

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

«Адыгейский государственный университет»



«Утверждаю»

Алиева М.Ф.

« 30 » июня 20 20 г.

РАБОЧАЯ ПРОГРАММА

2.9 Основы программирования

Дополнительная профессиональная программа

Педагогическое образование по программе «Физика» и «Информатика»

Программа профессиональной переподготовки

Составитель: О.А.Плисенко О.А.Плисенко

Рабочая программа обсуждена и утверждена на заседании кафедры
теоретической физики протокол № 9 от 30 июня 2020г.

Заведующий кафедрой В.Б. Тлячев В.Б. Тлячев

Содержание

1.	Планируемые результаты обучения дисциплины (цели и задачи освоения дисциплины и результаты обучения)	3
2.	Учебно-тематический план освоения дисциплины	5
2.1.	Содержание разделов дисциплины	5
2.2.	Распределение видов учебной работы по разделам дисциплины	7
3.	Учебно-методическое обеспечение дисциплины	8
3.1.	Основная и дополнительная литература	8
3.2.	Периодические издания	9
3.3.	Ресурсы сети Интернет	9
4.	Фонд оценочных средств	10

Пояснительная записка

Рабочая программа дисциплины составлена в соответствии с установленными требованиями по дополнительной профессиональной образовательной программе Педагогическое образование по программе «Физика» и «Информатика».

Рабочая программа представляет собой совокупность дидактических материалов, направленных на реализацию содержательных, методических и организационных условий подготовки слушателей.

Рабочая программа ориентирована на реализацию компетентного подхода в обучении.

Дисциплина относится к вариативной части учебного плана дополнительной профессиональной образовательной программы Педагогическое образование по программе «Физика» и «Информатика».

Трудоемкость дисциплины: 3 зачетных единиц (108 часов).

Ключевые слова: программирование, языки программирования, алгоритм, тип данных, вычислительный процесс, функциональное программирование, объектно-ориентированное программирование, программа, массивы, структуры.

Составитель: Плисенко О.А., ст. преподаватель кафедры Автоматизированных систем обработки информации и управления.

1. Планируемые результаты обучения дисциплины (цели и задачи освоения дисциплины и результаты обучения)

Цели преподавания дисциплины:

- обеспечить усвоение слушателями понятийного аппарата дисциплины;
- сформировать у слушателей современные представления о различных парадигмах программирования, таких как структурное программирование, объектно-ориентированное программирование;
- сформировать у слушателей представления об основных алгоритмических структурах;
- обеспечить освоение навыков программирования на языке C/C++.

Задачи изучения дисциплины:

- познакомить с методами и средствами обработки информации, информационными технологиями;
- познакомить слушателей с современными технологиями программирования, языками программирования;
- раскрыть содержание фундаментальных понятий алгоритмизации и программирования, основ современной технологии программирования;
- сформировать у слушателей представление о типовых алгоритмах обработки информации;
- дать представление о составлении программ на современных алгоритмических языках, об отладке и тестировании программ;
- познакомить с алгоритмами решения задач, возникающих в профессиональной деятельности

Цели и задачи дисциплины в терминах компетенций

Выпускник должен обладать следующими профессиональными компетенциями (ПК):

педагогическая деятельность:

- готовностью реализовывать образовательные программы по учебным предметам в соответствии с требованиями образовательных стандартов (ПК-1);
- готовностью применять современные методики и информационные технологии для обеспечения качества учебно-воспитательного процесса (ПК-2).

Показателями компетенций являются:

знания – этапы решения задачи на компьютере; типы данных; базовые конструкции языков программирования; методы и средства сбора, хранения, передачи, накопления информации; правила алгоритмизации и программирования различных прикладных задач; принципы структурного и модульного программирования; принципы объектно-ориентированного программирования

умения - описывать структуры данных, работать в средах программирования, реализовывать алгоритмы на языке программирования высокого уровня, реализовывать базовые конструкции языка программирования, реализовывать методы анализа и обработки данных, применять приемы объектно-ориентированного подхода для решения прикладных задач.

навыки – использования методов и технологий разработки алгоритмов, работы в различных средах программирования; описания структур данных и других базовых представлений данных, описания объектов и классов, программирования на языке высокого уровня.

Конечные требования к овладению дисциплиной «Основы программирования»

В результате изучения дисциплины «Основы программирования» слушатель должен:

- владеть понятийным аппаратом дисциплины;
- знать различные парадигмы программирования, такие как структурное программирование, объектно-ориентированное программирование;
- владеть методами и средствами обработки информации, информационными технологиями;
- уметь работать в средах программирования;
- выработать навыки описания структур данных, реализации алгоритмов на языке программирования высокого уровня, программирования базовых конструкций языка программирования;
- иметь навыки использования методов и технологий разработки алгоритмов;

2. Учебно-тематический план освоения дисциплины.

2.1. Содержание разделов дисциплины

Таблица 1.

Номер раздела	Наименование разделов	Содержание раздела	Форма текущего контроля
1	Основы языков программирования и технологии разработки программ	Язык программирования, общая характеристика, назначение. Классификация языков программирования. Понятия оператора, алгоритма, выражения. Основные операции языка. Правила записи выражений и операторов. Структура программы. Функции и задачи компилятора. Компиляция и интерпретация: основные этапы компиляции. Простые типы данных: целый, вещественный, символьный, логический и текстовый. Работа с переменными и константами. Организация ввода и вывода данных	ДЗ
2	Принципы структурного программирования. Базовые алгоритмы.	Разветвляющиеся программы. Условные операторы. Операции сравнения. Логические операции. Составные операторы. Оператор множественного ветвления. Циклы. Основные параметры цикла. Циклы со счетчиком. Циклы с предусловием. Циклы с постусловием. Вложенные циклы. Понятие массива, элемента и его индекса. Одномерные массивы. Алгоритмы работы с массивами: ввод и вывод элементов массивов, нахождение характеристик массивов. Сортировка массива. Матрицы (двумерные массивы). Выполнение поиска данных в массиве. Алгоритмы сортировки данных. Основные конструкции программы в структурном программировании. Оформление блоков кода в виде подпрограмм. Функции. Объявление и вызов функций. Параметры функции и возвращаемое функцией	ДЗ

		<p>значение. Локальные и глобальные переменные. Правила работы с функциями.</p> <p>Вычисление по формулам. Индуктивная обработка последовательностей данных. Рекуррентные вычисления. Понятие файла и файловой системы. Характеристики и типы файлов. Организация доступа к файлам. Виды работы с файлом. Функции работы с файлами и папками.</p>	
3	Основы объектно-ориентированного программирования.	<p>Идеология объектно-ориентированного программирования. Основные понятия. Инкапсуляция. Свойства. Классификация свойств. Методы класса. Объявление объектов. Работа с классами. Наследование. Правила наследования различных методов. Перекрытие методов. Полиморфизм. Перегрузка функций. Виртуальные функции. Правила описания и использования виртуальных функций. Создание собственных библиотек (модулей). Использование шаблонов функций и классов (компонент) для описания собственных библиотек классов. Структура среды разработки. Создание проекта. Компоновка проекта и настройка свойств компонентов. Компиляция и запуск программы. Обработчики событий и их параметры.</p>	ДЗ
4	Зачет		

Содержание лабораторных занятий

Основы языков программирования

Лабораторная работа № 1. Знакомство с интерфейсом и инструментами среды программирования. Создание линейной программы на языке программирования C/C++. Работа с простыми типами данных: целым, вещественным, символьным и логическим.

Лабораторная работа № 2. Создание разветвляющейся программы с помощью условных

операторов. Использование простых и составных условий.

Принципы структурного программирования.

Лабораторная работа № 3. Работа с различными формами циклических операторов: циклами со счетчиком, циклами с предусловием и с постусловием. Реализация алгоритмов с применением циклов со счетчиком и циклов с условиями.

Лабораторная работа № 4. Объявление одномерных массивов на основе простых типов данных. Ввод и вывод данных массивов. Нахождение характеристик массивов. Выполнение поиска заданных значений массива. Реализация сортировки элементов массива.

Лабораторная работа № 5. Изучение правил объявления и вызова функции в программе. Составление программ с использованием функций. Реализация концепций структурного программирования с помощью функций.

Основы объектно-ориентированного программирования

Лабораторная работа № 6. Изучение синтаксиса описания класса. Изучение правил описания и вызова конструкторов, деструкторов и других методов класса. Объявление объектов и работа с ними. Создание программы, выполняющей работу с классами и объектами. Использование объектно-ориентированного подхода для описания объектов в задачах моделирования предметных областей.

2.2. Распределение видов учебной работы по разделам дисциплины

Номер раздела	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			СР
			Л	ПЗ	ЛР	
1	Основы языков программирования и технологии разработки программ	18	2	0	2	14
2	Принципы структурного программирования. Базовые алгоритмы	54	6	0	6	42
3	Основы объектно-ориентированного программирования	36	4	0	4	28
4	Итого	108	12	0	12	84

Самостоятельная работа слушателей

Цели самостоятельной работы – освоить те разделы дисциплины, которые не были затронуты в процессе очных занятий, но предусмотрены рабочей программой, а также расширить границы получаемых знаний, умений и навыков в процессе дополнительного изучения отдельных тем, разработки алгоритмов и программ на языке программирования C/C++, подготовиться к лекционным и практическим занятиям.

Виды самостоятельной работы:

- выполнение домашних практических работ (составление алгоритмов, описание структур данных, классов и т.д.);
- изучение отдельных тем, вопросов, их конспектирование;
- подготовка к лекционным и практическим занятиям;
- подготовка к модульно-зачетным мероприятиям и экзамену;
- другие виды самостоятельной работы студентов.

3. Учебно-методическое обеспечение дисциплины

3.1. Основная и дополнительная литература

Основная

№ п/п	Наименование, библиографическое описание	Наличие грифа
1	Основы программирования: учебное пособие. /В.П. Хиценко. – Новосибирск: Изд-во НГТУ, 2015. -83с [Электронный ресурс] ЭБС: Режим доступа: http://biblioclub.ru/index.php?page=book_view_red&book_id=438365	
2	Лубашева, Т. В. Основы алгоритмизации и программирования: учеб. Пособие/ Т.В. Лубашева, Б.А. Жлезко. – Минск: РИПО, 2016. -378. [Электронный ресурс] ЭБС: Режим доступа: http://biblioclub.ru/index.php?page=book_view_red&book_id=463632	
3	Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Горячая линия-Телеком, 2012. ISBN: 978-5-9912-7001-4 [Электронный ресурс]	Гриф

Дополнительная

1. Аверскин, В. П. Программирование на C++ / В. П. Аверскин, А. Д. Хомоненко, А. И. Бобровский. – М.: Мир, 2001. – 256 с.
2. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Г. Буч. – СПб.: Невский диалект, 2000. – 359 с.
3. Вирт, Н. Алгоритмы + структуры данных = программы / Н. Вирт. – М.: Мир, 1989. – 360 с.

4. Дейтел, Х. Как программировать на языке С++ / Х. Дейтел, П. Дейтел. – М.: Бином, 2005. – 1244 с.
5. Керниган, Б. Язык программирования С / Б. Керниган, Д. Ритчи. – СПб.: Невский проспект, 2006. – 289 с.
6. Кнут, Д. Э. Искусство программирования: т.1-3. / Д. Кнут. – М.: Мир, 1976-1978.
7. Лафоре Р. Объектно-ориентированное программирование в С++ / Р. Лафоре. – СПб.: Питер, 2004. – 923 с.
8. Мейерс, С. Эффективное использование С++ / С. Мейерс. – М.: ДМК, 2000. – 240 с.
9. Павловская, Т.А. С/С++. Структурное программирование. Практикум / Т. А. Павловская, Ю. А. Щипак. – СПб.: Питер, 2002. – 240 с.
10. Страуструп, Б. Дизайн и эволюция языка С++ / Б. Страуструп. – М.: ДМК Пресс, 2006. – 444 с.
11. Шилдт, Г. Теория и практика С++ / Г. Шилдт. – СПб.: BHV, 1996. – 416 с.в.

3.2. Периодические издания

Периодические издания

1. **"Информатика и образование"** <http://infojournal.ru/> Научно-методический журнал по методике преподавания информатики и информатизации образования. Выходит 10 раз в год. Учредители журнала: Российская академия образования, издательство «Образование и информатика». Целевая аудитория журнала: учителя и преподаватели информатики; работники органов управления образованием, отвечающие за информатизацию образования; специалисты, отвечающие за информатизацию образовательных учреждений.
2. **"Информатика в школе"** <http://infojournal.ru/> Научно-методический журнал по методике преподавания информатики и информатизации образования. Выходит 10 раз в год. Учредители журнала: Российская академия образования, издательство «Образование и информатика». Целевая аудитория журнала: учителя и преподаватели информатики; методисты по информатике; студенты педагогических вузов и колледжей — будущие учителя информатики; специалисты, отвечающие за информатизацию образовательных учреждений.
3. **«Программирование»** <http://www.ispras.ru/programming/> Журнал основан в 1975 году и публикует статьи по всем проблемам, связанным с теоретическим и практическим программированием: операционные системы, технологии программирования, языки программирования и компиляторы, параллельное программирование, верификация и тестирование программ, машинная графика, компьютерная алгебра и т.п. Журнал предназначен для исследователей, практиков и студентов. Периодичность выпуска – 6 номеров в год. Журнал внесен в список ВАК, выпускается ФГУП «Издательство «Наука».

3.3. Ресурсы сети Интернет

1. Сайт Национального открытого университета ИНТУИТ [Электронный ресурс] : – Режим доступа: <http://www.intuit.ru/>
2. Сайт Учебно-методический комплект по информатике и ИКТ Натальи Владимировны Макаровой [Электронный ресурс] : – Режим доступа: <http://makarova.piter.com/>

3. Сайт Евразийского открытого института [Электронный ресурс] : – Режим доступа:
<http://www.eoi.ru/>
4. Компьютерные видеокурсы. Обучающие видео и видеоуроки [Электронный ресурс] : – Режим доступа: <http://www.teachvideo.ru/>
5. Образовательный сайт [Электронный ресурс]: – Режим доступа:
<http://www.exponenta.ru/>
6. Портал «Программирование для всех» [Электронный ресурс] : - Режим доступа:
<http://realcoding.net/>

4. Фонд оценочных средств

Перечень видов оценочных средств используемых для формирования ФОС по дисциплине «Основы программирования»:

- контрольные работы по теоретическому материалу (компьютерные тесты);
- лабораторный практикум по каждой теме курса.

Зачет по дисциплине выставляется при выполнении всех лабораторных работ по курсу и прохождении тестов по каждому теоретическому разделу не ниже 65% в каждом.

Вопросы теста

1. Укажите неправильно записанную операцию отношения

все операторы записаны правильно

>=

=! +

<=

2. Каков будет результат выражения **!(1 && !(0 || 1))?**

False

неоднозначность

True +

запись некорректна

3. Какие преобразования типов данных не возможны без потери данных?

float to int +

char to float

int to float

все перечисленные преобразования не возможны

4. Оператор вывода **cout** может печатать несколько значений или переменных в одной команде, используя следующий синтаксис:

```
cout << ("Привет" & name & "n");
```

```
cout << "Привет", name, "n";
```

```
cout << "Привет" + name + "n";
```

```
cout << "Привет" << name << "n"; +
```

5. Какое ключевое слово указывает, что целая переменная не может принимать отрицательные значения?

нет такого зарезервированного слова

positive

long

unsigned +

6. Какой заголовочный файл следует подключить, чтобы можно было пользоваться приведением типов данных?

cmath
cctype
iostream
Никакого +

7. Какое из следующих значений эквивалентно зарезервированному слову **true**?

-1
0.1
1
Все варианты ответов +

8. Какое из следующих значений эквивалентно зарезервированному слову **false**?

-1
0 +
1
Нет правильного ответа

9. Значение **5.9875e17** может быть сохранено в переменной, типа:

int
short
bool
float +

10. Ввод данных в C++

```
cin >> выражение1 >> выражение2 >> endl >> ...;  
cin >> выражение1 >> выражение2...;  
cin >> выражение1 >> выражение2...; +  
cin << выражение1 << выражение2...;
```

11. Результат выполнения следующего фрагмента кода: **!(1 || 0) && 0)**

результат не может быть заранее определен
1 +
0
запись не корректна

12. Укажите правильное приведение типа данных!

a(char);
char:a;
to(char, a);
(char)a; +

13. Вывод данных в C++

```
cout << переменная, выражение, endl;  
cout << переменная << выражение << endl; +  
cout << переменная, << выражение, << endl;  
cout >> переменная >> выражение >> endl;
```

14. Какой из следующих логических операторов - логический оператор И?

&
|

&& +
|&

15. Какой из следующих логических операторов - логический оператор ИЛИ?

|| +
!
|
&&

16. Какой из следующих логических операторов - логическое отрицание?

||
! +
|
&&

17. Какой из перечисленных типов данных не является типом данных в C++?

float
double
int
real +

18. Какими знаками заканчивается большинство строк кода в Си++?

, (запятая)
.(точка)
; (точка с запятой) +
: (двоеточие)

19. Выберите правильный вариант объявления константной переменной в C++, где **type** - тип данных в C++ **variable** - имя переменной **value** - константное значение

const type variable = value; +
const variable = value;
const type variable := value;
в C++ нет констант

20. Какой из следующих операторов - оператор сравнения двух переменных?

=
== +
equal
:=

21. Язык программирования C++ разработал

Бьерн Страуструп +
Кен Томпсон
Дональд Кнут
Никлаус Вирт

22. Какую функцию должны содержать все программы на C++?

main() +
system()
program()
start()

23. Чтобы подключить заголовочный файл **iostream** в программу на C++, необходимо написать:

```
include <iostream,h;
#include <> с iostream внутри скобок +
include (iostreamh)
#include <>; с iostream.h внутри скобок
```

24. Какие служебные символы в C++ используются для обозначения начала и конца блока кода?

```
{ } +
()
<>
begin end
```

25. Какая из следующих записей - правильный комментарий в C++?

```
*/ Комментарий */
{комментарий}
** Комментарий **
/* комментарий */ +
```

26. Укажите объектно-ориентированный язык программирования

```
C
Все варианты ответов
Pascal
C++ +
```

27. Простые типы данных в C++.

```
целые – int, вещественные – float или double, символьные – string
целые – int, вещественные – float или real, символьные – char
целые – bool, вещественные – float или double, символьные – string
целые – int, вещественные – float или double, символьные – char +
```

28. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на языке ЭВМ, называется:

```
исходный код
транслятор +
сканер
текстовый редактор
```

29. Зачем нужен тип данных?

```
Для обеспечения целостности данных
Для создания переменных с неизвестным типом
Для указания переменной типа ее содержимого +
В языке C++ тип данных не нужен
```

30. Что такое функция?

Некоторая часть программы имеющее собственное имя и которое может вызываться столько раз, сколько это нужно +

Некоторая часть программы содержащая вредоносный код, и блокирует определенные действия системы

Некоторая часть программы в которой происходит начальная инициализация всех полей структур, массивов, переменных и т.д.

Что-то из математики и в программировании не используется

31. Как написать следующее выражение "Переменной a присвоено значение b"?

a==b;
a=b; +
b=a;
a+=b;

32. Допустимо ли следующее выражение `int name_student = ivan;` ?

Да +
Нет
Не всегда, но возможно
Все зависит от обстоятельств

33. Выберите правильное определение алгоритма:

Алгоритм - это последовательность действий, направленных на решение задачи за конечное число шагов +

Алгоритм - это последовательность действий, направленных на решение задачи за бесконечное число шагов

Алгоритм - это хаотичная последовательность действий

Алгоритм - это последовательность задач, которые надо решить за определенный промежуток времени

34. Укажите способы представления алгоритма:

блок-схема
псевдокод
словесное описание
все перечисленные +

35. `#include` - это

указание препроцессору, что надо подключить библиотеку +
указание процессору, как надо выполнять программу
указание пользователю, как надо запускать программу
указание операционной системе, что делать с программой

36. Какая команда завершает выполнение функции и возвращает результат:

begin
end
return +
exit

37. В какой библиотеке хранится описание команд ввода/вывода `cin` и `cout`

iostream +
conio.h
math.h
Нет верного ответа

38. Какая из перечисленных программ является транслятором:

интерпретатор +
текстовый редактор

графический редактор
редактор видео файлов

39. Какая из перечисленных программ является транслятором:

компилятор +
текстовый редактор
графический редактор
программа просмотра видео файлов

40. Какая программа создает объектный файл:

компилятор +
интерпретатор
архиватор
компоновщик

41. Какими знаками ограничивается символьный тип данных:

одинарными кавычки +
двойные кавычки
круглые скобки
фигурные скобки

42. Какими знаками ограничивается строка:

одинарными кавычки
двойные кавычки +
круглые скобки
фигурные скобки

43. Каким словом в C++ обозначается логический тип данных:

bool +
boolean
logic
logical

44. Какими символами задается комментарий одной строки или ее части:

// +
/*
*/
\\

45. Чем ограничивается область видимости локальной переменной:

функцией, в которой переменная объявлена +
главной функцией main()
программой, содержащей локальную переменную
локальная переменная не имеет ограничений

46. Операция инкремента - это:

увеличение на 1 +
уменьшение на 1

присвоение
в C++ нет такой операции

47. Операция декремента - это:

увеличение на 1
уменьшение на 1 +
присвоение
в C++ нет такой операции

48. Как в C++ обозначается операция увеличения на 1:

++ +
--
+=
+

49. Как в C++ обозначается операция уменьшения на 1:

++
-- +
+=
+

50. В каких случаях операция деления является целочисленным:

при делении целого типа на целый +
при делении вещественного типа на целый
при делении вещественного типа на вещественный
при делении целого типа на вещественный

51. Как в C++ обозначается операция остатка от деления:

% +
&
mod
div

52. Какие значения примут переменные X и Y в результате выполнения действий: X = 5; Y = ++X

X = 5 Y = 5
X = 6 Y = 6 +
X = 5 Y = 6
X = 6 Y = 5

53. Какие значения примут переменные X и Y в результате выполнения действий: X = 5; Y = X++

X = 5 Y = 5
X = 6 Y = 6
X = 5 Y = 6
X = 6 Y = 5 +

54. К каким типам данных может быть применена операция вычисления остатка от деления:

целочисленным +
вещественным
строковым
любым числовым

55. Что обозначает выражение: **$x*=y$**

$x = x * y;$ +
 $y = x * y;$
 $x = y * y;$
 $y = x * x;$

56. Какие значения примут переменные Z и Y в результате выполнения целочисленного деления: **$X = 5; Y = X / 2; Z = (\text{float}) X / 2;$**

$Y = 2 \ Z = 2.5$ +
 $Y = 1 \ Z = 2.5$
 $Y = 2.5 \ Z = 2.5$
 $Y = 2.5 \ Z = 2$

57. Какая функция в C++ вычисляет корень квадратный:

$\text{sqr}()$
 $\text{sqrt}()$ +
 $\text{abs}()$
 $\text{fabs}()$

58. Как в C++ обозначается "НЕ РАВНО":

$<>$
 $!=$ +
 $=$
 $==$

59. Что обозначает выражение: **$x-=y$**

$x = x - y;$ +
 $y = x - y;$
 $x = y - y;$
 $y = x - x;$

60. Если переменная задается перед всеми функциями в программе, то она называется:

локальной
глобальной +
произвольной
главной

Лист регистрационных изменений

Номер изменения	Номера листов			Основание для внесения изменения	Подпись	Расшифровка подписи	Дата
	замененн ых	новых	аннулирова нных				