

Министерство образования и науки Российской Федерации  
 Федеральное государственное автономное образовательное  
 учреждение высшего образования  
 «СЕВЕРО-ВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ  
 имени М. К. АММОСОВА»  
 Институт математики и информатики  
 Кафедра информационных технологий

УТВЕРЖДАЮ  
 Директор ИМИ

\_\_\_\_\_/ В. И. Афанасьева /  
 «\_\_» \_\_\_\_\_ 20\_\_ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

**Б1.В.ОД.3 – Программирование**

для программы бакалавриата  
 по направлению подготовки  
 09.03.01 – Информатика и вычислительная техника

Автор: Павлов А.В., к.ф.-м.н., доцент кафедры информационных технологий ИМИ,  
 av.pavlov@s-vfu.ru

ОДОБРЕНО Заведующий кафедрой разработчика _____/_____/	ОДОБРЕНО Заведующий выпускаю- щей кафедрой ИТ _____/_____/	РЕКОМЕНДОВАНО Нормоконтроль в составе ОП пройден _____/_____/
Протокол № __ от «__» _____ 20__ г.	Протокол № __ от «__» _____ 20__ г.	Протокол № __ от «__» _____ 20__ г.
Рекомендовано к утверждению в составе ОП 09.03.01 «Информатика и и вычислительная техника» Председатель УМК ИМИ _____/И. В. Николаева/ Протокол УМК № _____ от «__» _____ 20__ г.		Эксперт УМК ИМИ _____/_____/
		«__» _____ 20__ г.

# 1. АННОТАЦИЯ

к рабочей программе дисциплины

**Б1.В.ОД.3 – Программирование**

Трудоемкость 10 з. е.

## 1.1. Цель освоения и краткое содержание дисциплины

Целью изучения дисциплины «Программирование» является: овладение основами программирования на языках Питон (Python) и Си/Си++.

*Краткое содержание дисциплины.* Основы синтаксиса и семантики императивных языков высокого уровня Python и C++. Переменные, типы, выражения и присваивание. Условные и циклические структуры. Текстовый ввод/вывод. Функции и передача параметров. Структурированная декомпозиция.

## 1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Таблица 1. *Перечень планируемых результатов обучения*

Планируемые результаты освоения программы (содержание и коды компетенций)	Планируемые результаты обучения по дисциплине
ОПК-2 : способностью осваивать методики использования программных средств для решения практических задач, ПК-1 : способностью разрабатывать модели компонентов информационных систем, включая модели баз данных и модели и интерфейсов «человек – электронно-вычислительная машина»	В результате изучения дисциплины обучающийся должен: <u>знать:</u> – способы передачи параметров в функцию в языке Си++; – несколько методов сортировки, включая быструю сортировку; <u>уметь:</u> – анализировать, объяснять поведение, модифицировать, тестировать небольшие программы на языках Питон и Си++, использующие любую комбинацию следующих понятий: простые вычисления, ветвление, итерация, простой ввод-вывод, в том числе в/из текстовых файлов, массивы (списки), функции; – сравнивать два алгоритма для решения несложной задачи; <u>владеть навыками:</u> – отладки программ на языке Си++ в современной инструментальной среде; – написания небольших программ на языках Питон и Си++ с использованием условного оператора и различных видов циклов;

### 1.3. Место дисциплины в структуре образовательной программы

Таблица 2. Содержательно-логические связи дисциплины

Индекс дисциплины	Наименование дисциплины	Коды учебных дисциплин, практик	
		на которые опирается содержание дисциплины	для которых содержание дисциплины выступает опорой
Б1.В.ОД.3	Программирование	—	Б1.Б.23 – Операционные системы, Б1.В.ОД.1 – Компьютерные сети и телекоммуникации, Б1.В.ОД.4 – Структуры и алгоритмы обработки данных, Б1.В.ОД.5 – Web-программирование, Б1.В.ОД.6 – Программирование .NET, Б1.В.ОД.7 – Объектно-ориентированное программирование, Б1.В.ОД.8 – Языки программирования и методы трансляции

### 1.4. Язык преподавания

Русский.

**2. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся**

Таблица 3. Выписка из учебного плана

Код и название дисциплины по учебному плану	Б1.В.ОД.3 – Программирование	
Курс изучения	1	
Семестр(ы) изучения	1, 2	
Форма промежуточной аттестации (зачет/экзамен)	экзамен / экзамен	
Курсовой проект / курсовая работа (указать вид работы при наличии в учебном плане), семестр выполнения		
Трудоемкость (в ЗЕТ)	5 / 5 (10)	
<b>Трудоемкость (в часах)</b> (сумма строк №1, 2, 3), в т. ч.:	180 / 180	
<b>№ 1. Контактная работа обучающихся с преподавателем (КР), в часах:</b>	Объем аудиторной работы, в часах	В т. ч. с применением ДОТ или ЭО, в часах
Объем работы (в часах) (1.1.+1.2.+1.3.)	75 / 79	
1.1. Занятия лекционного типа (лекции)	17 / 18	
1.2. Занятия семинарского типа, всего, в т.ч.:		
- семинары (практические занятия, коллоквиумы и т. п.)	17 / 18	
- лабораторные работы	34 / 36	
- практикумы		
1.3. КСР (контроль самостоятельной работы, консультации)	7 / 7	
<b>№ 2. Самостоятельная работа обучающихся (СРС) (в часах)</b>	69 / 65	
<b>№ 3. Количество часов на экзамен (при наличии экзамена в учебном плане)</b>	36 / 36	

### 3. Содержание дисциплины, структурированное по темам с указанием отведенного на них количества академических часов и видов учебных занятий

#### 3.1. Распределение часов по темам и видам учебных занятий

Таблица 4

Тема	Всего часов	Контактная работа, в часах									Часы СРС
		Лекции	из них с прим-м ЭО и ДОТ	Семинары (практические занятия, коллоквиумы)	из них с прим-м ЭО и ДОТ	Лабораторные работы	из них с прим-м ЭО и ДОТ	Практикумы	из них с прим-м ЭО и ДОТ	КСР (консультации)	
Тема 1. Язык Питон: простые типы и основы синтаксиса.	54	7	0	7	0	14	0	0	0	2	24
Тема 2. Язык Питон: структурированные типы.	56	6	0	6	0	12	0	0	0	2	30
Тема 3. Язык Питон: функции и рекурсия.	33	4	0	4	0	8	0	0	0	2	15
Тема 4. Язык Си: простые типы и основы синтаксиса.	32	4	0	4	0	8	0	0	0	2	14
Тема 5. Язык Си/Си++: структурированные типы.	30	4	0	4	0	7	0	0	0	2	13
Тема 6. Язык Си: функции и рекурсия.	30	4	0	4	0	7	0	0	0	2	13
Тема 7. Алгоритмы сортировки.	27	3	0	3	0	7	0	0	0	1	13
Тема 8. Язык Си++: классы. Знакомство с STL.	26	3	0	3	0	7	0	0	0	1	12
<b>ВСЕГО ЧАСОВ</b>	<b>288</b>	<b>35</b>	<b>0</b>	<b>35</b>	<b>0</b>	<b>70</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>14</b>	<b>134</b>

#### 3.2. Содержание тем программы дисциплины

##### Тема 1. Язык Питон: простые типы и основы синтаксиса.

Что такое программирование? Компилируемые и интерпретируемые языки. Статическая и динамическая типизация. Знакомство с Питоном. Среда IDLE. Окно интерпретатора. REPL-цикл. Написание, редактирование и запуск программ. Числовые типы. Логический тип. Арифметические и логические операторы. Оператор присваивания. Модуль math. Программы линейной структуры. Блок-схемы. Операторы ветвления и цикла, роль отступов. Строковый тип. Перебор строк. Ввод с клавиатуры. Простое преобразование типов. Встроенные средства документации, функция help(). Функции dir().

##### Тема 2. Язык Питон: структурированные типы.

Кортежи и списки в Питоне. Индексация, нотация диапазонов. Поиск минимального/максимального элемента. Строки. Unicode, Python 2 и Python 3. Словари. Семантика присваивания для сложных типов. Двумерные массивы как списки: индексация, ввод, заполне-

ние случайными значениями. Перебор списков. Перебор словаря, методы keys(), values(), items().

**Тема 3. Язык Питон: функции и рекурсия.**

Функции. Видимость переменных. Передача параметров. Значения параметров по умолчанию. Рекурсия. Стандартная библиотека, менеджер пакетов pip.

**Тема 4. Язык Си: простые типы и основы синтаксиса.**

Язык Си. Функция main(). Числовые типы. Объявления переменных. Компиляция программы на Си, заголовочные и библиотечные файлы. Стандартная библиотека. Логический тип. Арифметические и логические операторы. Оператор присваивания. Неявное преобразование типов. Блоки. Операторы ветвления и цикла, оператор выбора. Вычисление сумм и произведений. Символьный тип, символьные константы, коды символов. Unicode и широкий символьный тип. Ввод-вывод с текстовыми файлами.

**Тема 5. Язык Си/Си++: структурированные типы.**

Указатели, типизированные и нетипизированные. Разыменование. Средства отладки Visual Studio. Массивы. Строки. Структуры. Перечисления. malloc() и new().

**Тема 6. Язык Си: функции и рекурсия.**

Функции. Передача параметров по ссылке и по значению. Декомпозиция задачи «сверху вниз». Видимость переменных. Статические переменные. Глобальные переменные.

**Тема 7. Алгоритмы сортировки.**

Сортировка пузырьком. Сортировка вставками. Сортировка слиянием. Быстрая сортировка.

**Тема 8. Язык Си++: классы. Знакомство с STL.**

Перегрузка функций и перегрузка операторов. Классы и объекты; конструкторы и деструкторы. Методы. Классы потокового ввода-вывода. Некоторые контейнерные классы STL.

### **3.3. Формы и методы проведения занятий, применяемые учебные технологии**

При проведении занятий и организации СРС используются традиционные технологии обучающего обучения, предполагающие передачу информации в готовом виде: проведение лекционных занятий, самостоятельная работа с источниками. Предусмотрено использование активных и интерактивных форм обучения с целью формирования и развития профессиональных навыков студентов — выполнение лабораторных работ, подразумевающих применение компьютерных технологий.

## **4. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине**

№	Наименование раздела (темы) дисциплины	Вид СРС	Трудоемкость (в часах)	Формы и методы контроля
1	Язык Питон: простые типы и основы синтаксиса.	Решение задач	24	Сдача домашних заданий лично, неоконченных лаб. работ
2	Язык Питон: структурированные типы.	Решение задач	30	Сдача домашних заданий лично, неоконченных лаб. работ
3	Язык Питон: функции и рекурсия.	Решение задач	15	Сдача домашних заданий лично, неоконченных лаб. работ

№	Наименование раздела (темы) дисциплины	Вид СРС	Трудо- емкость (в часах)	Формы и методы контроля
4	Язык Си: простые типы и основы синтаксиса.	Решение задач	14	Сдача домашних заданий лично, неоконченных лаб. работ
5	Язык Си/Си++: структурированные типы.	Решение задач	13	Сдача домашних заданий лично, неоконченных лаб. работ
6	Язык Си: функции и рекурсия.	Решение задач	13	Сдача домашних заданий лично, неоконченных лаб. работ
7	Алгоритмы сортировки.	Решение задач	13	Сдача домашних заданий лично, неоконченных лаб. работ
8	Язык Си++: классы. Знакомство с STL.	Решение задач	12	Сдача домашних заданий лично, неоконченных лаб. работ
	ИТОГО		134	

## 5. Методические указания для обучающихся по освоению дисциплины

Важное значение в освоении дисциплины имеет самостоятельная работа. Ключевым ее видом является самостоятельное написание программ. Только самостоятельное практическое написание программ, поиск и исправление ошибок в них могут обеспечить действительное понимание тем курса.

Последовательное и добросовестное изучение курса является основой для выработки навыков алгоритмизации, чтения и отладки текстов программ, ключевых для данного направления подготовки.

### Рейтинговый регламент по дисциплине

Вид выполняемой учебной работы (контролирующие мероприятия)	Кол-во баллов (min)	Кол-во баллов (max)
Посещение занятий	6	10
Лабораторные работы	15	24
Домашние задания	11	17
Контрольные тесты	13	19
<b>Кол-во баллов для допуска к экзамену в 1 сем. (min–max)</b>	<b>45</b>	<b>70</b>
Посещение занятий	6	10
Лабораторные работы	12	18
Домашние задания	16	25
Контрольные тесты	11	17
<b>Кол-во баллов для допуска к экзамену во 2 сем. (min–max)</b>	<b>45</b>	<b>70</b>

## 6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

### 6.1. Показатели, критерии и шкала оценивания

Коды оцениваемых компетенций	Показатель оценивания (дескриптор) (по п.1.2)	Уровни освоения	Критерий оценивания	Оценка
ОПК-2, ПК-1	<p><u>знать:</u> способы передачи параметров в функцию в языке Си++; несколько методов сортировки, включая быструю сортировку;</p> <p><u>уметь:</u> анализировать, объяснять поведение, модифицировать, тестировать небольшие программы на языках Питон и Си++, использующие любую комбинацию следующих понятий: простые вычисления, ветвление, итерация, простой ввод-вывод, в том числе в/из текстовых файлов, массивы (списки), функции; сравнивать два алгоритма для решения несложной задачи;</p> <p><u>владеть навыками:</u> отладки программ на языке Си++ в современной инструментальной среде; написания небольших программ на языках Питон и Си++ с использованием условного оператора и различных видов циклов;</p>	высокий	<p>может правильно написать на языке Си функцию, изменяющую аргументы, переданные по ссылке, например заполняющую структуру;</p> <p>может объяснить механизм работы оператора присваивания для кортежей, списков, словарей в языке Питон;</p> <p>может реализовать сортировку пузырьком и быструю сортировку на языке Си, объяснить преимущество последней;</p> <p>может найти ошибку типа «ошибка на единицу» в некорректно реализованных на языках Си или Питон функциях сортировки списка или массива, агрегации списка или массива, обработки строки;</p> <p>может найти ошибку в циклической программе, используя отладку в инструментальной среде;</p> <p>может написать, запустить, протестировать и отладить на языках Си или Питон программу, вычисляющую агрегирующую функцию от строки или числового массива или списка, в том числе рекурсивно;</p> <p>может объяснить, какую агрегирующую функцию</p>	отлично



Коды оцениваемых компетенций	Показатель оценивания (дескриптор) (по п.1.2)	Уровни освоения	Критерий оценивания	Оценка
			<p>от числового массива или списка вычисляет данная функция (в том числе рекурсивная), в пределах 20 строк;</p> <p>имея документацию, может написать вызов на языках Си и Питон библиотечной функции для конкретной задачи, правильно организовав подключение нужных пакетов/заголовочных файлов и передав параметры;</p> <p>может построить пример ввода, на котором предьявленная дефектная реализация простой программы обработки массива/списка на языках Си или Питон дает неверный результат;</p> <p>может пользоваться для отладки программы на Си++: точками останова, в том числе условными, пошаговым исполнением, шагом с заходом и с обходом по вызовам функций, просмотром значений переменных и памяти остановленной программы.</p>	
		базовый	<p>может объяснить, почему изменяются значения переменных в вызывающей функции при передаче по ссылке на языке Си;</p> <p>может реализовать сортировку пузырьком на языке Си;</p> <p>может найти ошибку типа «ошибка на единицу» в некорректно реализованных на языках Си или Питон функциях сортировки списка или массива,</p>	хорошо

Коды оцениваемых компетенций	Показатель оценивания (дескриптор) (по п.1.2)	Уровни освоения	Критерий оценивания	Оценка
			<p>агрегации списка или массива, обработки строки; может написать, запустить, протестировать и отладить на языках Си или Питон программу, вычисляющую агрегирующую функцию от числового массива или списка;</p> <p>может объяснить, какую агрегирующую функцию от числового массива или списка вычисляет данная функция, в пределах 15 строк;</p> <p>имея документацию и примеры использования, может написать вызов на языках Си и Питон библиотечной функции для конкретной задачи, правильно организовав подключение нужных пакетов/заголовочных файлов и передав параметры;</p> <p>может вручную трассировать по тексту несложные циклические программы в пределах 4 итераций и правильно находить значения переменных после их завершения при различных входных данных;</p> <p>может пользоваться для отладки программы на Си++: точками останова, пошаговым исполнением, шагом с заходом и с обходом по вызовам функций, просмотром значений переменных остановленной программы.</p>	

Коды оцениваемых компетенций	Показатель оценивания (дескриптор) (по п.1.2)	Уровни освоения	Критерий оценивания	Оценка
		минимальный	<p>может объяснить, почему изменяются значения переменных в вызывающей функции при передаче по ссылке;</p> <p>может написать, запустить, протестировать и отладить на языках Си или Питон программу, вычисляющую минимум или максимум чисел, удовлетворяющих некоторому условию, из числового массива или списка;</p> <p>может объяснить, какую агрегирующую функцию от числового массива или списка вычисляет данная функция, в пределах 15 строк;</p> <p>может вручную трассировать по тексту несложные циклические программы в пределах 4 итераций и правильно находить значения переменных после их завершения при различных входных данных;</p> <p>может пользоваться для отладки программы на Си++: точками останова, пошаговым исполнением, просмотром значений переменных остановленной программы.</p>	удовл.
		не освоены	не способен выполнить три или более пунктов из вышеперечисленного	не зачтено

## 6.2. Типовые контрольные задания (вопросы) для промежуточной аттестации

Коды оцениваемых компетенций	Оцениваемый показатель (ЗУВ)	Тема	Образец типового (тестового или практического) задания (вопроса)
ОПК-2, ПК-1	знать способы передачи параметров в функцию в языке Си++;	6	<p>Какие значения будут напечатаны данной программой:</p> <pre>void f(int x) {     printf("%d",x); x++;} void g(int &amp;x) {     printf("%d",x); x++;} int main(){     int x = 1; f(x); g(x);     return 0;}</pre>
ОПК-2, ПК-1	знать несколько методов сортировки, включая быструю сортировку;	7	<p>Сколько раз будет выполнена функция swap() для массива <math>A = \{3, 2, 5, 4, 1\}</math> при следующей реализации сортировки пузырьком:</p> <pre>for (i=0; i&lt;N; ++i)     for (j=i; j&lt;N; ++j)         if (a[i]&gt;a[j])             swap(A, i, j);</pre> <p>Как можно улучшить эту реализацию?</p>
ОПК-2, ПК-1	уметь анализировать, объяснять поведение, модифицировать, тестировать и отлаживать небольшие программы на языках Питон и Си++, использующие любую комбинацию следующих понятий: простые вычисления, ветвление, итерация, простой ввод-вывод, в том числе в/из текстовых файлов, массивы (списки), функции;	1, 4	<p>Даны действительные числа <math>a, b, c</math>, причем <math>a \neq 0</math>. Даны ординаты <math>y_1, y_2, y_3</math> точек <math>L, M, N</math> на прямой <math>ax + by + c = 0</math>. Вывести координаты точек <math>L, M, N</math> в порядке слева направо (или сверху вниз, если прямая вертикальна).</p>

Коды оцениваемых компетенций	Оцениваемый показатель (ЗУВ)	Тема	Образец типового (тестового или практического) задания (вопроса)
ОПК-2, ПК-1	уметь анализировать, объяснять поведение, модифицировать, тестировать и отлаживать небольшие программы на языках Питон и Си++, использующие любую комбинацию следующих понятий: простые вычисления, ветвление, итерация, простой ввод-вывод, в том числе в/из текстовых файлов, массивы (списки), функции;	2	Приведите пример списка L из трех чисел, для которого данный код напечатает 2: <pre>i, M, iM = 0, 0, -1 for x in L:     if x%2 == 1:         if x &gt; M:             M = x             iM = i     i += 1 if iM &gt;= 0:     print(iM)</pre>
ОПК-2, ПК-1	анализировать, объяснять поведение, модифицировать, тестировать небольшие программы на языках Питон и Си++, использующие любую комбинацию следующих понятий: простые вычисления, ветвление, итерация, простой ввод-вывод, в том числе в/из текстовых файлов, массивы (списки), функции;	3	Имеется некорректно написанная рекурсивная функция, подсчитывающая количество вхождений числа x в список A: <pre>def occurs(x, A):     if A[0] == x:         return 1+occurs(x, A[1:])     return occurs(x, A[1:])</pre> Исправьте ее
ОПК-2, ПК-1	анализировать, объяснять поведение, модифицировать, тестировать небольшие программы на языках Питон и Си++, использующие любую комбинацию следующих понятий: простые вычисления, ветвление, итерация, простой ввод-вывод, в том числе в/из текстовых файлов, массивы (списки), функции;	6	Имеется некорректно написанная рекурсивная функция, подсчитывающая количество вхождений числа x в массив A длины n: <pre>int occurs(int x, int n, int* A) {     if (A[0] == x)         return 1+occurs(x, n-1, A+1);     return occurs(x, n-1, A+1); }</pre> Исправьте ее

Коды оцениваемых компетенций	Оцениваемый показатель (ЗУВ)	Тема	Образец типового (тестового или практического) задания (вопроса)
ОПК-2, ПК-1	владеть навыками отладки программ на языке Си++ в современной инструментальной среде;	4	<p>Имеется программа нахождения трех максимальных членов числовой последовательности, предполагается, что среди чисел могут быть одинаковые.</p> <pre>#include &lt;stdio.h&gt; int main() {     int i, n, x, max1, max2, max3;     scanf("%d", &amp;n);     if (n &lt; 3) return -1;     scanf("%d", &amp;x);     max1 = max2 = max3 = x;     for (i=1; i&lt;n; ++i){         scanf(" %d", &amp;x);         if (x &gt; max1) max1 = x; else         if (x &gt;= max2) max2 = x; else         if (x &gt;= max3) max3 = x;     }     printf("max=%d,max2=%d,max3=%d",         max1, max2, max3);     return 0; }</pre> <p>Приведите пример входных данных, для которых она работает правильно, и пример входных данных, для которых она работает неправильно. Исправьте программу.</p>
ОПК-2, ПК-1	владеть навыками написания небольших программ на языках Питон и Си++ с использованием условного оператора и различных видов циклов;	1, 4	<p>Натуральное число называется совершенным, если оно равно сумме всех своих делителей, включая единицу, но исключая себя. Например, число 28 — совершенное:</p> $1 + 2 + 4 + 7 + 14 = 28.$ <p>Напечатать все совершенные числа, меньшие заданного с клавиатуры числа <math>N</math>.</p>

Коды оцениваемых компетенций	Оцениваемый показатель (ЗУВ)	Тема	Образец типового (тестового или практического) задания (вопроса)
ОПК-2, ПК-1	владеть навыками написания небольших программ на языках Питон и Си++ с использованием условного оператора и различных видов циклов;	2, 5	Ввести строку. Гарантируется, что в ней только пробелы и латинские буквы. Подсчитать число слов в строке.
ОПК-2, ПК-1	владеть навыками написания небольших программ на языках Питон и Си++ с использованием условного оператора и различных видов циклов;	2	Ввести с клавиатуры положительное целое $n$ , а затем $n$ целых чисел. Найти любое наиболее часто встречающееся число, используя словарь.
ОПК-2, ПК-1	уметь сравнивать два алгоритма для решения несложной задачи;	5	<p>Сравните два алгоритма для подсчета количества различных чисел в массиве <math>A</math> длины <math>n</math> неотрицательных чисел, меньших <math>N</math>:</p> <pre> const int N = ...; int uniq1(int n, int *A) {     int i, j, res = 0;     for (i=0; i &lt; n; ++i){         bool last = true;         for (j=i+1; j &lt; n; ++j)             if (A[i]==A[j]) last=false;         if (last) res++;     }     return res; } int uniq2(int n, int *A) {     int i, res = 0;     bool* h = new bool[N];     for (i=0; i&lt;N; ++i) h[i] = 0;     for (i=0; i&lt;n; ++i) h[A[i]] = 1;     for (i=0; i&lt;n; ++i) res += h[i];     return res; } </pre> <p>Приведите примеры, когда каждый из алгоритмов совершает меньше действий, чем второй. При каких условиях лучше тот или другой?</p>

Коды оцениваемых компетенций	Оцениваемый показатель (ЗУВ)	Тема	Образец типового (тестового или практического) задания (вопроса)
ОПК-2, ПК-1	владеть навыками написания небольших программ на языках Питон и Си++ с использованием условного оператора и различных видов циклов;	8	Со стандартного ввода вводится число $n > 0$ , затем $n$ целых чисел. Найдите максимум отрицательных чисел, с обязательным использованием следующих элементов: потокового ввода ( <code>&lt;iostream&gt;</code> ), векторов и итераторов ( <code>&lt;vector&gt;</code> ). Если отрицательные числа не вводились, выведите "ERROR".

### Вопросы к экзамену за I семестр

1. Типы в языке Питон. Динамическая типизация. `int`, `float`, `bool`, `str`, `NoneType`. Явное преобразование типов. Функции `type()`, `dir()`, `help()`.
2. Числовые типы `int`, `bool`, `float`. Преобразование числовых типов друг в друга. Числовые операторы, в том числе степень, деление нацело, остаток по модулю, побитовые сдвиги влево и вправо.
3. Логический тип `bool`. Преобразование числовых, строковых значений и `None` в `bool`. Логические операторы.
4. Операторы `if`, `while`. Отступы и блоки. Пример.
5. Оператор `while`. Нахождение НОД двух положительных целых чисел.
6. Кортежи, присваивание и распаковка кортежей. Списки и строки. Присваивание списков. Изменяемые и неизменяемые типы. Функция `append()`. Длина. Индексация, отрицательные индексы, диапазоны, шаг. Пропуск начального, конечного значений диапазона индексов. Оператор `in`. Сложение и умножение на число для кортежей, списков и строк.
7. Списки. Присваивание списков. Функция `append()`. Индексация, длина, отрицательные индексы, диапазоны, шаг. Пропуск начального, конечного значений диапазона индексов. Ввод числового списка с клавиатуры. Перечисления, отбор значений, примеры.
8. Списки. Присваивание списков. Ввод числового списка с клавиатуры. Порождение списка случайных чисел заданной длины. Нахождение максимума, минимума и среднего арифметического для числового списка.
9. Оператор `for`. Объекты `range`. Начальное значение, шаг. Перебор символов строки. Перебор строк файла.
10. Двумерные массивы в виде списков. Инициализация нулевой матрицы  $m \times n$ . Ввод матрицы с клавиатуры,  $m$  строк по  $n$  значений в строке.
11. Функции. Параметры. Видимость переменных, глобальные и локальные переменные. Примеры.
12. Словари. Допустимые типы ключей. Проверка наличия ключа в словаре. Перебор словаря, функция `items()`.
13. Рекурсия. Примеры: рекурсивное вычисления максимума, минимума, факториала, чисел Фибоначчи; печать ключей бинарного дерева, заданного в виде словаря.
14. Пузырьковая сортировка. Количество действий для массива длины  $n$ .
15. Быстрая сортировка. Упорядочение числового массива на три части относительно выбранного срединного значения.



## Вопросы к экзамену за II семестр

1. Состав языка Си. Алфавит языка. Идентификаторы. Ключевые слова. Комментарии.
2. Типы данных. Стандартные типы (целые, со знаком и без, вещественные, `char`, `bool`).
3. Пользовательские типы данных. Переименование типов. Перечисления (`enum`), структуры (`struct`), объединения (`union`).
4. Операции. Операторы. Составной оператор.
5. Алгебраические выражения, их типы и правила вычисления. Функции стандартной библиотеки из `<math.h>`.
6. Логические выражения, их типы и правила вычисления.
7. Инициализация переменных (стандартных, строк, массивов).
8. Условные операторы. Оператор безусловного перехода. Оператор выбора.
9. Программы циклической структуры. Оператор цикла с параметром (`for`).
10. Программы циклической структуры. Оператор цикла с предусловием (`while`).
11. Программы циклической структуры. Оператор цикла с постусловием `do-while` (`repeat`).
12. Структурированные типы данных. Массивы. Описание типа. Действия над массивами. Действия над элементами массива.
13. Функции стандартной библиотеки для работы со строками и символами.
14. Динамические структуры. Очереди.
15. Текстовые файлы.
16. Бинарные файлы.
17. Объектно-ориентированное программирование. Классы. Наследование. Конструкторы и деструкторы.
18. Перегрузка операций.
19. Перегрузка функций, шаблоны функций.
20. Динамические структуры. Дерево.
21. Рекуррентные формулы. Вычисление конечных сумм и произведений.
22. Рекуррентные формулы. Приближенное Вычисление бесконечных сумм и произведений.
23. Перевод целого числа из десятичной системы счисления (СС) в десятичную СС. Перевод целого числа из десятичной системы счисления в  $p$ -ичную СС.
24. Решето Эратосфена.
25. Сортировка методом обмена (пузырька).
26. Сортировка методом выбора.
27. Сортировка методом вставки.
28. Быстрая сортировка Хоара.
29. Задачи поиска. Линейный поиск. Бинарный поиск.
30. Реализация динамических структур с помощью массивов. Стеки.
31. Обратная польская запись.
32. Схема Горнера. Перевод числа из  $p$ -ичной системы счисления в десятичную.
33. Определение площади произвольного  $n$ -угольника (без самопересечений).
34. Перевод целого числа из двоичной СС в десятичную СС.
35. Перевод целого числа из десятичной СС в  $p$ -ичную СС.
36. Перевод целого числа из  $p$ -ичной СС в десятичную СС.
37. Задачи поиска. Бинарный поиск.
38. Задача о восьми ферзях.
39. Схема Горнера. Вычисление значения многочлена  $n$ -й степени.

### 6.3. Методические материалы, определяющие процедуры оценивания

*Лабораторные работы.* Во время лабораторных занятий по каждой теме обучающиеся должны самостоятельно написать программы для задач в описании лабораторной работы. Решения сдаются лично преподавателю или отправляются через веб-браузер в автоматизированную проверяющую систему «Мультиметр». Задания очередной лабораторной работы могут быть сданы не позднее следующего лабораторного занятия.

*Домашние задания.* Домашние задания выполняются по индивидуальным вариантам. Срок сдачи очередного комплекта домашних заданий — как правило, не позднее двух недель после его выдачи. Каждое задание сдается преподавателю лично.

*Онлайн-тестирование.* Данная форма текущего контроля направлена на оценку основных теоретических знаний обучающегося по мере освоения разделов дисциплины. Предполагает ответы на вопросы теста через веб-формы, например Google Forms, либо исправление и сдачу выданных «заготовок» (незаконченных программ или программ с дефектами) в автоматизированной проверяющей системе «Мультиметр».

*Форма промежуточной аттестации: экзамен.* К экзамену допускаются студенты, выполнившие обязательный минимум учебной работы и набравшие в семестре не менее 45 баллов. Данный вид комплексного испытания предполагает ответ по билету, содержащему один теоретический и два практических вопроса. Последние предполагают написание программы. На экзамене можно набрать до 30 баллов: 9 за теоретический вопрос, 9 и 12 баллов за практические вопросы.

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

*Перечень литературы*

№	Автор, название, место издания, издательство, год издания учебной литературы, вид и характеристика иных информационных ресурсов	Наличие грифа, вид грифа	НБ СВФУ, кафедральная библиотека и кол-во экземпляров	Электронные издания: точка доступа к ресурсу (наименование ЭБС, ЭБ СВФУ)
<b>Основная литература</b>				
1	Мозговой М. В. Классика программирования: алгоритмы, языки, автоматы, компиляторы: практический подход. СПб: Наука и Техника, 2006. 320 с.		10	
2	Павловская Т. А. С/С++. СПб.: Питер, 2013.		13	
<b>Дополнительная литература</b>				
1	Кормен Т. Х. Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: МЦНМО, 1999.		1	
2	Кнут, Д. Искусство программирования Т. 1: Основные алгоритмы. М. 2004.		3	
3	Страуструп Б., Язык программирования С++. М.: Бином, 1999.		1	

## 8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее — сеть Интернет), необходимых для освоения дисциплины

1. Шень А. Х. Программирование: теоремы и задачи. М.: МЦНМО, 2004. 160 с.  
<http://www.mccme.ru/free-books/shen/shen-progbook.pdf>
2. Компания Sololearn. Мобильное приложение «Учим Python».  
<https://play.google.com/store/apps/details?id=com.sololearn.python> или  
<https://itunes.apple.com/us/app/id953972812> или  
<http://www.windowsphone.com/s?appid=7bb32109-b882-49c9-8fa2-a500b79a19ca>
3. Онлайн-игра для обучения программированию CodeCombat.  
<https://codecombat.com/>
4. Игра для обучения программированию Colobot.  
<https://colobot.info/>

## **9. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

Для проведения лекционных занятий требуется аудитория, оборудованная доской, мультимедийным проектором с экраном. Для проведения лабораторных занятий требуется компьютерный класс с подключением к интернету.

## **10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения**

### **10.1. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине**

При осуществлении образовательного процесса по дисциплине используются следующие информационные технологии:

- использование на занятиях электронных изданий (чтение лекций с использованием слайд-презентаций);
- ведение учета посещаемости и выполнения учебных заданий в системе Google Docs;
- написание программ на языках высокого уровня в инструментальных средах;
- сдача программ в автоматической проверяющей системе «Мультиметр»;
- организация взаимодействия с обучающимися посредством электронной почты, специализированного образовательного форума Piazza;
- компьютерное тестирование.

### **10.2. Перечень программного обеспечения**

При осуществлении образовательного процесса по дисциплине используются следующее программное обеспечение:

- Язык программирования Python, среда IDLE;
- Среда Visual Studio с компилятором Visual C++, или среда CodeBlocks с компилятором GNU C++;
- интернет-браузер.

## ЛИСТ АКТУАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Б1.В.ОД.3 — Программирование

[illegible]

*В таблице указывается только характер изменений (например, изменение темы, списка источников по теме или темам, средств промежуточного контроля) с указанием пунктов рабочей программы. Само содержание изменений оформляется приложением по сквозной нумерации.*

## Содержание

<b>1</b>	<b>АННОТАЦИЯ</b>	<b>2</b>
1.1	Цель освоения и краткое содержание дисциплины . . . . .	2
1.2	Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы . . . .	2
1.3	Место дисциплины в структуре образовательной программы . . . . .	3
1.4	Язык преподавания . . . . .	3
<b>2</b>	<b>Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся</b>	<b>4</b>
<b>3</b>	<b>Содержание дисциплины, структурированное по темам с указанием отведенного на них количества академических часов и видов учебных занятий</b>	<b>5</b>
3.1	Распределение часов по темам и видам учебных занятий . . . . .	5
3.2	Содержание тем программы дисциплины . . . . .	5
3.3	Формы и методы проведения занятий, применяемые учебные технологии . .	6
<b>4</b>	<b>Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине</b>	<b>6</b>
<b>5</b>	<b>Методические указания для обучающихся по освоению дисциплины</b>	<b>7</b>
<b>6</b>	<b>Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине</b>	<b>8</b>
6.1	Показатели, критерии и шкала оценивания . . . . .	8
6.2	Типовые контрольные задания (вопросы) для промежуточной аттестации . .	12
6.3	Методические материалы, определяющие процедуры оценивания . . . . .	18
<b>7</b>	<b>Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины</b>	<b>19</b>
<b>8</b>	<b>Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее — сеть Интернет), необходимых для освоения дисциплины</b>	<b>19</b>
<b>9</b>	<b>Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине</b>	<b>20</b>
<b>10</b>	<b>Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения</b>	<b>20</b>
10.1	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине . . . . .	20
10.2	Перечень программного обеспечения . . . . .	20