



Муниципальное бюджетное общеобразовательное  
учреждение городского округа Тольятти  
«Школа с углубленным изучением  
отдельных предметов № 41»

---

РАССМОТРЕНО

на заседании МО  
учителей математики и информатики  
Протокол № 5 от 17.05.2024

ПРИНЯТО

Педагогическим советом Школы  
Протокол № 41 от 31.05.2024

УТВЕРЖДЕНО

Приказ № 61-1 от 31.05.2024  
Директор МБУ «Школа № 41»  
Б.Э. Зимонина

**Дополнительная общеобразовательная  
общеразвивающая программа  
«Объектно-ориентированное программирование»**

**Направленность: техническая**

**Возраст обучающихся:** 15-18 лет (9, 10, 11 класс)

**Срок реализации:** 1 год (60 часов)

Тольятти, 2024

## **Программа «Объектно-ориентированное программирование»**

Направленность: техническая

Возраст: 15-18 лет (9/10/11 класс)

Срок реализации: 1 год

### **Аннотация**

Рабочая программа «Объектно-ориентированное программирование» составлена на основе авторской программы М.А. Дранкина. Программа направлена на развитие индивидуальности учащихся и высвобождение их творческого потенциала в процессе освоения средств информационных технологий.

### **Пояснительная записка**

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.

Умение выделять актуальные проблемы и решать их оптимальным способом средствами приложений является важным условием формирования информационной компетентности учащихся, выбравших «Объектно-ориентированное программирование». Программы-приложения, которые можно написать, являются конструктивными средствами для реализации творческих идей.

Программа «Объектно-ориентированное программирование» предназначена для учащихся старших классов.

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.

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

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

Основа программы – практическая и продуктивная направленность занятий, способствующая позитивному изменению индивидуально-личностных качеств учащихся. Одна из целей обучения работе в среде Lazarus заключается в предоставлении обучающимся возможности личного самовыражения и самореализации в программных продуктах, созданных в этой среде. Достижение цели становится возможным при создании лично значимой для обучающихся 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.

Рабочая программа разработана в соответствии с нормативными документами:

- Федеральный закон от 29.12.2012 г. №273-ФЗ «Об образовании в Российской Федерации»;
- Федеральный закон от 30.12.2020 г № 517-ФЗ «О внесении изменений в федеральный закон «Об образовании в Российской Федерации» и отдельные законодательные акты Российской Федерации»
- Концепция развития дополнительного образования в РФ (утверждена распоряжением Правительства РФ от 04. 09.2014 № 1726-Р);
- Приказ Министерства просвещения России от 9.11.2018 № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Постановление Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 года № 28 «Об утверждении СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организации воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Методические рекомендации по проектированию дополнительных общеразвивающих программ, направленных письмом Минобрнауки России от 18.11.2015 № 09-3242;
- «Методические рекомендации по разработке дополнительных общеобразовательных программ» (Приложение к письму министерства образования и науки Самарской области 03.09.2015 № МО -16-09-01/826-ТУ);
- «Методические рекомендации по проектированию разноуровневых дополнительных общеобразовательных общеразвивающих программ» (Приложение к письму министерства образования и науки Самарской области 24.06.2021 № МО/909-ТУ).

#### **Цели и задачи:**

- сформировать целостное представление о разработке программных приложений с помощью объектно-ориентированных подходов в программировании;
- реализовать способности 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.

#### **Образовательные результаты**

В процессе изучения курса «Объектно-ориентированное программирование» обучающиеся овладевают следующими знаниями, умениями и способами деятельности:

- знают инструментарий среды Lazarus, необходимый для разработки полнофункционального приложения;

- знают принципы работы основных компонентов Lazarus и умеют с ними работать;
- знают основы объектно-ориентированного подхода в программировании и умеют работать с некоторыми классами языка Free Pascal;
- умеют конструировать интерфейс приложений и разрабатывать их дизайн;
- умеют работать с графическими средствами среды;
- владеют приёмами организации и самоорганизации работы по созданию приложений;
- имеют положительный опыт коллективного сотрудничества при создании программ;
- овладевают процедурой самооценки знаний и деятельности и корректируют дальнейшую деятельность по созданию приложений.

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

#### **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**

### **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, OpenDialog;
- 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 PaintBox;
- 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 PaintBox. 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.**

**Проект**

## Тематическое планирование

№ п/п	Наименование разделов и тем	Всего часов	В том числе	
			Лекции	Практ. занятия
<b>Введение. Основные понятия</b>		<b>5</b>	<b>4</b>	<b>1</b>
1	Тема 1. Техника безопасности. Введение в программирование		1	
2	Тема 2. Алгоритмизация. Понятия ООП		2	
3	Тема 3. О программе Lazarus		1	
4	Практическая работа 1. Первый запуск программы			1
<b>Раздел 1. Компоненты и события</b>		<b>11</b>	<b>3</b>	<b>8</b>
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			1
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
<b>Раздел 2. Дизайн и обработка данных</b>		<b>12</b>	<b>4</b>	<b>8</b>
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		1	
19	Практическая работа 9. Обработка массива.			2
20	Итоговая работа 2 «Дизайн и обработка данных»			2
<b>Раздел 3. Интерфейс и работа с текстом</b>		<b>9</b>	<b>2</b>	<b>7</b>
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			1

28	Итоговая работа 3 «Интерфейс и работа с текстом»			2
<b>Раздел 4. Работа с графикой, решение задач</b>		<b>24</b>	<b>4</b>	<b>20</b>
29	Тема 12. Работа с графикой		2	
30	Практическая работа 15. Светофор			1
31	Практическая работа 16. Мишени			2
32	Итоговая работа 3 «Работа с графикой»			1
33	Тема 13. Системы счисления.		1	
34	Практическая работа 17. Решение задач.			1
35	Практическая работа 18. Решение задачи в программе.			2
36	Практическая работа 19. Решение математических задач программным способом..			3
37	Выполнение контрольного итогового проекта			6
<b>ВСЕГО</b>		<b>60</b>	<b>16</b>	<b>44</b>