

**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЛГПУ»)**

**СТАРОБЕЛЬСКИЙ ФАКУЛЬТЕТ (ФИЛИАЛ)
Федерального государственного бюджетного образовательного учреждения высшего
образования «Луганский государственный педагогический университет»
(Старобельский факультет (филиал) ФГБОУ ВО «ЛГПУ»)**

**Кафедра естественно-математических, технических дисциплин
и методик их преподавания**

УТВЕРЖДАЮ

Директор
Старобельского факультета (филиала)
ФГБОУ ВО «ЛГПУ»

 Маршуба И.В.
«15» _____ 2023 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

По направлению подготовки: 44.03.04 Профессиональное обучение (по отраслям)

Профиль подготовки – «Разработка программного обеспечения образовательных систем»

Квалификация выпускника – бакалавр

Форма обучения – очная, заочная

Курс – ОФО - 1 курс (1, 2 семестр), ЗФО – 1, 2 курс (3, 4 триместр)

Старобельск, 2023

Лист согласования

Рабочая программа учебной дисциплины «ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ» является частью основной образовательной программы для подготовки бакалавров по направлению подготовки 44.03.04 Профессиональное обучение (по отраслям) «Разработка программного обеспечения образовательных систем» очной и заочной форм обучения.

Составлена на основании Федерального государственного образовательного стандарта по направлению подготовки 44.03.04 Профессиональное обучение (по отраслям) и уровню высшего образования – программы бакалавриата, утверждённого приказом Министерства образования и науки Российской Федерации от 22.02.2018 г. № 124.

СОСТАВИТЕЛЬ:

ассистент кафедры естественно-математических, технических дисциплин и методик их преподавания Старобельского факультета (филиала) ФГБОУ ВО «ЛГПУ» Фоменко М.М.

Рабочая программа утверждена на заседании кафедры естественно-математических, технических дисциплин и методик их преподавания

«01» ноября 2023 г., протокол № 3

Заведующий кафедрой

 Н.А. Василенко

ОДОБРЕНА на заседании учебно-методической комиссии Старобельского факультета (филиал)

«15» ноября 2023 г., протокол № 3

Председатель

 О.В. Верховод

СОГЛАСОВАНО:

Заместитель директора
Старобельского факультета
(филиал)

 А.В. Стась

«15» ноября 2023 г.

Структура и содержание дисциплины

1. Цели и задачи дисциплины, ее место в учебном процессе

Цель изучения дисциплины «Информатика и программирование» – сформировать знания, умения и навыки, необходимые для написания программ, рационального использования оболочек для написания программ, а именно языка Python и среды программирования IDLE Python 3.7 и выше при обработке разнообразных задач, связанных с обработкой информации, расчетами, поиском, сортировкой.

Задачи:

- ознакомление с теорией алгоритмизации.
- ознакомление с теорией программирования.
- ознакомление со способами создания алгоритмов и программ.
- предоставить теоретические знания и практические навыки по языку программирования Python.
- Научить студентов программировать и составлять алгоритмы.

2. Место дисциплины в структуре ООП ВО. Требования к результатам освоения содержания дисциплины

учебная дисциплина «Информатика и программирование» относится к базовой части учебного плана (Б1.О.09.04). Дисциплину реализует кафедра естественно-математических, технических дисциплин и методик их преподавания.

Основывается на базе знаний, полученных при изучении дисциплины «Основы информатики и вычислительной техники» уровня средней школы.

Содержание дисциплины «Информатика и программирование» является основой для дальнейшего освоения дисциплин: «Алгоритмы и структуры данных», «Программирование для платформы Java», «Базы данных», «Основы программирования мобильных платформ», «Программирование для платформы .Net».

А также освоение данной дисциплины необходимо для выполнения выпускной квалификационной работы и сдачи комплексного квалификационного экзамена.

3. Требования к результатам освоения содержания дисциплины

Код по ФГОС ВО	Индикатор достижения	Результаты обучения по дисциплине
Общепрофессиональные компетенции		
ОПК-9. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-9.1 Знает: теоретические основы, принципы и закономерности современных информационных технологий;	Знать: как делать постановку задачи и определять основные пункты задачи при написании программы; какие задачи решаются с помощью программирования; как создавать алгоритмы, как писать код программы; как проводить отладку программы. Уметь: использовать
	ОПК-9.2 Умеет: решать задачи профессиональной деятельности с использованием специальных научных знаний принципов и закономерностей современных информационных технологий;	

	<p>ОПК-9.3 Владеет: алгоритмами и технологиями осуществления профессиональной педагогической деятельности на основе специальных научных знаний в области информационных технологий.</p>	<p>прикладное программное обеспечение общего и целевого назначения, в частности IDLE Python 3.7 и выше; работать с программами для разработки программных продуктов и вспомогательными программами, в частности IDLE Python 3.7 и выше.</p> <p>Владеть: методами составления, описания и реализации алгоритмов; принципами и методами написания программ; принципами отладки программ.</p>
--	--	---

4. Структура и содержание учебной дисциплины

4.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов (зач. ед.)	
	Очная форма	Заочная форма
Общая учебная нагрузка (всего)	216 (6 зач. ед)	144 (4 зач. ед)
Лекции	24	8
Семинарские занятия		
Практические занятия (в том числе интерактив)		
Лабораторные работы	48	8
Контрольные работы (модули)		
КСР		
Курсовая работа (курсовой проект)	36	-
Другие формы организации учебного процесса (контроль)	58	24
Самостоятельная работа студента (всего)	86	104
Итоговая аттестация	Экзамен	Экзамен

4.2. Содержание разделов учебной дисциплины

Тема 1. Жизненный цикл программного продукта и введение в Python

Основные этапы создания программного продукта. Особенности разработки на Python. Установка интерпретатора и среды разработки (IDE/Jupyter).

Тема 2. Алгоритмы и способы их описания

Определение алгоритма. Его свойства (дискретность, детерминированность, конечность, массовость, результативность). Виды алгоритмов. Способы описания алгоритмов (словесный, блок-схемы, «псевдокод»).

Тема 3. Основы синтаксиса Python. Типы данных

Элементы и структура языка Python (отступы, комментарии, PEP 8). «Переменные». Основные встроенные типы данных: целые (int), числа с плавающей точкой (float), логический (bool), строки (str). Преобразование типов (явное и неявное). Операторы.

Тема 4. Условный оператор и ветвление

Оператор if, elif, else. Синтаксис, назначение. Логические операторы (and,

or, not). Алгоритмы с ветвлением. Тернарный оператор в Python.

Тема 5. Ветвление с множеством условий (имитация switch/case)

Реализация логики множественного выбора: использование конструкции if-elif-else для последовательной проверки условий. Применение словарей (dict) для имитации switch/case в Python.

Тема 6. Циклы: while и for. Управление циклом

Цикл while (цикл с предусловием). Цикл for для итерации по последовательностям. Функция range(). Операторы break (прерывание цикла) и continue (пропуск итерации). Использование блока else с циклами. Циклические алгоритмы.

Тема 7. Коллекции I: Списки и Строки

Работа с list (списки): создание, индексация, срезы, методы (добавление, удаление, изменение). Итерирование по спискам. Генераторы списков (list comprehensions). Строки (str): неизменяемость, основные методы работы со строками, форматирование строк (f-строки, format).

Тема 8. Коллекции II: Кортежи, Словари и Множества

Кортежи (tuple): создание, индексация, неизменяемость, распаковка. Словари (dict): создание, доступ к элементам, методы, итерация. Множества (set): создание, операции над множествами (объединение, пересечение, разность).

Тема 9. Функции. Область видимости

Определение функции (def), вызов. Аргументы: позиционные и именованные. Значения по умолчанию. Возврат значений (return). Область видимости: локальные (local), внешние (enclosing), глобальные (global) переменные. Ключевое слово global.

Тема 10. Дополнительные возможности функций

Передача произвольного числа аргументов (*args, **kwargs). Анонимные функции (lambda). Рекурсия (определение, примеры, стек вызовов). Аннотации типов (type hints).

Тема 11. Работа с Модулями и Пакетами

Концепция модулей (файлы .py). Импорт модулей (import). Создание и использование пакетов. Установка и использование внешних библиотек (pip).

Тема 12. Алгоритмы поиска

Алгоритмы поиска в последовательностях: линейный поиск. Бинарный поиск (применительно к отсортированным спискам). Оценка сложности алгоритмов (O-нотация).

Тема 13. Алгоритмы сортировки

Встроенные механизмы сортировки в Python: метод sort() для списков и функция sorted(). Принципы работы основных алгоритмов сортировки (сортировка выбором, вставками, быстрая сортировка и др.) и их реализация на Python.

Тема 14. Итераторы и Генераторы

Итерируемые объекты и итераторы. Протокол итератора (методы __iter__ и __next__). Генераторы (функции-генераторы и генераторные выражения). Преимущества использования генераторов.

Тема 15. Объектно-ориентированное программирование (ООП) в Python I

Классы и объекты. Объявление класса. Атрибуты (переменные) и методы. Конструктор (__init__). Наследование.

Тема 16. Объектно-ориентированное программирование (ООП) в Python II

Полиморфизм. Инкапсуляция и модификаторы доступа (соглашения об именовании в Python). Магические методы (дандеры): `__str__`, `__repr__` и другие.

Тема 17. Работа с файлами

Открытие, чтение и запись файлов (текстовый и бинарный режимы). Конструкция `with open(...) as f:`. Обработка исключений при работе с файлами. Модуль `os` и `pathlib`.

Тема 18. Обработка Исключений

Концепция исключений. Обработка исключений: блоки `try`, `except`, `else`, `finally`. Иерархия исключений. Генерация исключений (`raise`) и создание пользовательских исключений.

Тема 19. Работа со специальными библиотеками

Введение в экосистему Python. Использование популярных библиотек: `numpy` (для работы с многомерными массивами), `pandas` (для анализа данных) или `matplotlib` (для визуализации).

4.3. Лекции

№ п/п	Название темы	Объем часов	
		Очная форма	Заочная форма
1, 2 семестр/ 1 семестр			
1	Тема 1. Жизненный цикл программного продукта и введение в Python	1	
2	Тема 2. Алгоритмы и способы их описания	1	2
3	Тема 3. Основы синтаксиса Python. Типы данных	1	
4	Тема 4. Условный оператор и ветвление	1	
5	Тема 5. Ветвление с множеством условий (имитация switch/case)	1	
6	Тема 6. Циклы: while и for. Управление циклом	1	1
7	Тема 7. Коллекции I: Списки и Строки	1	
8	Тема 8. Коллекции II: Кортежи, Словари и Множества	1	1
9	Тема 9. Функции. Область видимости	1	1
10	Тема 10. Дополнительные возможности функций	1	
11	Тема 11. Работа с Модулями и Пакетами	2	
12	Тема 12. Алгоритмы поиска	1	1
13	Тема 13. Алгоритмы сортировки	1	
14	Тема 14. Итераторы и Генераторы	2	

15	Тема 15. Объектно-ориентированное программирование (ООП) в Python I	1	
16	Тема 16. Объектно-ориентированное программирование (ООП) в Python II	1	
17	Тема 17. Работа с файлами	2	
18	Тема 18. Обработка Исключений	2	1
19	Тема 19. Работа со специальными библиотеками	2	1
Итого:		24	8

4.4. Практические занятия

Не предусмотрены учебным планом.

4.5. Лабораторные работы

№ п/п	Название темы	Объем часов	
		Очная форма	Заочная форма
1, 2 семестр/ 3, 4 триместр			
1	Раздел 1. Тема 1-2. Арифметические действия в различных системах счисления	2	
2	Раздел 1. Тема 5. Разработка и построение алгоритмов, блок-схемы	2	
3	Раздел 2. Тема 1. Знакомство со средой разработки IDLE Python	2	
4	Раздел 2. Тема 2. Типы данных. Операции над типами данных. Сложные структуры данных.	4	
5	Раздел 2. Тема 3. Условные операторы. Множественный выбор. Циклы.	4	2
6	Раздел 2. Тема 4. Определение функций. Аргументы функции.	4	
7	Раздел 2. Тема 4. Рекурсия.	2	
8	Раздел 2. Тема 5. Импорт модулей. Создание и использование модулей.	4	
9	Раздел 2. Тема 6. Работа с классами и объектами. Наследование. Полиморфизм.	4	2
10	Раздел 2. Тема 7. Чтение и запись файлов. Операции с файлами.	2	2
11	Раздел 2. Тема 7. Операции с файлами CSV/JSON.	4	
12	Раздел 2. Тема 7. Стандартные операции работы с базами данных.	4	
13	Раздел 2. Тема 8. Создание многопоточных приложений.	4	
14	Раздел 2. Тема 8. Тестирование и отладка ПО.	2	

15	Раздел 2. Тема 8. Создание графического интерфейса пользователя.	4	2
Итого:		48	8

4.6. Самостоятельная работа

№ п/ п	Название темы	Вид самостоятельной работы	Объем часов	
			Очная форма	Заочная форма
1, 2 семестр/ 1 семестр				
1	1. Литералы в Python. Иммутабельные и мутабельные типы.	Конспект лекций	6	6
2	2. Форматирование строк (f-strings) и методы строкового типа.	Конспект лекций	6	8
3	3. Системы счисления и встроенные функции преобразования.	Конспект лекций	6	8
4	4. Сравнение типов данных: Списки, Кортежи, Словари, Множества.	Конспект лекций	4	6
5	5. Битовые операции и их применение в Python.	Конспект лекций	4	6
6	6. Логические операции и "ленивые вычисления" (Short-circuit evaluation).	Конспект лекций	4	6
7	7. Генераторы списков, словарей и множеств (Comprehensions).	Конспект лекций	6	8
8	8. Декораторы функций.	Конспект лекций	4	6
9	9. Использование функции map(), filter(), reduce().	Конспект лекций	4	4
10	10. Организация кода: Модули, Пакеты и __init__.py.	Конспект лекций	6	8
11	11. Итераторы и Генераторы.	Конспект лекций	6	6
12	12. Словари как Ассоциативные массивы (Хеш-таблицы).	Конспект лекций	6	8
13	13. Работа с регулярными выражениями (модуль re).	Конспект лекций	8	8

14	14. Динамические структуры данных на базе классов: Стек и Очередь.	Конспект лекций	8	8
15	15. Работа с исключениями и отладка (Debugging) проектов.	Конспект лекций	8	8
Итого:			86	104

4.7. Курсовые работы.

Изучаемый курс заканчивается выполнением курсовой работы, тема выбирается из следующего списка (Список может быть скорректирован в процессе обучения).

Задание курсовой работы:

1. «Математическое моделирование в программировании»
2. «Геометрические алгоритмы в программировании»
3. «Жадные алгоритмы в программировании»
4. «Динамическое программирование. Плюсы и минусы в использовании»
5. «Рекурсия и перебор в программировании»
6. «Метод двух указателей. Теория и практика»
7. «Деревья – нелинейные структуры данных. Типы. Применение. Алгоритмы работы»
8. «Анонимные функции в программировании. Назначение, применение, приоритеты. Использование в разных языках»
9. «Индексно-последовательный поиск в программировании. Назначение, применение, примеры»
10. «Бинарный и тернарный поиск в программировании. Назначение, применение, примеры»
11. «Z-функция строки и ее вычисление. Назначение, применение, примеры»
12. «Алгоритм сжатия Хаффмана. Назначение, применение, примеры» Коналов А.
13. «Алгоритм Кнута — Морриса — Пратта. Назначение, применение, примеры»

5. Методическое обеспечение. Образовательные технологии

Преподавание дисциплины ведется с применением следующих видов образовательных технологий.

Наряду с методикой традиционной лекционно-практической работы предусмотрено использование активных форм и методов учебной деятельности, в том числе: учебные дискуссии, беседы, мозговой штурм.

Методика проблемно-диалогического обучения применяется в процессе лекционной работы над учебным материалом в каждой из тем учебной дисциплины.

Методика обучения в сотрудничестве с применением командных, групповых видов работы используется в процессе организации лабораторных работ.

Методика исследовательской деятельности используется как основа для организации самостоятельной работы студентов в объеме учебных тем. Применяются средства мультимедиа: презентации, видео, базы ЭОР.

Информационные технологии: использование электронных образовательных ресурсов (электронный конспект, размещенный во внутренней сети или т.п.) при подготовке к лекциям, лабораторным работам и самостоятельной работе.

Работа в команде, проектная деятельность: совместная работа студентов в группе при выполнении лабораторных работ.

6. Формы контроля освоения дисциплины.

Текущая аттестация студентов производится в дискретные временные интервалы в следующих формах: выполнение лабораторных работ; защита лабораторных работ.

Итоговый контроль по результатам освоения дисциплины проходит в форме устного экзамена (включает в себя ответ на теоретические вопросы или выполнение тестов) и написание программы согласно полученному заданию.

Баллы, которые получают студенты очной формы обучения

Вид текущей учебной работы	Количество баллов
1, 2 семестр/ 1 семестр	
Оформление отчетов по лабораторным работам и работа на лабораторных работах	40 баллов
Самостоятельная работа	20 баллов
Экзамен	40 баллов
Итого за семестр:	100 баллов

Накопительная система оценивания по 100-балльной шкале

Четырехбалльная система оценивания экзамена	100-балльная шкала	Буквенная шкала, соответствующая 100-балльной шкале	Система оценивания зачета
Отлично	90–100	А – отлично – теоретическое содержание курса освоено полностью, без пробелов; необходимые практические навыки работы с освоенным материалом сформированы; все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	Зачтено
Хорошо	83–89	В – очень хорошо – теоретическое содержание курса освоено полностью, без пробелов; необходимые практические навыки работы с освоенным материалом в основном сформированы; все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному	

Хорошо	75–82	С – хорошо – теоретическое содержание курса освоено полностью; некоторые практические навыки работы с освоенным материалом сформированы недостаточно; все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	
Удовлетворительно	63–74	D – удовлетворительно – теоретическое содержание дисциплины освоено частично, но пробелы не носят существенного характера; необходимые практические навыки работы с освоенным материалом в основном сформированы; большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, содержат ошибки	
Удовлетворительно	50–62	E – посредственно – теоретическое содержание курса освоено частично; некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному	
Неудовлетворительно	21–49	FX – неудовлетворительно – теоретическое содержание курса освоено частично; необходимые практические навыки работы не сформированы; большинство предусмотренных программой обучения учебных заданий не выполнено либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	Не зачтено
Неудовлетворительно	0–20	F – неудовлетворительно – теоретическое содержание курса не освоено; необходимые практические навыки работы не сформированы; все выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий	

7. Учебно-методическое и программно-информационное обеспечение учебной дисциплины

а) Основная литература:

- 1) **Луц, Дж. Изучаем Python : 5-е издание** / Джон Луц ; [пер. с англ. Е.М. Симоновой]. – Санкт-Петербург : Питер, 2019. – 833 с. : ил. – Текст : непосредственный.
- 2) **Доусон, М. Программируем на Python** / Майкл Доусон. – Санкт-Петербург : Питер, 2014. – 416 с. : ил. – Текст : непосредственный.
- 3) **Гагарина, Л. Г. Алгоритмы и структуры данных** / Л. Г. Гагарина, В. Д. Колдаев. — Москва : Инфра-М, 2009. — 304 с. — Текст : непосредственный.
- 4) **Вандер Плас, Д. Python для сложных задач: наука о данных и машинное обучение** / Джейк Вандер Плас. — Санкт-Петербург : Питер, 2018. — 576 с. — Текст : непосредственный.

б) Дополнительная литература:

- 1) **Чан, Джейми. Python: быстрый старт** / Джейми Чан. — Санкт-Петербург : Питер, 2021. — 224 с. : ил. — Текст : непосредственный.
- 2) **Лутц, М. Python: карманный справочник : 5-е изд.** / Марк Лутц ; [пер. с англ. В. В. Смирнова]. — Москва : ООО "И.Д. Вильямс", 2015. — 320 с. : ил. — Текст : непосредственный.
- 3) **Официальная документация Python.** — Текст : электронный // Python Software Foundation. — URL: <https://docs.python.org/3> (дата обращения: 14.02.2023).

в) Программное обеспечение и информационные ресурсы:

- 1) **Интерпретатор Python : актуальная версия 3.x.** — Текст : электронный. — URL: <https://www.python.org/downloads> (дата обращения: 14.02.2023). — [Свободное ПО].
- 2) **PyCharm Community Edition : среда разработки (IDE).** — Текст : электронный. — URL: <https://www.jetbrains.com/pycharm> (дата обращения: 18.05.2023) — [Свободная лицензия].
- 3) **Visual Studio Code : среда разработки (IDE).** — Текст : электронный. — URL: <https://code.visualstudio.com> (дата обращения: 18.05.2023) — [Свободное ПО].
- 4) **Jupyter Notebook / Jupyter Lab : интерактивные среды.** — Текст : электронный. — URL: <https://jupyter.org> (дата обращения: 18.05.2023) — [Свободное ПО].
- 5) **PEP (Python Enhancement Proposals) : стандарты и предложения по развитию языка.** — Текст : электронный. — URL: <https://peps.python.org> (дата обращения: 18.05.2023).

8. Материально-техническое обеспечение учебной дисциплины

В качестве материально-технического обеспечения дисциплины могут быть использованы мультимедийные средства: проектор.

Лекционные занятия: комплект электронных презентаций/слайдов, аудитория, оснащенная презентационной техникой (проектор, экран, компьютер).

Лабораторные работы: лаборатория кафедры ИОТС, оснащенная мультимедийным проектором, интерактивной доской, сетевой инфраструктурой и организованным доступом в Интернет, пакеты ПО MS Office, Corel Draw, Visual Studio 2017 и выше.

9. Лист дополнений и изменений

[illegible]