

Государственное бюджетное нетиповое общеобразовательное учреждение
«Губернаторский многопрофильный лицей-интернат»

Рассмотрено:

на заседании МО учителей точных наук

Протокол № 1 от «23» августа 2021 г.

Руководитель МО

_____ / _____

Утверждено:

педагогическим советом

Протокол № 1 от «23» августа 2021г.

Председатель педагогического совета

Директор ГБНОУ «ГМЛИ» _____ /

Мурышкина Е.В.

РАБОЧАЯ ПРОГРАММА
курса внеурочной деятельности
«Школа программиста (Разработка приложений средствами ООП)»

10-11 класс

Составитель: учитель информатики и ИКТ высшей квалификационной категории

Змысля Олеся Александровна

2021- 2022 учебный год

Кемерово 2021

Пояснительная записка

Рабочая программа курса внеурочной деятельности по информатике «Школа программиста (Разработка приложений средствами ООП)» предназначена для обучающихся 10-11 класса технологического профиля, освоивших базовый курс информатики в основной школе.

Данная рабочая программа разработана на основе следующих нормативных документов:

- Закона РФ «Об образовании»;
- Федерального государственного образовательного стандарта среднего общего образования (ФГОС СОО);
- основных подходов к развитию и формированию универсальных учебных действий (УУД) для среднего общего образования
- требования государственного образовательного стандарта среднего общего образования к результатам освоения основной образовательной программы среднего общего образования;
- требования к оснащению образовательного процесса в соответствии с содержательным наполнением учебных предметов Федерального компонента государственных стандартов образования;
- требования к результатам освоения основной образовательной программы (личностным, метапредметным, предметным).

Эпоха быстрых социальных изменений и стремительный прогресс в области информационных технологий предъявляют повышенные требования к развитию познавательной активности молодого поколения. Поэтому основными задачами любого учебного курса информационно-технологической направленности являются развитие индивидуальности учащихся и высвобождение их творческого потенциала в процессе освоения средств информационных технологий.

Технология объектно-ориентированного программирования (ООП) является современной технологией создания компьютерных программ-приложений. Интегрированная среда разработки Лазарус предоставляет программисту инструменты, которые являются конструктивными средствами для реализации творческих идей.

Активизация познавательного процесса позволяет учащимся более полно выразить свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению приложений, созданных в среде Lazarus, и в других учебных курсах, а также способствует возникновению дальнейшей мотивации, направленной на освоение профессии программиста.

Курс включает в себя практическое освоение техники создания программ-приложений различной направленности, с использованием объектно-ориентированного подхода в программировании.

Курс служит средством внутривидеальной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся.

Основа курса – практическая и продуктивная направленность занятий, способствующая позитивному изменению индивидуально-личностных качеств учащихся. Одна из целей обучения работе в среде Lazarus заключается в предоставлении школьникам возможности личностного самовыражения и самореализации в программных продуктах, созданных в этой среде. Достижение цели становится возможным при создании лично значимой для учащихся образовательной продукции в рамках программирования программ-приложений.

Реализация творческих замыслов учеников осуществляется поэтапно:

- на первом этапе создаются простейшие программы-приложения;
- на втором этапе уделяется особое внимание объектно-ориентированным

- принципам работы с компонентами ОС посредством среды Lazarus. Рассказывается о работе с некоторыми классами языка Lazarus;
- на третьем этапе создаются приложения для работы с графикой.

Цели и задачи курса:

- сформировать у школьников целостное представление о разработке программных приложений с помощью объектно-ориентированных подходов в программировании;
- реализовать способности учащихся в ходе создания различных приложений;
- сформировать элементы информационной компетенции по отношению к знаниям, умениям и опыту создания программных продуктов с использованием объектно-ориентированных технологий;
- понять принципы объектно-ориентированного подхода к составлению алгоритмов и программ, значение объектно-ориентированного программирования в современном множестве информационных технологий;
- овладеть базовыми понятиями объектно-ориентированного программирования и применять их при создании приложений;
- приобрести навыки разработки программных приложений в визуальной среде программирования Lazarus.

Личностные, метапредметные и предметные результаты освоения курса

Личностные результаты

1. сформированность мировоззрения, соответствующего современному уровню развития науки и техники;
2. готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;
3. навыки сотрудничества со сверстниками и взрослыми в образовательной, учебно-исследовательской, проектной и других видах деятельности;
4. эстетическое отношение к миру, включая эстетику научного и технического творчества;
5. осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов; отношение к профессиональной деятельности как возможности участия в решении личных, общественных, государственных, общенациональных проблем.

Метапредметные результаты

1. умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;
2. умение продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;
3. владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания;

4. готовность и способность к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;

5. умение использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены.

Предметные результаты

1. сформированность представлений об инструментарию среды Lazarus, необходимом для разработки полнофункционального приложения;

2. владение основами объектно-ориентированного подхода в программировании и умение работать с некоторыми классами языка Object Pascal;

3. сформированность представлений о принципах работы основных компонентов Lazarus и умение с ними работать;

4. сформированность навыков конструирования интерфейса приложений и разработки их дизайна;

5. сформированность навыков работы с графическими средствами среды ООП;

6. владение приемами организации и самоорганизации работы по созданию приложений;

7. сформированность положительного опыта коллективного сотрудничества при создании программ;

8. овладение процедурой самооценки знаний и результатов деятельности по созданию приложений.

Технологии, используемые в учебном процессе: личностно-ориентированное обучение, дифференцированное обучение, проектные технологии.

Методы обучения:

- Объяснительно-иллюстративные.
- Репродуктивные.
- Проблемные.
- Частично-поисковые (при выполнении практических и лабораторных работ).
- Метод программированного обучения.
- Исследовательские (при выполнении лабораторных, проектных работ).
- Метод проектов.

Формы обучения:

Обще-классные формы:

- урок;
- лекция;
- лабораторно-практические занятия;
- зачетный урок;

Групповые формы обучения:

- групповая работа;
- групповые творческие задания;
- групповая лабораторно-практическая работа.

Индивидуальные формы работы в классе и дома:

- индивидуальные задания.

Содержание курса

Введение. Основные понятия

Тема 1. Техника безопасности. Введение в программирование

Техника безопасности и правила поведения в компьютерном классе.

Структура курса. Взаимосвязь курса со смежными дисциплинами.

Обзор истории развития вычислительной техники. Технические средства. Основные составные части компьютера и физические процессы, лежащие в основе его работы. Понятие о компьютерных сетях, виды сетей.

Эволюция программного обеспечения. Понятие об операционной системе. Программа, выполнение программ. Программное обеспечение. Цели и методы программирования. Основы современных технологий программирования, направления их развития в настоящее время. Роль, значение и преимущества использования объектно-ориентированного программирования. Обоснование выбора среды объектно-ориентированного программирования Lazarus.

Тема 2. Алгоритмизация. Понятия ООП

Основы разработки программ. Информация и данные. Обработка данных. Этапы разработки программ. Понятие об алгоритме, свойства алгоритмов (выполнимость, определенность, массовость, конечность, результативность). Тестирование.

Алгоритмические языки программирования. Принципы структурного программирования. Этапы решения задач с помощью ПК.

Допустимые операции и значения (арифметические и логические функции). Постоянные величины (константы) и переменные величины. Массивы. Формы записи алгоритма. Блок-схема. Основные блоки. Алгоритмические структуры. Линейные алгоритмы.

Учащиеся должны знать/понимать:

- три принципа объектно-ориентированного программирования: инкапсуляцию, наследование, полиморфизм;
- понятия объекта и класса, их различия;
- назначение методов, свойств и событий объекта;
- принципы работы с готовыми объектами и классами языка Lazarus.

Учащиеся должны уметь:

- приводить примеры классов, объектов, наследования классов.

Объект как сущность реального мира. Его моделирование посредством свойств, методов и событий. Классы как шаблоны объектов. Преимущества объектно-ориентированного подхода при организации взаимосвязи внутри одного объекта и между несколькими объектами с помощью принципов инкапсуляции, наследования и полиморфизма. Связь классов и объектов реального мира с компонентами и классами среды Lazarus.

Тема 3. О программе Lazarus

Интерфейс среды визуального программирования Lazarus.

Учащиеся должны знать/понимать:

- назначение окон среды программирования Lazarus;
- назначение основных команд меню.

Учащиеся должны уметь:

- запускать среду Lazarus;
- настраивать интерфейс среды Lazarus;
- переключаться между окнами среды Lazarus.

Запуск среды программирования Lazarus на исполнение. Основные окна среды Lazarus: главное окно, окно редактора форм, окно инспектора объектов, окно дерева объектов, окно редактора кода. Минимальная настройка среды Lazarus.

Проект в среде Lazarus

Учащиеся должны знать / понимать:

- структуру проекта.

Учащиеся должны уметь:

- создавать новый проект;
- добавлять в проект новую форму;
- переключаться на нужную форму либо на нужный модуль;
- запускать проект;
- сохранять проект.

Создание проекта в Lazarus. Добавление в проект новой формы. Переключение между формами. Структура проекта Lazarus. Сохранение проекта. Запуск проекта на исполнение.

Практическая работа 1. Первый запуск программы.

Раздел 1. Компоненты и события

Тема 4. Компоненты

Учащиеся должны знать/понимать:

- способы работы с компонентами;
- основные свойства компонентов.

Учащиеся должны уметь:

- выделять и копировать компоненты;
- помещать компоненты на форму;
- изменять свойства компонентов;
- изменять порядок расположения компонентов;
- работать с компонентами: Button, Label.

Понятие и особенности визуального программирования в среде Lazarus. Палитра компонентов, редактор форм и инспектор объектов. Создание приложения с помощью компонентов. Изменение свойств компонентов.

Элементы языка

Учащиеся должны знать/понимать:

- важность комментирования исходного кода, виды комментариев;
- классификацию типов данных языка программирования Lazarus;
- порядковые типы;
- вещественные типы.

Учащиеся должны уметь:

- преобразовывать целочисленный и вещественный типы в строковый и обратно;
- работать с основными математическими функциями среды Lazarus.

Практическая работа 2. Первые успехи.

Практическая работа 3. Диалог.

Тема 5. Арифметические операции.

Учащиеся должны знать/понимать:

- как запросить у пользователя информацию с клавиатуры и выполнить над ней арифметическую операцию либо операцию конкатенации.

Учащиеся должны уметь:

- программировать ввод, вывод чисел и арифметические операции;
- переводить из строкового типа данных в целочисленный и обратно.

Конкатенация строк. Преобразование типов (функции IntToStr и StrToInt).

Арифметические операции.

Практическая работа 4. Калькулятор.

Тема 6. Обработка исключений.

Учащиеся должны знать/понимать:

- как отобразить на форме произвольный текст программным способом;
- что такое событие и обработчик события;

- структуру программного модуля;
- основные события компонентов;
- что можно делать на этапе конструирования и на этапе написания кода.

Учащиеся должны уметь:

- создавать обработчик события;
- назначать компонентам осмысленные имена;
- редактировать исходный код в обработчике события;
- программно изменять свойства компонента и вызывать его методы;
- работать с компонентами: Edit, Button, Label;
- определять, какие события необходимы при реализации поставленной задачи, и какая у них должна быть реализация.

Обработка события OnClick. Программное изменение свойств компонентов и вызов их методов. Исходный код модуля. Однострочный редактор.

Понятие события и обработчика события в среде Lazarus. Создание обработчика события для компонента. Структура программного модуля среды Lazarus. Процесс конструирования и процесс написания кода. Программное изменение свойств компонентов и вызов их методов.

Управление событиями

Учащиеся должны знать/понимать:

- в каких случаях рационально использовать один обработчик событий для нескольких событий.

Учащиеся должны уметь:

- выполнять приведение типов для компонентов;
- назначать одно событие нескольким компонентам;
- работать с компонентом SpeedButton.

Параметр Sender в процедурах обработки событий. Приведение типов.

Практическая работа 5. Ваш вес.

Практическая работа 6. Тест по географии.

Итоговая работа 1. «Компоненты и события».

Раздел 2. Дизайн и обработка данных.

Тема 7. Работа с формами.

Учащиеся должны знать/понимать:

- в каких случаях необходимо использовать несколько форм;
- каковы преимущества и недостатки использования дополнительных форм в приложении;
- как перейти к нужной форме, к соответствующему ей модулю в проекте.

Учащиеся должны уметь:

- создавать новую и корректно удалять существующую форму;
- подключать модуль формы к модулю другой формы;
- производить обмен информацией между несколькими формами;
- работать с компонентами: StringGrid, UpDown, RadioButton, GroupBox, SpeedButton.

Создание и удаление формы. Переключение между формами. Зависимый переключатель.

Практическая работа 7. Много форм.

Тема 8. Логика.

Учащиеся должны знать / понимать:

- основные определения, связанные с алгеброй логики;
- законы алгебры логики;
- принципы работы с логическими выражениями и действия над ними;
- принципы построения таблиц истинности.

Учащиеся должны уметь:

- понимать смысл логических выражений;
- читать логические формулы;
- производить действия над логическими формулами и логическими выражениями;
- строить таблицы истинности;
- решать логические задачи

Составление логических выражений, преобразование логических выражений, действия с логическими функциями, составление таблиц истинности.

Практическая работа 8. Решение задач.

Тема 9. Массивы

Учащиеся должны знать/понимать:

- основные понятия по теме Массивы;
- принципы построения массивов;
- принципы нумерации массивов;
- возможность производить с массивами различные операции преобразования.

Учащиеся должны уметь:

- создавать одномерные массивы;
- производить над массивами операции поиска максимального и минимального элементов;
- производит сортировку одномерного массива по возрастанию и по убыванию;
- менять местами элементы массива;
- копировать один массив в другой;
- решать задачи с использованием массивов.

Ввод элементов массива с клавиатуры. Чтение элементов массива. Операции над элементами массива. Перестановка элементов массива. Печать элементов массива в заданном порядке.

Практическая работа 9. Обработка массива.

Итоговая работа 2. «Дизайн и обработка данных».

Раздел 3. Интерфейс и работа с текстом

Тема 10. Стандартный интерфейс пользователя

Учащиеся должны знать/понимать:

- функциональность стандартных элементов пользовательского интерфейса приложений.

Учащиеся должны уметь:

- создавать стандартный интерфейс пользователя для приложений;
- использовать в приложениях всплывающие подсказки (Hint);
- работать с компонентами: MainMenu, PopupMenu, ToolBar, ImageList.

Стандартный интерфейс пользователя. Главное и контекстное меню. Панель инструментов. Всплывающая подсказка (Hint). Строка состояния. Пиктограмма.

Практическая работа 10. Альбом

Практическая работа 11. Работа с меню

Практическая работа 12. Вкладки

Тема 11. Текстовые редакторы

Учащиеся должны знать/понимать:

- способы работы с компонентами-контейнерами;
- назначение класса TStrings.

Учащиеся должны уметь:

- работать со строками;
- работать с параметрами шрифта;
- работать с компонентами: Memo, CheckBox, FontsDialog, OpenFileDialog;
- создавать «резиновый» интерфейс.

Многострочный текст. «Резиновый» интерфейс. Многострочный редактор. Стандартный диалог. Компонент-контейнер. Независимый переключатель.

Практическая работа 13. Справочник

Практическая работа 14. Словарь

Итоговая работа 3. «Интерфейс и работа с текстом».

Раздел 8. Работа с графикой, решение задач

Тема 12. Работа с графикой

Теоретические сведения

Учащиеся должны знать/понимать:

- различия между компонентами Image и PictureBox;
- графические методы класса TCanvas: MoveTo, LineTo, Rectangle, Ellipse, Pie, Arc, Draw, OutText;
- свойство холста Pen;
- свойство холста Brush.

Учащиеся должны уметь:

- программировать рисование графических изображений.

Компоненты Image и PictureBox. Canvas (Канва). Линия, контур, закрашенная фигура.

Цвет и стиль линий, заливки. Графический формат файла.

Основы графики

Учащиеся должны знать/понимать:

- основы рисования линий на холсте при помощи мыши.

Учащиеся должны уметь:

- отслеживать движение мыши по компоненту и определять состояние кнопок мыши.

Рисование линий. Отслеживание движения мыши. Отслеживание состояния кнопок мыши.

Классы TPen и TBrush

Учащиеся должны знать/понимать:

- основные принципы рисования в цвете.

Учащиеся должны уметь:

- управлять параметрами пера, кисти;
- очищать канву;
- выбирать требуемый курсор для компонента;
- сохранять нарисованное изображение в формате BMP на диск.

Свойства Pen и Brush. Очистка канвы. Графический формат BMP. Вид курсора.

Класс TPicture.

Просмотр графических файлов

Учащиеся должны знать/понимать:

- уметь создавать простое приложение для просмотра графических файлов;
- принцип работы свойства Anchors.

Учащиеся должны уметь:

- загружать и отображать графические файлы;
- подключать графический формат JPG;
- работать с компонентами выбора файлов и директорий: FileListBox, DirectoryListBox и DriveComboBox.

Компоненты выбора файлов и директорий. Загрузка графических файлов с помощью компонента Image. Графический формат JPG.

Практическая работа 15. Светофор

Практическая работа 16. Мишени

Итоговая работа 3. «Работа с графикой».

Тема 13. Системы счисления.

Учащиеся должны знать/понимать:

- понятия о системах счисления;
- возможность представления чисел в различных системах счисления;
- правила действий над числами с различных системах счисления.

Учащиеся должны уметь:

- переводить числа из одной системы счисления в другую;
- производить основные арифметические действия над числами в разных системах счисления;
- определять основание системы счисления и используемые в ней символы;
- решать задачи с числами в разных системах счисления.

Практическая работа 17. Решение задач

Практическая работа 18. Решение задачи в программе.

Практическая работа 19. Решение математических задач программно.

Выполнение контрольного итогового проекта

Тематическое планирование

№ п/п	Наименование разделов и тем	Всего часов	В том числе		Характеристика видов деятельности обучающихся
			Лекции	Практ. занятия	
Введение. Основные понятия		6	5	1	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> анализировать интерфейс программного средства с позиций исполнителя, его среды функционирования, системы команд и системы отказов; выделять и определять назначения элементов окна системы программирования (область записи алгоритма, область ввода данных); рассмотрение этапов решения задач с использованием программирования. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> строить алгоритмы решения задачи с использованием основных алгоритмических конструкций; составлять блок-схему решения задачи; преобразовывать один способ записи алгоритма в другой; исполнять алгоритм; строить различные алгоритмы решения задачи как реализацию различных методов решения данной задачи; отлаживать и тестировать программы; работать с программами-конструкторами, обучающими
1	Тема 1. Техника безопасности. Введение в программирование		2		
2	Тема 2. Алгоритмизация. Понятия ООП		2		
3	Тема 3. О программе Lazarus		1		
4	Практическая работа 1. Первый запуск программы			1	
Раздел 1. Компоненты и события		14	3	11	
5	Тема 4. Компоненты		1		
6	Практическая работа 2. Первые успехи.			1	
7	Практическая работа 3. Диалог.			1	
8	Тема 5. Арифметические операции.		1		
9	Практическая работа 4. Калькулятор			2	
10	Тема 6. Обработка исключений.		1		
11	Практическая работа 5. Ваш вес.			1	
12	Практическая работа 6. Тест по географии.			2	
13	Итоговая работа 1 «Компоненты и события»			2	
Раздел 2. Дизайн и обработка данных		14	5	9	
14	Тема 7. Работа с формами.		1		
15	Практическая работа 7. Много форм			2	
16	Тема 8. Логика		2		
17	Практическая работа 8. Решение задач			2	
18	Тема 9. Массивы		2		
19	Практическая работа 9. Обработка массива.			2	

20	Итоговая работа 2 «Дизайн и обработка данных»			2	<p>программами и проводить их анализ с позиций исполнителя; •работать с программами, моделирующими деятельность</p> <ul style="list-style-type: none"> • исполнителей; • проводить компьютерные эксперименты. • знакомство с разными формами отказов, их сравнение; • составлять последовательность предписаний, описывающих ход • разработка творческих индивидуальных и групповых проектов на разные темы; • формально выполнять действия в соответствии с инструкцией. 	
Раздел 3. Интерфейс и работа с текстом		10	2	8		
21	Тема 10. Стандартный интерфейс пользователя		1			
22	Практическая работа 10. Альбом			1		
23	Практическая работа 11. Работа с меню			1		
24	Практическая работа 12. Вкладки			1		
25	Тема 11. Текстовые редакторы		1			
26	Практическая работа 13. Справочник			1		
27	Практическая работа 14. Словарь			2		
28	Итоговая работа 3 «Интерфейс и работа с текстом»			2		
Раздел 4. Работа с графикой, решение задач		28	4	24		
29	Тема 12. Работа с графикой		2			
30	Практическая работа 15. Светофор			2		
31	Практическая работа 16. Мишени			2		
32	Итоговая работа 3 «Работа с графикой»			2		
33	Тема 13. Системы счисления.		2			
34	Практическая работа 17. Решение задач.			2		
35	Практическая работа 18. Решение задачи в программе.			2		
36	Практическая работа 19. Решение математических задач программно.			6		
37	Выполнение контрольного итогового проекта			6		
38	Резерв			2		
ВСЕГО		69	19	50		

Учебно-методическое обеспечение курса

- Основы программирования в среде Lazarus. Мансуров К.Т., 2010
- Free Pascal и Lazarus: Учебник по программированию. Алексеев Е. Р., Чеснокова О. В., Кучер Т. В, 2010
- Чернов А.А. Конспекты уроков информатики в 9-11-х классах: практикум по программированию. – Волгоград: Учитель, 2006
- Шауцукова Л.З. Информатика: Учеб. Пособие для 10-11 кл. общеобразова. Учреждений. – М.: Просвещение, 2003
- <http://gospodaretsva.com/category/lazarus>
- <http://pascal-cod.ru/laz>
- <http://www.edu.sbor.net/mars/info/proekt/gambas/index.html>
- <http://www.intuit.ru/studies/courses/13745/1221/lecture/23276>
- <http://youclass.ru/clubs/18>

Программное обеспечение курса



Lazarus – свободная среда разработки программного обеспечения на языке Object Pascal для компилятора Free Pascal (часто используется сокращение FPC – Free Pascal Compiler, свободно распространяемый компилятор языка программирования Pascal). Интегрированная среда разработки предоставляет возможность кроссплатформенной разработки приложений в Delphi-подобном окружении.