

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Колледж радиозлектроники имени П.Н. Яблочкова



**Рабочая программа учебной дисциплины**

Объектно- ориентировочное программирование

09.02.07 Информационные системы и программирование

Профиль подготовки  
технологический  
Квалификация выпускника  
программист  
Форма обучения  
очная

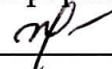
Саратов

2020

Разработчики: преподаватель И.З. Дервянченко 

Рассмотрено на заседании ЦК программирования, информатики и  
вычислительной техники

от «25» 05 2020 г. Протокол № 9

Председатель ЦК программирования, информатики и вычислительной  
техники  Е.Д.Шаманаева

Директор Колледжа  
радиоэлектроники  
имени П.Н. Яблочкова



О.В.Бреус

Заместитель директора по УР



Н.Н.Чернова

Рабочая программа учебной дисциплины разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование.

Организация- разработчик: ФГБОУ ВО «СГУ имени Н.Г. Чернышевского»  
Колледж радиоэлектроники имени П.Н. Яблочкова СГУ.

Разработчик: Дервянченко З.И. преподаватель Колледжа  
радиоэлектроники имени П.Н. Яблочкова СГУ.

## СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	Стр. 4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	5
3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	11
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	12

# 1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

## Объектно-ориентированное программирование

### 1.1. Область применения программы

Программа учебной дисциплины является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.07 Информационные системы и программирование.

1.2. Место дисциплины в структуре основной профессиональной образовательной программы: дисциплина входит в общепрофессиональные дисциплины общепрофессионального учебного цикла.

### 1.3. Цели и задачи дисциплины – требования к результатам освоения дисциплины:

В результате освоения дисциплины студенты должны уметь:

- разрабатывать алгоритмы для конкретных задач;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

В результате освоения дисциплины студенты должны знать:

- основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

ПК и ОК, которые актуализируются при изучении учебной дисциплины:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК.1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Выполнять требования нормативно-технической документации, иметь опыт оформления проектной документации.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

**1.4. Количество часов на освоение программы дисциплины:**

учебной нагрузки обучающегося 68 часов, в том числе:

учебной нагрузки обучающегося во взаимодействии с преподавателем 56 часов;

самостоятельной учебной работы обучающегося 4 часов.

промежуточная аттестация 8 часов

## 2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем в часах
Учебной нагрузки	68
Учебной нагрузки обучающегося во взаимодействии с преподавателем	56
в том числе:	
теоретическое обучение	20
лабораторные работы	26
Самостоятельная работа	4
Промежуточная аттестация в форме экзамена	8

## 2.2 Тематический план и содержание учебной дисциплины Объектно-ориентированное программирование

Наименование разделов и тем	Содержание учебного материала, лабораторные и практические занятия, самостоятельная учебная работа обучающихся, курсовая работа (проект)	Объем часов	Уровень освоения
1	2	3	4
Раздел I Визуальное программирование в среде Lazarus		38	
	Содержание учебного материала	8	
	<ol style="list-style-type: none"> <li>1. Требования к аппаратным и программным средствам интегрированной среды разработчика.</li> <li>2. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.</li> <li>3. Панель компонентов и их свойства. Окно кода проекта.</li> <li>4. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.</li> <li>5. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.</li> </ol>	4	1
Тема 1.1 Интегрированная среда разработчика Lazarus.			
	Лабораторные занятия	2	
	Лабораторное занятие №1 <i>Разработка простого приложения Lazarus.</i>		
	Самостоятельная работа	2	
	Тематика внеаудиторной самостоятельной работы (комплект) Основные понятия языка. Object Pascal		



<p>Тема 1.2 Визуальное событийно-управляемое программирование</p>	<p>Содержание учебного материала</p> <p>12</p> <ol style="list-style-type: none"> <li>1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.</li> <li>2. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектов через свойства.</li> <li>3. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.</li> </ol> <p>Лабораторные занятия</p> <p>8</p> <p>Лабораторное занятие №2</p> <p>Основные приемы визуального программирования. Разработка приложения «Калькулятор».</p> <p>Лабораторное занятие №3</p> <p>Использование компонентов для ввода-вывода многострочного текста.</p> <p>Лабораторное занятие №4</p> <p>Компоненты для отображения данных в виде таблицы в Lazarus. Класс TStringGrid</p> <p>Лабораторное занятие №5</p> <p>Использование компонентов диалогов TOpenDialog и TSaveDialog.</p> <p>Содержание</p> <p>4</p>	
<p>Тема 1.3 Обработка исключительных ситуаций</p>	<p>1. Виды ошибок. Синтаксические, логические и динамические ошибки.</p> <p>2. Виды обработок исключений. Глобальная и локальная обработка.</p> <p>3. Конструкции try ... finally и try...except. Анализ обработчиков исключений</p> <p>Лабораторные занятия</p> <p>2</p> <p>Лабораторное занятие №6</p> <p>Обработка исключительных ситуаций в Lazarus. Восприятие ввода от пользователя</p> <p>Содержание учебного материала</p> <p>8</p>	<p>2</p> <p>1</p> <p>2</p>
<p>Тема 1.4 Разработка оконного приложения</p>	<p>1. Разработка функционального интерфейса приложения. Создание интерфейса приложения.</p> <p>2. Разработка функциональной схемы работы приложения.</p> <p>3. Разработка приложения.</p> <p>Лабораторные занятия</p> <p>6</p>	<p>2</p> <p>1</p>

	Лабораторное занятие №7 Многооконные приложения. Модальные и немодальные окна.		
	Лабораторное занятие №8 Модули и классы		
	Содержание учебного материала	6	
Тема 1.5 Этапы разработки приложений	1. Разработка приложения.		
	2. Проектирование объектно-ориентированного приложения. 3. Создание интерфейса пользователя. 4. Тестирование, отладка приложения.	2	1
	Лабораторные занятия	2	
	Лабораторное занятие №9 Разработка Windows-приложения в среде Lazarus.		
	Лабораторное занятие №10 Создание и обработка меню		
	Самостоятельная работа	2	
	Тематика внеаудиторной самостоятельной работы (конспект)		
	1. Компиляция. Средства отладки на этапе компиляции. 2. Средства отладки на этапе выполнения. 3. Виды обработки ошибок на этапе выполнения.		
	Раздел 2. Основные концепции объектно-ориентированного программирования	24	
	Содержание учебного материала	4	
	1. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. 2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. 3. Определение класса, объекта.	2	1
Тема 2.1. Объект, класс	Самостоятельная работа	2	
	Тематика внеаудиторной самостоятельной работы (конспект) Написать иерархию объектов класса «Точка» - геометрические фигуры: «Окружность», «Квадрат», «Ромб»		
Тема 2.2. Методы, свойства,	Содержание учебного материала	8	

полю.	<ol style="list-style-type: none"> <li>1. Методы, свойства, поля.</li> <li>2. Синтаксис и программирование свойств.</li> <li>3. Конструкторы, деструкторы.</li> <li>4. Области видимости.</li> </ol>	4	1
<p>Лабораторные занятия</p> <p>Лабораторное занятие №11 Объекты Иерархия объектов</p> <p>Содержание учебного материала</p> <p>Тема 2.3. События. Виды и способы управления проектом и объектами</p>	<ol style="list-style-type: none"> <li>1. Событийно-управляемая модель программирования. Компонентно-ориентированный подход.</li> <li>2. Взаимодействие программы с ОС.</li> <li>3. Управление проектом.</li> <li>4. Управление объектом.</li> </ol> <p>Промежуточная аттестация</p> <p>Итого:</p>	10	10
		8	
		68	

Для характеристики уровня освоения учебного материала используются следующие обозначения:

1. – ознакомительный (узнавание ранее изученных объектов, свойств);
2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)
3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

### 3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

#### 3.1. Материально - техническое обеспечение

Реализация программы дисциплины требует наличия учебного кабинета информатики.

- рабочее место преподавателя;
- посадочные места обучающихся (по количеству обучающихся);
- учебные наглядные пособия (таблицы, плакаты);
- тематические папки дидактических материалов;
- комплект учебно-методической документации;
- комплект учебников (учебных пособий) по количеству обучающихся.

Технические средства обучения:

- компьютер, мультимедиа комплекс, интерактивная доска.

Оборудование учебной лаборатории (по учебному плану):

- рабочие станции для работы обучающихся;
- рабочее место преподавателя;
- комплект учебно-методической документации.

Технические средства обучения:

- компьютер, мультимедиа комплекс, интерактивная доска.

#### 3.2. Информационное обеспечение обучения

Перечень учебных изданий, Интернет-ресурсов, дополнительной литературы

##### Основные источники:

1. О. Л. Голицына, И. И. Попов, Т. Л. Партыка. Языки программирования: Учебное пособие для ССУЗов:- 3, перераб. и доп. - Москва: Издательство "ФОРУМ"; Москва: Издательский Дом "ИНФРА-М", 2015. - 400 с.
2. С.Р. Гуриков. Программирование в среде Lazarus для школьников и студентов. Учебное пособие Высшее образование Форум, Инфра-М, 2017. – 336 с.
3. Т.И. Немцова, С.Ю. Голова, И.В. Абрамова. - Москва: Издательский Дом "ФОРУМ"; Москва: ООО "Научно-издательский центр ИНФРА-М", 2013. - 496 с.
4. Г.М. Эйдлина, К.А. Милорадов. Delphi: Программирование в примерах и задачах. Практикум: Учебное пособие - М.: ИЦ РИОР: НИЦ Инфра-М, 2015 - 116 с

##### Дополнительные источники:

1. В.Д. Колдаев. Основы алгоритмизации и программирования: Учебное пособие. Глава 4. ООП / - Москва: Издательский Дом "ФОРУМ"; Москва: Издательский Дом "ИНФРА-М", 2014. - 416 с.
2. Т.И. Немцова; Под ред. Л.Г. Гагариной. Программирование на языке высокого уровня. Программирование на языке Object Pascal: Учебное пособие / - М.: ФОРУМ: ИНФРА-М, 2015. - 496 с.

##### Интернет-ресурсы:

1. А. Н. Вальвачев, К. А. Сурков, Д. А. Сурков, Ю. М. Четырько. Объектно-ориентированное программирование на языках Delphi и C++: учебное пособие для студентов [Электронный ресурс] – БГУИР, 2016 . – 432 с Режим доступа: <http://libeldoc.bsuir.by/handle/123456789/850>
2. Ю. И. Кудинов, Ф. Ф. Пашенко, А. Ю. Келина. - Практикум по основам современной информатики [Электронный ресурс] / Москва: Лань, 2014. - 352 с.: Режим доступа: <http://e.lanbook.com/books/element.php>

3. Соколова Ю. С. Разработка приложений в среде Delphi. Компоненты и их использование. Часть 1 / - 2013. - 142 с. [Электронный ресурс]: учебное пособие. - Москва: Горячая линия – Телеком Режим доступа: Книга находится в базовой версии ЭБС IPRbooks
4. Мансуров К.Т. Основы программирования в среде Lazarus Электронная книга, 2014. — 772 с. — ISBN 978-9967-03-646-8

#### 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий и лабораторных занятий, тестирования, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Результаты обучения (освоенные умения, усвоенные знания)	Основные показатели оценки результата
<p>разрабатывать алгоритмы для конкретных задач;</p> <p>определять сложность работы алгоритмов;</p> <p>работать в среде программирования;</p> <p>реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;</p> <p>оформлять код программы в соответствии со стандартом кодирования;</p> <p>выполнять проверку, отладку кода программы.</p> <p>основные алгоритмические конструкции;</p> <p>эволюцию языков программирования, их классификацию, понятие системы программирования;</p> <p>основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;</p> <p>подпрограммы, составление библиотек подпрограмм;</p> <p>объектно-ориентированную модель программирования;</p> <p>основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.</p>	<p>понимание алгоритмов для конкретных задач;</p> <p>анализ сложности работы алгоритмов;</p> <p>работать в среде программирования;</p> <p>реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;</p> <p>владение методикой оформления кода программы в соответствии со стандартом кодирования;</p> <p>выполнять проверку, отладку кода программы.</p> <p>понимание основных алгоритмических конструкции;</p> <p>эволюцию языков программирования, их классификацию, понятие системы программирования;</p> <p>знать элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;</p> <p>подпрограммы, составление библиотек подпрограмм;</p> <p>понимание объектно-ориентированной модели программирования;</p> <p>основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.</p>