии», «Безопасное энергопотребление», «Использование и сохранение энергии»,

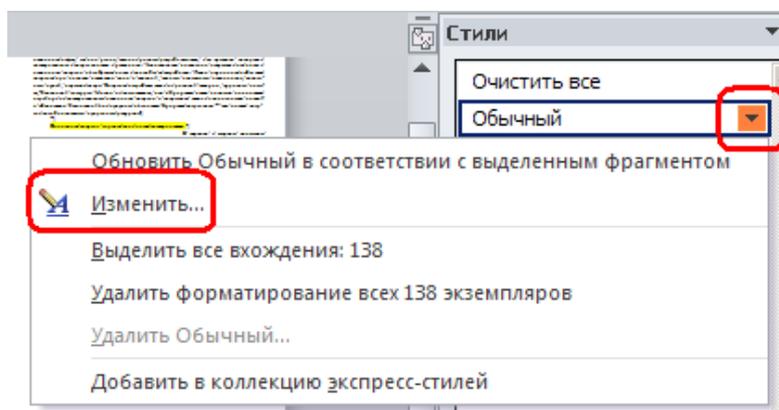
«Справочник Особых Служб», – назначьте стиль **Заголовок 1**. Абзацам с подзаголовками, выделенным желтым цветом, назначьте стиль **Заголовок 2**. Для этого откройте окно стилей, нажав на раскрывающую кнопку (указатель стрелочки в правом нижнем углу) группы **Стили** вкладки **Главная**. Затем выделите нужный фрагмент текста и щелкните указателем мыши на название стиля в окне стилей (рисунок 10).

3.



**Рисунок 10.** Окно для просмотра и изменения параметров форматирования

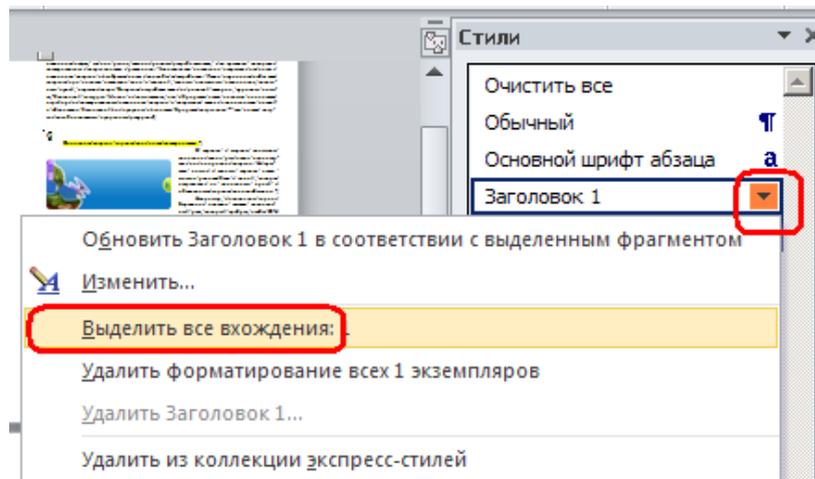
4. Измените параметры стиля **Обычный**. Установите форматирование шрифта: размер шрифта – *14 pt*; форматирование абзаца: *выравнивание по ширине*, отступ первой строки абзаца – *1,25 см*, междустрочный интервал – *Одинарный*. Для этого воспользуйтесь пунктом **Изменить** раскрывающегося меню стиля (рисунок 11).



**Рисунок 11.** Пункт для изменения параметров стиля «Обычный»

5. Измените параметры форматирования заголовков. Для стиля **Заголовок 1** установите: размер шрифта – *20 pt*, цвет шрифта – *зеленый*, начертание – *полужирный*. Для стиля **Заголовок 2**: размер шрифта – *16 pt*, цвет шрифта – *черный*, начертание – *полужирный*. Для этого используйте пункт **Изменить** раскрывающегося меню стиля.

6. Отмените выделение цветом. Для этого воспользуйтесь пунктом **Выделить все вхождения** раскрывающегося меню стилей **Заголовок 1** и **Заголовок 2** (рисунок 12).



*Рисунок 12.* Пункт для выделения всех вхождений стиля **Заголовок 1**

7. Вставьте в начало документа разрыв страницы.
8. На первой странице добавьте оглавление с помощью пиктограммы **Оглавление** вкладки **Ссылки**.
9. Сохраните полученный результат.

## ПРИЛОЖЕНИЕ 4

### Технология обработки аудио- и видеoinформации: методические рекомендации<sup>1</sup>

Работа с аудио- и видеозаписями в последнее время осложнилась повышением контроля за соблюдением лицензионных требований при использовании данного контента. Поскольку необходимо учить учащихся их соблюдать, то использовать нелицензионные записи на уроках нельзя.

На сайте *YouTube* создана фонотека музыкальных композиций, которые не имеют ограничений на скачивание и использование. Именно на таких музыкальных композициях построены задания 1–6 данной темы.

Чтобы скачать файлы трех композиций, упомянутых в заданиях, надо зарегистрировать свой канал на *YouTube*. Для этого нужно иметь (или создать) аккаунт в *Google* и соответствующий аккаунт на *YouTube*. Открыть страницу своего канала на *YouTube*. В правом верхнем углу страницы щелкнуть по значку канала (он соответствует значку вашего аккаунта в *Google*). В открывшемся меню выбрать кнопку **Творческая студия**. На новой странице в левой области располагается меню **Творческая студия**. В меню выбрать пункт **Создать**. В основной части окна появятся три дополнительных пункта, первый из которых – **Фонотека** – выбирается автоматически. На странице открывается список композиций бесплатной музыки для проектов. Над этим списком удобнее выбрать пункт **Указание авторства**. Щелкнуть по нему и выбрать подпункт **Указание авторства не требуется**. Произведения из данного списка можно проигрывать, скачивать и работать с ними. Авторские права при этом не нарушаются. Поле поиска находится над списком композиций в меню справа. В это поле надо ввести название композиции и нажать на значок для начала поиска. Скачать появившуюся композицию.

Для работы с учащимися понадобятся композиции *Highway\_Wildflowers.mp3*; *Giant\_Leap.mp3*.

#### **Задания 1–3. Создание фонограмм**

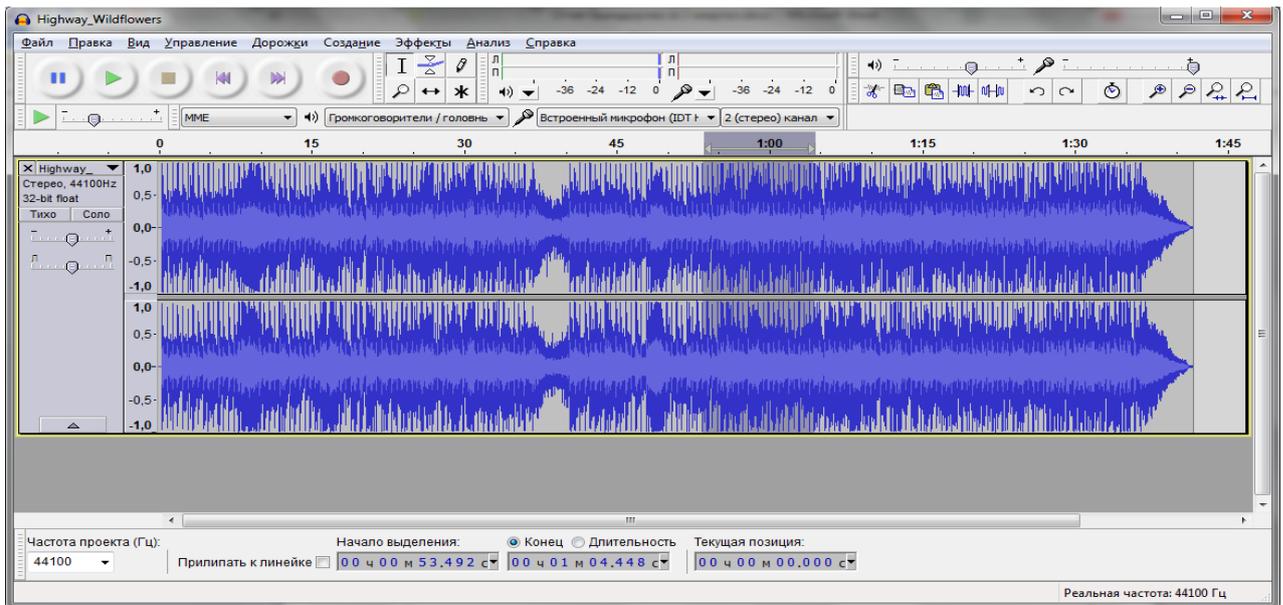
Задания предназначены для формирования умений по редактированию аудиозаписей. В заданиях 1–2 рассматриваются учебные практико-ориентированные ситуации: надо создать фонограмму к видеофильму (видеоклипу) для случаев, когда имеющаяся музыкальная композиция по длительности больше и меньше соответствующего видеофильма. В задании 3 предполагается на основе имеющегося аккомпанемента известной песни подготовить музыкальную композицию (фонограмму), в которой предусмотрено большее количество куплетов с припевами. Задание 3 направлено на интегрированное закрепление приемов, отработанных учащимися в заданиях 1–2.

---

<sup>1</sup>Использованы материалы пособия: Информатика. 8–9 классы. Дидактические и диагностические материалы : пособие для учителей учреждений общ. среднего образования с белорус. и рус. языками обучения / С. И. Зенько [и др.] ; под ред. С. И. Зенько. – Мозырь : Выснова, 2018. – 191 с. – (Компетентностный подход).

В заданиях 1–3 используется программное средство *Audacity*. Это редактор аудиозаписей, который находится в свободном доступе. Его можно скачать по следующему адресу: <https://sourceforge.net/projects/audacity/>.

Для успешной работы с аудиоредактором учащихся надо сначала ознакомить с элементами его интерфейса (рисунок 13). После запуска редактора и выбора в меню **Файл** пункта **Открыть...** появляется окно **Выберите один или несколько звуковых файлов**. Выбрать файл *Highway\_Wildflowers.mp3*.

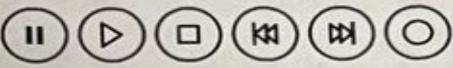


**Рисунок 13.** Интерфейс редактора аудиозаписей *Audacity* с загруженной композицией *Highway Wildflowers*

Две линейные синие фигуры из штрихов представляют запись двух звуковых дорожек (стереоканалов) музыкальной композиции. Вместе они составляют трек. Имеет смысл рассмотреть с учащимися только необходимые элементы интерфейса.

Под строкой меню на панели слева расположены кнопки для управления

воспроизведением и записью:



Во время воспроизведения по треку движется вертикальная полоса, которая называется *текущей позицией*.

В нижней части окна под треком расположены три цифровых табло:



Правое табло показывает численное значение времени текущей позиции, считая от начала композиции. Важное значение в интерфейсе редактора имеет *курсор редактирования*, который также изображается вертикальной чертой. Его числовое положение на треке отображается на левом табло под треком. Первоначально он находится в начале композиции. Поставить курсор редактирования можно в любое место трека простым щелчком по треку. Воспроизведение трека будет теперь неизменно начинаться с этого места.

Кнопки управления воспроизведением  перемещают курсор редактирования в начало и в конец композиции соответственно. Любую часть трека можно выделить протяжкой курсора мыши как фрагмент тестовой строки. При этом часть трека выделяется темным цветом. В этом случае воспроизводить можно только выделенный фрагмент. Курсор редактирования находится в начале фрагмента, а положение конца выделенного фрагмента отображается на среднем табло под треком.

Выделенный фрагмент можно скопировать, вырезать или удалить. Скопированный фрагмент вставить с местоположения курсора редактирования. Команды для выполнения этих действий расположены в пункте **Правка** главного меню, а для отдельных операций также имеются

соответствующие пиктограммы на панели: .

Для выбора точного положения курсора редактирования на треке полезны два инструмента в правой зоне панели под меню:  – **Уместить выделенное** и **Уместить проект**. Первый позволяет любой малый выделенный фрагмент растянуть на ширину всего окна (увеличить). Вторым – вернуть привычный вид, в котором на ширину окна видна запись всей композиции. Рассмотренных элементов интерфейса достаточно для работы с заданиями 1–3.

#### **Задание 4. Изменение формата видеофайла**

Задание предназначено для рассмотрения возможностей работы со свободным программным обеспечением для конвертирования имеющегося видеофайла, поскольку в настоящее время имеется (а также создается с помощью цифровых мобильных устройств) большое количество видеофайлов различных форматов. Не все форматы поддерживают разные программы для видеодемонстрации такого содержимого. Также сложности могут возникать и при создании видеопродукта на основании видеоматериалов разных форматов.

#### **Задание 5. Уменьшение длительности видеофайла**

Задание направлено на создание возможностей для учащихся по приобретению опыта редактирования видеофайлов за счет удаления из него некоторых ненужных фрагментов. В частности, для подготовки видеодемонстраций к презентации.

В заданиях 4 и 5 предполагается использование программного средства *Freemake Video Converter*. Это утилита для работы с видео, аудио и изображениями. Конвертер находится в свободном доступе. Его можно скачать по адресу: [http://www.freemake.com/ru/free\\_video\\_converter/](http://www.freemake.com/ru/free_video_converter/).

Для успешной работы с конвертером учащихся надо сначала познакомить с элементами его интерфейса. После запуска появляется основное окно (рисунок 14).

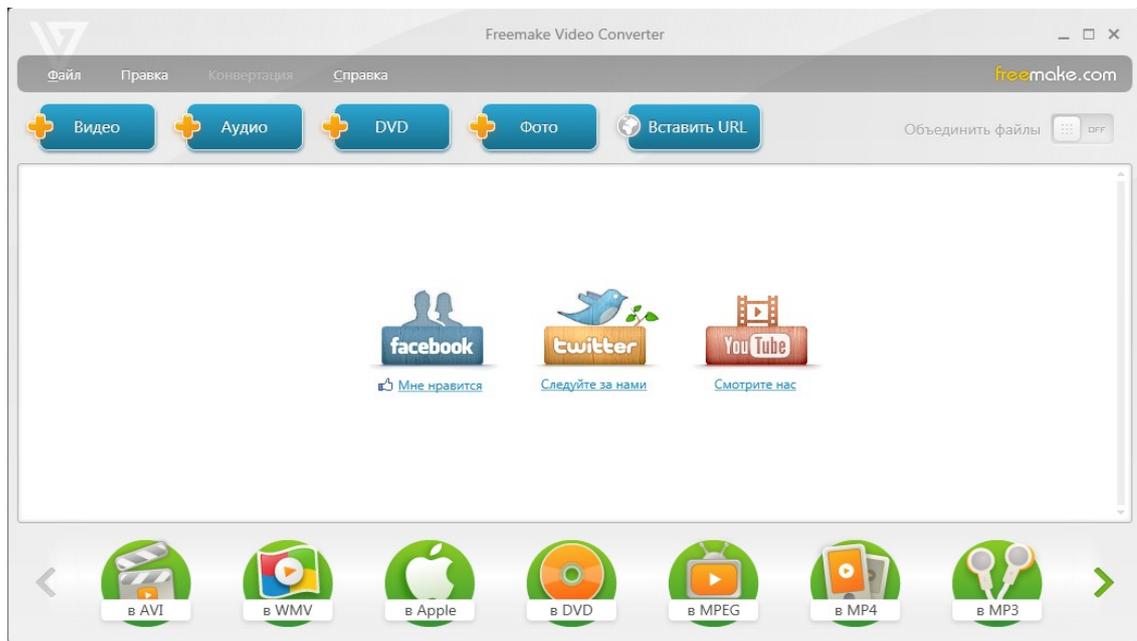


Рисунок 14. Основное окно программы *Freemake Video Converter*

Для загрузки видеофайла (или нескольких видеофайлов) используется кнопка **+Видео**. В диалоговом окне для каждого файла выделяется горизонтальная полоса с его названием и параметрами (рисунок 15).

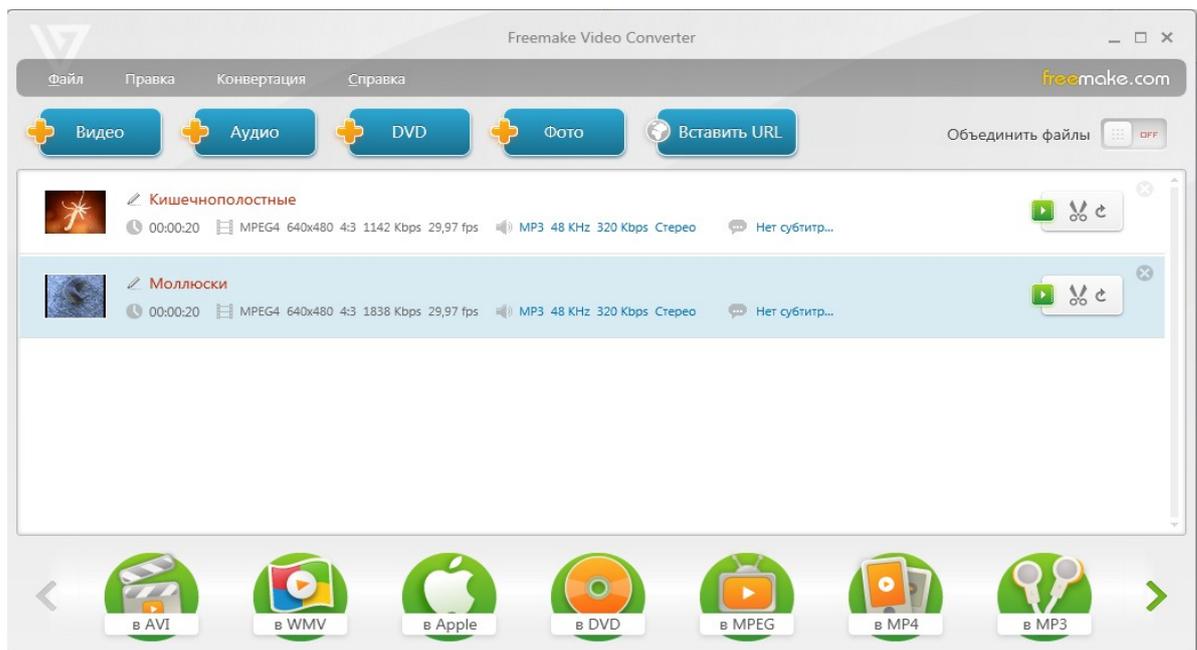


Рисунок 15. Окно программы *Freemake Video Converter* с параметрами файлов

Информация об одном из файлов на полосе подсвечивается голубым цветом. Это означает, что работа будет вестись с параметрами данного объекта. Для объединения видеофайлов в один нажмите переключатель **Объединить файлы** в правом верхнем углу (рисунок 16).

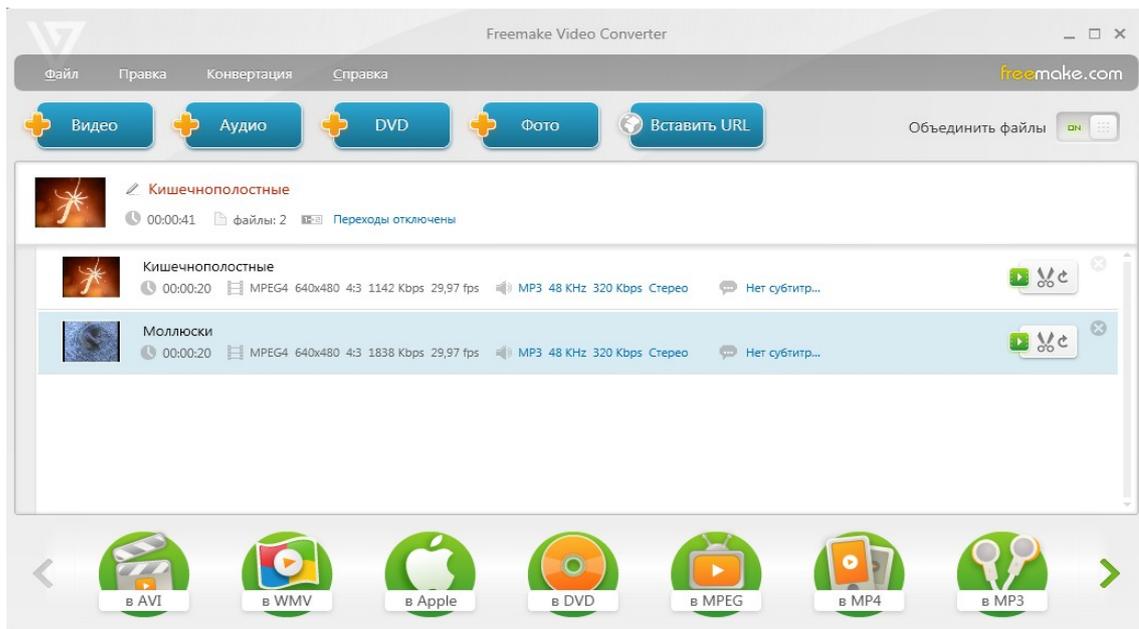


Рисунок 16. Окно программы *Freemake Video Converter* с новым объединенным файлом

Щелчком по пиктограмме  вызывается окно видеоредактора для соответствующего файла. Об этом в решении *задания 5* рассказано подробно.

Конвертацию видеофайла можно сделать одним из двух способов.

Первый способ – использовать команду **Конвертация** главного меню. Появляется окно **Параметры конвертирования**, в котором надо из списка в разделе **Профиль** выбрать формат результата.

Второй способ – щелкнуть по кнопке с названием формата на ленте с различными типами форматов в нижней части окна конвертера. Также откроется окно **Параметры конвертирования**.

В разделе **Профиль** в основном выбирается предполагаемый размер видеофайла результата. При конвертации видео используются определенные видео- и аудиокодеки (способы кодирования). Кодеки можно изменить,

вызвав окно редактора профиля кнопкой  под разделом **Профиль**. При этом видеокодек желательно не изменять. В качестве аудиокодека использовать лучше кодек *MP3*, если в оригинале он был другим.

### **Задание 6. Создание видеофильма**

Задание предназначено для формирования у учащихся умений реализовывать последовательность этапов, необходимых для создания видеофильма из готовых видеофрагментов.

Предварительно учителю необходимо подготовить перечень видеофайлов: «Кишечнополостные», «Черви», «Моллюски», «Ракообразные», «Паукообразные», «Насекомые», «Рыбы», «Земноводные», «Пресмыкающиеся», «Птицы» и «Млекопитающие». Также для работы учащимся понадобится музыкальная композиция *Lake Eerie*.

В задании 6 предполагается использование программных средств *Audacity* и *VideoPad*. *VideoPad* – это видеоредактор, имеющий определенный набор инструментов, которые позволяют осуществлять нелинейный монтаж видео- и аудиофайлов. Видеоредактор находится в свободном доступе. Его можно скачать по адресу: <http://www.nchsoftware.com/videopad/ru/>.

Программное средство *VideoPad* можно использовать для работы в операционной системе *Windows* (начиная с версии *XP*). Для успешной работы с видеоредактором учащихся надо сначала познакомить с элементами его интерфейса (рисунок 17).

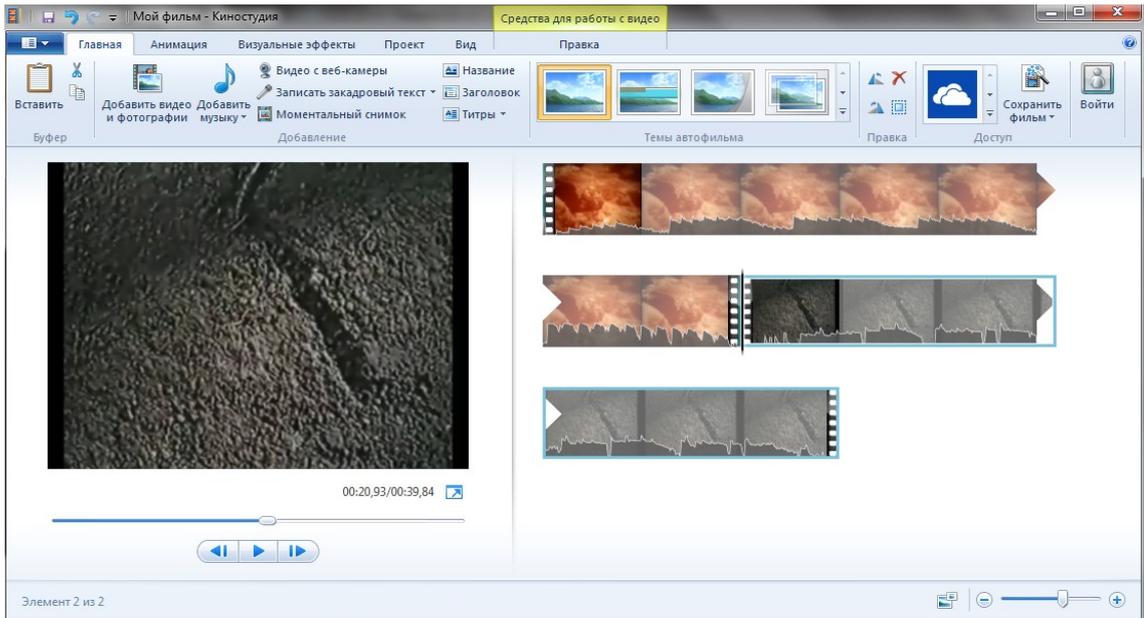


Рисунок 17. Основное окно программы *VideoPad*

В заголовке окна слева размещена *Панель быстрого доступа*. Только в *Windows XP* под заголовком окна находится строка **Меню**, а еще ниже – строка **Закладки**. В остальных версиях *Windows* строки меню нет, но слева в строке **Закладки** есть зеленая кнопка  – **Показать все меню**.

Ниже строки **Закладки** расположены панели с инструментами, еще ниже – рабочие окна:

- *окно файлов* (вверху слева);
- *окно предпросмотра* (вверху справа);
- *окно видеоряда* (внизу).

Границы рабочих окон можно изменять, перетаскивая за их пределы с помощью мыши.

Исходные материалы называются *клипами*. Для загрузки клипов перейти на вкладку **Клипы** и выбрать инструмент **Добавить файл(ы)**. Появится окно для добавления файлов в проект. Файлы можно загружать по одному или группой. В результате загрузки файлы появляются как эскизы в *окне файлов* в разделе **Видеофайлы**.

Изображение набора файлов в *окне файлов* можно поменять

кнопкой  – **Переключиться на вид списка** – под названиями вкладок. В списке с именами файлов появляются колонки с дополнительной информацией о каждом клипе, включая длительность клипа, путь к файлу и т. д.

Выбор клипа в *окне файлов* позволяет просмотреть его в *окне предпросмотра* с помощью управляющих кнопок. Под кадром клипа на *временной диаграмме* заметен вертикальный красный курсор редактирования. При воспроизведении он автоматически передвигается. Щелкнув указателем мыши на временной диаграмме, курсор редактирования можно перемещать в другое место временной диаграммы.

Клип в *окне предпросмотра* можно разделить на две части по положению курсора редактирования. Для этого нажмите кнопку  – **Разделить**. В *окне файлов* появляется еще один файл, к имени которого добавлены символы «(1)». Файл со старым именем хранит первую часть клипа, с новым – вторую. Части клипа отображаются в окне предпросмотра на фоне временной диаграммы исходного клипа. Таким образом всегда можно выделить нужную часть клипа.

Просмотреть аудиодорожку, связанную с клипом, тоже можно в *окне предпросмотра*, если выбрать вкладку **Аудио**, а потом – в *окне файлов* клипа.

Заметим, что любое действие в видеоредакторе можно отменить кнопкой **Отменить** на *Панели быстрого доступа*.

*Окно видеоряда* имеет два режима просмотра: **Шкала времени** и **Раскадровка**. Собирать видеофильм из клипов удобнее в режиме **Раскадровка**, при этом клипы из *окна файлов* просто перетаскивают в *окно видеоряда*. В *окне видеоряда* для каждого клипа отображается эскиз видео и эскиз связанного с клипом аудио (если оно есть).

Важно, что клипы перетаскивают в нужной последовательности и обязательно в конец видеоряда.

В последней строке *окна видеоряда* справа есть инструменты масштабирования изображения в этом окне (отображается в режиме **Шкала времени**).

Просмотр видеоряда как единого фильма доступен в *окне предпросмотра* в разделе **Просмотр видеоряда**. В этом случае курсор редактирования находится в *окне видеоряда*.

Если видеофильм будет иметь единую музыкальную фонограмму, то надо добавить новую фонограмму и выключить звук аудиодорожек, связанных с клипами.

### Технология обработки аудио- и видеoinформации: дидактические материалы

**Задание 1.** Пусть имеется музыкальная композиция длительностью 3 минуты 7 секунд и видеофильм длительностью 2 минуты 38 секунд. С помощью аудиоредактора *Audacity* создайте фонограмму для видеофильма.

**Решение.** Нужно сократить длительность композиции до длительности видеофильма, применив эффект затухания в конце сокращенной части композиции.

Решение проведем на примере конкретной композиции *GiantLeap*, которая хранится в файле *Giant\_Leap.mp3*.

Рекомендации по выполнению практической части задания:

1. Загрузите композицию в аудиоредактор *Audacity*.
2. Выделите на треке часть композиции от момента 2 минуты 38 секунд до конца трека:

- момент начала выделения на треке сначала определяем примерно – визуально установите курсор мыши в предполагаемое начало и протяните его до конца трека;

- затем задайте начало области выделения точно, используя цифровое поле **Начало выделения** в нижней части окна редактора:

- а) щелкните по позиции поля – выделяется первая позиция момента времени (2 минуты 38 секунд), в которую с клавиатуры введите нужную цифру (2);

- б) автоматически активной становится позиция поля справа – введите вторую цифру момента времени (3) и т. д.

3. Удалите выделенную часть композиции. Для этого выберите в меню **Правка** пункт **Удалить** или на панели пиктограмму **Вырезать**.

4. Выделите 5 секунд в конце композиции. Для выделения проведите указателем курсора мыши, определяя начальный момент выделения приблизительно.

5. Примените к выделенной части эффект плавного затухания, выбрав в меню **Эффекты** пункт **Плавное затухание**.

6. Сохраните полученную фонограмму в формате *MP3*, используя в меню **Файл** пункт **Export Audio...** В открывшемся окне определите папку, в которую вы будете сохранять фонограмму, укажите имя нового файла (*Фонограмма\_1*) и выберите тип файла (*Файлы MP3*). Нажмите **Сохранить**.

**Задание 2.** Пусть имеется музыкальная композиция длительностью 1 минута 41 секунда и видеофильм длительностью 2 минуты 38 секунд. С помощью аудиоредактора *Audacity* создайте фонограмму для видеофильма без заметных на слух разрывов звучания.

**Решение.** Нужно увеличить длительность композиции до длительности видеоклипа за счет добавления в композицию повтора некоторой ее части. При этом момент стыка частей не должен проявляться заметным на слух разрывом звучания. Увеличенную композицию следует сократить до необходимой длительности и применить в ее конце эффект плавного затухания.

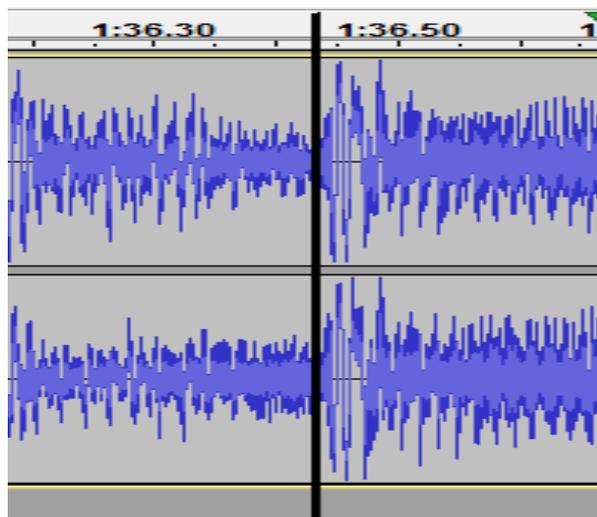
Решение проведем на примере конкретной композиции *Highway Wildflowers*, которая хранится в файле *Highway\_Wildflowers.mp3*.

Рекомендации по выполнению практической части задания:

1. Загрузите композицию в аудиоредактор *Audacity*.

2. Определите в конце трека момент, в котором заканчивается музыкальная фраза (припев, куплет, проигрыш и т. п.). Для этого:

- прослушайте композицию и выберите примерно необходимый момент на треке (1 минута 36 секунд);
- выделите часть трека, в которой отмеченный момент должен быть примерно посередине (длина выделения на экране – приблизительно 1–2 см);
- используя инструмент **Уместить выделенное**, увеличьте выделенную часть трека на все окно. При этом выделение не снимайте;
- прослушайте выделенный фрагмент несколько раз, останавливая прослушивание кнопкой **Приостановить**, и выберите требуемый момент перед всплеском амплитуды (1 минута 36,435 секунд) в соответствии с рисунком 18.



*Рисунок 18.* Положение вертикального курсора при редактировании композиции

3. Верните трек целиком в окно редактора, используя инструмент **Уместить проект**.

4. Выделите часть трека от найденного момента до конца трека (по аналогии со шагом 2 решения задания 1).

5. Удалите выделенную часть трека (по аналогии с шагом 3 решения задания 1).

6. Определите на треке момент, в котором начинается музыкальная фраза (припев, куплет, проигрыш и т. п.) и с которого будет начинаться повторный фрагмент трека – по аналогии с шагом 2 выполнения данного задания (примерный момент – 20,35 секунд; точный момент – 20,391 секунд).

7. Выделите часть трека от найденного момента до конца трека (по аналогии с шагом 2 решения задания 1).

8. Скопируйте выделенную часть композиции в электронный буфер, используя в меню **Правка** пункт **Скопировать** или на панели пиктограмму **Скопировать**.

9. Установите курсор редактора в конец композиции с помощью кнопки **Перейти к концу дорожки**.

10. Вставьте фрагмент из электронного буфера, используя в меню **Правка** пункт **Вставить**.

11. Выделите на треке часть композиции от момента 2 минуты 38 секунд до конца трека по аналогии с шагом 2 решения задания 1).

12. Удалите выделенную часть композиции, выбрав в меню **Правка** пункт **Удалить** или на панели пиктограмму **Вырезать**.

13. Выделить 5 секунд в конце композиции. Для выделения проведите указателем курсора мыши, определив начальный момент выделения приблизительно.

14. Примените к выделенной части эффект плавного затухания, выбрав в меню **Эффекты** пункт **Плавное затухание**.

15. Сохраните полученную фонограмму в формате *MP3*, используя в меню **Файл** пункт **Export Audio...** В открывшемся окне определите папку, в которую вы будете сохранять фонограмму, укажите имя нового файла (*Фонограмма\_2*) и выберите тип файла (*Файлы MP3*). Нажмите **Сохранить**.

**Задание 3.** Пусть имеется музыкальная композиция, которая является аккомпанементом известной песни. Композиция включает 3 куплета с припевами. Создать на ее основе композицию, которая содержит 6 куплетов с припевами без заметных на слух разрывов звучания.

**Решение.** В имеющуюся композицию следует добавить копию одного или нескольких фрагментов (куплет с припевом) так, чтобы стык частей не проявлялся заметным на слух разрывом звучания. Задание решается с помощью приемов, которые были использованы при выполнении первого и второго заданий.

Рекомендации по выполнению практической части задания:

1. Загрузите композицию в аудиоредактор *Audacity*.
2. Определите на треке момент, в котором начинается музыкальная фраза (куплет).
3. Определите на треке момент, в котором заканчивается музыкальная фраза (конец припева и проигрыш до нового куплета, если он есть).
4. Выделите часть трека между выделенными моментами.
5. Скопируйте выделенный фрагмент в электронный буфер обмена.
6. Установите курсор редактора на второй выделенный момент (конец припева).
7. Вставьте скопированный фрагмент нужное количество раз.
8. Сохраните полученную композицию в формате *MP3*.

**Задание 4.** Изменить формат видеофайла с *MP4* на *AVI* с помощью конвертера *Freemake Video Converter*.

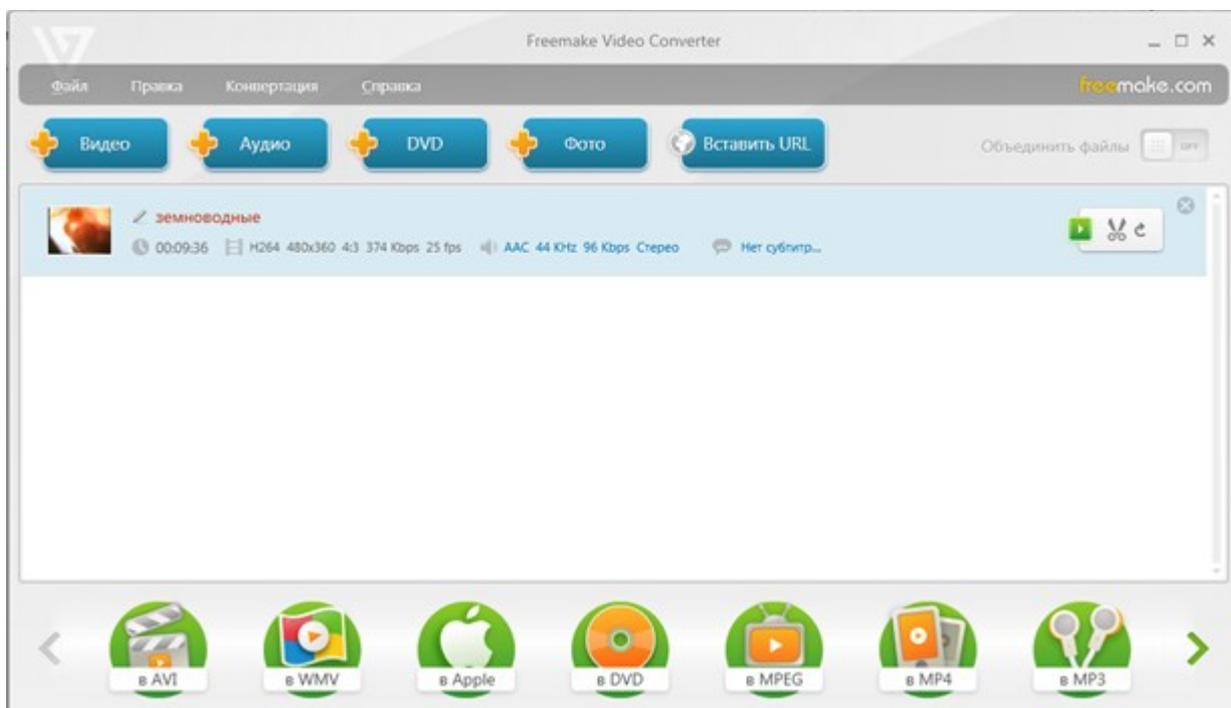
**Решение.** Изменение формата нескольких видеоклипов на какой-то один формат может потребоваться в случае включения клипов разных форматов в единый видеофильм.

Рекомендации по выполнению практической части задания:

1. Запустите конвертер *Freemake Video Converter*.

2. Загрузите видеофайл формата *MP4* в конвертер. Для этого используйте кнопку **+Видео** под строкой меню. В появившемся окне **Открыть** выберите файл для загрузки.

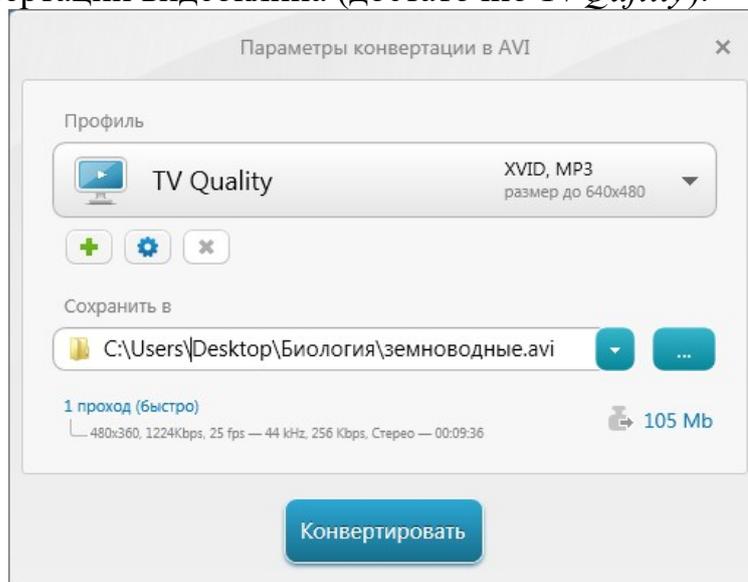
В окне конвертера появляется голубая полоса с названием и параметрами видеофайла (рисунок 19).



**Рисунок 19.** Информация с параметрами видеофайла в программе *Freemake Video Converter* к заданию 4 темы «Технология обработки аудио- и видеoinформации»

3. Подберите параметры конвертации и запустите конвертацию, используя в меню **Конвертация** пункт **в AVI...**

4. В разделе **Профиль** появившегося окна **Параметры конвертации в AVI** (рисунок 20) выберите из раскрывающегося списка необходимое качество результата конвертации видеоклипа (достаточно *TVQuality*).



**Рисунок 20.** Окно «Параметры конвертации» в программе *Freemake Video Converter*

5. В разделе **Сохранить в** выберите место сохранения преобразованного видеофайла на компьютере.

6. Запустите конвертацию, нажав кнопку **Конвертировать** в нижней части окна. Конвертация может занять некоторое время. Дождитесь ее окончания.

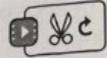
**Задание 5.** Сократите длительность видеофайла с помощью конвертера *Freemake Video Converter*.

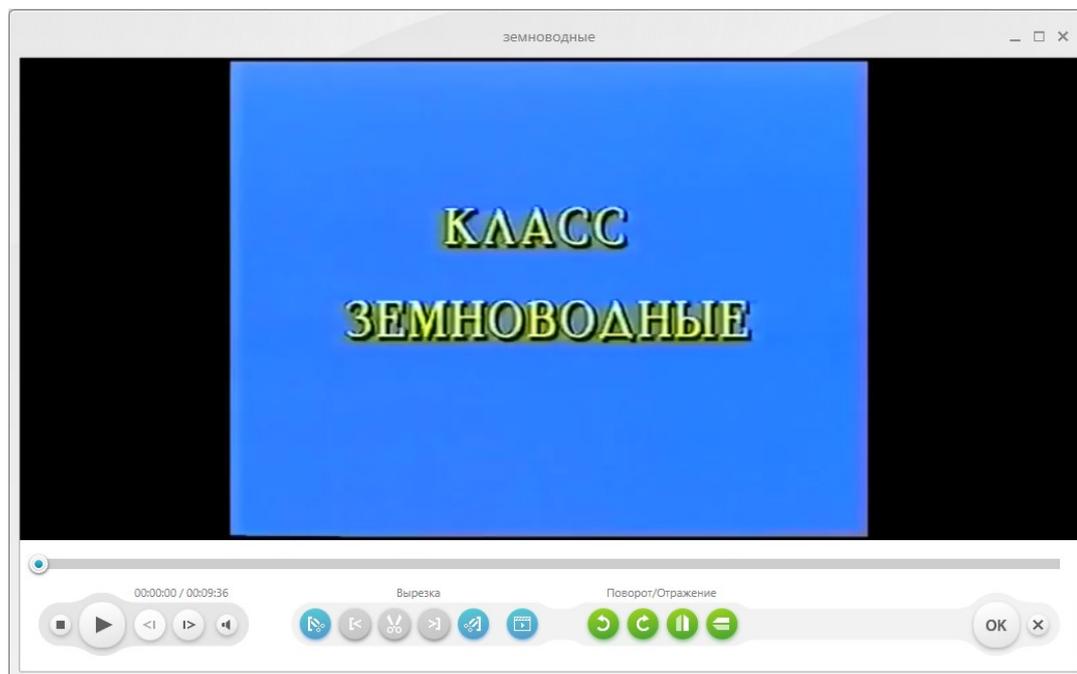
**Решение.** Уменьшение длительности видеофайла предполагает составление нового видеофайла из некоторых фрагментов данного. Например, видеофайлы с описанием различных групп животных в Интернете имеют длительность от 2 до 20 минут. А для включения в мультимедийную презентацию лучше всего подходят видеофайлы (видеоклипы) длительностью 20–35 секунд.

Рекомендации по выполнению практической части задания:

1. Запустите конвертер *Freemake Video Converter*.
2. Загрузите видеофайл в формате *MP4* в конвертер. Для этого используйте кнопку **+Видео** под строкой меню. В появившемся окне **Открыть** выберите файл для загрузки.

В окне конвертера появится голубая полоса с названием и параметрами видеофайла.

3. Запустите редактор для этого видеофайла. Щелкните по пиктограмме  в правой части голубой полосы для клипа. Откроется окно видеоредактора: слева внизу расположены кнопки управления воспроизведением, далее – группа кнопок **Вырезка** для редактирования видеофайла (рисунок 21).



**Рисунок 21.** Окно видеоредактора в программе *Freemake Video Converter*

4. Отредактируйте видео, вырезав из клипа ненужные фрагменты:

- установите указатель на полосе времени в начало удаляемого фрагмента;
- нажмите на первую слева кнопку в группе **Вырезка**;
- переместите указатель на полосе времени в конец удаляемого фрагмента;
- нажмите на пятую слева кнопку в группе **Вырезка** – в результате будет выделен фрагмент видеофайла на полосе времени;
- нажмите на кнопку с ножницами для удаления выделенного фрагмента.

5. Сохраните отредактированное видео:

- нажмите на кнопку **ОК** для выхода из видеоредактора;
- проводите конвертирование измененного видеофайла (по аналогии с шагами 3–6 задания 4).

**Задание 6.** С помощью программного средства *VideoPad* создайте видеофильм «Основные группы животных» с музыкальным сопровождением на основе данного набора видеофайлов (видеоклипов) и музыкальной композиции *Lake Eerie*.

Видеофильм должен включать название фильма, видеоклипы, предваряемые кадрами с названиями групп животных, последний кадр с текстом «Конец фильма».

Предполагается, что все видеофайлы уже имеют подходящую длительность.

**Решение.** Для выполнения задания следует пройти ряд этапов:

1. Создание сценария видеофильма.
2. Загрузка видеофайлов (видеоклипов) и фонограммы в *VideoPad*.
3. Создание текстовых клипов.
4. Сбор клипов в фильм в окне видеоряда.
5. Сохранение проекта видеофильма.
6. Создание к фильму фонограммы соответствующей длительности.
7. Добавление фонограммы к фильму.
8. Сохранение видеофильма.

Рассмотрим реализацию каждого этапа.

**Этап 1. Создание сценария видеофильма**

Видеофильм будет включать следующие клипы (в терминологии программы *VideoPad*):

Клип 1 – текстовый клип с названием фильма «Основные группы животных».

Клип 2 – текстовый клип с названием группы «Беспозвоночные/Кишечнополостные».

Клип 3 – видеоклип «Кишечнополостные».

Клип 4 – текстовый клип с названием группы «Беспозвоночные/Черви».

Клип 5 – видеоклип «Черви».

Клип 6 – текстовый клип с названием группы «Беспозвоночные/ Моллюски».

Клип 7 – видеоклип «Моллюски».

Клип 8 – текстовый клип с названием группы «Беспозвоночные/ Членистоногие/ Ракообразные».

Клип 9 – видеоклип «Ракообразные».

Клип 10 – текстовый клип с названием группы «Беспозвоночные/ Членистоногие/ Паукообразные».

Клип 11 – видеоклип «Паукообразные».

Клип 12 – текстовый клип с названием группы «Беспозвоночные/ Членистоногие/ Насекомые».

Клип 13 – видеоклип «Насекомые».

Клип 14 – видеоклип с названием группы «Позвоночные/ Рыбы».

Клип 15 – видеоклип «Рыбы».

Клип 16 – текстовый клип с названием группы «Позвоночные/ Земноводные».

Клип 17 – видеоклип «Земноводные».

Клип 18 – текстовый клип с названием группы «Позвоночные/ Пресмыкающиеся».

Клип 19 – видеоклип «Пресмыкающиеся».

Клип 20 – текстовый клип с названием группы «Позвоночные/ Птицы».

Клип 21 – видеоклип «Птицы».

Клип 22 – текстовый клип с названием группы «Позвоночные/ Млекопитающие».

Клип 23 – видеоклип «Млекопитающие».

Клип 24 – текстовый клип с текстом «Конец фильма».

*Этап 2. Загрузка видеофайлов (видеоклипов) и фонограммы в VideoPad*  
Запустите видеоредактор *VideoPad*.

Загрузите видеоклипы «Кишечнополостные», «Черви», «Моллюски», «Ракообразные», «Паукообразные», «Насекомые», «Рыбы», «Земноводные», «Пресмыкающиеся», «Птицы» и «Млекопитающие». Для загрузки клипов перейдите на вкладку **Клипы** и выберите инструмент **Добавить файл(ы)**. Появится окно для добавления файлов в проект. Файлы можно загружать по одному или группой. В результате загрузки файлы появляются как эскизы в *окне файлов* в разделе **Видеофайлы**.

*Этап 3. Создание текстовых клипов*

Создайте *текстовый клип с названием фильма*.

Для правильного отображения текстов на экране нужно зафиксировать свойства видеоклипов. В *окне файлов* щелкните правой клавишей мыши по файлу клипа. Откроется контактное меню. Выберите в нем пункт **Свойства...** В окне **Свойства** установите значение свойств разрешения – **640x480**.

На вкладке **Клип** щелкните по кнопке **Добавить текст**. Открывается диалоговое окно для ввода текста, в *окне файлов* открывается вкладка **Изображения** с новым файлом *Текстовый клип*, а в *окне предпросмотра* – изображение его кадра.

Щелкните правой клавишей мыши по кадру в *окне предпросмотра*. Откроется контекстное меню. Выберите пункт **Изменить настройки предпросмотра...** В пункте **Предустановка** выберите набор параметров **640x480 (4:3)**, который соответствует свойствам видеоклипов.

Диалоговое окно редактирования текста можно открывать в любой момент в том же контекстном меню.

В диалоговом окне редактирования, как в текстовом редакторе, введите текст «Основные группы животных». Текст и его положение дублируются на кадре в *окне предпросмотра*. В диалоговом окне подберите размер, цвет фона, определяясь по кадру в *окне предпросмотра*.

По умолчанию длительность клипа равна трем секундам. Она отображается под временной диаграммой на сером фоне *окна предпросмотра*. Увеличьте ее до пяти секунд. Для изменения щелкните указателем мыши по заданной длительности и внесите изменения.

Проверьте *тестовый клип с названием фильма в окне предпросмотра*, нажав на кнопку **Воспроизведение**.

Аналогичным образом создайте *текстовые клипы с названиями групп животных* и текстом «*Конец фильма*».

Новые пустые текстовые клипы с настройками первого клипа добавляются инструментом **Добавить текст** на вкладке **Клипы**. Остается только ввести текст и выбрать его размер и положение в кадре.

#### *Этап 4. Сбор клипов в фильм в окне видеоряда*

В *окне видеоряда* перейти в режим **Раскадровка**. Соберите видеофильм из имеющихся клипов. Для этого перетащите с помощью мыши текстовые и видеоклипы из *окна файлов* в *окно видеоряда* в нужной последовательности согласно сценарию (см. Этап 1). В *окне видеоряда* для каждого клипа отобразится эскиз видео.

#### *Этап 5. Сохранение проекта видеофильма*

Сохраните проект видеофильма, чтобы потом иметь возможность его изменить. Для этого на вкладке **Главная** выберите кнопку **Сохранить проект**. Укажите место его сохранения и название. Проект сохранится в файле с расширением *.vrj*.

#### *Этап 6. Создание к фильму фонограммы соответствующей длительности*

С помощью аудиоредактора *Audacity* создайте фонограмму к фильму соответствующей длительности на основании имеющейся музыкальной композиции *Lake Eerie*. Длительность заданной композиции равна 2 мин (по аналогии с выполнением задания 2).

#### *Этап 7. Добавление фонограммы к фильму*

Загрузите полученную фонограмму в проект в видеоредакторе *VideoPad*. Для загрузки аудиофайла перейдите на вкладку **Аудио** и выберите инструмент **Добавить файл(ы)**.

Для работы с фонограммами *окно видеоряда* переведите в режим **Шкала времени**.

Файл фонограммы из *окна файлов* перетащите в *окно видеоряда* под фонограмму первого клипа точно в начало безымянной дорожки. Дорожка с фонограммой автоматически получает имя **Аудиодорожка 2**.

Далее выключите звук связанных с клипами аудиодорожек. Для этого в *окне видеоряда* в начале диаграммы **Аудиодорожки 1** щелкните по кнопке **Выключить звук дорожки**.

Осталось установить нарастание и затухание фонограммы (в начале и в конце). Для этого в *окне видеоряда* щелкните по эскизу фонограммы на **Аудиодорожке 2**. Затем выберите вкладку **Аудио**, нажмите на инструмент **Затухание**. В меню инструмента, используя пункты **Проявление** и **Исчезновение** (нарастание и затухание), установите соответствующее время.

#### *Этап 8. Сохранение видеофильма*

Сохраните проект видеофильма, чтобы потом иметь возможность его изменить. Для этого на вкладке **Главная** выберите кнопку **Сохранить проект**.

Для сохранения фильма в видеоформате на вкладке **Главная** нажмите кнопку **Экспортировать видео**. Из открывшегося перечня выберите пункт **Видеофайл**. Откроется диалоговое окно настроек экспорта. Выберите для фильма универсальный формат *.avi*. Разрешение увеличивать не имеет смысла, можно оставить исходное или уменьшить его.

После выбора настроек нажмите кнопку **Создать**.

## ИНФОРМАЦИОННЫЕ РЕСУРСЫ СЕТИ ИНТЕРНЕТ

**Задание 1.** Среди указанных вариантов найдите неправильные записи IP-адреса. Ответ поясните.

**А.** 27.0.1.250.43

**Б.** 89.256.142.131

**В.** 34.78.43.188

**Г.** 240.156.170.54

**Решение.**

IP-адрес записывают с помощью четырех десятичных чисел от 0 до 255, отделенных друг от друга точкой.

В IP-адресе 27.0.1.250.43 допущена ошибка: превышено количество используемых чисел.

В адресе 89.256.142.131 допущена ошибка: число 256 превышает допустимое.

Адрес 34.78.43.188 записан верно.

Адрес 240.156.170.54 записан верно.

**Ответ:** А, Б.

**Задание 2.** Учитель записал Пете IP-адрес школьного сервера на листке бумаги. Петя по рассеянности разорвал его на 4 кусочка. Помогите Пете восстановить IP-адрес, если отдельные фрагменты обозначены буквами А, Б, В и Г. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

**А** – .64

**Б** – 3.13

**В** – 3.133

**Г** – 20

**Решение.**

IP-адрес записывают с помощью четырех десятичных чисел от 0 до 255, отделенных друг от друга точкой.

Фрагмент, обозначенный буквой А, не может быть первым, т. к. перед числом 64 стоит точка.

Если после фрагмента А поставить любой из фрагментов Б, В или Г, то адрес будет ошибочным (получатся числа 643 или 6420). Следовательно, фрагмент А должен стоять последним.

После фрагмента В не могут находиться ни фрагмент Б, ни фрагмент Г, т. к. будут получаться числа 1333 или 13320. Следовательно, фрагмент В должен стоять предпоследним (перед фрагментом А).

Если перед фрагментом В поставить фрагмент Г, то адрес также будет ошибочным (получится число 13203). Следовательно, фрагмент Г должен стоять перед фрагментом В.

В результате получаем 203.133.133.64 – IP-адрес, записанный для Пети.

**Ответ:** ГБВА.

**Задание 3.** С помощью электронных словарей сети Интернет (таблица 1) найдите лексические значения слов метонимия, видеокарта, железо, папирус, скальпель, дебет (бухг. термин).

Таблица 1

**Перечень электронных словарей сети Интернет**

<b>Название ресурса</b>	<b>Адрес интернет-ресурса</b>
Электронные словари и энциклопедии	slovar.plib.ru/dictionary
Лингвистика в Интернете	www.slovari.ru
Глоссарий	www.glossary.ru
Толковый словарь Т. Ф. Ефремовой	www.efremova.info
Словарь С. И. Ожегова	www.ozhegov.org
Большой толковый словарь современного русского языка Д. Н. Ушакова	ushdict.narod.ru
Коллекция толковых словарей и энциклопедий онлайн	vseslova.ru
Переводчик онлайн	www.translate.ru
Электронный словарь	www.lingvo.ru
Мультитран	www.multitran.ru

**Решение.**

На диске **D** в папке *IX класс* создайте текстовый документ под названием «Значение слов.docx».

Создайте таблицу из трех столбцов и семи строк (таблица 2). В первую строку введите названия столбцов: «Слово», «Лексическое значение», «Адрес электронного словаря».

Таблица 2

**Шаблон таблицы для выполнения задания**

<b>Слово</b>	<b>Лексическое значение</b>	<b>Адрес электронного словаря</b>

В первый столбец таблицы введите слова, указанные в условии задания.

Откройте браузер *Google Chrome, Microsoft Edge, Mozilla Firefox, Opera, Safari* или любой другой. В адресную строку введите адрес одного из электронных словарей (таблица 1). На веб-странице электронного словаря введите слово, лексическое значение которого вам надо определить.

Скопируйте лексическое значение и вставьте в соответствующую строку второго столбца таблицы (таблица 2) текстового документа. В третьем столбце укажите адрес электронного словаря.

Полученный результат сохраните.

**Задание 4.** Используя различные поисковые системы ([www.yandex.ru](http://www.yandex.ru), [www.rambler.ru](http://www.rambler.ru), [www.google.com](http://www.google.com) и др.), заполните таблицу сведениями о выдающихся личностях XX века (таблица 3).

Таблица 3

**Выдающиеся личности XX века**

<b>Имя, фамилия</b>	<b>Фотография</b>	<b>Годы жизни</b>	<b>Род занятий</b>
Джеф Раскин			
Лев Ландау			
Юрий Гагарин			

**Решение.**

На диске **D** в папке *IX класс* создайте текстовый документ под названием «*Выдающиеся личности.docx*».

В текстовом документе создайте таблицу (таблица 3).

Для поиска информации о перечисленных деятелях откройте несколько поисковых систем.

В поле поиска введите имя и фамилию: Джеф Раскин. Нажмите кнопку **OK**.

Дождитесь, когда результаты поиска появятся на экране. Сравните полученную информацию в двух-трех поисковых системах. Из предоставленного множества ссылок выберите наиболее подходящие и откройте их.

Соберите необходимые сведения и скопируйте их в таблицу (таблица 3).

Аналогичным образом найдите информацию о Л. Ландау, Ю. Гагарине.

Полученный результат сохраните.

## АЛГОРИТМЫ ОБРАБОТКИ СТРОКОВЫХ ВЕЛИЧИН

### *Общая вводная сюжетная часть*

Агент *A2F* получил зашифрованное сообщение от своего информатора *C4D*. И первое, что он должен был сделать, – это проверить письмо на подлинность (возможно, его написал кто-то другой, используя чужой e-mail). По договоренности между *A2F* и *C4D* в текст письма вместе с зашифрованным донесением вставлялось число, состоящее из  $n$  цифр (число  $n$  передавалось по другим каналам связи). Письмо можно было считать подлинным, если данное число делилось без остатка на какое-то другое число (к примеру, сегодня это 12). Все шло хорошо, пока число  $n$  было небольшим, – спасал обычный калькулятор. Когда же числа стали большими, возникла необходимость в написании компьютерной программы, которая могла бы проверить подлинность письма.

**Задание 1.** Цифры числа вводятся с клавиатуры в том порядке, в котором они следуют в числе. Программисту надо написать программу, способную определить, делится ли указанное в письме число  $n$  на 12. Количество цифр – от 2 до 255.

Для решения задачи программисту пришлось пройти ряд этапов:

1. Определение исходных данных.
2. Определение результатов.
3. Составление алгоритма решения задачи.
4. Описание переменных.
5. Написание программы.
6. Тестирование программы.

### **Описание этапов, выполненных программистом**

*Этап 1. Определение исходных данных.*

Переменная  $a$  – строка для хранения цифр числа.

*Этап 2. Определение результатов.*

Ответ «Да», если число делится на 12, или ответ «Нет», если не делится на 12.

*Этап 3. Составление алгоритма решения задачи.*

Ввод исходных данных.

Если бы количество цифр в числе было небольшим, то можно было бы ввести переменную типа *integer* и проверить делимость. Но число из 255 цифр не поместится даже в самый большой целочисленный тип данных *int64*, хранящий до 19 цифр числа. Поэтому воспользуемся признаками делимости. Разложив число 12 на взаимно простые множители, то получим:  $12=3\cdot 4$ . Значит для того, чтобы делиться на 12, число должно одновременно делиться и на 3, и на 4. Значит, должны выполняться два условия:

*Условие 1.* Для проверки делимости на 3 найдем сумму цифр числа. Для этого превратим каждый символ в число с помощью функции *StrToInt*. Если полученная сумма цифр числа будет делиться на 3, то и исходное число делится на 3.

*Условие 2.* Для проверки делимости на 4 сконструируем двузначное число  $d$  из двух последних цифр исходного числа. Если это число будет делиться на 4, то и исходное число делится на 4.

Если *условие 1* и *условие 2*, описанные в пункте 2, выполняются одновременно, то выводим сообщение «Да», иначе – сообщение «Нет».

*Этап 4. Описание переменных.*

Переменные  $n$ ,  $S$ ,  $d$  имеют тип *integer*, переменная  $a$  имеет тип *string*. Для работы также необходима целая переменная  $i$  – счетчик цикла.

*Этап 5. Написание программы.*

```
var n,i,s,d:integer;
    a: string;
begin
  writeln('Введи число');
  readln(a);
  n:=length(a);
  s:=0;
  for i:=1 to n do
    s:= s+StrToInt(a[i]);
  d:=10*StrToInt(a[n-1])+StrToInt(a[n]);
  if (s mod 3=0) and (d mod 4=0) then
    writeln('Да')
  else
    writeln('Нет');
end.
```

Однако программист не успел выполнить шестой этап – *этап тестирования программы*. Помогите ему это сделать.

*Упражнение 1.* Проверьте работу программы в соответствии с данными, представленными в таблице (таблица 4).

Таблица 4

#### Данные для тестирования кода программы

Ввод $a$	Вывод
2641	Нет
1498796006400	Да

*Упражнение 2.* Заполните таблицу для тестирования кода программы (таблица 5). Значения для двух последних тестов придумайте самостоятельно.

Таблица 5

#### Шаблон заполнения данными для тестирования программы

Номер теста	Ввод $a$	Вывод
1	235	
2	12336	
3	1323432432	
4		
5		

*Упражнение 3.* В решении задачи замените ввод цифр числа с клавиатуры на их генерацию случайным образом. При этом учтите, что первая цифра числа не должна равняться нулю. Используйте функцию *IntToStr*.

*Упражнение 4.* Проведите исследование, как часто при случайной генерации выпадают числа, которые делятся на 12. При этом количество цифр зафиксируйте, например, оно равно шести).

*Упражнение 5.* Измените код программы так, чтобы она запрашивала количество чисел, необходимых для исследования частоты встречаемости чисел, кратных 12. Определите процентное отношение чисел, кратных и не кратных 12. Что оказывает большее влияние на результат: количество цифр в числе или количество испытаний?

*Упражнение 6.* Измените диапазон возможных значений цифр числа (например, от 3 до 6). Как изменится процентное соотношение чисел, кратных и не кратных 12? Протестируйте программу для других числовых промежутков (например, 2–5 или 6–9), содержащих 4 цифры.

**Решение.**

1. Для выполнения *упражнения 1* необходимо скопировать программу, сохранить ее, а затем дважды запустить и ввести соответственно значения 2641 и 1498796006400. В каждом случае надо сверить полученный ответ с приведенным в таблице.

2. Результат в *упражнении 2* (таблица 6) получен впоследствии выполнения с использованием калькулятора третьего пункта *Этапа 3. Составление алгоритма решения задачи*, проделанного программистом.

Для тестов 4 и 5 могут быть разные варианты. Два из возможных представлены в таблице (таблица 6).

Таблица 6

**Данные для тестирования кода программы**

Номер теста	Ввод <i>a</i>	Вывод
1	2235	Нет
2	12336	Да
3	1323432432	Да
4	22222	Нет
5	2222220	Да

**3. Код программы:**

```

var n, i, s, d: integer;
    a: string;
begin
  writeln ('Введите количество цифр');
  readln(n);
  a:=IntToStr(random(9)+1);
  for i:=2 to n do
    a:=a+IntToStr(random(10));
  writeln('Число - ', a);
  s:=0;
  for i:=1 to n do
    s:= s+StrToInt(a[i]);
  d:=10*StrToInt(a[n-1])+StrToInt(a[n]);
  if (s mod 3=0) and (d mod 4=0) then
    writeln('Да')
  else
    writeln('Нет');
end.

```

4. Результат может быть разным. Теоретически вариант «Да» может появиться 1 раз при 12 испытаниях.

Код программы:

```

var n, i, s, d, k, k1, k2, j: integer;
    a: string;
    z_da, z_net: real;
begin
  writeln('Введите количество испытаний');
  readln(k);
  writeln('Введите количество цифр');
  readln(n);
  k1 := 0;
  k2 := 0;
  for j := 1 to k do
  begin
    a := IntToStr(random(9) + 1);
    for i := 2 to n do
      a := a + IntToStr(random(10));
    s := 0;
    for i := 1 to n do
      s := s + StrToInt(a[i]);
    d := 10 * StrToInt(a[n - 1]) + StrToInt(a[n]);
    if (s mod 3 = 0) and (d mod 4 = 0) then
      k1 := k1 + 1
    else
      k2 := k2 + 1;
    end;
    z_da := k1 / k * 100;
    z_net := k2 / k * 100;
    writeln('Да - ', z_da:7:3, '%. Нет - ', z_net:7:3, '%');
  end.

```

Код программы:

```

var n, i, s, d, k, k1, k2, j: integer;
    a: string;
    z_da, z_net: real;
begin
  writeln('Введите количество испытаний');
  readln(k);
  writeln('Введите количество цифр');
  readln(n);
  k1 := 0;
  k2 := 0;
  for j := 1 to k do
  begin
    a:='';
    for i := 1 to n do
      a := a + IntToStr(random(4)+3);
    writeln(a);
    s := 0;
    for i := 1 to n do
      s := s + StrToInt(a[i]);
    d := 10 * StrToInt(a[n - 1]) + StrToInt(a[n]);
    if (s mod 3 = 0) and (d mod 4 = 0) then
      k1 := k1 + 1
    else
      k2 := k2 + 1;
    end;
    z_da := k1 / k * 100;
  end.

```

```

z_net := k2 / k * 100;
writeln('Да - ', z_da:7:3, '%. Нет - ', z_net:7:3, '%');
end.

```

**Задание 2.** Программисту было дано следующее техническое задание: «С клавиатуры введите цифры числа в том порядке, в котором они следуют в числе. Определите, делится ли данное число на 132. Количество цифр от 2 до 255. Предполагаемый формат ввода и вывода данных представлен в таблице (таблица 7)».

Таблица 7

**Пример исходных данных и результата**

Пример ввода	Пример вывода
2641	Нет
1498796006400	Да

Программист подготовил отчет о проделанной части работы. К сожалению, в момент его сохранения произошел сбой, и восстановленный отчет имеет следующий вид:

Для решения задачи было запланировано выполнить следующие этапы:

1. Определение исходных данных.
2. Определение результатов.
3. Составление алгоритма решения задачи.
4. Описание переменных.
5. Написание программы.
6. Тестирование программы.

**Описание выполненных этапов**

*Этап 1. Определение исходных данных.*

Переменная \_\_\_\_\_ – строка для хранения цифр числа.

*Этап 2. Определение результатов.*

Ответ «Да», если число делится на 132, или ответ «Нет», если не делится на 132.

*Этап 3. Составление алгоритма решения задачи.*

Ввод исходных данных.

Если бы количество цифр в числе было небольшим, то можно было бы проверить его делимость с помощью операции *div*. Но число из 255 цифр не поместится даже в самый большой целочисленный тип данных \_\_\_\_\_, хранящий до \_\_\_\_\_ цифр числа. Поэтому воспользуемся признаками делимости. Если разложить число 132 на взаимно простые множители, то получим, что  $132=3 \cdot 4 \cdot 11$ . Следовательно, для того, чтобы делиться на 132, число должно одновременно делиться и на 3, и на 4, и на 11. Значит, должны выполняться три условия:

*Условие 1.* Для проверки делимости на \_\_\_\_\_ найдем сумму цифр числа. Для этого превратим каждый символ в число с помощью функции

*StrToInt*. Если полученная сумма цифр числа будет делиться на \_\_\_\_\_, то и исходное число делится на \_\_\_\_\_.

*Условие 2.* Для проверки делимости на \_\_\_\_\_ сконструируем двузначное число  $d$  из двух последних цифр исходного числа. Если это число будет делиться на \_\_\_\_\_, то и исходное число делится на \_\_\_\_\_.

*Условие 3.* Для проверки делимости на 11 необходимо сконструировать число, равное модулю разности между суммой цифр, занимающих нечетные позиции, и суммой цифр, занимающих четные позиции. Если это число будет делиться на 11, то и исходное число делится на 11.

Если *условие 1*, *условие 2* и *условие 3*, описанные в пункте 2, выполняются одновременно, то выводим сообщение «Да», иначе – сообщение «Нет».

*Этап 4. Описание переменных.*

Переменные  $n$ ,  $S$ ,  $S1$ ,  $S2$ ,  $d$  имеют тип *integer*, переменная  $a$  имеет тип \_\_\_\_\_.

Для работы также необходима целая переменная  $i$  – счетчик цикла.

*Этап 5. Написание программы.*

```

var n, i, s, s1, s2, d: integer;
    __: string;
begin
  writeln('Введи число');
  readln(__);
  n := length(a);
  s := 0;
  for i := 1 to n do
    s := s + _____;
  s1 := 0;
  for i := 1 to n div 2 do
    s1 := s1 + _____(a[i * 2]);
  s2 := s - s1;
  d := _____;
  if (s _____ 3 = 0) and (abs(s2 - s1) mod _____ = 0)
and (d mod _____ = 0) then
    writeln('Да')
  else
    writeln('Нет');
end.

```

После восстановления поврежденного файла оказалось, что некоторые места в отчете не сохранились. Помогите заполнить пропуски в документе и реализовать оставшийся этап – *этап тестирования программы*, выполнив следующие упражнения.

*Упражнение 1.* Запустите программу и введите значения из условия. Проверьте правильность результата.

*Упражнение 2.* Заполните таблицу для тестирования кода программы (таблица 8). Значения для двух последних тестов придумайте самостоятельно.

**Шаблон заполнения данными для тестирования программы**

Номер теста	Ввод <i>a</i>	Вывод
1	235	
2	12540	
3	1323432432	
4		
5		

*Упражнение 3.* Проведите исследования, аналогичные *заданию 1* (упражнения 3–6).

**Решение.**

Восстановленный отчет программиста:

Описание выполненных этапов.

*Этап 1. Определение исходных данных.*

Переменная *a* – строка для хранения цифр числа.

*Этап 2. Определение результатов.*

Ответ «Да», если число делится на 132, или ответ «Нет», если не делится на 132.

*Этап 3. Составление алгоритма решения задачи.*

Ввод исходных данных.

Если бы количество цифр в числе было небольшим, то можно было бы проверить его делимость с помощью операции *div*. Но число из 255 цифр не поместится даже в самый большой целочисленный тип данных `int64`, хранящий 9 цифр числа. Поэтому воспользуемся признаками делимости. Если разложить число 132 на взаимно простые множители, то получим, что  $132=3\cdot 4\cdot 11$ . Следовательно для того, чтобы делиться на 132, число должно одновременно делиться и на 3, и на 4, и на 11. Значит, должны выполняться три условия:

*Условие 1.* Для проверки делимости на 3 найдем сумму цифр числа. Для этого превратим каждый символ в число с помощью функции *StrToInt*. Если полученная сумма цифр числа будет делиться на 3, то и исходное число делится на 3.

*Условие 2.* Для проверки делимости на 4 сконструируем двузначное число *d* из двух последних цифр исходного числа. Если это число будет делиться на 4, то и исходное число делится на 4.

*Условие 3.* Для проверки делимости на 11 необходимо сконструировать число, равное модулю разности между суммой цифр, занимающих нечетные позиции, и суммой цифр, занимающих четные позиции. Если это число будет делиться на 11, то и исходное число делится на 11.

Если *условие 1*, *условие 2* и *условие 3*, описанные в пункте 2, выполняются одновременно, то выводим сообщение «Да», иначе – сообщение «Нет».

*Этап 4. Описание переменных.*

Переменные *n*, *S*, *S1*, *S2*, *d* имеют тип *integer*, переменная *a* имеет тип *string*.

Для работы также необходима целая переменная  $i$  – счетчик цикла.

*Этап 5. Написание программы.*

```

var n, i, s, s1, s2, d: integer;
    a: string;
begin
  writeln('Введите число');
  readln(a);
  n := length(a);
  s := 0;
  for i := 1 to n do
    s := s + StrToInt(a[i]);
  s1 := 0;
  for i := 1 to n div 2 do
    s1 := s1 + StrToInt(a[i * 2]);
  s2 := s - s1;
  d := 10 * StrToInt(a[n - 1]) + StrToInt(a[n]);
  if (s mod 3 = 0) and (abs(s2 - s1) mod 11 = 0)
and (d mod 4 = 0) then
    writeln('Да')
  else
    writeln('Нет');
end.

```

**Выполнение упражнений.**

1. Для выполнения упражнения 1 необходимо скопировать программу, сохранить ее, а затем дважды запустить и ввести соответственно значения 2641 и 1498796006400. В каждом случае полученный ответ следует сверить с приведенным в таблице.

Результат в упражнении 2 (таблица 9) можно получить, выполнив с помощью калькулятора третий пункт Этапа 3. Составление алгоритма решения задачи, проделанного программистом.

Для тестов 4 и 5 могут быть разные варианты. Два из возможных представлены в таблице (таблица 9).

Таблица 9

Данные для тестирования кода программы

Номер теста	Ввод $a$	Вывод
1	235	Нет
2	12540	Да
3	1323432432	Да
4	2222222222	Нет
5	222222222220	Да

**Задание 3.** Напишите программу для решения задачи. С клавиатуры введите цифры числа в том порядке, в котором они следуют в числе. Определите, делится ли данное число на 396. Количество цифр от 3 до 255.

**Решение.**

```

var n, i, s, s1, s2, d: integer;
    a: string;
begin
  writeln('Введите число');

```

```

readln(a);
n := length(a);
s := 0;
for i := 1 to n do
    s := s + StrToInt(a[i]);
s1 := 0;
for i := 1 to n div 2 do
    s1 := s1 + StrToInt(a[i * 2]);
s2 := s - s1;
d := 10 * StrToInt(a[n - 1]) + StrToInt(a[n]);
if (s mod 9 = 0) and (abs(s2 - s1) mod 11 = 0)
    and (d mod 4 = 0) then
    writeln('Да')
else
    writeln('Нет');
end.

```

**Задание 4.** Напишите программу для решения задачи. С клавиатуры введите цифры числа в том порядке, в котором они следуют в числе. Определите, делится ли данное число на 72. Количество цифр от 2 до 255.

**Решение.**

```

var n, i, s, d: integer;
    a: string;
begin
    writeln('Введите число');
    readln(a);
    n := length(a);
    s := 0;
    for i := 1 to n do
        s := s + StrToInt(a[i]);
    if n > 2 then
        d := 100 * StrToInt(a[n - 2]) +
            10 * StrToInt(a[n - 1]) + StrToInt(a[n])
    else
        if d = 2 then
            d := 10 * StrToInt(a[n - 1]) + StrToInt(a[n]);
    if (s mod 9 = 0) and (d mod 8 = 0) then
        writeln('Да')
    else
        writeln('Нет');
end.

```

## ПРИЛОЖЕНИЕ 7

## ОБРАБОТКА ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

**Задание.** Оплата за двухкомнатную квартиру в 2015 году за месяц в летний период (с 15 апреля по 14 октября) составила 181 260 руб. (неденоминированных). В зимний период (с 15 октября по 14 апреля) оплата возрастает на 53,48 %. Оплату за каждый месяц необходимо вносить до 25 числа следующего месяца. В случае задержки платежа данных услуг ежедневно начисляется пеня в размере 3 % от общей суммы за квартиру.

Постройте компьютерную модель «Оплата коммунальных услуг» в MS Excel и ответьте на следующие вопросы:

1. Сколько понадобится денег для оплаты коммунальных услуг в квартире в зимний период?

2. Вычислите величину годовой оплаты за квартиру для двух случаев: при своевременной оплате и при задержке платежей (в феврале оплата коммунальных услуг была задержана на 9 дней, в марте – на 3 дня и в августе – на 15 дней). На сколько рублей возрастет оплата за год при задержке платежей?

3. Как изменится ответ на предыдущий вопрос, если пеня в размере 3 % будет рассчитываться не от количества денег, необходимых к оплате за месяц, а от суммы долга на каждый день просрочки?

Расчетные таблицы расположите на листах 1–3 (название листа 1 – «Вопрос 1», листа 2 – «Вопрос 2» и листа 3 – «Вопрос 3»).

**Решение.**

1. Разработаем математическую модель для ответа на вопрос 1.

Пусть  $Sl$  руб. – денежная сумма, которую необходимо уплачивать за квартиру в летний период,  $Sz$  руб. – денежная сумма, которую необходимо уплачивать в зимний период,  $h$  % – разница между оплатой квартиры в зимний и летний периоды.

Если размер платы в летний период известен, то денежную сумму для оплаты квартиры в зимний период можно определить по формуле:

$$Sz = Sl + Sl \cdot \frac{h}{100} \quad \text{или} \quad Sz = \left(1 + \frac{h}{100}\right) \cdot Sl$$

При ответе на первый вопрос используется формула для нахождения простого процента.

2. Построим компьютерную модель для ответа на вопрос 1.

В начале листа «Вопрос 1» оформим исходные данные (рис. 22).

Компьютерная модель "Оплата коммунальных услуг"		
Исходные данные:		
$Sl$ , руб.	81260	– количество денег, необходимых для оплаты за квартиру в летний период.
$h$ %	53,48	– разница между оплатой за квартиру в зимний и летний периоды.
Вопрос:		
1. Сколько понадобится денег для оплаты двухкомнатной квартиры за зимний период?		

Рисунок 22. Исходные данные для ответа на вопрос 1

Построим расчетную таблицу, в которой ежемесячно отобразим суммы для оплаты коммунальных услуг в зимний период (рис. 23). Введем в ячейку C16 формулу  $=\text{ОКРУГЛ}((1+B\$6/100)*B\$5;10)$ .

Полученное значение необходимо округлить до десятков. Воспользуйтесь функцией ОКРУГЛ(). Обратите внимание: в формуле следует использовать смешанные ссылки. Это позволит корректно скопировать ее для остальных ячеек.

Месяц	Оплата за квартиру
Ноябрь	124 720
Декабрь	124 720
Январь	124 720
Февраль	124 720
Март	124 720
Апрель/Октябрь	124 720

Рисунок 23. Построение расчетной таблицы

Вычислим общую сумму для оплаты коммунальных услуг за зимний период в ячейке C23 (рис. 24). Используйте функцию СУММ(). Ее можно вставить комбинацией клавиш Alt + =.

Февраль	124 720
Март	124 720
Апрель/Октябрь	124 720
Итого:	748 320

Рисунок 24. Вычисление общей суммы

### 3. Разработаем математическую модель для ответа на вопрос 2.

Пусть  $p\%$  – пеня за каждый день просрочки квартплаты, а  $n$  – число просроченных дней,  $S_n$  – сумма, которую необходимо заплатить после  $n$  дней просрочки. Тогда за  $n$  дней просрочки пеня составит  $pn\%$  от  $S$  или  $\frac{pn}{100}S$ , а всего придется заплатить  $S + \frac{pn}{100}S$ , где  $S$  – денежная сумма, которую необходимо уплачивать за квартиру в летний или зимний период. Таким образом,  $S_n = \left(1 + \frac{pn}{100}\right)S$ .

При ответе на второй вопрос используется формула для нахождения простого процента.

### 4. Построим компьютерную модель для ответа на вопрос 2.

В начале листа «Вопрос 2» оформим исходные данные (рис. 25). Сделаем ссылки на все исходные данные листа «Вопрос 1». Это позволит повысить эффективность использования разрабатываемой компьютерной модели.

Row	Label	Value	Description
3	Исходные данные:		
5	$S_1$ , руб.	81 260	— количество денег, необходимых для оплаты за квартиру в летний период.
6	$S_2$ , руб.	124 720	— количество денег, необходимых для оплаты за квартиру в зимний период.
7	$p$ , %	3	— пеня за каждый день просрочки квартплаты.
8	$n_1$ , дней	9	— количество дней, на которые была задержана оплата за квартиру в феврале.
9	$n_2$ , дней	3	— количество дней, на которые была задержана оплата за квартиру в марте.
10	$n_3$ , дней	15	— количество дней, на которые была задержана оплата за квартиру в августе.
12	Вопрос:		
14	На сколько рублей возрастет годовая оплата за двухкомнатную квартиру, если в феврале оплата за коммунальные услуги была задержана на 9 дней, в марте – на 3 дня и в августе – на 15 дней?		

Рисунок 25. Исходные данные для ответа на вопрос 2

Вычислим суммы для оплаты коммунальных услуг для месяцев, в которых начислялась пеня.

В зимний период для оплаты коммунальных услуг за месяц потребуется 124 720 руб. Согласно математической модели, денежные суммы за февраль и март будут вычисляться следующим образом:  $\left(1 + \frac{3 \cdot 9}{100}\right) \cdot 124720$  и  $\left(1 + \frac{3 \cdot 3}{100}\right) \cdot 124720$ , где  $\frac{3 \cdot 9}{100}$  и  $\frac{3 \cdot 3}{100}$  – пени за 9 и 3 дня просрочки, соответственно (рис. 26).

Row	Label	Value
16	Подсчет суммы квартплаты за февраль	
19		158 394

а) февраль

Row	Label	Value
16	Подсчет суммы квартплаты за март	
19		135 945

б) март

Рисунок 26. Подсчет необходимой суммы для платы за квартиру с учетом пени в зимний период

Количество денег, необходимое в месяц для оплаты квартиры в летний период, составляет 81 260 руб. Согласно математической модели, денежная сумма для августа будет  $\left(1 + \frac{3 \cdot 15}{100}\right) \cdot 81260$ , где  $\frac{3 \cdot 15}{100}$  – пеня за 15 дней просрочки (рис. 27).

Row	Label	Value
16	Подсчет суммы квартплаты за август	
19		117 827

Рисунок 27. Подсчет необходимой суммы для платы за квартиру с учетом пени в летний период

Построим расчетную таблицу (рис. 28), в которой ежемесячно отобразим суммы для оплаты коммунальных услуг в зимний и летний периоды в случаях своевременной оплаты и при задержке платежей. В ячейки C26, C27, C34 введем формулы, возвращающие результаты округления значений ячеек A19, E19, I19. Точность округления, как и в случае ответа на вопрос 1, до десятков.

Для определения сумм, необходимых для оплаты квартиры без пени и с пеней, в ячейки B39 и C39 введем формулы =СУММ(B23:B28;B30:B35) и =СУММ(C23:C28;C30:C35).

Месяц	Денежные средства, необходимые	
	при своевременной оплате	при задержке платежей
Ноябрь	124 720	124 720
Декабрь	124 720	124 720
Январь	124 720	124 720
Февраль	124 720	158 390
Март	124 720	135 940
Апрель/Октябрь	124 720	124 720
Апрель/Октябрь	81 260	81 260
Май	81 260	81 260
Июнь	81 260	81 260
Июль	81 260	81 260
Август	81 260	117 830
Сентябрь	81 260	81 260
Годовая оплата за квартиру		
	Без пени	С наличием пени
	1 235 880	1 317 340

Рисунок 28. Расчетная таблица для ответа на вопрос 2

Заполним область «*Ответ*» (рис. 29).

40										
41	Ответ:									
42										
43		При своевременной оплате величина годовой оплаты за двухкомнатную квартиру составит								
44		1 235 880 рублей.								
45		При задержке платежей величина годовой оплаты за двухкомнатную квартиру составит								
46		1 317 340 рублей.								
47		Годовая оплата за двухкомнатную квартиру с учетом задержки оплаты в феврале, марте								
48		и августе возрастет на 81 460 рублей.								
49										

Рисунок 29. Оформление ответа на вопрос 2

### 5. Разработаем математическую модель для ответа на вопрос 3.

В отличие от предыдущего случая речь идет о том, что пеня  $p\%$  начисляется на общую сумму долга *ежедневно* в течение  $n$  дней. За первый день сумма оплаты  $S$  увеличится на  $p\%$ , т. е. на  $\frac{p}{100} \cdot S$  рублей, и составит

$S_1 = \left(1 + \frac{p}{100}\right) \cdot S$  рублей. Таким образом, начальная сумма увеличится в  $1 + \frac{p}{100}$  раза.

За следующий день сумма  $S_1$  увеличится во столько же раз, и поэтому через два дня необходимо будет оплатить

$$S_2 = \left(1 + \frac{p}{100}\right) \cdot S_1 = \left(1 + \frac{p}{100}\right) \cdot \left(1 + \frac{p}{100}\right) \cdot S = \left(1 + \frac{p}{100}\right)^2 \cdot S.$$

Аналогично  $S_3 = \left(1 + \frac{p}{100}\right)^3 \cdot S$  и так далее. Другими словами, справедливо равенство:

$$S_n = \left(1 + \frac{p}{100}\right)^n \cdot S.$$

При ответе на третий вопрос используется формула для нахождения сложного процента.

*б. Построим компьютерную модель для ответа на вопрос 3.*

Сделаем копию листа «Вопрос 2». Переименуем скопированный лист «Вопрос 3».

В диапазон ячеек  $B5:B10$  вставим относительные ссылки на соответствующие данные листа «Вопрос 2». Изменим текст вопроса в ячейке  $A14$  согласно условию (рис. 30).

	A	B	C	D	E	F	G	H	I	J	K	L	M
2													
3	Исходные данные:												
4													
5	$S_1$ , руб.	81 260	— количество денег, необходимых для оплаты за квартиру в летний период.										
6	$S_2$ , руб.	124 720	— количество денег, необходимых для оплаты за квартиру в зимний период.										
7	$p$ , %	3	— пеня за каждый день просрочки квартплаты.										
8	$n_1$ , дней	9	— количество дней, на которые была задержана оплата за квартиру в феврале.										
9	$n_2$ , дней	3	— количество дней, на которые была задержана оплата за квартиру в марте.										
10	$n_3$ , дней	15	— количество дней, на которые была задержана оплата за квартиру в августе.										
11													
12	Вопрос:												
13													
14	Как изменится ответ на предыдущий вопрос, если пеня будет рассчитываться в размере 3 % не от количества денег, необходимых к оплате за месяц, а от суммы долга на каждый день просрочки?												
15													

**Рисунок 30.** Исходные данные для ответа на вопрос 3

В строке  $B19$  удалим все данные и обратим внимание, что в ячейках  $C26$ ,  $C27$  и  $C34$  значения станут равными нулю.

Вычислим суммы, для оплаты коммунальных услуг за месяцы, в которых начислялась пеня, используя формулу для нахождения капитализированного процента.

После строки  $B19$  вставим 16 строк. Число строк определяется количеством дней, для которых необходимо найти пеню. В нашем случае наибольшее количество дней просрочки в августе – 15.

Для определения размера долга необходимо проводить его подсчет по окончании каждого дня, т. к. эта сумма влияет на определение величины пени на последующий день (в случае неоплаты коммунальных услуг). С учетом этого расчеты для февраля проведите следующим образом (рис. 31):

Подсчет суммы квартплаты за февраль				Подсчет суммы квартплаты за февраль			
18я	День	Сумма	Процент	18	День	Сумма	Процент
19я	1	$=(1+C20/100)*B6$	$=B6$	19	1	128 462	
20я	$=A20+1$	$=(1+C20/100)*B20$	$=C20$	20	2	132 315	
21я	$=A21+1$	$=(1+C21/100)*B21$	$=C21$	21	3	136 285	
22я	$=A22+1$	$=(1+C22/100)*B22$	$=C22$	22	4	140 373	
23я	$=A23+1$	$=(1+C23/100)*B23$	$=C23$	23	5	144 585	
24я	$=A24+1$	$=(1+C24/100)*B24$	$=C24$	24	6	148 922	
25я	$=A25+1$	$=(1+C25/100)*B25$	$=C25$	25	7	153 390	
26я	$=A26+1$	$=(1+C26/100)*B26$	$=C26$	26	8	157 992	
27я	$=A27+1$	$=(1+C27/100)*B27$	$=C27$	27	9	162 731	
28я				28			

а) формулы

б) значения

**Рисунок 31.** Подсчет суммы оплаты за февраль с пеней, рассчитанной с учетом капитализированного процента

Обратите внимание, что при заполнении столбца «Процент» используется ссылка на процент предыдущего дня.

Аналогичным образом проводится подсчет суммы оплаты за март и август. В ячейках G20 и K20 находятся ссылки на ячейки C28 и G22 соответственно (рис. 32).

Построение расчетных таблиц:												
Подсчет суммы квартплаты за февраль				Подсчет суммы квартплаты за март				Подсчет суммы квартплаты за август				
18	День	Сумма	Процент	18	День	Сумма	Процент	18	День	Сумма	Процент	
20	1	128 462	3	20	1	128 462	3	20	1	83 698	=G22	
21	2	132 315	3	21	2	132 315	3	21	2	86 209	3	
22	3	136 285	3	22	3	136 285	3	22	3	88 795	3	
23	4	140 373	3	23	4			23	4	91 459	3	
24	5	144 585	3	24	5			24	5	94 203	3	
25	6	148 922	3	25	6			25	6	97 029	3	
26	7	153 390	3	26	7			26	7	99 940	3	
27	8	157 992	3	27	8			27	8	102 938	3	
28	9	162 731	3	28	9			28	9	106 026	3	
								29	10	109 207	3	
								30	11	112 483	3	
								31	12	115 857	3	
								32	13	119 333	3	
								33	14	122 913	3	
								34	15	126 600	3	

**Рисунок 32** Подсчет суммы, необходимой для оплаты за февраль, март и август, с пеней, рассчитанной с учетом капитализированного процента

В ячейках C41, C42 и C49 вставим соответственно ссылки на ячейки B28, F22 и J34.

Остальные расчеты будут проведены автоматически (рис. 33).

	A	B	C	D	E	F	G	H	I	J
51										
52	Годовая оплата за квартиру									
		Без пени	С наличием пени							
53										
54		1 235 880	1 330 790							
55										
56	Ответ:									
57										
58		При своевременной оплате величина годовой оплаты за двухкомнатную квартиру составит								
59		1 235 880 рублей.								
60		При задержке платежей величина годовой оплаты за двухкомнатную квартиру составит								
61		1 330 790 рублей.								
62		Годовая оплата за двухкомнатную квартиру с учетом задержки оплаты в феврале, марте								
63		и августе возрастет на		94 910	рублей.					
64										

**Рисунок 33.** Ответ на вопрос 3

После рассмотрения задачи с неденоминированными данными целесообразно предложить учащимся произвести в соответствующих ячейках необходимое форматирование данных и редактирование формул и пересчитать полученные результаты в современных денежных единицах.

### КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ МОДЕЛИ

**Задание 1.** Проанализируйте табличную информационную модель (таблица 10) и ответьте на вопрос: «Каким браузером пользуется каждый из ребят?»

Таблица 10

#### Табличная информационная модель

	Михась	Ясь	Левон	Кастусь	Янка
Яндекс браузер	0	0	0	1	0
<i>Firefox</i>	0	1	0	0	0
<i>Opera</i>	0	0	1	0	0
<i>Google Chrome</i>	1	0	0	0	0
<i>Safari</i>	0	0	0	0	1

**Ответ:** Михась – *Google Chrome*, Ясь – *Firefox*, Левон – *Opera*, Кастусь – Яндекс браузер, Янка – *Safari*.

**Задание 2.** Ученики IX класса принимали участие в олимпиаде по информатике. Василь решил 1, 2 и 4-ю задачи, Михась – 2, 3 и 5-ю задачи, Антось – 1, 3 и 4-ю. Степан не решил 1-ю и 4-ю задачи. Семен решил все, кроме 2-й и 4-й.

Какая задача на олимпиаде оказалась самой сложной, а какая – самой легкой? Решение оформите в виде графа с вершинами (рис. 34).

Василь	Михась	Антось	Степан	Семен
1 задача	2 задача	3 задача	4 задача	5 задача

Рисунок 34. Шаблон для построения графа

**Ответ:** 4 задача самая сложная, 3 задача самая легкая (рис. 35).

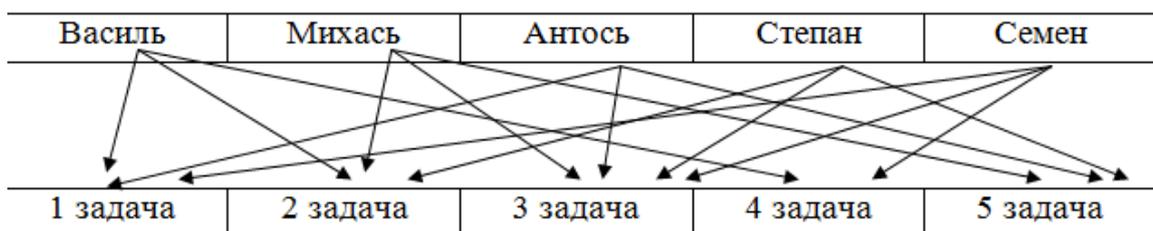


Рисунок 35. Решение в виде графа

**Задание 3.** Одноклассники Максим, Сергей, Артем, Илья зарегистрировали аккаунты в социальной сети *ВКонтакте*. Максим добавил в друзья Сергея и Артема, Артем добавил в друзья Илью, Илья – Максима.

Артем разместил на своей странице ссылку на ролик в *YouTube*. Кто из ребят не увидит видео? Решение оформите в виде графа.

**Ответ:** Сергей.

**Задание 4.** Марина заботится о безопасности информации на своем компьютере, поэтому перед использованием флэшки всегда проверяет ее на наличие вредоносных программ.

Антивирусная программа на компьютере Марины за первую минуту просканировала 10 файлов. За каждую последующую минуту количество просканированных файлов увеличивается на 25% по сравнению с предыдущей.

Через сколько минут Марина сможет работать с флэшкой, содержащей 427 файлов? Решите задачу средствами электронных таблиц.

**Ответ:** 18 мин.

**Задание 5.** Учащиеся украшали школу к Новому году. У каждого класса были гирлянды разного цвета: у VII класса – красные, у VIII класса – бирюзовые и синие, у IX класса – оранжевые и синие. У X класса есть гирлянды всех перечисленных цветов, а у XI класса нет бирюзового, синего и салатового.

При оформлении школы каждый класс использовал гирлянды только одного цвета. Помогите определить, гирлянды какого цвета использовал каждый из классов? Для решения задачи создайте табличную модель.

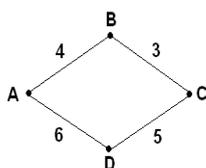
**Ответ:** VII класс – красные, VIII класс – бирюзовые, IX класс – синие, X класс – салатовые, XI класс – оранжевые.

**Задание 6.** В таблице приведена стоимость перевозок между соседними железнодорожными станциями (таблица 11). Укажите схему (рис. 36), соответствующую таблице.

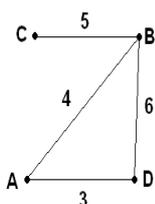
Таблица 11

Стоимость перевозок

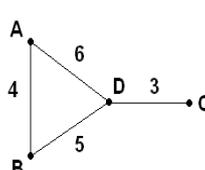
	A	B	C	D
A		4		5
B	4		3	6
C		3		
D	5	6		



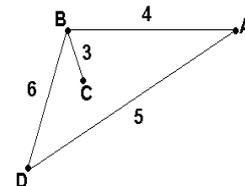
а) схема 1



б) схема 2



в) схема 3



г) схема 4

**Рисунок 36** Набор схем с информацией о стоимости перевозок

**Решение.**

В формулировке задания приводится табличная модель стоимости перевозок между двумя соседними железнодорожными станциями и 4 модели в виде схем. Выполнение задания сводится к отысканию схемы, соответствующей табличной модели.

Перепишем таблицу в следующем виде: пара соседних станций – стоимость перевозок между ними. Поскольку в данном случае стоимость перевозок между соседними станциями не зависит от направления, написав станцию  $XU$ , симметричную ей станцию  $UX$  можно не писать. Итак, имеем:  $AB - 4$ ;  $AD - 5$ ;  $BC - 3$  и  $BD - 6$ .

Получили полный список элементов схемы. Только одна схема соответствует этому списку – схема 4.

**Ответ:** схема 4.

**Задание 7.** Между четырьмя местными аэропортами – «Луговое», «Дятлово», «Никитино» и «Орехово» ежедневно выполняются авиарейсы. Приведен фрагмент расписания перелетов между ними (рис. 37).

Аэропорт вылета	Аэропорт прилета	Время вылета	Время прилета
ДЯТЛОВО	ЛУГОВОЕ	10:15	10:55
ЛУГОВОЕ	НИКИТИНО	10:20	11:00
ОРЕХОВО	ЛУГОВОЕ	10:25	12:05
ЛУГОВОЕ	ДЯТЛОВО	10:30	11:15
НИКИТИНО	ЛУГОВОЕ	10:55	11:40
ОРЕХОВО	ДЯТЛОВО	11:10	11:55
ЛУГОВОЕ	ОРЕХОВО	11:50	13:30
ДЯТЛОВО	ОРЕХОВО	12:00	12:50
НИКИТИНО	ОРЕХОВО	12:05	12:55
ОРЕХОВО	НИКИТИНО	12:10	12:55

**Рисунок 37.** Фрагмент расписания перелетов между аэропортами

Путешественник оказался в аэропорту «Луговое» в полночь. Определите самое раннее время, когда он сможет попасть в аэропорт «Орехово». Считается, что путешественник успевает совершить пересадку в аэропорту, если между временем прилета в этот аэропорт и временем вылета проходит не менее часа.

**Решение.**

Можно решить эту задачу, глядя на таблицу и перебирая подходящие варианты, но есть риск ошибиться или пропустить нужную строчку. Поэтому рекомендуется нарисовать дерево всех возможных путей из аэропорта «Луговое» в «Орехово» (рис. 38).



**Рисунок 38.** Дерево всех возможных путей из аэропорта «Луговое» в «Орехово»

Средняя ветка не подходит, т. к. между прилетом в аэропорт «Дятлово» (11:15) и вылетом из него в «Орехово» (12:00) интервал меньше часа.

Из оставшихся двух выбираем более раннее время прилета: 12:55.

**Ответ:** 12:55.