

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Ростовский государственный экономический университет (РИНХ)»

УТВЕРЖДАЮ
Директор Таганрогского института
имени А. П. Чехова (филиала)
РГЭУ (РИНХ)
_____ С. А. Петрушенко
«25» мая 2026 г.

**Рабочая программа дисциплины
Языки программирования**

Направление подготовки
44.04.01 Педагогическое образование

Направленность (профиль) программы магистратуры
44.04.01.15 Информатика. Цифровая трансформация образования

Для набора 2026 года

Квалификация
Магистр

КАФЕДРА информатики**Распределение часов дисциплины по семестрам / курсам**

Семестр (<Курс>.<Семестр на курсе>)	1 (1.1)		2 (1.2)		Итого	
	Неделя		13 5/6			
Вид занятий	УП	РП	УП	РП	УП	РП
Лекции	6	6	6	6	12	12
Лабораторные	26	26	26	26	52	52
Итого ауд.	32	32	32	32	64	64
Контактная работа	32	32	32	32	64	64
Сам. работа	40	40	76	76	116	116
Часы на контроль	36	36	36	36	72	72
Итого	108	108	144	144	252	252

ОСНОВАНИЕ

Учебный план утвержден учёным советом вуза от 03.03.2026, протокол № 9.

Программу составил(и): канд. техн. наук, Доц., Белоконова С.С.

Зав. кафедрой: Тюшнякова И.А.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	подготовка студентов к осуществлению качественного преподавания основ программирования в школе с использованием популярных языков программирования и сред разработки, таких как Scratch, КуМир, Python и Кулибин.
-----	---

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

ПКР-1:	Способен осуществлять педагогическую деятельность в образовательных организациях в соответствии с уровнем и направленностью полученного образования
ПКР-1.1:	Ориентируется в современной образовательной среде, осведомлен о требованиях федеральных государственных образовательных стандартов
ПКР-1.2:	Осуществляет педагогическую деятельность в образовательных организациях различного уровня с учётом последних достижений методики преподавания

В результате освоения дисциплины обучающийся должен:

Знать:	основные концепции и принципы программирования; интерфейс и основные возможности сред программирования Scratch, КуМир, Python и «Кулибин»; структуру и логику построения алгоритмов линейных, ветвящихся и циклических процессов; основные типы данных и операторы языка Python; возможности использования датчиков и исполнительных механизмов в робототехнике (платформа «Кулибин»); современное состояние методики преподавания программирования в образовательных организациях, ориентируясь на Федеральные государственные образовательные стандарты (ФГОС); методы и подходы к организации занятий по программированию в зависимости от уровня подготовки школьников (соотнесено с индикаторами ПКР-1.1, ПКР-1.2).
Уметь:	составлять и исполнять программы в средах Scratch, КуМир и Python; разрабатывать программы, решающие типовые задачи школьного курса информатики; организовывать и проводить занятия по программированию в школе; эффективно использовать специальные модули и библиотеки языка Python; устанавливать и загружать программное обеспечение платформы «Кулибин», работать с датчиками и устройствами передвижения; разрабатывать и проводить занятия по программированию с использованием сред Scratch, КуМир и Python, учитывая психолого-возрастные особенности учащихся (соотнесено с индикаторами ПКР-1.1, ПКР-1.2).
Владеть:	навыками работы с программными средами Scratch, КуМир и Python; приемами создания и исполнения программ для школьников; методами организации учебного процесса с применением робототехнического комплекса «Кулибин»; умениями представлять материал доступно и понятно для школьников; навыками работы с программными средами Scratch, КуМир, Python и устройством аппаратных компонентов платформы «Кулибин» (соотнесено с индикаторами ПКР-1.1, ПКР-1.2).

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Раздел 1. Программирование в начальных классах. Scratch

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
1.1	Введение в Scratch. Основы программирования в Scratch	Лекционные занятия	1	2	ПКР-1 ПКР-1.1 ПКР-1.2
1.2	Знакомство со средой программирования Scratch. Интерфейс и главное меню Scratch. Понятия «скрипт», «сцена», «спрайт». Система команд исполнителя Scratch. Блоки и команды. Движение, звук, цвет спрайтов. Управление и контроль над спрайтом, анимация. Управление несколькими объектами. Последовательное и одновременное выполнение. Линейный алгоритм. Разветвляющийся алгоритм. Циклический алгоритм. Случайные числа. Диалог с пользователем. Использование слоев. Анимация полета. Создание плавной анимации. Разворот в направлении движения. Изучаем повороты. Изменение движения в зависимости от условия. Графические эффекты картинок. Создание личного проекта в Scratch	Лабораторные занятия	1	14	ПКР-1 ПКР-1.1 ПКР-1.2
1.3	Введение в Scratch. Основы программирования в Scratch. Знакомство со средой программирования Scratch. Интерфейс и главное меню Scratch. Понятия «скрипт», «сцена», «спрайт». Система команд исполнителя Scratch. Блоки и команды. Движение, звук, цвет спрайтов. Управление и контроль над спрайтом, анимация. Управление несколькими объектами. Последовательное и	Самостоятельная работа	1	20	ПКР-1 ПКР-1.1 ПКР-1.2

	одновременное выполнение. Линейный алгоритм. Разветвляющийся алгоритм. Циклический алгоритм. Случайные числа. Диалог с пользователем. Использование слоев. Анимация полета. Создание плавной анимации. Разворот в направлении движения. Изучаем повороты. Изменение движения в зависимости от условия. Графические эффекты картинок. Создание личного проекта в Scratch Методика преподавания Scratch.				
Раздел 2. Среда «КуМир»					
№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
2.1	Понятие исполнителя. Среда «КуМир». Учебные исполнители. Понятие алгоритма. Типы алгоритмов для управления исполнителями Кузнечик, Водолей, Робот, Черепаха. Составление алгоритмов и программ	Лекционные занятия	1	4	ПКР-1 ПКР-1.1 ПКР-1.2
2.2	Понятие исполнителя. Среда «КуМир». Учебные исполнители. Понятие алгоритма. Типы алгоритмов для управления исполнителями Кузнечик, Водолей, Робот, Черепаха. Составление алгоритмов и программ	Лабораторные занятия	1	12	ПКР-1 ПКР-1.1 ПКР-1.2
2.3	Понятие исполнителя. Среда «КуМир». Учебные исполнители. Понятие алгоритма. Типы алгоритмов для управления исполнителями Кузнечик, Водолей, Робот, Черепаха. Составление алгоритмов и программ	Самостоятельная работа	1	20	ПКР-1 ПКР-1.1 ПКР-1.2
Раздел 3. Язык Python					
№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
3.1	Язык Python. Типы данных. Особенности ввода/вывода. Линейные алгоритмы. Особенности подключения и использования модулей. Разветвляющиеся процессы. Условный оператор. Циклические алгоритмы. Операторы цикла. Работа со строками. Кортежи, списки словари, множества. Одномерные массивы. Кортежи, списки словари, множества. Одномерные массивы. Работа с файлами.	Лекционные занятия	2	6	ПКР-1 ПКР-1.1 ПКР-1.2
3.2	Язык Python. Типы данных. Особенности ввода/вывода. Линейные алгоритмы. Особенности подключения и использования модулей. Разветвляющиеся процессы. Условный оператор. Циклические алгоритмы. Операторы цикла. Работа со строками. Кортежи, списки словари, множества. Одномерные массивы. Кортежи, списки словари, множества. Одномерные массивы. Работа с файлами.	Лабораторные занятия	2	20	ПКР-1 ПКР-1.1 ПКР-1.2
3.3	Язык Python. Типы данных. Особенности ввода/вывода. Линейные алгоритмы. Особенности подключения и использования модулей. Разветвляющиеся процессы. Условный оператор. Циклические алгоритмы. Операторы цикла. Работа со строками. Кортежи, списки словари, множества. Одномерные массивы. Кортежи, списки словари, множества. Одномерные массивы. Работа с файлами.	Самостоятельная работа	2	40	ПКР-1 ПКР-1.1 ПКР-1.2
Раздел 4. Образовательная робототехника. Платформа Кулибин					
№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
4.1	Загрузка и установка среды «Кулибин». Основные элементы интерфейса. Разделы «управление» и «движение». Датчики. Движение по линии. Квадрокоптер.	Лабораторные занятия	2	6	ПКР-1 ПКР-1.1 ПКР-1.2
4.2	Загрузка и установка среды «Кулибин». Основные элементы интерфейса. Разделы «управление» и «движение». Датчики. Движение по линии. Квадрокоптер.	Самостоятельная работа	2	36	ПКР-1 ПКР-1.1 ПКР-1.2
Раздел 5. Экзамен					
№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
5.1	Подготовка к промежуточной аттестации	Экзамен	1	36	ПКР-1 ПКР-1.1 ПКР-1.2
5.2	Подготовка к промежуточной аттестации	Экзамен	2	36	ПКР-1 ПКР-1.1 ПКР-1.2

4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Структура и содержание фонда оценочных средств для проведения текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Учебные, научные и методические издания

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
1	Златопольский Д. М.	Программирование: типовые задачи, алгоритмы, методы: учебное пособие	Москва: Лаборатория знаний, 2020	http://biblioclub.ru/index.php?page=book&id=222873
2	Анеликова Л. А., Гусева О. Б.	Программирование на алгоритмическом языке КуМир: учебное пособие	Москва: СОЛОН-ПРЕСС, 2012	http://biblioclub.ru/index.php?page=book&id=226976
3	Sweigart A.	Разработка компьютерных игр на языке Python	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	http://biblioclub.ru/index.php?page=book&id=429009
4	Северенс Ч.	Введение в программирование на Python	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	http://biblioclub.ru/index.php?page=book&id=429184
5	Нагаева И. А., Кузнецов И. А.	Алгоритмизация и программирование. Практикум: учебное пособие	Москва Берлин: Директ-Медиа, 2019	http://biblioclub.ru/index.php?page=book&id=570287
6	Сузи Р.	Python	Санкт-Петербург: БХВ-Петербург, 2015	https://ibooks.ru/reading.php?short=1&productid=18432
7	Прохоренок Н.А., Дронов В.А.	Python 3 и PyQt 5. Разработка приложений	Санкт-Петербург: БХВ-Петербург, 2016	https://ibooks.ru/reading.php?short=1&productid=353570
8	Прохоренок Н.А.	Python 3 и PyQt. Разработка приложений	Санкт-Петербург: БХВ-Петербург, 2011	https://ibooks.ru/reading.php?short=1&productid=24833
9	Плас Дж. Вандер	Python для сложных задач: наука о данных и машинное обучение	Санкт-Петербург: Питер, 2018	https://ibooks.ru/reading.php?short=1&productid=356721
10	Шелудько, В. М.	Основы программирования на языке высокого уровня Python: учебное пособие	Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017	http://www.iprbookshop.ru/87461.html
11	Шелудько, В. М.	Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие	Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017	http://www.iprbookshop.ru/87530.html
12	Маккинли, Уэс, Слинкина, А.	Python и анализ данных	Саратов: Профобразование, 2019	http://www.iprbookshop.ru/88752.html
13	Анеликова, Л. А., Гусева, О. Б., Кушниренко, А. Г.	Программирование на алгоритмическом языке КуМир	Москва: СОЛОН-Пресс, 2019	http://www.iprbookshop.ru/90306.html
14	Рик, Гаско	Простой Python просто с нуля	Москва: СОЛОН-Пресс, 2019	http://www.iprbookshop.ru/94940.html
15	Дроботун, Н. В., Рудков, Е. О., Баев, Н. А.	Алгоритмизация и программирование. Язык Python: учебное пособие	Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2020	http://www.iprbookshop.ru/102400.html

5.2. Профессиональные базы данных и информационные справочные системы

rs1.ru – Российская государственная библиотека
elibrary.ru – Научная электронная библиотека
biblioclub.ru – Университетская библиотека онлайн
intuit.ru – Национальный Открытый Университет «ИНТУИТ»
Научная электронная библиотека «КиберЛенинка» <https://cyberleninka.ru/>

5.3. Перечень программного обеспечения

Python
OpenOffice

5.4. Учебно-методические материалы для обучающихся с ограниченными возможностями здоровья

При необходимости по заявлению обучающегося с ограниченными возможностями здоровья учебно-методические материалы предоставляются в формах, адаптированных к ограничениям здоровья и восприятия информации. Для лиц с нарушениями зрения: в форме аудиофайла; в печатной форме увеличенным шрифтом. Для лиц с нарушениями слуха: в форме электронного документа; в печатной форме. Для лиц с нарушениями опорно-двигательного аппарата: в форме электронного документа; в печатной форме.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Помещения для всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения:

- столы, стулья;
- персональный компьютер / ноутбук (переносной);
- проектор;
- экран / интерактивная доска.

Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными и/или свободно распространяемыми программными средствами и выходом в Интернет, и/или в специализированных лабораториях, предусмотренных образовательной программой.

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

1 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

1.1 Показатели и критерии оценивания компетенций:

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
ПКР-1: Способен осуществлять педагогическую деятельность в образовательных организациях в соответствии с уровнем и направленностью полученного образования			
<p>Знать: основные концепции и принципы программирования; интерфейс и основные возможности сред программирования Scratch, КуМир, Python и «Кулибин»; структуру и логику построения алгоритмов линейных, ветвящихся и циклических процессов; основные типы данных и операторы языка Python; возможности использования датчиков и исполнительных механизмов в робототехнике (платформа «Кулибин»); современное состояние методики преподавания программирования в образовательных организациях, ориентируясь на Федеральные государственные образовательные стандарты (ФГОС); методы и подходы к организации занятий по программированию в зависимости от уровня подготовки школьников</p>	<p>Осуществление поиска и сбора необходимой литературы, изучение лекционного материала, основной и дополнительной литературы. Выполняет лабораторные задания.</p>	<p>Полнота и содержательность ответа; умение приводить примеры; соответствие представленной в ответах информации материалам лекции и учебной литературы, сведениям из информационных ресурсов Интернет; целенаправленность поиска и отбора</p>	<p>Вопросы к экзамену (вопросы 1-40) ЛР (1-28) Доклад (1-15)</p>
<p>Уметь: составлять и исполнять программы в средах Scratch, КуМир и Python; разрабатывать программы, решающие типовые задачи школьного курса информатики; организовывать и проводить занятия по программированию в школе; эффективно использовать специальные модули и библиотеки языка Python; устанавливать и загружать программное обеспечение платформы «Кулибин», работать с датчиками и устройствами передвижения; разрабатывать и проводить занятия по</p>	<p>Выполнение домашнего задания, выполняет лабораторные задания. Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.</p>	<p>Достоверность решения, полнота и содержательность ответа; умение приводить примеры; соответствие представленной в ответах информации материалам лекции и учебной литературы, сведениям из информационных ресурсов Интернет; целенаправленность поиска и отбора</p>	<p>Вопросы к экзамену (вопросы 1-40) ЛР (1-28) Доклад (1-15)</p>

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
программированию с использованием сред Scratch, КуМир и Python, учитывая психолого-возрастные особенности учащихся			
Владеть методами и инструментами поиска, критического анализа и синтеза информации для решения задач программирования, навыками программирования на языках Pascal и Python для реализации алгоритмов разной сложности, средствами отладки и тестирования программ, умея находить и исправлять ошибки, способность реализовывать программы на языках Pascal и Python в соответствии с требованиями школьного курса информатики, обеспечивая качественное выполнение учебных программ	Выполнение домашнего задания, выполняет лабораторные задания. Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.	Достоверность решения, полнота и содержательность ответа; умение приводить примеры; соответствие представленной в ответах информации материалам лекции и учебной литературы, сведениям из информационных ресурсов Интернет; целенаправленность поиска и отбора	Вопросы к экзамену (вопросы 1-40) ЛР (1-28) Доклад (1-15)

1.2 Шкалы оценивания:

Текущий контроль успеваемости и промежуточная аттестация осуществляется в рамках накопительной балльно-рейтинговой системы в 100-балльной шкале:

- 1,2 семестр – экзамен
- 84-100 баллов (оценка «отлично»)
- 67-83 баллов (оценка «хорошо»)
- 50-66 баллов (оценка «удовлетворительно»)
- 0-49 баллов (оценка «неудовлетворительно»)

2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Вопросы к экзамену

1. Что такое Scratch и каковы его основные элементы?
2. Что означают термины «скрипт», «сцена», «спрайт»?
3. Как управлять цветом и звуком спрайта в Scratch?
4. Как задать произвольное движение спрайту?
5. Что такое одновременное выполнение скриптов и как это используется в Scratch?
6. Какие блоки отвечают за анимацию и создание плавного движения?
7. Как организовать диалог с пользователем в проекте Scratch?
8. Как создается анимация полета и поворота спрайта?
9. Какие существуют графические эффекты картинок в Scratch?
10. Приведите пример простой игры или анимации, сделанной в Scratch.
11. Что такое исполнитель и среда «КуМир»?
12. Какие учебные исполнители представлены в среде «КуМир»?
13. Что такое линейный, разветвленный и циклический алгоритмы?

14. Приведите примеры алгоритмов для исполнителя «Кузнечик» и «Водолей».
15. Как работают команды исполнителя «Робот»?
16. Как выглядит схема составления алгоритма в среде «КуМир»?
17. Какие типы данных можно использовать в программах «КуМир»?
18. Приведите пример программы с использованием вложенных циклов.
19. Что означает слово «исполнитель» в контексте среды «КуМир»?
20. Как формулируются задачи для исполнителя «Черепашка»?
21. Что такое язык программирования Python и какие его особенности?
22. Какие типы данных существуют в Python?
23. Как устроен механизм ввода и вывода данных в Python?
24. Что такое модуль и как подключить его в Python?
25. Приведите примеры использования линейных алгоритмов на Python.
26. Когда и как используется условный оператор (if-else) в Python?
27. Какие циклы предусмотрены в Python и как они работают?
28. Что такое кортежи, списки, словари и множества в Python?
29. Как организован доступ к элементам массива и строкам в Python?
30. Какие операции возможны при работе с файлами в Python?
31. Что представляет собой платформа «Кулибин»?
32. Как устанавливаются и обновляются драйверы платформы «Кулибин»?
33. Какие основные элементы интерфейса присутствуют в среде «Кулибин»?
34. Как настроены датчики и камеры в среде «Кулибин»?
35. Что значит «движение по линии» в робототехнике?
36. Как работает квадрокоптер в среде «Кулибин»?
37. Как настроить поведение робота при движении по чёрной линии?
38. Какие режимы управления доступны для робота в среде «Кулибин»?
39. Приведите примеры программ, работающих с датчиками и камерой в среде «Кулибин».
40. Как организованы эксперименты с моделью квадрокоптера в среде «Кулибин»?

Экзаменационное задание включает теоретические вопросы («Вопросы к экзамену») и практическое задание (формируется из перечня заданий, представленных в разделе «Лабораторные задания»).

Критерии оценивания:

Максимальное количество баллов за экзаменационное задание – 100 (50 баллов максимально за теоретический вопрос, 50 баллов максимально за практико-ориентированное задание).

Критерии оценивания теоретического вопроса.

Критерии оценивания теоретического вопроса	Баллы
Изложенный материал фактически верен, наличие глубоких исчерпывающих знаний; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе	41-50
Наличие твердых и достаточно полных знаний, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, неуверенность и неточность ответов на дополнительные и наводящие вопросы	26-40
Неполный ответ на вопросы; затрудняется ответить на дополнительные вопросы	1-25
Ответ не связан с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы	0
<i>Максимальный балл за ответ на теоретический вопрос</i>	<i>50</i>

Критерии оценивания практического задания.

Критерии оценивания практического задания	Баллы
Задание выполнено в полном объеме, в представленном решении обоснованно получены правильные ответы, проведен анализ, дана грамотная интерпретация полученных результатов, сделаны выводы	41-50
Задание выполнено в полном объеме, но при анализе и интерпретации полученных результатов допущены незначительные ошибки, выводы – достаточно обоснованы, но неполны	26-40
Задание выполнено не в полном объеме, при анализе и интерпретации полученных результатов допущены ошибки, выводы – но неполные или отсутствуют	1-25
Задание выполнено полностью неверно или отсутствует решение	0
<i>Максимальный балл за решение практического задания</i>	<i>50</i>

Итоговый результат формируется из суммы набранных баллов за выполнение зачетного задания (1 теоретический вопрос и 1 практико-ориентированное задание) и соответствует шкале:

84-100 баллов (оценка «отлично»)

67-83 баллов (оценка «хорошо»)

50-66 баллов (оценка «удовлетворительно»)

0-49 баллов (оценка «неудовлетворительно»)

Темы докладов

1. Использование среды Scratch для раннего обучения программированию младших школьников. Возможности и преимущества Scratch, примеры успешных проектов.
2. Типы алгоритмов и их применение в среде «КуМир». Линейные, разветвляющиеся и циклические алгоритмы на примерах исполнителей «Кузнечик», «Водолей», «Робот», «Черепаша».
3. Средства и технологии разработки программ на языке Python. Структура языка, типы данных, операторы и стандартные модули, примеры программ для школьников.
4. Применение языка Python в решении школьных задач по информатике. Реализация классических задач и головоломок на Python, подходящих для начальной и средней школы.
5. Робототехнические комплексы в школьном образовании: среда «Кулибин». Установка и настройка среды, основы программирования роботов, особенности движущихся объектов и датчиков.
6. Примеры использования Scratch в междисциплинарных проектах (математика, физика, искусство). Иллюстрация использования Scratch для решения задач из разных областей знаний.
7. Оптимизация процесса обучения программированию с помощью Python. Методы эффективного преподавания программирования с использованием языка Python.
8. Эффективные стратегии обучения программированию в начальной школе с использованием Scratch. Обоснование важности ранней разработки навыков программирования, обсуждение удачного опыта применения Scratch.
9. Разбор практических случаев применения «КуМир» в среднем школьном возрасте. Рассмотрение уроков, основанных на различных исполнителях, представленных в среде «КуМир».
10. Особенности работы с файлами в языке Python и их значимость в обучении школьников. Работа с файлами как важный элемент преподавания программирования, конкретные примеры и случаи из школьной практики.
11. Методика организации конкурсов и соревнований по программированию для школьников на базе Scratch и Python. Правила проведения турниров, отбор заданий, стратегия подготовки школьников к подобным состязаниям.
12. Опыт внедрения робототехники на платформе «Кулибин» в образовательный процесс школы. Реалии и проблемы внедрения робототехники в российскую школу, рассмотрение позитивных изменений и трудностей.
13. Алгоритмические головоломки и задачи на языках Scratch и Python. Подбор и анализ алгоритмических задач различной степени сложности, которые подойдут для школьников разных возрастов.
14. Способы использования «КуМир» для углубленной подготовки к ЕГЭ по информатике. Подробный разбор тем, связанных с решением задач ЕГЭ, и помощь студентам в понимании фундаментальных идей программирования.

15. Этапы создания проектов в среде Scratch с нуля до завершения. Алгоритм разработки проектов, полезные идеи и подсказки по улучшению работ школьников.

Критерии оценивания. Максимальное количество баллов – 40:

- 30-40 баллов – наличие глубоких исчерпывающих знаний в области изучаемого вопроса, грамотное и логически стройное изложение материала, широкое использование дополнительной литературы, демонстрация основных компетенций;
- 20-30 баллов – наличие твердых и достаточно полных знаний в рамках поставленного вопроса; четкое изложение материала; допускаются отдельные логические и стилистические погрешности;
- 0-20 баллов – наличие слабых знаний в рамках поставленного вопроса, изложение ответов с отдельными ошибками, исправленных после замечаний.

Лабораторные работы

1. **Лабораторная работа №1. Знакомство с интерфейсом Scratch.** *Цель:* Изучение окна программы, панели инструментов, палитры блоков и настроек. Основные понятия: сцена, спрайт, костюм, фон, звук.
 2. **Лабораторная работа №2. Первый скрипт в Scratch.** *Цель:* Создание первого скрипта с движением спрайта по экрану. Изучение базовых команд движения, вращения и озвучивания.
 3. **Лабораторная работа №3. Работа с цветами и формами.** *Цель:* Практическое исследование возможностей изменения размера, формы и цвета спрайтов. Применение фильтров и графических эффектов.
 4. **Лабораторная работа №4. Управление несколькими объектами.** *Цель:* Программирование одновременного управления двумя и более спрайтами. Синхронное исполнение действий.
 5. **Лабораторная работа №5. Линейные и разветвляющие алгоритмы.** *Цель:* Разработка программ с простыми линейными алгоритмами и условием if-else.
 6. **Лабораторная работа №6. Циклические алгоритмы и случайные события.** *Цель:* Применение циклов repeat и forever. Добавление случайных значений для создания непредсказуемости в поведении спрайтов.
 7. **Лабораторная работа №7. Диалог с пользователем и слои изображений.** *Цель:* Взаимодействие со зрителем через ввод текста и нажатие клавиш. Применение слоев для создания многослойных композиций.
 8. **Лабораторная работа №8. Индивидуальный проект в Scratch.** *Цель:* Завершить небольшой личный проект (мини-игра, мультфильм, музыкальная композиция), используя все изученные навыки и инструменты.
-
1. **Лабораторная работа №1. Знакомство с средой «КуМир».** *Цель:* Изучение интерфейса среды «КуМир», основные элементы и их назначение. Понятие исполнителя, среды, сценария и программной оболочки.
 2. **Лабораторная работа №2. Исполнитель «Кузнечик».** *Цель:* Изучение основных команд и свойств исполнителя «Кузнечик». Разработка простых программ, управляя передвижением кузнечика.
 3. **Лабораторная работа №3. Исполнитель «Водолей»**
Цель: Знакомство с командами исполнителя «Водолей». Решение задач по переливанию жидкости, создавая и выполняя короткие программы.
 4. **Лабораторная работа №4. Исполнитель «Робот»**
Цель: Постановка и решение задач с использованием исполнителя «Робот». Работа с оператором цикла и условиями, составление программ с ветвлением и итерациями.
 5. **Лабораторная работа №5. Исполнитель «Черепашка»**
Цель: Изучение команд исполнителя «Черепашка». Рисование геометрических фигур и линий с помощью черепашки. Знакомство с работой координатной плоскости.
 6. **Лабораторная работа №6. Составление и отладка программ в среде «КуМир»**
Цель: Создание комплексной программы, объединяющей свойства нескольких исполнителей. Проверка правильности работы программы и выявление возможных ошибок.
-
1. **Лабораторная работа №1. Знакомство с языком Python**

- Цель: Установка интерпретатора Python, создание первой программы ("Hello World"), знакомство с выводом на экран и вводом данных.
2. **Лабораторная работа №2. Типы данных в Python**
Цель: Изучение примитивных типов данных (целые числа, вещественные числа, строки, булевы значения), операции сравнения и преобразования типов.
 3. **Лабораторная работа №3. Линейные алгоритмы**
Цель: Реализация простых линейных программ, выполняющих вычисления и обработку данных без ветвлений и циклов.
 4. **Лабораторная работа №4. Разветвляющиеся процессы и условный оператор**
Цель: Изучение оператора ветвления (if-elif-else), написание программ с выборочным выполнением операций.
 5. **Лабораторная работа №5. Циклические алгоритмы и операторы цикла**
Цель: Изучение циклов (for, while) и применение их для многократного выполнения инструкций.
 6. **Лабораторная работа №6. Структуры данных: кортежи, списки, словари, множества**
Цель: Изучение способов хранения и обработки данных с помощью списков, кортежей, словарей и множеств.
 7. **Лабораторная работа №7. Работа со строками**
Цель: Изучение методов и операций со строками (разбиение, соединение, замена символов, поиск и удаление фрагментов).
 8. **Лабораторная работа №8. Работа с модулями и библиотеками**
Цель: Импорт стандартных и сторонних модулей (например, math, random), подключение и использование библиотечных функций.
 9. **Лабораторная работа №9. Работа с файлами**
Цель: Открытие, чтение и запись данных в файл, использование текстового формата и бинарных файлов.
 10. **Лабораторная работа №10. Итоговый проект на Python**
Цель: Создание комплексного проекта, использующего знания предыдущих лабораторных работ (циклы, условия, работа с файлами, структуры данных).
1. **Лабораторная работа №1. Загрузка и установка среды «Кулибин»**
Цель: Скачать дистрибутив среды «Кулибин», установить и запустить программу. Первичное знакомство с главным окном, изучение элементов интерфейса.
 2. **Лабораторная работа №2. Элементы управления и команда «Движение»**
Цель: Изучение команд управления моторами, блока «Управление» и элемента «Движение». Запуск робота по прямой траектории, повороты, остановка.
 3. **Лабораторная работа №3. Работа с датчиками и линиями**
Цель: Изучение сенсоров платформы «Кулибин». Реализация программы движения робота вдоль черной линии с использованием светочувствительных датчиков.
 4. **Лабораторная работа №4. Летательные аппараты (квадрокоптеры)**
Цель: Знакомство с управлением квадрокоптерами в среде «Кулибин». Изучение команд для взлета, посадки, горизонтального и вертикального маневрирования.

Критерии оценивания. Максимальное количество баллов – 60:

- 47-60 баллов, если студент правильно выполнил 84-100% практических заданий;
- 31-48 баллов, если студент правильно выполнил 67-83% практических заданий;
- 21-30 баллов, если студент правильно выполнил 50-66% практических заданий;
- 0-20 баллов, если студент правильно выполнил менее 50 % практических заданий.

3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.

Текущий контроль успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации.

Промежуточная аттестация проводится в форме экзамена. Экзамен проводится по расписанию промежуточной аттестации в устной форме. Объявление результатов проводится в день зачета. Результаты аттестации заносятся в ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику промежуточной аттестации, должны ликвидировать задолженность в установленном порядке.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины адресованы студентам всех форм обучения. Учебным планом предусмотрены следующие виды занятий:

- лекции;
- лабораторные работы.

Важным условием успешного освоения дисциплины «Программирование» является создание системы правильной организации труда, позволяющей распределить учебную нагрузку равномерно в соответствии с графиком образовательного процесса. Большую помощь в этом может оказать составление плана работы на семестр, месяц, неделю, день. Его наличие позволит подчинить свободное время целям учебы, трудиться более успешно и эффективно. С вечера всегда надо распределять работу на завтрашний день. В конце каждого дня целесообразно подвести итог работы: тщательно проверить, все ли выполнено по намеченному плану, не было ли каких-либо отступлений, а если были, по какой причине они произошли. Нужно осуществлять самоконтроль, который является необходимым условием успешной учебы. Если что-то осталось невыполненным, необходимо изыскать время для завершения этой части работы, не уменьшая объема недельного плана. Все задания к лабораторным работам, а также задания, вынесенные на самостоятельную работу, рекомендуется выполнять непосредственно после соответствующей темы лекционного курса, что способствует лучшему усвоению материала, позволяет своевременно выявить и устранить «пробелы» в знаниях, систематизировать ранее пройденный материал, на его основе приступить к овладению новыми знаниями и навыками.

Знакомство с дисциплиной происходит уже на первой лекции, где от студента требуется не просто внимание, но и самостоятельное оформление конспекта. При работе с конспектом лекций необходимо учитывать тот фактор, что одни лекции дают ответы на конкретные вопросы темы, другие – лишь выявляют взаимосвязи между явлениями, помогая студенту понять глубинные процессы развития изучаемого предмета, как в истории, так и в настоящее время.

Конспектирование лекций – сложный вид вузовской аудиторной работы, предполагающий интенсивную умственную деятельность студента. Конспект является полезным тогда, когда записано самое существенное и сделано это самим обучающимся. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Целесообразно вначале понять основную мысль, излагаемую лектором, а затем записать ее. Желательно запись осуществлять на одной странице листа или оставляя поля, на которых позднее, при самостоятельной работе с конспектом, можно сделать дополнительные записи, отметить непонятные места.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает лектор, отмечая наиболее важные моменты в лекционном материале замечаниями «важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек, подчеркивая термины и определения.

Целесообразно разработать собственную систему сокращений, аббревиатур и символов. Однако при дальнейшей работе с конспектом символы лучше заменить обычными словами для быстрого зрительного восприятия текста.

Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

В процессе подготовки к лабораторным занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал из-за лимита аудиторных часов. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме.

Изучение дисциплины проходит с акцентом на практическую работу. Для успешного овладения предлагаемым курсом студент должен обладать определённой информационной культурой: навыками работы с литературой, умением определять и находить информационные ресурсы, соответствующие целям и задачам образовательного процесса, получать к ним доступ и использовать в целях повышения эффективности своей профессиональной деятельности. При изучении данного курса необходимо максимально использовать компьютер, изучать дополнительные информационные ресурсы.

Подготовка к промежуточной аттестации.

При подготовке к промежуточной аттестации целесообразно:

- внимательно изучить перечень вопросов и определить, в каких источниках находятся сведения, необходимые для ответа на них;
- внимательно прочитать рекомендованную литературу;
- составить краткие конспекты ответов (планы ответов).