

ФГБОУ ВО «УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»
СИБАЙСКИЙ ИНСТИТУТ (ФИЛИАЛ) УУиТ
ЕСТЕСТВЕННО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

Утверждено:

на заседании кафедры
протокол № 11 от «31» мая 2023 г.

Зав. кафедрой  Гумеров И.С.

Согласовано:

Председатель УМК естественно-
математического факультета



/Ильбулова Г.Р.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Дисциплина **ПРАКТИКУМ НА ЭВМ**

(наименование дисциплины)

Часть, формируемая участниками образовательных отношений

(обязательная часть или часть, формируемая участниками образовательных отношений, факультатив)

программа бакалавриата

Направление подготовки

01.03.02 Прикладная математика и информатика

(указывается код и наименование направления подготовки)

Направленность (профиль) подготовки

Прикладная математика и информационные технологии

(указывается наименование направленности (профиля) подготовки)

Квалификация

бакалавр

(указывается квалификация)

Разработчик (составитель)

Доцент кафедры, к.ф.-м.н.

(должность, ученая степень, ученое звание)



/ Хисаметдинов Ф.З.

Для приема: 2023 г.

Сибай 2023 г.

Составитель: Хисаметдинов Ф.З.

Рабочая программа дисциплины утверждена на заседании кафедры прикладной математики и информационных технологий, протокол № 11 от «31» мая 2023 г.

И.о. заведующего кафедрой  / Гумеров И.С. /

Дополнения и изменения, внесенную в рабочую программу дисциплины

утверждены на заседании кафедры

протокол №__ от «__» _____ 20__ г.

Заведующий кафедрой _____ / _____ /

Дополнения и изменения, внесенную в рабочую программу дисциплины

утверждены на заседании кафедры

протокол №__ от «__» _____ 20__ г.

Заведующий кафедрой _____ / _____ /

Дополнения и изменения, внесенную в рабочую программу дисциплины

утверждены на заседании кафедры

протокол №__ от «__» _____ 20__ г.

Заведующий кафедрой _____ / _____ /

Список документов и материалов

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными в образовательной программе индикаторами достижения компетенций
2. Цель и место дисциплины в структуре образовательной программы
3. Содержание рабочей программы (объем дисциплины, типы и виды учебных занятий, учебно-методическое обеспечение самостоятельной работы обучающихся)
4. Фонд оценочных средств по дисциплине
 - 4.1. Перечень компетенций и индикаторов достижения компетенций с указанием соотнесенных с ними запланированных результатов обучения по дисциплине. Описание критериев и шкал оценивания результатов обучения по дисциплине
 - 4.2. Типовые контрольные задания или иные материалы, необходимые для оценивания результатов обучения по дисциплине, соотнесенных с установленными в образовательной программе индикаторами достижения компетенций. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине.
5. Учебно-методическое и информационное обеспечение дисциплины
 - 5.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины
 - 5.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» и программного обеспечения, необходимых для освоения дисциплины
6. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными в образовательной программе индикаторами достижения компетенций

В результате освоения образовательной программы обучающийся должен овладеть

Категория (группа) компетенций	Формируемая компетенция (с указанием кода)	Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине
-Профессиональные компетенции	ПК-4. Владеет навыками проектирования программного обеспечения	ПК-4.1. Знает: -принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; - типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; -методы и средства проектирования программного обеспечения; -методы и средства проектирования баз данных; -методы и средства проектирования программных интерфейсов.	<i>Знать</i> принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; методы и средства проектирования программного обеспечения; методы и средства проектирования баз данных; методы и средства проектирования программных интерфейсов.
		ПК-4.2. Умеет: - использовать существующие типовые решения и шаблоны проектирования программного обеспечения; -применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; -осуществлять коммуникации с заинтересованными сторонами.	<i>Уметь</i> использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; осуществлять коммуникации с заинтересованными сторонами.
		ПК-4.3. Владеет: -навыками разработки и согласования технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения; - навыками распределения заданий между программистами в соответствии с техническими спецификациями; -навыками осуществления контроля выполнения заданий; -навыками осуществления обучения и наставничества; -навыками формирования и предоставления отчетности в соответствии с установленными регламентами; -навыками оценки и согласования сроков выполнения поставленных задач.	<i>Владеть</i> навыками разработки и согласования технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения; навыками распределения заданий между программистами в соответствии с техническими спецификациями; навыками осуществления контроля выполнения заданий; навыками осуществления обучения и наставничества; навыками формирования и предоставления отчетности в соответствии с установленными регламентами; навыками оценки и согласования сроков выполнения поставленных задач.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Практикум на ЭВМ» относится к части, формируемой участниками образовательных отношений.

Дисциплина изучается на 1,2 курсе в (2, 3, 4 семестр) очной и очно-заочной форм обучения.

Курс «Практикум на ЭВМ» является прикладным курсом, обеспечивающим практическую часть лекционных курсов «Основы информатики» и «Языки и методы программирования», поэтому курс разделен на 2 части – «Практикум на ЭВМ», «Практикум на ЭВМ. Часть 2» соответственно.

Основными целями дисциплины являются:

- овладение студентами основными понятиями теории алгоритмов, знаниями и умениями по составлению программ на различных языках программирования, принципами и методами разработки алгоритмов и программ, навыками составления программ и крупных программных комплексов;

- развитие логического и алгоритмического мышления;

- повышение уровня информационной культуры.

В процессе преподавания учебной дисциплины и ее самостоятельного изучения студентами решаются следующие основные задачи:

- освоение основных положений теории алгоритмов, основ алгоритмизации и программирования;

- формирование практических умений и навыков по разработке алгоритмов и программ.

3. Содержание рабочей программы (объем дисциплины, типы и виды учебных занятий, учебно-методическое обеспечение самостоятельной работы обучающихся)

Содержание рабочей программы представлено в Приложении № 1.

4. Фонд оценочных средств по дисциплине

4.1. Перечень компетенций и индикаторов достижения компетенций с указанием соотношенных с ними запланированных результатов обучения по дисциплине. Описание критериев и шкал оценивания результатов обучения по дисциплине

Код и формулировка компетенции: **ПК – 4** – способность осуществлять целенаправленный поиск информации о новейших научных и технологических достижениях в информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет") и в других источниках

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Критерии оценивания результатов обучения	
		«Не зачтено»	«Зачтено»
ПК-4.1. Знает: -принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; -типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при	Знает: -принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; -типовые решения, библиотеки программных модулей, шаблоны, классы объектов, ис-	Не знает принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке	Обладает на хорошем уровне знаниями о принципах построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовых решениях, библиотеках программных модулей, шаблонах, классах объектов,

разработке программного обеспечения; -методы и средства проектирования программного обеспечения; -методы и средства проектирования баз данных; -методы и средства проектирования программных интерфейсов.	пользуемые при разработке программного обеспечения; -методы и средства проектирования программного обеспечения; -методы и средства проектирования баз данных; -методы и средства проектирования программных интерфейсов.	программного обеспечения; методы и средства проектирования программного обеспечения; методы и средства проектирования баз данных; методы и средства проектирования программных интерфейсов.	используемых при разработке программного обеспечения; методах и средствах проектирования программного обеспечения; методах и средствах проектирования баз данных; методах и средствах проектирования программных интерфейсов
ПК-4.2. Умеет: - использовать существующие типовые решения и шаблоны проектирования программного обеспечения; -применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; -осуществлять коммуникации с заинтересованными сторонами.	Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; осуществлять коммуникации с заинтересованными сторонами.	Не умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; осуществлять коммуникации с заинтересованными сторонами.	Хорошо умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; осуществлять коммуникации с заинтересованными сторонами.
ПК4.3. Владеет навыками разработки и согласования технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения; - навыками распределения заданий между программистами в соответствии с техническими спецификациями; -навыками осуществления контроля выполнения заданий; -навыками осуществления обучения и наставничества; -навыками формирования и предоставления отчетности в соответствии с установленными регламентами; -навыками оценки и согласования сроков выполнения поставленных задач.	Владеет навыками разработки и согласования технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения; - навыками распределения заданий между программистами в соответствии с техническими спецификациями; навыками осуществления контроля выполнения заданий; навыками осуществления обучения и наставничества; навыками формирования и предоставления отчетности в соответствии с установленными регламентами; навыками оценки и согласования сроков выполнения поставленных задач.	Не владеет разработки и согласования технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения; - навыками распределения заданий между программистами в соответствии с техническими спецификациями; навыками осуществления контроля выполнения заданий; навыками осуществления обучения и наставничества; навыками формирования и предоставления отчетности в соответствии с установленными регламентами; навыками оценки и согласования сроков выполнения поставленных задач.	Хорошо владеет навыками разработки и согласования технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения; - навыками распределения заданий между программистами в соответствии с техническими спецификациями; навыками осуществления контроля выполнения заданий; навыками осуществления обучения и наставничества; навыками формирования и предоставления отчетности в соответствии с установленными регламентами; навыками оценки и согласования сроков выполнения поставленных задач.

Критериями оценивания являются баллы, которые выставляются преподавателем за виды деятельности (оценочные средства) по итогам изучения модулей (разделов дисциплины), перечисленных в рейтинг-плане дисциплины (для экзамена: текущий контроль – максимум 40 баллов; рубежный контроль – максимум 30 баллов, поощрительные баллы – максимум 10; для зачета: текущий контроль – максимум 50 баллов; рубежный контроль – максимум 50 баллов, поощрительные баллы – максимум 10).

для зачета:

зачтено – от 60 до 110 рейтинговых баллов (включая 10 поощрительных баллов),
не зачтено – от 0 до 59 рейтинговых баллов).

4.2. Типовые контрольные задания или иные материалы, необходимые для оценивания результатов обучения по дисциплине, соотнесенных с установленными в образовательной программе индикаторами достижения компетенций. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине.

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные средства
ПК-4.1. Знает: -принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; - типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; -методы и средства проектирования программного обеспечения; -методы и средства проектирования баз данных; -методы и средства проектирования программных интерфейсов. ПК-4.2. Умеет: - использовать существующие типовые решения и шаблоны проектирования программного обеспечения; -применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; -осуществлять коммуникации с заинтересованными сторонами. ПК-4.3. Владеет: -навыками разработки и согласования технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения; - навыками распределения заданий между программистами в соответствии с техническими спецификациями; -навыками осуществления контроля выполнения заданий; -навыками осуществления обучения и наставничества; -навыками формирования и предоставления отчетности в соответствии с установленными регламентами; -навыками оценки и согласования сроков выполнения поставленных задач.	Знает: -принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; - типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; -методы и средства проектирования программного обеспечения; -методы и средства проектирования баз данных; -методы и средства проектирования программных интерфейсов.	Индивидуальный опрос; Групповой опрос; Решение задач; Тесты; Контрольные работы
	Умеет: - использовать существующие типовые решения и шаблоны проектирования программного обеспечения; -применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; -осуществлять коммуникации с заинтересованными сторонами.	Индивидуальный опрос; Групповой опрос; Решение задач; Тесты; Контрольные работы
	Владеет: -навыками разработки и согласования технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения; - навыками распределения заданий между программистами в соответствии с техническими спецификациями; -навыками осуществления контроля выполнения заданий; -навыками осуществления обучения и наставничества; -навыками формирования и предоставления отчетности в соответствии с установленными регламентами; -навыками оценки и согласования сроков выполнения поставленных задач.	Подготовка докладов (рефератов); Индивидуальный опрос; Групповой опрос; Решение задач; Тесты; Контрольные работы; Вопросы экзамена

Критериями оценивания при *модульно-рейтинговой системе* являются баллы, которые выставляются преподавателем за виды деятельности (оценочные средства) по итогам изучения модулей (разделов дисциплины), перечисленных в рейтинг-плане дисциплины (*для экзамена*: текущий контроль – максимум 70 баллов; рубежный контроль – максимум 30 баллов, поощрительные баллы – максимум 10).

Рейтинг-план дисциплины

Рейтинг-план дисциплины представлен в приложении 2.

Примерные темы лабораторных работ (2 семестр)

1. Работа в интегрированной среде CodeBlocks на примере программ линейной структуры на языке программирования Си
2. Программы разветвленной структуры на языке программирования Си
3. Программы циклической структуры на языке программирования Си
4. Обработка статических массивов на языке программирования Си
5. Работа с символьными данными на языке программирования Си
6. Функции и рекурсии на языке программирования Си
7. Программные средства для работы с файлами на языке программирования Си
8. Динамические массивы и динамические структуры данных на языке программирования Си

Примерные темы лабораторных работ (3 семестр)

1. Введение. Знакомство с Python. Ввод-вывод данных. Команды print и input. Параметры sep и end.
2. Задачи целочисленной арифметики.
3. Условный оператор. Выбор из двух. Логические операции. Вложенные и каскадные условия.
4. Типы данных. Числовые типы данных: int, float. Модуль math. Строковый тип данных.
5. Циклы for и while. Цикл for. Цикл for: функция range. Частые сценарии. Цикл while. Цикл while: обработка цифр числа. Опции break, continue и else. Вложенные циклы.
6. Строковый тип данных. Индексация. Срезы. Методы строк. Строки в памяти компьютера, таблица символов Unicode
7. Списки. Введение в списки. Основы работы со списками. Методы списков. Вывод элементов списка. Методы строк: split, join. Списочные выражения. Сортировка списков.
8. Функции. Функции без параметров. Функции с параметрами. Локальные и глобальные переменные. Функции с возвратом значения.
9. Работа над мини-проектами. Модуль random. Числовая угадайка.
10. Магический шар. Генератор безопасных паролей. Шифр Цезаря. Калькулятор систем счисления.

Примерные темы лабораторных работ (4 семестр)

1. Повторение основных конструкций языка Python . Тип данных bool и NoneType. Тип данных bool. Тип данных NoneType.
2. Вложенные списки.
3. Матрицы. Математические операции над матрицами.
4. Кортежи. Введение в кортежи. Основы работы с кортежами.
5. Множества. Множества в математике. Операции над множествами, диаграммы Эйлера-Венна. Введение в множества в Python.
6. Основы работы с множествами. Методы множеств. Генераторы множеств и frozenset.
7. Словари. Введение в словари в Python. Основы работы со словарями. Методы

словарей.

8. Вложенные словари и генераторы словарей.
9. Модули random и string. Модуль random.. Метод Монте-Карло и Bogosort.
10. Модули decimal, fraction и complex. Тип данных complex.
11. Модуль turtle. Модуль черепашки.
12. Функции. Необязательные и именованные аргументы. Функции с переменным количеством аргументов.
13. Парадигмы программирования. Функции как объекты. Функции высшего порядка. Встроенные функции map(), filter(), reduce(). Анонимные функции. Встроенные функции any(), all(), zip(), enumerate().
14. Работа с файлами. Файловый ввод и вывод. Работа с текстовыми файлами.

Примеры лабораторных работ Лабораторная работа №2 Программы разветвлённой структуры

1. ЦЕЛЬ РАБОТЫ: приобретение практических навыков в составлении алгоритмов и программ разветвленной структуры.

2. ОСНОВНЫЕ СВЕДЕНИЯ

Алгоритм разветвленной структуры - это алгоритм, в котором вычислительный процесс осуществляется по одной из ветвей. Если количество ветвей две – то используется условный блок, если больше – то множественный выбор. В программах используют соответственно условный оператор IF или оператор SWITCH для реализации разветвленного алгоритма.

2.1. Условный оператор IF

Условный оператор в языке Си имеет формат:

```
if (условие)
    оператор1;
else
    оператор2;
(полная форма) или
if (условие) оператор1; (сокращенная форма).
```

Если оператор1 и оператор2 состоят из одного оператора, то в фигурные скобки они не заключаются. Если оператор1 и/или оператор2 являются составным оператором (несколько операторов), то он заключается в фигурные скобки. Условие всегда заключается в круглые скобки.

В качестве примера приведем программу вычисления наибольшего из значений функции $y_1=x^2+1$, $y_2=7-x^2$, $y_3=x+1$ для любого x .

```
//lab2_1
#include <stdio.h>
main()
{ float x,y1,y2,y3,max;
  printf("Введите x\n"); scanf("%f",&x);
  y1=x*x+1; y2=7-x*x; y3=x+1;
  if (y1>y2)
    max=y1;
  else
```

```

    max=y2;
    if (y3>max)
        max=y3;
    printf ("x=%8.2f y1=%8.2f y2=%8.2f y3=%8.2f max=%8.2f\n",x,y1,y2,y3,max);
}

```

2.2. Условная трехместная операция

Условная тернарная операция используется с тремя операндами. В изображении условной операции применяются два символа '?' и ':' и три выражения – операнда:

выражение_1 ? выражение_2 : выражение_3;

Первым вычисляется значение выражения_1. Если оно истинно, т. е. не равно 0, то вычисляется значение выражения_2, которое становится результатом. Если при вычислении выражения_1 получится 0, то в качестве результата берётся значение выражения_3, например:

$x < 0 ? -x : x$;

Выражение возвращает абсолютную величину переменной x.

Пример программы вычисления наибольшего из значений функции $y_1 = x^2 + 1$, $y_2 = 7 - x^2$, $y_3 = x + 1$ для любого x.

```

#include <stdlib.h>
#include <stdio.h>
int main(int argc, char *argv[])
{ float x,y1,y2,y3,max;
  printf("Input x "); scanf("%f",&x);
  y1=x*x+1; y2=7-x*x; y3=x+1;
  y1>y2? max=y1:max=y2;
  y3>max? max=y3:max=max;
  printf ("x=%8.2f y1=%8.2f y2=%8.2f y3=%8.2f max=%8.2f\n",x,y1,y2,y3,max);
  system("PAUSE");
  return EXIT_SUCCESS;
}

```

2.3 Множественный выбор

В программах с множественным выбором используется переключатель switch , который сравнивает значение выражения, указанного за ним, и выполняет один из операторов, метка которого совпадает с этим значением. Общий вид:

Switch (выражение)

{ case метка_1: список_операторов_1;

.....

case метка_n: список_операторов_n;

default : операторы; }

Значения выражения и меток должны быть целочисленными константами.

Например, определим количество дней по введенному номеру месяца.

```

//lab2_2 множественный выбор
#include<stdio.h>
#include<conio.h> // Заголовочный файл для работы с терминалом
int m;
main()
{
  printf("\n Введите номер месяца :"); scanf("%d",&m);

```

```
printf("\n В  %4d месяце дней: ",m);
switch(m){
case 1:case 3:case 5: case 7:case 8:case 10:case 12:printf("тридцать один\n");break;
case 2: printf ("двадцать восемь \n");break;
case 4: case 6: case 9: case 11: printf("тридцать\n");break;
default: printf("\n Номер месяца неверен \n");}
getch (); }
```

В примере программы, если номер месяца превышает 12, выводится сообщение о неверном вводе месяца, для чего используется default. Оператор break служит для прерывания цикла проверки и перехода в конец переключателя. В случае отсутствия break, происходит переход на следующую ветвь.

Функция getch, подключаемая заголовочным файлом conio.h, ожидает нажатия любой клавиши. Это позволяет просмотреть результаты, не используя ALT+F5 для просмотра экрана пользователя.

3. ВЫПОЛНЕНИЕ РАБОТЫ

Составить программы разветвляющейся структуры согласно вариантам задания 1 (используя IF) и задания 2 (используя SWITCH) .

Варианты задания 1

- 1) Даны четыре числа. На сколько их сумма меньше их произведения?
- 2) Даны четыре числа. Вычислить сумму положительных среди них чисел.
- 3) Даны четыре числа. Вычислить произведение отрицательных среди них чисел.
- 4) Даны четыре числа. Все отрицательные среди них числа заменить на 0.
- 5) Даны четыре числа. Определить сколько среди них отрицательных и сколько положительных.
- 6) Даны четыре числа. Определить порядковый номер наименьшего среди них.
- 7) Даны два числа. Большее из этих двух чисел заменить их полусуммой, а меньшее удвоенным произведением.
- 8) Даны три числа. Меньшее среди них заменить на 0 .
- 9) Даны четыре числа. Найти разность между наибольшим и наименьшим среди них.
- 10) Даны три числа K, M и N. Поменять их значения местами таким образом, чтобы $K < M < N$.
- 11) Даны четыре разных числа. Найти среди них два наибольших.
- 12) Даны три числа. Поменять местами большее и меньшее из этих чисел.

Варианты задания 2

Предусмотреть ситуацию неправильного ввода исходных данных.

- 1) Вводится число программ $N \leq 20$. Напечатать фразу "Я разработал N программ", согласовав слово "программа" с числом N.
- 2) Вводится число экзаменов $N \leq 20$. Напечатать фразу "Мы успешно сдали N экзаменов", согласовав слово "экзамен" с числом N.
- 3) Вводится число лет ($N \leq 25$). Напечатать фразу "Мне N лет", согласовав слово "лет, год, года" с числом N.
- 4) Вводится число студентов $N \leq 30$. Вывести фразу "В нашей группе N студентов", согласовав слово "студент" с числом N.
- 5) Вводится число M - номер месяца. Определить номер квартала по введенному номеру месяца и номер полугодия.

- 6) Вводится число М - номер месяца. Определить время года по введенному номеру месяца и номер полугодия.
- 7) Вводится целое число С. Если $-9 \leq c \leq 9$ вывести величину числа в словесной форме с учетом знака, в противном случае - предупреждающее сообщение и повторный ввод.
- 8) Вводится номер семестра $N \leq 10$. Вывести фразу "Я проучился N семестров", согласовав слово "семестр" с числом N. Определить номер курса.
- 9) В китайском гороскопе года носят следующие названия: крыса, корова, тигр, заяц, дракон, змея, лошадь, овца, обезьяна, петух, собака, свинья. Учitando, что 1996 – год крысы, написать программу, определяющую название года по его номеру.
- 10) Вводится число Т – порядковый день в году. Определить номер месяца М и дня недели D, соответствующих Т. (Например, если $T=365$. то $M=12$, а $D=31$)
- 11) Вводится номер месяца М и дня D. Определить порядковый номер дня в году Т соответствующий этой дате.
- 12) Вводится номер месяца М и дня D. Определить день недели с датой М и D, считая, что год начинается с понедельника.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 4.1. Сравните виды условных операторов Си.
- 4.2. Всегда ли ставятся скобки и какие в условном операторе?
- 4.3. Зачем ставятся в переключателе операторы break и default?
- 4.4. Как будет работать программа из примера без break?

Лабораторная работа №3 Программы циклической структуры

1. ЦЕЛЬ РАБОТЫ: приобретение практических навыков в составлении алгоритмов и программ циклической структуры.

2. ОСНОВНЫЕ СВЕДЕНИЯ

2.1. Операторы циклов

В языке Си существует 3 вида циклов: 1) цикл с параметром или цикл типа for; 2) цикл с предусловием или цикл типа while, 3) цикл с постусловием или цикл типа do ... while . Во всех этих циклах условие продолжения цикла заключается в круглые скобки. В циклах типов for и while повторяющаяся часть состоит из одного оператора, если требуется выполнить в цикле несколько операторов, они заключаются в фигурные скобки, образуя составной оператор. В цикле с постусловием тело цикла помещается между словами do и while. В отличие от цикла с предусловием, цикл с постусловием выполнится хотя бы один раз. Циклы с пред- и постусловием продолжаются, если условие продолжения истинно.

2.2. Цикл с параметром

Цикл **for** используется тогда, когда количество повторений цикла заранее известно или может быть вычислено. Цикл **for** состоит из заголовка и тела цикла. В заголовке после слова **for** в круглых скобках записываются через точку с запятой три выражения:

А) **начальные значения:** операторы присваивания, которые выполняются один раз перед выполнением цикла;

В) **условие, при котором выполняется следующий шаг цикла;** если условие неверно, работа цикла заканчивается; если оно неверно в самом начале, цикл не выполняется ни одного

раза (говорят, что это *цикл с предусловием*, то есть условие проверяется перед выполнением цикла);

С) **действия в конце каждого шага** цикла (в большинстве случаев это операторы присваивания).

В каждой части заголовка может быть несколько операторов, разделенных запятыми.

Примеры заголовков:

```
for ( i = 0; i < 10; i ++ ) { ... }
```

```
for ( i = 0, x = 1.; i < 10; i += 2, x *= 0.1 ){ ... }
```

- Тело цикла заключается в фигурные скобки; если в теле цикла стоит всего один оператор, скобки можно не ставить.
- В тело цикла могут входить любые другие операторы, в том числе и другие циклы (такой прием называется «*вложенные циклы*»).
- Для того, чтобы легче разобраться в программе, все тело цикла и ограничивающие его скобки сдвигаются вправо на 2-3 символа (запись «лесенкой»).

Пример вычисления суммы квадратов натурального ряда чисел от 1 до n.

$$S = \sum_{i=1}^n i^2 = 1^2 + 2^2 + \dots + n^2.$$

```
//lab3_1 сумма квадратов натурального ряда
#include <stdio.h>
main()
{ int S,n,i;
  printf("Введите n "); scanf("%d",&n);
  for (S=0, i=1; i<=n; i++)
    S+=i*i; // соответствует S=S+i*i
  printf ("n=%d S=%d",n,S);
}
```

В разделе инициализации присваиваются начальные значения переменным S и i. При i, не превышающем n, цикл повторяется, иначе выполняется выход из цикла. Оператор инкремента i++ (или декремента i--), эквивалентен i=i+1 (i=i-1). Для досрочного выхода из цикла и перехода на следующий оператор после цикла используется оператор break. Для пропуска всех операторов, оставшихся до конца тела цикла и перехода к следующему повторению цикла используется оператор continue.

2.3 Циклы с неизвестным количеством повторений

Циклы while или repeat используются тогда, когда количество повторений цикла заранее неизвестно и не может быть вычислено.

Цикл while состоит из заголовка и тела цикла.

В заголовке после слова while в круглых скобках записывается условие, при котором цикл продолжает выполняться. Когда это условие нарушается (становится ложно), цикл заканчивается.

В условии можно использовать знаки логических отношений и операций

> < больше, меньше

>= <= больше или равно, меньше или равно

== равно

!= не равно

Если условие неверно в самом начале, то цикл не выполняется ни разу (это цикл с предусловием). Если условие никогда не становится ложным (неверным), то цикл никогда не

заканчивается; в таком случае говорят, что программа «зациклилась» — это серьезная логическая ошибка. В языке Си любое число, не равное нулю, обозначает истинное условие, а ноль — ложное

Условие

```
while ( 1 ){ ... } // бесконечный цикл
```

```
while ( 0 ){ ... } // цикл не выполнится ни разу
```

Тело цикла заключается в фигурные скобки; если в теле цикла стоит всего один оператор, скобки можно не ставить. В тело цикла могут входить любые другие операторы, в том числе и другие циклы (такой прием называется «вложенные циклы»). Для того, чтобы легче разобраться в программе, все тело цикла и ограничивающие его скобки сдвигаются вправо на 2-3 символа.

Цикл do-while также используется тогда, когда количество повторений цикла заранее неизвестно и не может быть вычислено.

Цикл do-while состоит из заголовка do, тела цикла и завершающего условия.

Условие записывается в круглых скобках после слова while, цикл продолжает выполняться, пока условие верно; когда условие становится ложным, цикл заканчивается. Условие проверяется только в конце очередного шага цикла (это цикл с постусловием), таким образом, цикл всегда выполняется хотя бы один раз. Если условие никогда не становится ложным (неверным), то цикл никогда не заканчивается, т.е. программа зацикливается. Тело цикла также заключается в фигурные скобки; если в теле цикла стоит всего один оператор, скобки можно не ставить. В тело цикла могут входить любые другие операторы, в том числе и другие циклы.

Рассмотрим пример из предыдущей работы - вычисление суммы квадратов натурального ряда чисел от 1 до n.

$$S = \sum_{i=1}^n i^2 = 1^2 + 2^2 + \dots + n^2.$$

Изменим условие в этом примере: Определить число n, при котором сумма чисел S предыдущего ряда не превысит величину K, введенную с клавиатуры. Такую программу можно реализовать с помощью циклов предусловия или постусловия следующим образом:

//lab3_2 циклы while и do...while

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
main()
```

```
{int K,S, i;
```

```
clrscr(); // очистка экрана перед выводом
```

```
printf ("Введите K "); scanf ("%d", &K);
```

```
// цикл с предусловием
```

```
S=0;
```

```
i= 0;
```

```
while ( S<K) { i++ ; S+=i*i };
```

```
printf ("n=%d S=%d\n", i, S);
```

```
getch();
```

```
}
```

```
//цикл с постусловием
```

```
S=0; i=0;
```

```
do
```

```
{ i++; S+=i*i; }
```

```
while ( S<K) ;
```

Отличие этих программ: если ввести K<1, то в программе с предусловием получим S=0, i=0 ; а в программе с постусловием S=1, i=1.

3. ВЫПОЛНЕНИЕ РАБОТЫ

Составить 3 варианта программ циклической структуры типа for и while, сравнить полученные результаты.

Варианты заданий

- | | | | |
|-----|---|-----|--|
| 1. | $\sum_{i=1}^N \frac{x^2 + 1}{i}$ | 2. | $\sum_{i=1}^N x \frac{\ln 3}{i}$ |
| 3. | $\sum_{i=1}^N \frac{x^2}{2i}$ | 4. | $\sum_{i=1}^N \frac{x^2}{2i + 1}$ |
| 5. | $\sum_{i=1}^N \frac{x}{\sin ix}$ | 6. | $\sum_{i=1}^N \frac{-1 + \cos ix}{i^3}$ |
| 7. | $\sum_{i=1}^N 3 \cos ix^3$ | 8. | $\sum_{i=1}^N \frac{\cos ix}{i}$ |
| 9. | $\sum_{i=1}^N \frac{\ln i - i^2}{4 + i}$ | 10. | $\sum_{i=1}^N \frac{3 - \operatorname{tg} i}{2 + i^2}$ |
| 11. | $\sum_{i=1}^N (-1)^{i+1} \frac{\cos ix}{i}$ | 12. | $\sum_{i=1}^N \frac{x^i}{i!}$ |

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 4.1. Какой алгоритм является алгоритмом циклической структуры?
- 4.2. Типы циклов в языке Си.
- 4.3. Какой цикл выполнится хотя бы один раз? А какой - ни разу?

5. Учебно-методическое и информационное обеспечение дисциплины

5.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. 1) Керниган, Б.В. Язык программирования С [Электронный ресурс] : учебник / Б.В. Керниган. — М.: Издательство «Лань», 2018. — 248 с. — Режим доступа: <https://e.lanbook.com/book/104883>. — Загл. с экрана.
2. Кудинов, Ю.И. Практикум по основам современной информатики [Электронный ресурс] : учебное пособие / Ю.И. Кудинов, Ф.Ф. Пащенко, А.Ю. Келина. — М.: Издательство «Лань», 2011. — 352 с. — Режим доступа: <https://e.lanbook.com/book/68471>. — Загл. с экрана.

Дополнительная литература:

3. Соловьянюк, Л.Г. Практикум решения задач на ЭВМ [Электронный ресурс] : учебное пособие / Л.Г. Соловьянюк. — Уфа : БГПУ имени М. Акмуллы, 2011. — 68 с. — Режим доступа: <https://e.lanbook.com/book/49519>. — Загл. с экрана.
4. Левкин, В.Е. NeoBook. Практикум по быстрому программированию с нуля : учебное пособие / В.Е. Левкин. - Москва ; Берлин : Директ-Медиа, 2018. - 191 с. : ил. - Библиогр.: с. 188

- ISBN 978-5-4475-9464-0 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=486254> (15.04.2019).
5. Соловьев, Н.А. Введение в программную инженерию : учебное пособие / Н.А. Соловьев, Л.А. Юркевская ; Министерство образования и науки Российской Федерации, Оренбургский Государственный Университет. - Оренбург : ОГУ, 2017. - 112 с. : схем., табл., ил. - Библиогр.: с. 83 - ISBN 978-5-7410-1685-5 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=481815> (15.04.2019).

1.2.Перечень ресурсов информационно-телекоммуникационной сети «Интернет» и программного обеспечения, необходимых для освоения дисциплины

- 1) <http://www.mcsme.ru> - сайт Московского центра непрерывного образования;
- 2) <http://www.etudes.ru> – научно-популярный сайт по математике;
- 3) <http://www.mathedu.ru> – сайт «Математическое образование: прошлое и настоящее»;
- 4) <http://www.math.ru>.
- 5) www.lib.bashedu.ru – сайт библиотеки БашГУ;
- 6) «Электронный читальный зал» (ЭБС «Библиотех»);
- 7) ЭБС «Университетская библиотека online» - www.biblioclub.ru;
- 8) ЭБС изд-ва «Лань» - www.e.lanbook.com;
- 9) <http://www.exponenta.ru> –образовательный математический сайт;

6. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
1	2	3
Аудитория 201	Лекции	Демонстрационное оборудование: доска, проектор – 1 шт., переносной экран – 1 шт. Специализированная мебель: столы, стулья (28 посадочных мест).
Аудитория 201	Практические занятия	Демонстрационное доска, проектор – 1 шт., переносной экран – 1 шт. Специализированная мебель: столы, стулья (28 посадочных мест).

Перечень специальных помещений и используемого лицензионного программного обеспечения представлен в справке о материально-техническом обеспечении ОП ВО по направлению подготовки 01.03.02 Прикладная математика и информатика (<http://www.sibsu.ru/sveden/education>).

МИНОБРНАУКИ РОССИИ
ФГБОУ ВО «УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»
СИБАЙСКИЙ ИНСТИТУТ (ФИЛИАЛ) УУНИТ
ЕСТЕСТВЕННО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

СОДЕРЖАНИЕ РАБОЧЕЙ ПРОГРАММЫ

дисциплины Практикум на ЭВМ на 2, 3, 4 семестры

очная форма обучения

Вид работы	Объем дисциплины		
	2 сем	3 сем	4 сем
Общая трудоемкость дисциплины (ЗЕТ / часов)	3 / 108	3 / 108	3 / 108
Учебных часов на контактную работу с преподавателем:			
Лекций	18	28	20
практических/ семинарских			
лабораторных	22	42	30
других (групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие работу обучающихся с преподавателем) (ФКР)	0,2	0,2	1,2
Учебных часов на самостоятельную работу обучающихся (СР)	67,8	37,8	2,8
Учебных часов на подготовку к экзамену/зачету/дифференцированному зачету (Контроль)			54

Форма(ы) контроля:

Зачет 2,3,4 семестры

2 семестр (1 курс, летняя сессия)								
№ п/п	Тема и содержание	Форма изучения материалов: лекции, практические занятия, семинарские занятия, лабораторные работы, самостоятельная работа и трудоемкость (в часах)				Основная и дополнительная литература, рекомендуемая студентам (номера из списка)	Задания по самостоятельной работе студентов	Форма текущего контроля успеваемости (коллоквиумы, контрольные работы, компьютерные тесты и т.п.)
		ЛК	ПР/СЕМ	ЛР	СР			
1	2	3	4	5	6	7	8	9
1.	Работа в интегрированной среде CodeBlocks на примере программ линейной структуры на языке программирования Си	2		2	7	1-7	– проработка лекций и работа с литературой по теме;	– опрос (тестирование) по теории; – проверка д.з.;
2.	Программы разветвленной структуры на языке программирования Си	2		2	7	1-7	– проработка лекций и работа с литературой по теме; – решение задач;	– опрос (тестирование) по теории; – проверка д.з.;
3.	Программы циклической структуры на языке программирования Си	2		2	7	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – контрольная работа; – сдача лабор. работ;
4.	Обработка статических массивов на языке программирования Си	2		2	7	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;

5.	Работа с символьными данными на языке программирования Си	2		2	7	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
6.	Функции и рекурсии на языке программирования Си	2		2	8	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
7.	Программные средства для работы с файлами на языке программирования Си	2		2	8	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
8.	Динамические массивы и динамические структуры данных на языке программирования Си	2		4	8	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
	Всего часов:	18		22	67,8			

3 семестр (2 курс, зимняя сессия)								
№ п/п	Тема и содержание	Форма изучения материалов: лекции, практические занятия, семинарские занятия, лабораторные работы, самостоятельная работа и трудоемкость (в часах)				Основная и дополнительная литература, рекомендуемая студентам (номера из списка)	Задания по самостоятельной работе студентов	Форма текущего контроля успеваемости (коллоквиумы, контрольные работы, компьютерные тесты и т.п.)
		ЛК	ПР/СЕМ	ЛР	СР			
1	2	3	4	5	6	7	8	9
1.	Введение. Знакомство с Python. Ввод-вывод данных. Команды print и input. Параметры sep и end.	2		6	5	1-7	– проработка лекций и работа с литературой по теме;	– опрос (тестирование) по теории; – проверка д.з.;
2.	Задачи целочисленной арифметики.	2		6	4	1-7	– проработка лекций и работа с литературой по теме; – решение задач;	– опрос (тестирование) по теории; – проверка д.з.;
3.	Условный оператор. Выбор из двух. Логические операции. Вложенные и каскадные условия.	2		6	4	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – контрольная работа; – сдача лабор. работ;
4.	Типы данных. Числовые типы данных: int, float. Модуль math. Строковый тип данных.	4		6	4	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;

5.	Циклы for и while. Цикл for. Цикл for: функция range. Частые сценарии. Цикл while. Цикл while: обработка цифр числа. Опции break, continue и else. Вложенные циклы.	4			4	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
6.	Строковый тип данных. Индексация. Срезы. Методы строк. Строки в памяти компьютера, таблица символов Unicode	4		6	4	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
7.	Списки. Введение в списки. Основы работы со списками. Методы списков. Вывод элементов списка. Методы строк: split, join. Списочные выражения. Сортировка списков.	4		4	4	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
8.	Функции. Функции без параметров. Функции с параметрами. Локальные и глобальные переменные. Функции с возвратом значения.	4		4	4,8	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
9	Работа над мини-проектами. Модуль random. Числовая угадайка. Магический шар. Генератор безопасных паролей. Шифр Цезаря. Калькулятор систем счисления.	2		4	4		– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
Всего часов:		28		42	37,8			

4 семестр (2 курс, летняя сессия)

№ n/n	Тема и содержание	Форма изучения материалов: лекции, практические занятия, семинарские занятия, лабораторные работы, самостоятельная работа и трудоемкость (в часах)				Основная и дополнительная литература, рекомендуемая студентам (номера из списка)	Задания по самостоятельной работе студентов	Форма текущего контроля успеваемости (коллоквиумы, контрольные работы, компьютерные тесты и т.п.)
		ЛК	ПР/СЕМ	ЛР	СР			
1	2	3	4	5	6	7	8	9
1.	Повторение основных конструкций языка Python . Тип данных bool и NoneType. Тип данных bool. Тип данных NoneType.	2		4		1-7	– проработка лекций и работа с литературой по теме;	– опрос (тестирование) по теории; – проверка д.з.;
2.	Вложенные списки. Матрицы. Математические операции над матрицами.	2		4		1-7	– проработка лекций и работа с литературой по теме; – решение задач;	– опрос (тестирование) по теории; – проверка д.з.;
3.	Кортежи. Введение в кортежи. Основы работы с кортежами.	2		4		1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – контрольная работа; – сдача лабор. работ;

4.	Множества. Множества в математике. Операции над множествами, диаграммы Эйлера-Венна. Введение в множества в Python. Основы работы с множествами. Методы множеств. Генераторы множеств и frozenset.	4		4		1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
5.	Словари. Введение в словари в Python. Основы работы со словарями. Методы словарей. Вложенные словари и генераторы словарей.	2		4	1	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
6.	Модули random и string. Модуль random.. Метод Монте-Карло и Bogosort. Модули decimal, fraction и complex. Тип данных complex. Модуль turtle. Модуль черепашки.	2		4	1	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
7.	Функции. Необязательные и именованные аргументы. Функции с переменным количеством аргументов.	2		2	0,8		– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
8.	Парадигмы программирования. Функции как объекты. Функции высшего порядка. Встроенные функции map(), filter(), reduce(). Анонимные	2		2			– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;

	функции. Встроенные функции any(), all(), zip(), enumerate().							
9.	Работа с файлами. Файловый ввод и вывод. Работа с текстовыми файлами.	2		2			– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лаборатор. работ;
	Всего часов:	20		30	2,8			

ФГБОУ ВО «УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»
 СИБАЙСКИЙ ИНСТИТУТ (ФИЛИАЛ) УУиТ
 ЕСТЕСТВЕННО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

СОДЕРЖАНИЕ РАБОЧЕЙ ПРОГРАММЫ

дисциплины Практикум на ЭВМ на 3, 4, 5 семестры

очно-заочная форма обучения

Вид работы	Объем дисциплины		
	3 сем	4 сем	5 сем
Общая трудоемкость дисциплины (ЗЕТ / часов)	2 / 72	5 / 180	2 / 72
Учебных часов на контактную работу с преподавателем:			
Лекций	14	22	16
практических/ семинарских			
лабораторных	16	32	20
других (групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие работу обучающихся с преподавателем) (ФКР)	0,2	0,2	1,2
Учебных часов на самостоятельную работу обучающихся (СР)	41,8	125,8	7,8
Учебных часов на подготовку к экзамену/зачету/дифференцированному зачету (Контроль)			27

Форма(ы) контроля:
 Зачет 3, 4, 5 семестры

2 семестр (1 курс, летняя сессия)								
№ п/п	Тема и содержание	Форма изучения материалов: лекции, практические занятия, семинарские занятия, лабораторные работы, самостоятельная работа и трудоемкость (в часах)				Основная и дополнительная литература, рекомендуемая студентам (номера из списка)	Задания по самостоятельной работе студентов	Форма текущего контроля успеваемости (коллоквиумы, контрольные работы, компьютерные тесты и т.п.)
		ЛК	ПР/СЕМ	ЛР	СР			
1	2	3	4	5	6	7	8	9
1.	Работа в интегрированной среде CodeBlocks на примере программ линейной структуры на языке программирования Си	2		2	7	1-7	– проработка лекций и работа с литературой по теме;	– опрос (тестирование) по теории; – проверка д.з.;
2.	Программы разветвленной структуры на языке программирования Си	2		2	7	1-7	– проработка лекций и работа с литературой по теме; – решение задач;	– опрос (тестирование) по теории; – проверка д.з.;
3.	Программы циклической структуры на языке программирования Си	2		2	7	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – контрольная работа; – сдача лабор. работ;
4.	Обработка статических массивов на языке программирования Си	2		2	7	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;

5.	Работа с символьными данными на языке программирования Си	2		2	7	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
6.	Функции и рекурсии на языке программирования Си	2		2	8	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
7.	Программные средства для работы с файлами на языке программирования Си	2		2	8	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
8.	Динамические массивы и динамические структуры данных на языке программирования Си	2		4	8	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
	Всего часов:	18		22	67,8			

3 семестр (2 курс, зимняя сессия)								
№ п/п	Тема и содержание	Форма изучения материалов: лекции, практические занятия, семинарские занятия, лабораторные работы, самостоятельная работа и трудоемкость (в часах)				Основная и дополнительная литература, рекомендуемая студентам (номера из списка)	Задания по самостоятельной работе студентов	Форма текущего контроля успеваемости (коллоквиумы, контрольные работы, компьютерные тесты и т.п.)
		ЛК	ПР/СЕМ	ЛР	СР			
1	2	3	4	5	6	7	8	9
1.	Введение. Знакомство с Python. Ввод-вывод данных. Команды print и input. Параметры sep и end.	2		4	4	1-7	– проработка лекций и работа с литературой по теме;	– опрос (тестирование) по теории; – проверка д.з.;
2.	Задачи целочисленной арифметики.	2		4	4	1-7	– проработка лекций и работа с литературой по теме; – решение задач;	– опрос (тестирование) по теории; – проверка д.з.;
3.	Условный оператор. Выбор из двух. Логические операции. Вложенные и каскадные условия.	2		6	4	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – контрольная работа; – сдача лабор. работ;
4.	Типы данных. Числовые типы данных: int, float. Модуль math. Строковый тип данных.	4		6	4	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;

5.	Циклы for и while. Цикл for. Цикл for: функция range. Частые сценарии. Цикл while. Цикл while: обработка цифр числа. Опции break, continue и else. Вложенные циклы.	4			4	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
6.	Строковый тип данных. Индексация. Срезы. Методы строк. Строки в памяти компьютера, таблица символов Unicode	4		6	4	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
7.	Списки. Введение в списки. Основы работы со списками. Методы списков. Вывод элементов списка. Методы строк: split, join. Списочные выражения. Сортировка списков.	4		6	5	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
8.	Функции. Функции без параметров. Функции с параметрами. Локальные и глобальные переменные. Функции с возвратом значения.	4		6	4,8	1-7	– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
9	Работа над мини-проектами. Модуль random. Числовая угадайка. Магический шар. Генератор безопасных паролей. Шифр Цезаря. Калькулятор систем счисления.	2		4	4		– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
Всего часов:		28		42	37,8			

4 семестр (2 курс, летняя сессия)

№ n/n	Тема и содержание	Форма изучения материалов: лекции, практические занятия, семинарские занятия, лабораторные работы, самостоятельная работа и трудоемкость (в часах)				Основная и дополнительная литература, рекомендуемая студентам (номера из списка)	Задания по самостоятельной работе студентов	Форма текущего контроля успеваемости (коллоквиумы, контрольные работы, компьютерные тесты и т.п.)
		ЛК	ПР/СЕМ	ЛР	СР			
1	2	3	4	5	6	7	8	9
1.	Повторение основных конструкций языка Python . Тип данных bool и NoneType. Тип данных bool. Тип данных NoneType.	2		4		1-7	– проработка лекций и работа с литературой по теме;	– опрос (тестирование) по теории; – проверка д.з.;
2.	Вложенные списки. Матрицы. Математические операции над матрицами.	2		4		1-7	– проработка лекций и работа с литературой по теме; – решение задач;	– опрос (тестирование) по теории; – проверка д.з.;
3.	Кортежи. Введение в кортежи. Основы работы с кортежами.	2		4		1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – контрольная работа; – сдача лабор. работ;

4.	Множества. Множества в математике. Операции над множествами, диаграммы Эйлера-Венна. Введение в множества в Python. Основы работы с множествами. Методы множеств. Генераторы множеств и frozenset.	2		4		1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
5.	Словари. Введение в словари в Python. Основы работы со словарями. Методы словарей. Вложенные словари и генераторы словарей.	2		4	1	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
6.	Модули random и string. Модуль random.. Метод Монте-Карло и Bogosort. Модули decimal, fraction и complex. Тип данных complex. Модуль turtle. Модуль черепашки.	4		4	1,8	1-7	– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
7.	Функции. Необязательные и именованные аргументы. Функции с переменным количеством аргументов.	2		2			– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;
8.	Парадигмы программирования. Функции как объекты. Функции высшего порядка. Встроенные функции map(), filter(), reduce(). Анонимные	2		2			– проработка лекций и работа с литературой по теме; – выполнение лабор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лабор. работ;

	функции. Встроенные функции any(), all(), zip(), enumerate().							
9.	Работа с файлами. Файловый ввод и вывод. Работа с текстовыми файлами.	2		2			– проработка лекций и работа с литературой по теме; – выполнение лаборатор. работ;	– опрос (тестирование) по теории; – проверка д.з.; – сдача лаборатор. работ;
	Всего часов:	20		30	2,8			

Рейтинг-план дисциплины

Виды учебной деятельности студентов	Балл за конкретное задание	Число заданий за семестр	Баллы	
			Минимальный	Максимальный
Модуль 1 (Разделы 1. 2 по РПД)				
Текущий контроль			12	20
1. Работа на занятиях	2	15	12	20
Рубежный контроль				
1.Контрольная работа	3	5	10	15
Модуль 2 (Разделы 3, 4 по РПД)				
Текущий контроль			13	20
1. Работа на занятиях	4	5	13	20
Рубежный контроль				
1.Контрольная работа	3	5	10	15
Поощрительные баллы				
1. Выполнение заданий повышенной трудности	2	5	0	10
Посещаемость (баллы вычитаются из общей суммы набранных баллов)				
Посещение лекционных и практ. занятий			-7	0
Итоговый контроль				
1.Экзамен			0	30
ИТОГО			45	110