Управление образования администрации г. Белгорода
Муниципальное бюджетное учреждение дополнительного образования
«Центр технологического образования и детского технического творчества»
г. Белгорода

Согласовано:
Руководителем МО
«Дополнительное образование»
Л.А.Баронова
Протокол № 5 от 26.05.2022 г.

Согласовано:
Заместитель директора
МБУДО ЦТОиДТТ
Ю.С.Феоктистова
« 27» июня 2022 г.

Утверждаю
Директор МБУДО ЦТОиДТТ
Ю.Н. Кумейк
« 27» июня 2022
Приказ № 96 от 27.06.2022

Дополнительная
общеобразовательная (общеразвивающая) программа

«Функциональное программирование»

Направленность: техническая
Уровень программы: стартовый
Возраст учащихся: 12 - 17 лет
Срок реализации: 1 год

Автор – составитель:
педагог дополнительного образования
Пронькин Андрей Владимирович

г. Белгород,
2022 г.

Дополнительная общеобразовательная (общеразвивающая) программа: авторская **«Функциональное программирование»** технической направленности

Автор программы: Пронькин Андрей Владимирович

Программа рассмотрена и утверждена на заседании педагогического совета МБУДО ЦТОиДТТ
от « 27 » июня 2022 г., протокол № 10 .

Председатель _____ Ю.Н. Кумейко
(подпись) Ф.И.О.

Оглавление:

1. Комплекс основных характеристик программы
1.1. Пояснительная записка 3
1.2. Учебный план 10
1.3. Содержание программы 14
1.4. Календарный учебный график 20
1.5. Формы аттестации 21
2. Комплекс организационно – педагогических условий реализации программы
2.1. Система оценки образовательной результатов 22
2.2. Оценочные материалы 23
2.3. Материально - техническое обеспечение 29
2.4. Методическое обеспечение 30
2.5. Информационное обеспечение 30
2.6. Список методической литературы 31

Приложение
№ 1. Календарно – тематический план

1. Комплекс основных характеристик программы

1.1. Пояснительная записка

Совсем недавно компьютерное программирование казалось таинственным ремеслом, уделом специалистов. Мысль о том, что программирование может быть увлекательным занятием для каждого, большинству людей и в голову не приходила. Но мир изменился. Интернет, электронная почта, социальные сети, смартфоны и мобильные приложения, ворвавшись в нашу жизнь, преобразили ее всего за несколько лет. Мы охотно поглощаем любые плоды компьютеризации – от интернет-шопинга и развлечений до новостей и игр. Однако мы можем не только использовать эти технологии, научившись программировать, мы можем развивать их, создавать собственные произведения цифрового искусства.

Всем, что делает компьютер, управляют строки программного кода, введенные с клавиатуры. Компьютерный код похож на иностранный язык, но язык этот может освоить каждый. Учиться программировать очень интересно, ведь можно получить результаты сразу же. Более того, создание игр и программ – это отличная возможность для творчества, развития логики и интеллекта, которые важны в самых разных областях – от науки до юриспруденции. Количество вакансий, где нужно умение программировать растет с каждым днем.

Последовательное и планомерное обучение на пути освоения технического мастерства в области программирования дополняется знакомством с созданием трендов в программировании, в области сайтостроения, что имеет особое значение для развития и раскрытия своего мастерства в технической направленности в школьном возрасте.

Учащимся предоставляется возможность выбора творческой формы, художественных средств выразительности. Они приобретают опыт технической деятельности в строении сайтов, программировании, совмещают правила технической направленности с элементами фантазии.

Теоретические знания, практические занятия и развитие технического восприятия представлены в программе в их содержательном единстве.

Программа «Функциональное программирование» разработана на основе авторской программы «Видеодизайн и компьютерная графика» (автор: Пронькин А.В.).

Программа «Функциональное программирование» авторская, **технической направленности.**

Отличительные особенности данной дополнительной общеобразовательной (общеразвивающей) программы заключается в том, что она не только развивает общеучебные и общеинтеллектуальные умения и навыки, но и формирует у учащихся интерес к профессиям, связанных с программированием, способствующих и повышающих развитие технико-творческих способностей детей. Расширение кругозора учащихся,

самостоятельного подхода к решению технических вопросов под контролем педагога.

Новизна дополнительной общеобразовательной (общеразвивающей) программы состоит в поэтапном изучении основ теории программирования и применении этих основ на практике.

Актуальность дополнительной общеобразовательной (общеразвивающей) программы заключается в том, что предложенный материал способствует:

• обеспечению необходимых условий для личностного развития;
• профессиональному самоопределению;
• адаптации их к жизни в обществе;
• организации содержательного досуга;
• способствует запросу современной жизни, требованиям учебно-воспитательного процесса.

Программа «Логическое программирование» актуальна еще в том, что занятия стимулируют стремление к самостоятельной деятельности, создают условия для развития личностных качеств учащихся, способствуют подготовке детей к новым тенденциям в программировании.

Предложенный вид деятельности вызывает мотивацию познания и творчества. Занятия развивают эстетический вкус, мышление, воображение, формируют конструктивные навыки. Повышают качество проводимого после школьных занятий времени, что развивает коммуникативные умения, содействуют профилактике асоциального поведения детей и подростков.

Педагогическая целесообразность дополнительной общеобразовательной (общеразвивающей) программы состоит в создании условий для получения учащимися новых знаний и актуализацию уже имеющихся теоретических сведений по основам рекламных веб-макетов и компьютерного дизайна.

Дополнительная программа направлена на:

• дизайнерское развитие учащихся в сфере веб-рекламы, дизайна и сайтостроения;
• ознакомление с современными и перспективными технологиями дизайна;
• творческое и эстетическое развитие учащихся;
• умение находить и использовать необходимую информацию;
• выдвигать идеи решения возникающих задач при разработке веб-рекламы.

Цель программы: формирование и развитие компетенций учащихся в области программирования, приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.

Образовательные:

- сформировать навыки разработки конструирования несложных программ веб-сайта;
- дать базовые и глубокие навыки программирования на языках PHP, C++ C# Python и других языков функционального программирования;

- реализовать коммуникативные, технические способности учащихся в ходе проектирования и конструирования сайтов;
- сформировать отношение к компьютеру как инструменту для творчества, созидания, реализации своих потребностей;
- сформировать навыки элементарного проектирования, конструирования, размещения и сопровождения мультимедийного проекта;
- ознакомление учащихся с основными направлениями истории рекламы дизайна.

Развивающие:

- развивать пространственное, образное мышление учащихся;
- развивать творческий подход к делу и поиску нестандартных решений в процессе создания Web-сайта;
- развивать эстетический вкус и дизайнерское мышление;
- развивать творческие авторские начала через создание самостоятельных проектов, участие в конкурсах и олимпиадах;
- способствовать обогащению жизненного опыта и формированию системы ценностей учащихся;
- повышать грамотность учащихся в области Интернет-технологий.

Воспитательные:

- формирование у учащихся социальной активности, гражданской позиции, культуры общения и поведения в социуме, навыков здорового образа жизни;
- воспитание умения эффективно работать в команде и индивидуально над решением нестандартных задач;
- воспитание трудолюбия, коллективизма, обязательности, честности и культуры поведения, как в объединении, так и в обществе;
- воспитание бережного отношения к оборудованию.

Основные принципы обучения, используемые программой**:**

- принцип творческой активности,
- принцип сознательности,
- принцип последовательности,
- связь теории с практикой.

Использование разнообразных форм и методов обучения повышает продуктивность занятий, повышает интерес учащихся к учебному процессу. В процессе обучения применяются *групповые занятия, индивидуальные, теоретические и практические.*

При реализации программы используются следующие ***современные образовательные технологии***, основанные на личностно-ориентированном подходе: технология индивидуализации обучения, проблемное обучение, игровые технологии, технология коллективного взаимообучения, технология создания успеха, здоровьесберегающие технологии. Применение технологии создания успеха дает ребенку возможность осознать свою творческую ценность, продвигает к новым высотам творческих достижений.

Взаимодействие с родителями

Наибольшую эффективность работы в дополнительном образовании дает ***способ совместной деятельности педагога и родителей.***

Формы работы с родителями:

• Родительское собрание**.**

• Совместное посещение выставок.

• Участие в мероприятиях, проводимых в рамках образовательной программы.

Возрастные особенности детей (данная программа рассчитана на детей среднего и старшего школьного возраста (12 - 17 лет)).

Средний школьный возраст - это возраст перехода от детства к юности. В этом возрасте происходит рост и развитие всего организма. Неравномерное физическое развитие детей оказывает влияние на их поведение: они часто жестикулируют, движения порывисты, плохо координированы. Характерная черта восприятия детей среднего школьного возраста – специфическая избирательность, поэтому содержание общеобразовательной (общеразвивающей) программы подобрано с учетом интересов и познавательных возможностей учащихся. В этом возрасте идет интенсивное нравственное и социальное формирование личности. Объем данной общеобразовательной (общеразвивающей) программы соответствует возможностям и уровню развития детей данного возраста.

Старший школьный возраст - все познавательные процессы, сформированные еще в подростковом возрасте, в старшем школьном возрасте будут только укрепляться и совершенствоваться. Главное для старшего школьника теперь - выход во взрослый мир, овладение профессией, а значит, нахождение своего места в мире. Доминантой становится выбор и овладение профессией, поскольку от этого зависит дальнейшая жизнь, которую избирает человек на пороге взрослой жизни. Соответственно новая доминанта изменяет отношение к учению, заставляя соотносить свои старания и практическое их применение.

Организация образовательного процесса

Срок реализации программы: 1 год

Рекомендуемый возраст учащихся: 12-17 лет

Количество часов: 144 часа

Режим занятий: 2 раза в неделю по 2 часа

Наполняемость групп: 12-15 человек

Программа может быть использована педагогами учреждений дополнительного образования.

Основная форма проведения занятий - *учебное занятие*. Занятия состоят из теоретической и практической частей. *Теоретическая часть* занятия включает просмотр видеоуроков, просмотр и анализ работ. *Практическая часть* занятия включает общие практические занятия, индивидуальные занятия.

Возможна реализация дополнительной общеобразовательной (общеразвивающей) программы «Функциональное программирование» с применением электронного обучения и дистанционных образовательных технологий.

Уровень освоения программы – ***стартовый,*** предназначен для получения учащимися первоначальных знаний в области программирования и сопутствующих дисциплин (электроника и информатика).

Планируемые результаты программы

В результате завершения обучения по программе «Логическое программирование» учащиеся должны:

Знать:

- принципы построения Java документа и работы с его элементами;
- алгоритмы применения технологии Java
- сущность, назначение и структуру объектной модели браузера и документа;
- элементы и конструкции языка Java и способы их применения;
- принципы функционирования виртуального сервера;
- элементы и конструкции языка PHP и способы их применения для работы с сетью;
- современные технологии и средства разработки WEB-приложений;
- способы и средства публикации, поддержки, поисковой оптимизации и обновления WEB–документа.

Уметь:

- создавать описания стилей элемента документа различными способами;
- создавать разработки на языке Python
- создавать разработки серверных сценариев фреймворков на языке PHP.
- создаватьWEB-документ и работать с базовыми его элементами;
- изменять свойства документа и его элементов средствами технологии PHP
- создавать клиентские сценарии, осуществлять их внедрение в проект и тестирование;
- загружать в интернет рекламные веб-баннеры и веб-рекламу.

Личностные и метапредметные результаты освоения программы

Результаты	Формируемые умения	Средства формирования
личностные	- формирование у детей мотивации к обучению, о помощи им в самоорганизации и саморазвитии;	организация на занятий парно-групповой работы

	- развитие познавательных навыков учащихся, умений самостоятельно конструировать свои знания, ориентироваться в информационном пространстве, развитие критического и творческого мышления	
Метапредметные результаты		
регулятивные	- планировать свои действия в соответствии с поставленной задачей и условиями ее реализации, в том числе и во внутреннем плане; - осуществлять итоговый и пошаговый контроль по результату	-в сотрудничестве с педагогом ставить новые учебные задачи; - преобразовывать практическую задачу в познавательную; - проявлять познавательную инициативу в учебном сотрудничестве
познавательные	- умения учиться навыкам решения творческих задач и навыкам поиска, анализа и интерпретации информации; - добывать необходимые знания и с их помощью проделывать конкретную работу; - осуществлять поиск необходимой информации для выполнения учебных заданий с использованием учебной литературы; - осуществлять анализ объектов с выделением	- осуществлять расширенный поиск информации с использованием различных источников информации и Интернета

	существенных и несущественных признаков	
коммуникативные	- учиться выполнять различные роли в группе (лидера, исполнителя, критика); - умение координировать свои усилия с усилиями других; - формулировать собственное мнение и позицию; - договариваться и приходить к общему решению в совместной деятельности; - задавать вопросы; - допускать возможность существования у людей различных точек зрения, не совпадающих с собственной, и ориентироваться на позицию партнера в общении и взаимодействии; - учитывать разные мнения и стремиться к координации различных позиций в сотрудничестве	- учитывать разные мнения и интересы и обосновывать собственную позицию; - понимать относительность мнений и подходов к решению проблемы; - аргументировать свою позицию и координировать ее с позицией партнеров в сотрудничестве при выработке общего решения в совместной деятельности; - продуктивно разрешать конфликты на основе учета интересов и позиций всех его участников; - с учетом целей коммуникации достаточно точно, полно и последовательно передавать необходимую информацию партнеру, как ориентир для построения действия

1.2. Учебный план

№ п/п	Разделы программы и темы учебных занятий	Количество часов			Формы контроля (аттестации)
		Теория	**Практика**	**Всего**	
1.	**Вводное занятие. Инструктаж по технике безопасности. Знакомство с программами программирования**	***2***	**-**	***2***	Беседа
2.	**Django. Веб-сайты на Python (Начало)**	***10***	***10***	**20**	
2.1.	Установка виртуального окружения. Установка Django. Приложения в Django	1	1	2	Устный опрос, просмотр работ
2.2.	MVC в Django. Контроллеры и маршруты. Модели. Миграции. Основы работы с моделями. CRUD. Связи моделей.	1	1	2	
2.3.	Шаблоны. Админка Django. Внешний вид шаблона. Директивы, теги и фильтры.	1	1	2	
2.4.	Параметры в URL-запросах. Имена маршрутов. Наследование шаблонов.	1	1	2	
2.5.	Обратное разрешение адресов. Статические файлы.	1	1	2	
2.6.	Кастомные валидаторы. Класс ListView.	1	1	2	
2.7.	Класс CreateView. Подключение MySQL	1	1	2	
2.8.	Django ORM. Django Debug Toolbar	1	1	2	
2.9.	Кастомизация админки.	1	1	2	
2.10.	Постраничная навигация.	1	1	2	

3	**Фреймворк Laravel**	***20***	***20***	**40**	
3.1.	История фреймворка Laravel	1	1	2	Тестирование, просмотр работ
3.2.	Создание новых страниц на Laravel	1	1	2	
3.3.	Создание форм на основе Laravel	1	1	2	
3.4.	Контроллеры и валидация	1	1	2	
3.5.	Создание моделей и работа с базой данных	1	1	2	
3.6.	Получение данных из БД. Выполнение SQL-запросов. Конструктор запросов.	1	1	2	
3.7.	Редактирование и удаление записей	1	1	2	
3.8.	Роутинг в Laravel	1	1	2	
3.9.	Миграции в Laravel	1	1	2	
3.10.	Модели и связи моделей в Laravel	1	1	2	
3.11.	Шаблоны и сессии в Laravel. Работа с кешем.	1	1	2	
3.12.	Отправка E-mail в Laravel. Регистрация	1	1	2	
3.13.	Аутентификация	1	1	2	
3.14.	Middleware и загрузка картинок	1	1	2	
3.15.	View Composers	1	1	2	
3.16.	Pagination	1	1	2	
3.17.	Seeding	1	1	2	
3.18.	Mutators	1	1	2	
3.19.	Логирование	1	1	2	
3.20.	Шаблон админки	1	1	2	
4.	**Фреймворк Angular**	**18**	**18**	**36**	
4.1.	Установка фреймворка Angular	1	1	2	
4.2.	Определение структуры проекта	1	1	2	
4.3.	Компонент меню	1	1	2	
4.4.	Прелоадер	1	1	2	
4.5.	Маршрутизатор проекта	1	1	2	
4.6.	Форма авторизации пользователей	1	1	2	

4.7.	Передача информации об авторизированном пользователе в запросах	1	1	2	Тестирование, просмотр работ
4.8.	Альтернативный способ авторизации пользователя	1	1	2	
4.9.	Страница создания и редактирование лида или задачи.	1	1	2	
4.10.	Страница Dashboard. Вывод лидов на странице архива	1	1	2	
4.11.	Страница аналитики лидов	1	1	2	
4.12.	Компонент задач и их комментариев	1	1	2	
4.13.	Раздел управления пользователями	1	1	2	
4.14.	Стили компонентов. Интерполяция	1	1	2	
4.15.	Привязка свойств, привязка к атрибутам и событиям	1	1	2	
4.16.	Маршрутизация. Создание мультистраничности	1	1	2	
4.17.	Пайпы. Создание пайпов	1	1	2	
4.18.	Обработка форм. Template-Driven. Директива NgModel	1	1	2	
5.	**Django. Веб-сайты на Python (Продолжение)**	**10**	**10**	**20**	
5.1.	Регистрация в Django.	1	1	2	Тестирование, просмотр работ
5.2.	Авторизация в Django.	1	1	2	
5.3.	Отправка писем на E-mail	1	1	2	
5.4.	Визуальный редактор CKEditor	1	1	2	
5.5.	Капча	1	1	2	
5.6.	Кэширование	1	1	2	
5.7.	Пользовательские теги шаблона.	1	1	2	
5.8.	Работа с формами.	1	1	2	

5.9.	Класс DetailView.	1	1	2	
5.10.	Миксины	1	1	2	
6.	**Typescript**	**4**	**4**	**8**	
6.1.	Установка, компиляция и базовая настройка. Типы данных и функции	1	1	2	Тестирование, просмотр работ
6.2.	Классы. Модификаторы доступа	1	1	2	
6.3.	Наследование классов	1	1	2	
6.4.	Интерфейсы. Преобразование типов в ООП. Пространства имен. Обобщения. Миксины. Заголовочные файлы. Декораторы классов. Декораторы методов и свойств.	1	1	2	
7.	**Аттестация**	***3***	***3***	**6**	
7.1	Начальная аттестация	1	1	2	Тестирование, практическая работа
7.2	Промежуточная аттестация	1	1	2	Тестирование, практическая работа
7.3	Промежуточная аттестация	1	1	2	Тестирование, практическая работа
8.	**Подготовка к выставкам и конкурсам**	***5***	***5***	**10**	Просмотр
9.	**Итоговое занятие**	***1***	***1***	**2**	Просмотр работ
	Всего часов:	***73***	***71***	**144**	

1.3. Содержание программы

1. Вводное занятие. Инструктаж по технике безопасности. Знакомство с программами программирования – 2 часа

Теоретические сведения: Техника безопасности, правила поведения в работе с компьютерной техникой. Правила безопасного труда при работе с приборами, питающимися от сети переменного тока. Оказание первой медицинской помощи при травмах и электротравмах. Правила противопожарной безопасности и эвакуации при возникновении пожара. Обязанности учащихся по окончании занятий. Общее знакомство учащихся с программами программирования.

2. Django. Веб-сайты на Python (Начало) – 20 часов

2.1. Установка виртуального окружения. Установка Django. Приложения в Django – 2 часа

Теоретическая часть: Познакомить учащихся как устанавливается виртуальное окружение фреймворка Django языка Python.

Практическая часть: Научиться устанавливать виртуальное окружение фреймворка Django языка Python.

2.2. MVC в Django. Контроллеры и маршруты. Модели. Миграции. Основы работы с моделями. CRUD. Связи моделей – 2 часа

Теоретическая часть: Знакомство учащихся с контроллерами и маршрутами, моделями в Django.

Практическая часть: Сделать предложенные проекты веб-дизайна.

2.3. Шаблоны. Админка Django. Внешний вид шаблона. Директивы, теги и фильтры – 2 часа

Теоретическая часть: Рассказать как создаются шаблоны, админка Django, внешний вид шаблона. директивы, теги и фильтры.

Практическая часть: Сделать предложенные проекты веб-дизайна.

2.4. Параметры в URL-запросах. Имена маршрутов. Наследование шаблонов – 2 часа

Теоретическая часть: Рассказать о наследовании шаблонов

Практическая часть: Сделать предложенные проекты веб-дизайна.

2.5. Обратное разрешение адресов. Статические файлы – 2 часа

Теоретическая часть: Рассказать, что такое обратное разрешение адресов и статические файлы.

Практическая часть: Сделать предложенные проекты веб-дизайна.

2.6. Кастомные валидаторы. Класс ListView – 2 часа

Теоретическая часть: Рассказать, что такое кастомные валидаторы и класс ListView.

Практическая часть: Сделать предложенные проекты веб-дизайна.

2.7. Класс CreateView. Подключение MySQL – 2 часа

Теоретическая часть: Рассказать о классе CreateView и как подключать MySQL в Django.

Практическая часть: Сделать предложенные проекты веб-дизайна.

2.8. Django ORM. Django Debug Toolbar – 2 часа

Теоретическая часть: Рассказать, что такое Dango ORM и Django Debug Toolbar.

Практическая часть: Сделать предложенные проекты веб-дизайна.

2.9. Кастомизация админки – 2 часа

Теоретическая часть: Рассказать как делается кастомизация админки.

Практическая часть: Сделать предложенные проекты веб-дизайна.

2.10. Постраничная навигация – 2 часа

Теоретическая часть: Рассказать о создании постраничной навигации.

Практическая часть: Сделать предложенные проекты веб-дизайна.

3. Фреймворк Laravel – 40 часов

3.1. История фреймворка Laravel – 2 часа

Теоретическая часть: Рассказать об истории фреймворка Laravel языка PHP.

Практическая часть: Сделать предложенные проекты.

3.2. ***Создание новых страниц на Laravel – 2 часа***

Теоретическая часть: Изучение создания новых страниц на Laravel.

Практическая часть: Сделать предложенные проекты.

3.3. Создание форм на основе Laravel – 2 часа

Теоретическая часть: Изучение создания форм на основе Laravel

Практическая часть: Сделать предложенные проекты.

3.4. Контроллеры и валидация – 2 часа

Теоретическая часть: Рассказать, как применять контроллеры и валидацию.

Практическая часть: Выполнение предложенных проектов.

3.5. Создание моделей и работа с базой данных – 2 часа

Теоретическая часть: Рассказать о создании моделей и работе с базой данных.

Практическая часть: Выполнение предложенных проектов.

3.6. Получение данных из БД. Выполнение SQL-запросов. Конструктор запросов. – 2 часа

Теоретическая часть: Рассказать, как сделать получение данных из БД. Выполнение SQL-запросов. Конструктор запросов.

Практическая часть: Выполнить предложенные проекты.

3.7. Редактирование и удаление записей – 2 часа

Теоретическая часть: Рассказать о редактировании и удалении записей.

Практическая часть: Изучение и выполнение проектов по теме.

3.8. Роутинг в Laravel – 2 часа

Теоретическая часть: Изучение роутинга в Laravel

Практическая часть: Сделать предложенные проекты.

3.9. Миграции в Laravel – 2 часа

Теоретическая часть: Рассказать миграциях в Laravel

Практическая часть: Выполнить предложенные проекты.

3.10. Модели и связи моделей в Laravel – 2 часа

Теоретическая часть: Рассказать о связях моделей

Практическая часть: Сделать предложенные проекты.

3.11. Шаблоны и сессии в Laravel. Работа с кешем – 2 часа

Теоретическая часть: Рассказать, как работают шаблоны и сессии.

Практическая часть: Сделать предложенные проекты.

3.12. Отправка E-mail в Laravel. Регистрация – 2 часа

Теоретическая часть: Как с помощью PHP посылаются письма.

Практическая часть: Сделать предложенные проекты.

3.13. Аутентификация – 2 часа

Теоретическая часть: Рассказать о аутентификации.

Практическая часть: Сделать предложенные проекты.

3.14. Middleware и загрузка картинок – 2 часа

Теоретическая часть: Рассказать, как загружать картинки.

Практическая часть: Сделать предложенные проекты.

3.15. View Composers – 2 часа

Теоретическая часть: Рассказать о работе View Composers.

Практическая часть: Сделать предложенные проекты.

3.16. Pagination – 2часа

Теоретическая часть: Изучение темы о Pagination.

Практическая часть: Сделать предложенные проекты.

3.17. Seeding – 2 часа

Теоретическая часть: Рассказать, что такое Seeding.

Практическая часть: Выполнить предложенные проекты.

3.18. Mutators – 2 часа

Теоретическая часть: Рассказать о Mutators.

Практическая часть: Сделать предложенные проекты.

3.19. Логирование – 2 часа

Теоретическая часть: Рассказать, что такое логирование.

Практическая часть: Выполнить предложенные проекты.

3.20. Шаблон админки – 2 часа

Теоретическая часть: Рассказать о шаблоне админки в Laravel

Практическая часть: Сделать предложенные проекты.

4. Установка фреймворка Angular *– 36 часов*

4.1. Установка фреймворка Angular – 2 часа

Теоретическая часть: Рассказать, как установить фреймворк Angular.

Практическая часть: Закрепление материала на практике.

4.2. Определение структуры проекта – 2 часа

Теоретическая часть: Рассказать определение структуры проекта

Практическая часть: Сделать предложенные проекты.

4.3. Компонент меню – 2 часа.

Теоретическая часть: Рассказать, как сделать компонент меню.

Практическая часть: Сделать предложенные проекты.

4.4. Прелоадер – 2 часа

Теоретическая часть: Рассказать, что такое прелоадер.

Практическая часть: Сделать предложенные проекты.

4.5. Маршрутизатор проекта – 2 часа

Теоретическая часть: Рассказать, что такое маршрутизатор проекта.

Практическая часть: Сделать предложенные проекты.

4.6. Форма авторизации пользователей – 2 часа

Теоретическая часть: Рассказать, как авторизовать пользователей.

Практическая часть: Сделать предложенные проекты.

4.7. Передача информации об авторизированном пользователе в запросах – 2 часа

Теоретическая часть: Рассказать о передаче информации об авторизированном пользователе в запросах.

Практическая часть: Выполнить предложенные проекты.

4.8. Альтернативный способ авторизации пользователя – 2 часа

Теоретическая часть: Рассказать об альтернативном способе авторизации пользователя.

Практическая часть: Сделать предложенные проекты.

4.9. Страница создания и редактирование лида или задачи. – 2 часа

Теоретическая часть: Рассказать, как создается и редактируется страница лида или задачи.

Практическая часть: Сделать предложенные проекты.

4.10. Страница Dashboard. Вывод лидов на странице архива – 2 часа

Теоретическая часть: Рассказать, как создается страница Dashboard и как вывести лидов на странице архива.

Практическая часть: Сделать предложенные проекты.

4.11. Страница аналитики лидов – 2 часа

Теоретическая часть: Рассказать, что такое страница аналитики лидов.

Практическая часть: Выполнить предложенные проекты.

4.12. Компонент задач и их комментариев – 2 часа

Теоретическая часть: Рассказать о компоненте задач и их комментариев.

Практическая часть: Сделать предложенные проекты.

4.13. Раздел управления пользователями – 2 часа

Теоретическая часть: Рассказать о разделе управления пользователями.

Практическая часть: Сделать предложенные проекты.

4.14. Стили компонентов. Интерполяция – 2 часа

Теоретическая часть: Рассказать о стилях компонентов и интерполяции.

Практическая часть: Сделать предложенные проекты.

4.15. Привязка свойств, привязка к атрибутам и событиям – 2 часа

Теоретическая часть: Рассказать о привязке свойств к атрибутам и событиям.

Практическая часть: Сделать предложенные проекты

4.16. Маршрутизация. Создание мультистраничности – 2 часа

Теоретическая часть: Рассказать о создании мультистраничности.

Практическая часть: Сделать предложенные проекты.

4.17. Пайпы. Создание пайпов – 2 часа

Теоретическая часть: Рассказать о пайпах и их создании.

Практическая часть: Сделать предложенные проекты.

4.18. Обработка форм. Template-Driven. Директива NgModel – 2 часа

Теоретическая часть: Рассказать, как обрабатывать формы.

Практическая часть: Сделать предложенные проекты.

5. Django. Веб-сайты на Python (Продолжение)– 20 часов

5.1. Регистрация в Django – 2 часа.

Теоретическая часть: Рассказать, как регистрироваться в Django.

Практическая часть: Сделать предложенные проекты.

5.2. Авторизация в Django – 2 часа

Теоретическая часть: Рассказать, что такое авторизация в Django.

Практическая часть: Сделать предложенные проекты.

5.3. Отправка писем на E-mail – 2 часа

Теоретическая часть: Рассказать, как делается отправка писем на E-mail в Django.

Практическая часть: Сделать предложенные проекты.

5.4. Визуальный редактор CKEditor - 2 часа

Теоретическая часть: Рассказать, что такое визуальный редактор CKEditor.

Практическая часть: Сделать предложенные проекты.

5.5. Капча – 2 часа

Теоретическая часть: Рассказать, как делается капча.

Практическая часть: Сделать предложенные проекты.

5.6. Кэширование – 2 часа

Теоретическая часть: Рассказать, что такое кеширование и как его делать.

Практическая часть: Сделать предложенные проекты.

5.7. Пользовательские теги шаблона – 2 часа

Теоретическая часть: Рассказать, что такое пользовательские теги шаблона.

Практическая часть: Сделать предложенные проекты.

5.8. Работа с формами – 2 часа

Теоретическая часть: Рассказать как работать с формами .

Практическая часть: Сделать предложенные проекты.

5.9. Класс DetailView - 2 часа

Теоретическая часть: Рассказать, что такое класс DetailView.

Практическая часть: Сделать предложенные проекты.

5.10. Миксины – 2 часа

Теоретическая часть: Рассказать, что такое миксины.

Практическая часть: Сделать предложенные проекты

6. Typescript – 8 часов

6.1. Установка, компиляция и базовая настройка. Типы данных и функции – 2 часа

Теоретическая часть: Рассказать об установке, компиляции и базовой настройке языка программирования.

Практическая часть: Сделать предложенные проекты.

6.2. Классы. Модификаторы доступа – 2 часа

Теоретическая часть: Рассказать о классах и модификаторах доступа.

Практическая часть: Сделать предложенные проекты.

6.3. Наследование классов – 2 часа

Теоретическая часть: Научиться наследовать классы.

Практическая часть: Сделать предложенные проекты.

6.4. Интерфейсы. Преобразование типов в ООП. Пространства имен. Обобщения. Миксины. Заголовочные файлы. Декораторы классов. Декораторы методов и свойств – 2 часа.

Теоретическая часть: Рассказать, как работать интерфейсы, преобразование типов в ООП и т.д.

Практическая часть: Сделать предложенные проекты.

7. Аттестация – 6 часов

7.1. Начальная аттестация (сентябрь) – 2 часа

7.2. Промежуточная аттестация (декабрь) – 2 часа

7.3. Промежуточная аттестация (май) – 2 часа

8. Подготовка к выставкам и конкурсам – 10 часов

9. Итоговое занятие – 2 часа

Теоретическая часть: Подведение итогов работы.

Практическая часть: Просмотр лучших работ учащихся.

1.4. Календарный учебный график реализации дополнительной общеобразовательной (общеразвивающей) программы «Функциональное программирование»

Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
сентябрь	май	32	72	144	2 раза в неделю по 2 часа

1.5. Формы аттестации учащихся

Для определения уровня усвоения программы учащимися, ее дальнейшей корректировки и определения путей достижения каждым ребенком максимального творческого и личностного развития предусмотрена *аттестация учащихся*.

Аттестация:

- начальная аттестация (сентябрь);
- промежуточная аттестация (декабрь, май).

При наборе учащихся проводится **начальная аттестация**, в ходе которой педагог проводит *тестирование и практическую работу*, по результатам которого узнает уровень подготовки учащихся к занятиям.

Формы промежуточной аттестации: теоретическая часть - *тестирование,* практическая часть - *практическая работа*.

Тестирование состоит из вопросов по содержанию разделов программы с вариантами ответов. *Практическая работа* предполагает задания по пройденному материалу.

Аттестация	Сроки	Теория	Практика
Начальная	сентябрь	Тестирование	Практическая работа
Промежуточная	декабрь	Тестирование	Практическая Работа
Промежуточная	май	Тестирование	Практическая Работа

2. Комплекс организационно – педагогических условий реализации программы

2.1. Система оценки образовательных результатов

Оценка теоретических знаний и практических умений и навыков учащихся по теории и практике проходит по трем уровням: **высокий, средний, низкий.**

Высокий уровень – учащиеся должны грамотно излагать программный материал, и самостоятельно выполнять практическое задание.

Средний уровень – учащиеся должны уметь правильно, грамотно и по существу излагать программный материал, не допуская существенных неточностей в работе.

Низкий уровень – учащиеся не знают значительной части материала, допускают существенные ошибки, с большими затруднениями выполняют практическое задание.

При обработке результатов учитываются **критерии** для выставления уровней:

Высокий уровень – выполнение 100% - 70%;

Средний уровень – выполнение от 50% до 70%;

Низкий уровень - выполнение менее 50%.

Подведение итогов реализации дополнительной общеобразовательной (общеразвивающей) программы осуществляется в форме конкурсов, участия в тематических выставках различного уровня, где выставляются лучшие видеопроекты.

2.2. Оценочные материалы

Вопросы для начальной аттестации
(сентябрь)

Теоретическая часть: ***тестирование***

1. Что такое Photoshop?
1) профессиональная программа редактирования фото
2) программа создания видеообложек
3) программа цифрового скульптинга
4) видеомонтажная программа

2. Что такое клавиша NumLock?
1) клавиша удаления текста
2) клавиша восстановления файла
3) клавиша вспомогательной панели введения цифр
4) клавиша восстановления компьютера к заводским настройкам

3. Что из себя представляет первоначально любой язык программирования?
1) структура кода
2) структура текста
3) объемное изображение

4. Чем отличается компилятор от IDE?
1) переводчик кода
2) параллельный интерфейс подключения накопителей (жёстких дисков и оптических приводов) к компьютеру.
3) обозначение системы непосредственного впрыска топлива в бензиновых двигателях от компании Renault.

4) система программных средств, используемая программистами для разработки программного обеспечения.

5. Что такое Sata 3?
1) последовательный интерфейс обмена данными с накопителями информации.
2) сокет
3) ЦПУ
4) Один из разделов BIOSa

6. Что такое Boot-menu?
1) Диспетчер загрузки системы Windows в BIOSe
2) название элемента транзистора
3) Реле советского лампового телевизора
4) название части электрического предохранителя

7. Чему равен 1 Гб?
1) 8 бит
2) 1024 байт
3) 1024 Мбайт
4) 1000 байт

8. Чему равен 1 ТБ?
1) 1024 Гб
2) 1024 Тб
3) 1024 байт
4) 8 бит

Практическая часть: *практическая работа*
Создать афишу в Photoshope.

Вопросы для промежуточной аттестации
(декабрь)
1 полугодие

Теоретическая часть: ***тестирование***

1.Что такое роутинг?
1) Маршруты
2) Дороги
3) Блок питания

2.Что такое пагинация?
1) высокопроизводительный способ разбиения на страницы
2) JUnit
3) JHipster

3. Что такое миграции?
1) контроль версий для вашей базы данных, позволяющий вашей команде изменять и совместно использовать схему базы данных приложения
2) способы улучшения жизни
3) способы ухудшения состояния

4. Что такое кэширование?
1) процесс сохранения данных локально, который позволяет быстрее получить к ним доступ при будущих запросах
2) название компании, выпускающая компьютерную технику
3) видеокомпания

5. Что такое модификаторы доступа

1) сокрытие состояние объекта от внешнего доступа и управлять доступом к этому состоянию.
2) Arc2D.Float
3) PiePanel

6.Что такое DetailView?
1) Класс
2) Метод
3) Функция

7. Что такое интерполяция?
1) метод построения точек данных между заданными точками данных
2) построение списков
3) построение кортежей

8. Что такое Dashboard в Angular?
1) Страница
2) Электронная запись
3) Функция

9. Кто такие лиды?
1) потенциальный клиент, который заинтересовался товаром или услугой и оставил свои контакты
2) субкультура
3) группа подростков

10. что такое декораторы?
1) обычная функция-обертка
2) методы
3) классы

Практическая часть: ***практическая работа***
Создать простой одностраничник на Python

Вопросы для промежуточной аттестации
(май)
2 полугодие

1.Что такое фреймворк Laravel?
1) программная платформа, определяющая структуру программной системы
2) автомобильная компания
3) марка производства компьютеров

2. Какого языка программирования фреймворк Angular?
1) PHP
2) Python
3) CSS

3. Что такое Django?
1) Фреймворк
2) Язык программирования
3) Программное обеспечение

4. Какому языку программирования принадлежит Django?
1) HTML
2) Java
3) Python

5. Что такое Typescript?
1) Язык программирования
2) Фреймворк для C++
3) Библиотека для Kotlin

6. Чем отличается Typescript?
1) возможностью явного статического назначения типов
2) объектами с возможностью интроскепции;
3) автоматической сборкой мусора

7.Что такое логирование?
1) обработка ошибок
2) Java-пакет классов для управления учетными записями
3) Java-пакет набора классов для работы с технологией Bluetooth

8. Что такое пайпы в Angulare?
1) декоратор, который позволяет форматировать отображаемое значение.

2) подключаемый модуль Android Studio, который помогает создавать Android-приложения
3) подключаемый модуль Visual Studio, который помогает создавать Android-приложения

9. Что такое прелоадер?
1) анимированные изображения, представляющие процесс загрузки страницы сайта.
2) поршневой механизм
3) механизм турбины

10. Что такое миксины?
1) исключение дублирования кода
2) ускоряются электроны
3) происходит выброс радиоактивного мусора в безвоздушное пространство

Практическая часть: ***практическая работа***
Создать простую страницу на Django

2.3. Материально-техническое обеспечение

1. Компьютерное оборудование.
2. Программы веб-дизайна.
3. Компиляторы программирования.
4. Проектор.
5. Тетради и ручки для конспекта, флеш-накопители.

2.4. Методическое обеспечение программы

За основу программы «Функциональное программмирование» взяты такие методы и формы занятий, как *мотивация и стимулирование*, когда на первых занятиях педагог формирует интерес учащихся к обучению, используя при этом:

- словесные, наглядные, аудиовизуальные, практические занятия;
- методы эмоционального стимулирования;
- творческие задания;
- анализ, обобщение, систематизацию полученных знаний и умений;
- выполнение работ под руководством педагога; самостоятельную работу;
- контроль в виде анализа и коррекции работы.

Методы, используемые на занятии характеризуются постепенным смещением акцентов с *репродуктивных на саморазвивающиеся, продуктивные,* с фронтальных на *групповые и индивидуальные*.

Наряду с традиционными *формами организации занятий,* такими, как *рассказ, беседа*, программой предусмотрены и *нетрадиционные:*

- интегрированные занятия, основанные на межпредметных связях;

В основу всех форм учебных занятий заложены *общие характеристики*:

- каждое занятие имеет цель, конкретное содержание, определенные методы организации учебной деятельности;
- построение учебного занятия осуществляется по определенной логике, когда тип занятия соответствует его цели и задачам.

2.5. Информационное обеспечение программы

Интернет-ресурсы:

1. https://www.helloworld.ru/texts/comp/lang/java/java5/vol1/ch5.html

2. https://revall.info/
3. https://codernet.ru/books/
4. https://obuchalka.org/knigi-po-programmirovaniu/
5. https://aldebaran.ru/genre/kompyuteryi/programmirovanie/

2.6. Список методической литературы

1. Стаффер Мэтт Laravel. Полное руководство. 2-е изд. — СПб.: Питер, 2020. — 512 с.: ил. (Серия «Бестселлеры O'Reilly»)
2. Файн Я., Моисеев А. Angular и TypeScript. Сайтостроение для профессионалов. — СПб.: Питер, 2018. — 464 с.: ил. — (Серия «Библиотека программиста»)
3. Дронов В. А. Django 3.0. Практика создания веб-сайтов на Python. — СПб.: БХВ-Петербург, 2021. — 704 с.: ил. — (Профессиональное программирование)
4. Форсье Дж, Чан У. Django: Разработка веб-приложений на Python – Пер. с англ. – СПб: Символ-плюс, 2010 – 456 с.
5. Головатый А., Каплан-Мосс Дж. Django. Подробное руководство, 2-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2010. – 560 с.