

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Учебный центр по ДПО ИМИТ

УТВЕРЖДАЮ

Проректор по ВРДО

Е.Н. Гончарова

20 21 г.




ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

Основы программирования на С

Программа рассмотрена и одобрена на заседании методической комиссии Института математики и информационных технологий

протокол № 5 от «15» 02 2021 г.,

Председатель методической
комиссии Института


/И.В. Пономарев/
(подпись)

1. Общая характеристика программы

Данный курс направлен на изучение основ языка Си, на котором написано подавляющее большинство операционных систем. Си является языком системного уровня, на котором написаны драйверы и библиотеки для работы с оборудованием и компьютерной периферией. Сейчас С популярен преимущественно благодаря «интернету вещей». Огромное количество «умных» девайсов действительно работают на базе программного обеспечения, которое написано на С.

Знание основ языка Си будет полезно не только начинающим программистам, но и программистам со стажем. Это знание позволит понять внутренние механизмы и устройство большинства современных высокоуровневых языков, поскольку данный язык является их прародителем. Язык программирования Си - это универсальный язык с богатым набором операторов и компактным способом записи выражений. По данным TIOBE (TIOBE programming community index — индекса, оценивающего популярность языков программирования, на основе подсчёта результатов поисковых запросов) язык Си занимает по популярности первое место в мире (декабрь 2020). Благодаря данному курсу, вы познакомитесь с основными понятиями используемыми в программировании, научитесь напрямую работать с оперативной памятью, получите навык написания простых системных программ и библиотек, а также применять полученные знания при работе с любым другим языком программирования. Для работы с Си существует большое количество сред разработки: Visual Studio, Dev-C++, NetBeans, Code::Blocks. Среди них есть среды с открытым исходным кодом, а также бесплатные on-line оболочки в сети. (примеры приводятся в курсе).

1.1. Цель реализации программы

- формирование базовых знаний в области основ алгоритмизации и программирования;
- выработка навыков решения типичных задач с использованием ЭВМ;
- овладение приемами разработки и отладки программ в современных средах программирования.

Изучение дисциплины направлено

- на развитие у обучающихся алгоритмического мышления, систематизацию принципов построения языков программирования и подходов к разработке программ для ЭВМ;
- на формирование навыков реализации алгоритмов на высокоуровневом императивном языке программирования; разработки, отладки и тестирования программ;

- на подготовку обучающихся к системному восприятию дальнейших дисциплин из учебного плана, использующих навыки алгоритмизации и программирования;

- на получение представлений об основных идеях структурного программирования и развитие способностей сознательно использовать материал курса, умение разбираться в существующих языковых и программных средствах и условиях их применения.

1.2. Планируемые результаты обучения

В результате освоения данного курса обучающиеся должны **знать**:

- основные понятия в области алгоритмизации, свойства алгоритмов, методы анализа сложности алгоритмов; синтаксис и базовые конструкции языка C, основные возможности и приемы программирования на этом языке; принципы структурного программирования;
- назначение, устройство и свойства основных структур данных, алгоритмы поиска и сортировки.

Уметь:

- разрабатывать алгоритмы решения типичных задач и реализовывать их на языке C; использовать для разработки современные интегрированные среды разработки.

Владеть:

- навыками построения математической модели и алгоритма для прикладной задачи; написания и отладки программ в интегрированной среде разработки.

1.3. Категория слушателей.

Все заинтересованные лица. Уровень образования: высшее образование, среднее специальное. Требования к уровню подготовки поступающего на обучение – школьный курс информатики. *Трудоемкость обучения 72 часа.*

1.4. Форма обучения дистанционная.

2. Содержание программы

2.1. Учебный план программы повышения квалификации

№ п/п	Наименование разделов, (дисциплин, модулей)	Всего час.	Дистанционные занятия, час.			СРС, час.	Форма контроля
			лекции	практич., семинары	Лаборатор.		
1	Синтаксис языка Си.	6	2	-	2	2	тест
2	Управляющие конструкции языка Си.	14	4	-	4	6	тест
3	Функции.	10	2	-	4	4	тест
4	Массивы.	10	2		4	4	Отчет по лабораторной работе
5	Структуры.	10	2		4	4	Отчет по лабораторной

							работе
6	Строки.	10	2		4	4	тест
7	Файлы.	10	2		4	4	Отчет по лабораторной работе
8	Итоговая аттестация	2		2			Итоговое тестирование
	Итого	72	16	2	26	28	

2.2. Учебно-тематический план

№ п/п	Наименование разделов, тем	Всего час.	Дистанционные занятия, час.			СРС, час.
			лекции	практич., семинары	Лаборатор.	
1.	Синтаксис языка Си. Алфавит, идентификаторы, ключевые слова. Комментарии. Типы, константы. Операции и выражения. Приоритет операций. Общая структура программы.	6	2	-	2	2
2.	Управляющие конструкции языка Си. Условный оператор. Оператор ветвления. Операторы циклов. Прерывание циклов. Инвариант цикла	14	4	-	4	6
3.	Функции. Параметры функций. Инвариантная функция. Рекурсия.	10	2	-	4	4
4.	Массивы. Многомерные массивы. Массивы как параметры функций. Типичные операции над массивами. Указатели. Связь указателей и массивов. Динамические массивы. Указатели на функции. Указатели и	10	2		4	4

	параметры функций.					
5.	Структуры. Битовые поля. Указатели на структуры. Объединения. Размещение структур в памяти	10	2		4	4
6.	Строки. Функции для работы со строками. Типичные операции над строками. Строки как параметры функций	10	2		4	4
7.	Файлы. Файловая система. Текстовые и бинарные файлы. Функции для работы с файлами. Типичные приемы работы с файлами.	10	2		4	4
8.	Итоговая аттестация	2	-	2	-	-
	Итого	72	16	2	26	28

2.3. Календарный учебный график.

3. Календарный учебный график не требуется, поскольку данный учебный курс является дистанционным, обучение может быть начато в любой момент по индивидуальной заявке. Расчетное время обучения 2 календарные недели. В качестве примера, представляется предварительный календарный учебный план-график реализации образовательной программы (дата начала обучения 01.11.2020 – 15.11.2020 дата завершения обучения).

Дата	Время (часов)	Дисциплина	Преподаватель
01.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна
02.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна
03.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна
04.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна

05.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна
06.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна
07.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна
08.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна
09.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна
10.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна
11.11.2021	6	Основы программирования на С	Смолякова Лариса Ленгардовна
12.11.2021	4	Основы программирования на С	Смолякова Лариса Ленгардовна
13.11.2021	2	Основы программирования на С	Смолякова Лариса Ленгардовна

3.1. Рабочая программа раздела (дисциплины, модуля) Основы программирования на С.

Раздел 1. Синтаксис языка Си. (6 часов)

Тема 1.1. Алфавит, идентификаторы, ключевые слова. Комментарии. Типы, константы. (2 часа)

Тема 1.2. Операции и выражения. Приоритет операций. Общая структура программы. (4 часа)

Раздел 2. Управляющие конструкции языка Си. (14 часов)

Тема 2.1. Условный оператор. Оператор ветвления. (4 часа)

Тема 2.2. Операторы циклов. Прерывание циклов. Инвариант цикла (10 часа)

Раздел 3. Функции. (10 часов)

Тема 3.1. Понятие функции. Параметры функций. (6 часов)

Тема 3.2. Инвариантная функция. Рекурсия. (4 часа)

Раздел 4. Массивы. Типичные операции над массивами. (10 часов)

Тема 4.1. Одномерные массивы (2 часа)

Тема 4.2. Многомерные массивы. (2 часа)

- Тема 4.3. Массивы как параметры функций. (2 часа)
- Тема 4.4. Типичные операции над массивами. (2 часа)
- Тема 4.5. Указатели. Связь указателей и массивов. Динамические массивы. Указатели на функции. Указатели и параметры функций. (2 часа)
- Раздел 5. Структуры. Объединения. Размещение структур в памяти (10 часов)
- Тема 5.1. Структуры. Битовые поля. (4 часа)
- Тема 5.2. Указатели на структуры. (2 часов)
- Тема 5.3. Объединения. Размещение структур в памяти (4 часов)
- Раздел 6. Строки. (10 часов)
- Тема 6.1. Строки. Функции для работы со строками. (6 часа)
- Тема 6.2. Типичные операции над строками. Строки как параметры функций. (4 часов)
- Раздел 7. Файлы. (10 часов)
- Тема 7.1. Файловая система. Текстовые и бинарные файлы. (6 часа)
- Тема 7.2. Функции для работы с файлами. Типичные приемы работы с файлами. (4 часов)
- Итоговая аттестация (2 часа)

Перечень лабораторных работ и практических (семинарских) занятий

№ темы	Наименование практических (семинарских) занятий лабораторных работ	Дистанционных занятий, час.
1.	Управляющие конструкции языка Си.	4
2.	Функции.	4
3.	Массивы.	6
4.	Структуры.	6
5.	Строки.	6
6.	Файлы.	6

Виды самостоятельной работы слушателей (СРС)

№ п/п	Вид СРС	Трудоемкость, час.
1.	Самостоятельная работа реконструктивно-вариативного типа (Индивидуальное задание)	20
2.	Подготовка к тесту по дисциплине	16
3.	Изучение дополнительной литературы по дисциплине и Интернет источников (список приведен в программе дисциплины).	4

4. Условия реализации программы (организационно-педагогические)

4.1. Материально-технические условия

Компьютер с комплектацией:

- ОС: WINDOWS 7 или выше;
- Браузер: Google Chrome, Yandex.Browser;
- Интегрированная среда разработки (MS Visual Studio, Netbeans, Eclipse).
- Microsoft word. Power Point
- доступ к образовательному пространству, расположенному по адресу: <https://public.edu.asu.ru/>, где расположены материалы для самостоятельного изучения и выполнения.

4.2. Учебно-методическое и информационное обеспечение программы

Литература

Список основной литературы

1. Подбельский В.В., Фомин С.С. Курс программирования на языке Си М.: ДМК Пресс, 2012
2. Вирт Н. Алгоритмы и структуры данных М.: ДМК Пресс, 2010
3. Дейл Н., Уимз Ч., Хедингтон М. Программирование на С++ М.: ДМК Пресс, 2007
4. Паронджанов В.Д. Учись писать, читать и понимать алгоритмы. Алгоритмы для правильного мышления. Основы алгоритмизации М.: ДМК Пресс, 2012
5. Смирнов А. А. Технологии программирования М.: ДМК Пресс, 2010
6. Абрамов С. А. Лекции о сложности алгоритмов М.: МЦНМО, 2009

Интернет-ресурсы.

1. <http://www.nns.ru/> Национальная электронная библиотека.
2. <http://www.microinform.ru/> Учебный центр компьютерных технологий «Микроинформ».
3. <http://www.tests.specialist.ru/> Центр компьютерного обучения МГТУ им. Н.Э.Баумана.
4. <http://www.intuit.ru/> Национальный открытый университет ИНТУИТ
5. <http://www.window.edu.ru/> Библиотека учебной и методической литературы
6. <http://www.osp.ru/> Журнал «Открытые системы»
7. C — The ISO Standard — Rationale, Revision 5.10 (англ.) (апрель 2004). — Обоснование и пояснения для стандарта C99. — <http://www.open-std.org/JTC1/SC22/WG14/www/docs/C99RationaleV5.10.pdf>
8. ISO/IEC JTC1/SC22/WG14 official home – Официальная страница международной рабочей группы по стандартизации языка программирования Си. – <http://www.open-std.org/JTC1/SC22/WG14/>

5. Оценка качества освоения программы (форма аттестации, оценочные и методические материалы)

Мониторинг степени освоения учебного материала слушателями осуществляется в форме лабораторных заданий, промежуточных тестов и итогового теста.

Практические задания к каждому занятию предметно-методического раздела являются обязательными и подлежат размещению на образовательном портале. Результаты проверенных заданий обучающихся также размещаются на образовательном портале с возможными комментариями по поводу выявленных ошибок.

Освоение программы завершается итоговым онлайн-тестированием.

На итоговую работу отведено 2 часа учебного времени.

Основанием допуска к выполнению итогового задания является положительный результат индивидуального выполнения слушателем контрольных тестов и практических заданий по каждому разделу Программы.

5.3. Оценочные средства

Основные показатели оценки планируемых результатов – ответы на вопросы теста и выполнение практических работ.

Процедура тестирования включает в себя ответы на вопросы электронного тестирования в системе Moodle.

Итоговый тест включает 20 вопросов, которые относятся к типу «Множественный выбор» (позволяет выбирать один или несколько правильных ответов из заданного списка).

Тест ограничен по времени и по числу попыток его прохождения (3 попытки).

Порядок представления вопросов в тесте и вариантов ответов в вопросах – случайный (база достаточно большая более 300 вопросов).

Примерный вариант итогового теста

1. Что выведет следующая программа?

```
#include <stdio.h>
void main() {
    int a;
    unsigned b;
    if (sizeof(a)==sizeof(b))
        printf("YES");
    else
        printf("NO");
}
```

Выберите один ответ:

- Yes
- Нет правильного варианта ответа
- No
- Возникнет ошибка компиляции

2. Что выведет следующая программа?

```
#include <stdio.h>
void main() {
    float f=2.0;
    printf("%d", sizeof(f));
}
```

Ответ: 4

3. Если $x=4$ и $y=7$, то результатом вычисления выражения

$x < 3 \ \&\& \ y < 7 \ || \ y != 6$

будет число:

- 4
- 2
- 3
- 0

4. Как обозначаются логические операции в языке Си?
Выберите один ответ:

- && || !
- && || !!
- && // !
- & | !
- AND OR NOT

5. Что выведет следующая программа?

```
#include <stdio.h>
void main() {
    unsigned char p=0x0F;
    if (p>>4)
        printf("%d", p);
    else
        printf("%d", p+1);
}
```

Выберите один ответ:

- 1
- 0
- 16
- 0x10
- 4

6. Какое из следующих выражений является примером корректного употребления условной операции?

Выберите один ответ:

- x<1 ? printf("x<1"); printf("x>=1");
- x<1 ? printf("x<1") : printf("x>=1");
- x<1 : printf("x<1") ? printf("x>=1");
- x<1 ? printf("x<1") ? printf("x>=1");
- x < 1 ? printf("x<1"); : printf("x>=1");

7. Что выведет следующая программа?

```
#include <stdio.h>
void main() {
    int p=10;
    p=p>50? p++ : if (p<0) p/=2 else p*=2;
    printf("%d", p);
}
```

Выберите один ответ:

- 20
- 0
- 5
- 10
- Нет правильного варианта ответа
- 50

8. Что выведет следующая программа?

```
#include <stdio.h>
void main() {
    float f=5;
    int x;
    x=f%2;
    printf("%d", x);
}
```

Выберите один ответ:

- При компиляции программы возникнут ошибки
- 5.250000
- 1.000000
- Нет правильного варианта ответа
- 2.500000

9. В каких строках программы содержатся ошибки?

1. #include <stdio.h>
- 2.
3. void main() {
4. int a,b;
5. printf("input a=");
6. scanf("%f", a);
7. printf("input b=");
8. scanf("%d", &b);
9. c=a+b
10. printf("a+b=%c, c");
11. }

Выберите один или несколько ответов:

- 1
- 2

- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11

10. Сколько первых символов в идентификаторах должны различаться для того, чтобы идентификаторы считались различными согласно стандарту "ANSI C"?

Выберите один ответ:

- все
- 8
- 32
- 64
- 16

11. Какие из следующих значений записаны в соответствии с синтаксисом языка Си?

Выберите один или несколько ответов:

- 1.2009
- 12009,0e-001
- 1,2009
- 12009.0e-1
- 12009e-001
- 12009.0E-001

12. Какое значение получит переменная x после выполнения следующего оператора?
`float x=1+1e+1e+1;`

Выберите один ответ:

- a. Оператор синтаксически не корректен
- b. 12.000000
- c. 4.000000
- d. 2e2

13. Как называется директива препроцессора для подключения внешних файлов (библиотек):

Выберите один ответ:

-

- income
- include
- extern
- define
- exclude

14. В каких частях программы на языке Си могут располагаться объявления переменных?

Выберите один или несколько ответов:

- Вне описаний функций
- В любой точке тела функции `main()`
- В начале тела функций `main()`
- После заголовка функции `main()`, но перед ее телом
- В любом месте программы

15. Что выведет на экран следующая программа, если ее откомпилировать и исполнить?

```
#include <stdio.h>
#define max=5;

void main() {
    int a=0;
    a=max++;
    printf("%d", a++);
}
```

Выберите один ответ:

- 6
- 0
- 7
- 5
- Возникнет ошибка компиляции

16. Укажите, какие из следующих описаний массивов являются корректными.

Выберите один или несколько ответов:

- `int D[1..3];`
- `int A[3]={1,3,5};`
- `int B[3]={11,22};`
- `int M[3];`
- `int C[3]={0,1,2,3};`

17. Что выведет следующая программа, если ее откомпилировать и исполнить?

```
#include <stdio.h>
void main() {
    int i, M[3]={1} ;
    for (i=0; i<3;i++)
        printf("%d", M[i]);
}
```

Выберите один ответ:

- 111
- 100
- Нет правильного варианта ответа
- 000
- Выводимые программой значения не определены
- 1 1 1

18. Какое значение получит переменная st после выполнения следующего фрагмента программы, если при выполнении программы на ввод ей будет подана указанная в комментарии строка (без кавычек)?

```
char st[20];
gets(st); //Введенная строка "Hello world!"
```

Выберите один ответ:

- Нет правильного варианта ответа
- Hello world
- Hello
- Программа синтаксически некорректна
- Helloworld!
- Hello world!

19. Сколько раз исполнится цикл в следующей программе?

```
void main() {
    int i=1, j=1 ;
    for (; ;) {
        printf("%d%d", i,j);
    }
}
```

Выберите один ответ:

- Ни разу
- При компиляции программы возникнет ошибка
- Нет правильного варианта ответа
- 1
- Бесконечное количество раз

20. Что выведет следующая программа?

```
#include <stdio.h>
void main() {
    int i;
    float s=0;
    while (i>1) {
        s=s+1/i;
        i=i-1;
    }
    printf("%f", s);
}
```

Выберите один ответ:

- Нет правильного варианта ответа
- 1.000000
- 0.200000
- 2.283333
- 0.000000
- При компиляции программы возникнет ошибка

Примеры практических заданий

Практические задания оформляются в виде отчета. Отчет должен содержать следующие элементы:

- Титульный лист с указанием фамилии, имени, отчества автора, названия и номера лабораторной работы.
- Для каждой решаемой задачи: формулировка задания.
- Для каждой решаемой задачи: алгоритм решения задачи в словесно-пошаговой форме.
- Для каждой решаемой задачи: исходный код программы с комментариями.
Обязательные комментарии:
 - описание каждой функции (назначение, параметры, возвращаемые значения);
 - описание всех используемых переменных и констант;
 - описание отдельных логически завершенных фрагментов программы.
 - Размер матрицы вводит пользователь. Массив задавать динамический. Написать функции для ввода матрицы пользователем вручную и случайным образом, а также функцию для вывода матрицы.
 - Тесты, иллюстрирующие все основные варианты работы

Задание 1. Массивы

Инструкция. Прочитайте внимательно задание. Ответ представьте в одном файле Word. Пример оформления задания находится ниже (файл "SampleLabReport", чтобы скачать файл нажмите на него). Для того, чтобы добавить файл (Ваш ответ на задание), нажмите кнопку "добавить ответ на задание" и загрузите файл. Убедительная просьба в названии файла указывать ФИО.

Задача. В заданной прямоугольной матрице с количеством столбцов $2n$ переставить столбцы в следующем порядке: $1, 2n, 2, 2n-1, 3, 2n-2, \dots, n, n+1$. Например, в матрице с шестью столбцами итоговый порядок столбцов должен быть следующим: $1, 6, 2, 5, 3, 4$.

Задание 2. Структуры

Инструкция. Прочитайте внимательно задание. Ответ представьте в одном файле Word. Пример оформления задания находится ниже (файл "SampleLabReport", чтобы скачать файл нажмите на него). Для того, чтобы добавить файл (Ваш ответ на задание), нажмите кнопку "добавить ответ на задание" и загрузите файл. Убедительная просьба в названии файла указывать ФИО.

Задача. Английская денежная единица фунт стерлингов состоит из 20 шиллингов, а каждый шиллинг — из 12 пенсов. Описать тип `struct Pound`, задающий сумму, выраженную в фунтах, шиллингах и пенсах. Реализовать в виде отдельных функций следующие операции над переменными этого типа: а) проверка корректности значения (составляющие суммы должны быть выражены в максимально допустимых единицах, т.е. не 0 фунтов 0 шиллингов 1000 пенсов, а 4 фунта 3 шиллинга 4 пенса); б) увеличение заданной суммы на заданное количество пенсов, шиллингов, фунтов; в) сложения двух денежных сумм; г) вычисления разности денежных сумм; д) перевод денежной суммы в пенса; е) вывод денежной суммы в формате «99-99-99».

С использованием описанных типа и функций разработать программу, которая для заданного набора из N денежных сумм в английских единицах находит

- 1) среднее значение, выраженное в том же виде;
- 2) пары сумм, наиболее близких и наиболее далеких по значению.

Задание 3. Текстовые файлы


Инструкция. Прочитайте внимательно задание. Ответ представьте в одном файле Word. Пример оформления задания находится ниже (файл "SampleLabReport", чтобы скачать файл нажмите на него). Для того, чтобы добавить файл (Ваш ответ на задание), нажмите кнопку "добавить ответ на задание" и загрузите файл. Убедительная просьба в названии файла указывать ФИО.

Задача. В текстовом файле (`input.txt`) построчно содержатся слова или предложения. Создать текстовый файл (`output.txt`), содержащий слова и предложения- палиндромы (ШАЛАШ, ЛЁША НА ПОЛКЕ КЛОПА НАШЁЛ, и т.п.), встречающиеся в исходном файле.

6. Кадровые условия (составители программы)

Составитель рабочей программы – Смолякова Лариса Ленгардовна, ст.преподаватель, кафедры информатики Алтайского государственного университета.

Директор ИДПО

 Т.Г. Строителева

Директор Учебного центра по ДПО ИМИТ
(подпись)

 О.А. Жданова.
И.О.Ф.