


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

Колледж радиоэлектроники имени П.Н. Яблочкова

УТВЕРЖДАЮ

« 15 » июля 2020 г.

Рабочая программа учебной дисциплины

Адаптационное объектно-ориентированное программирование

09.02.07 Информационные системы и программирование

Профиль подготовки
технический
Квалификация выпускника
программист
Форма обучения
очная

Саратов
2020

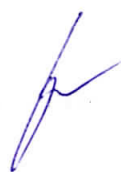
Разработчики: преподаватель И.З. Дервянченко

Рассмотрено на заседании ЦК программирования, информатики и
вычислительной техники

от «15» 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

Адаптационное объектно-ориентированное программирование

1.1. Область применения программы

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

1.2. Место дисциплины в структуре программы подготовки специалистов среднего звена: дисциплина относится к общепрофессиональному учебному циклу.

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

В результате освоения дисциплины обучающийся должен уметь:

- общие принципы построения алгоритмов, 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.

В результате освоения дисциплины обучающийся должен знать:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

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 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.

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 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

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 часа

промежуточная аттестация 8 часов.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объём 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, внеаудиторная (самостоятельная) учебная работа обучающихся, курсовая работа (проект).	Объем часов	Уровень освоения
1	2	3	4
Введение. Роль дисциплины в становлении специалистов.	Содержание 1. Роль дисциплины в становлении специалистов. 2. Взаимосвязь с другими дисциплинами.	2	1
Раздел 1. Стандартная библиотека классов STL C++		14	
Тема 1.1 Контейнерные классы C++ библиотеки STL	Содержание 1. Контейнерные классы в C++. 2. Библиотека STL. Преимущества STL. 3. Основные элементы стандартной библиотеки STL: контейнеры, итераторы, алгоритмы и распределители памяти.	2	1
Тема 1.2 Последовательные контейнеры	Содержание 1. Последовательные контейнеры. 2. Вектор, стек, очередь, дек, очередь с приоритетом, список. 3. Назначение и основные функции контейнеров. Лабораторная работа №1 Программирование с использованием последовательных контейнеров 1. Разработка блок-схемы решения задачи 2. Разработка и отладка программы	4 2	
Тема 1.3 Ассоциативные контейнеры	Содержание 1. Ассоциативные контейнеры. 2. Словари, словари с дубликатами, множества, множества с дубликатами, битовые множества. 3. Назначение и основные функции контейнеров. 1. Прочие классы стандартной библиотеки STL 2. Класс pair (пара) стандартной библиотеки C++ 3. Библиотека алгоритмов	8 2	2 1

	<p>Лабораторная работа №2 Программирование с использованием ассоциативных контейнеров. 1. Разработка блок-схемы решения задачи 2. Разработка и отладка программы.</p> <p>Тематика внеаудиторной самостоятельной работы: Выполнение практической работы по теме «Программирование с использованием библиотеки алгоритмов»</p>	2	2
<p>Раздел 2. Реализация объектно-ориентированного программирования в языке С++.</p>		44	
<p>Тема 2.1 Основные принципы ООП</p>	<p>Содержание</p> <ol style="list-style-type: none"> 1. Понятие объекта и класса. 2. Терминология ООП. Принципы ООП. 3. Понятие инкапсуляции, наследования, абстрагирования и полиморфизма. 4. Преимущества и недостатки ООП. 	2	1
	<p>Содержание</p> <ol style="list-style-type: none"> 1. Объявление класса. 2. Уровни доступа членов и методов класса. 3. Модификаторы. 4. Статические элементы класса: статические поля и статические методы.. 	10	
<p>Тема 2.1. Реализация классов в С++</p>	<ol style="list-style-type: none"> 1. Конструкторы. 2. Основные правила описания конструкторов. 3. Виды конструкторов. 	2	1
	<ol style="list-style-type: none"> 1. Деструкторы. 2. Основные правила описания деструкторов. <p>Лабораторная работа №3 Программирование с использованием классов. Создание и работа с классами. 3. Разработка блок-схемы решения задачи 4. Разработка и отладка программы</p>	2	1
		2	2

	<p>Лабораторная работа №4 Программирование с использованием классов. Перегрузка операций.</p> <ol style="list-style-type: none"> 1. Разработка блок-схемы решения задачи 2. Разработка и отладка программы 	2	2
	<p>Содержание</p> <ol style="list-style-type: none"> 1. Простое наследование. 2. Виртуальные методы. 3. Множественное наследование. 	6	1
Тема 2.2 Наследование	<p>Лабораторная работа №5 Программирование с использованием простого наследования.</p> <ol style="list-style-type: none"> 1. Разработка блок-схемы решения задачи 2. Разработка и отладка программы 	4	2
	<p>Содержание</p> <ol style="list-style-type: none"> 4. Дружественные функции. 5. Дружественные классы. 6. Дружественные методы. 	6	1
Тема 2.3 Дружественные функции и классы	<p>Лабораторная работа №6 Программирование с использованием дружественных классов и методов.</p> <ol style="list-style-type: none"> 1. Разработка блок-схемы решения задачи 2. Разработка и отладка программы 	4	2
	<p>Содержание</p> <ol style="list-style-type: none"> 1. Назначение и элементы диаграммы классов. 2. Видимость и область действия. 3. Взаимосвязи объектов классов. Мощность отношений. 	4	1
Тема 2.4	<p>Лабораторная работа №7 Построение UML-диаграммы решения задачи с помощью средств Visual Studio</p> <ol style="list-style-type: none"> 1. Разработка UML-диаграммы решения задачи 2. Разработка и отладка программы 	2	2
	<p>Содержание</p> <ol style="list-style-type: none"> 4. Создание и использование шаблона класса. 5. Специализация шаблона класса. 6. Достоинства и недостатки шаблонов. 	8	1
Тема 2.5 Шаблоны классов		2	1

	Лабораторная работа №8 Программирование с использованием шаблона класса. 3. Разработка блок-схемы решения задачи 4. Разработка и отладка программы.	4	2
	Тематика внеаудиторной самостоятельной работы: Написание реферата по теме «Примеры использования шаблонов классов в программировании»	2	3
	Содержание	8	
Тема 2.6 Обработка исключительных событий.	1. Обработка исключительных ситуаций. 2. Механизм обработки исключений.	2	1
	1. Синтаксис и перехват исключений. 2. Иерархия исключений. 3. Методы генерации исключительных ситуаций.	2	1
	Лабораторная работа №9 Обработка исключительных событий. 3. Разработка блок-схемы решения задачи 4. Разработка и отладка программы	4	2
	Промежуточная аттестация в виде экзамена	8	3
	Итого:	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://libelddoc.bsuir.by/handle/123456789/850>
2. Ю. И. Кудинов, Ф. Ф. Пащенко, 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 / - 2018. - 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; 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. 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.</p>	<p>понимание алгоритмов для конкретных задач; анализ сложности работы алгоритмов; работать в среде программирования; реализовывать построенные алгоритмы в виде программ на конкретном языке программирования; владение методикой оформления кода программы в соответствии со стандартом кодирования; выполнять проверку, отладку кода программы. понимание основных алгоритмических конструкции; эволюцию языков программирования, их классификацию, понятие системы программирования; знать элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти; подпрограммы, составление библиотек подпрограмм; понимание объектно-ориентированной модели программирования; основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.</p>