

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Ващенко Андрей Александрович  
Должность: Ректор  
Дата подписания: 16.07.2024 11:10:42  
Уникальный программный ключ:  
51187754f94e37d00c9236cc9eaf21a22f0a3b731acd32879ec947ce3c66589d

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОЛГОГРАДСКИЙ ИНСТИТУТ БИЗНЕСА»**

Утверждаю  
Проректор по учебной работе  
и управлению качеством  
\_\_\_\_\_ Л.В. Шамрай-Курбатова  
«15» июля 2024 г.

**Аннотации модулей дополнительной общеобразовательной общеразвивающей программы для детей в возрасте 10-15 лет и взрослых  
технической направленности «Программирование»**

**Сроки освоения – 24 часа.**

**Краткая характеристика профессиональной деятельности с учетом положений профессионального стандарта 06.001 «Программист»,  
утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 года N 424н, заявленными в  
образовательной программе:**

**Область профессиональной деятельности слушателей, освоивших дополнительную общеобразовательную общеразвивающую  
программу для детей в возрасте 10-15 лет и взрослых технической направленности «Программирование»: обучающиеся в возрасте 10-  
15 лет и взрослые, интересующиеся программированием.**

**I. Общие сведения**

**Вид профессиональной деятельности:**

62.01 - Разработка компьютерного программного обеспечения

**Основная цель вида профессиональной деятельности:**

- Разработка, отладка, проверка работоспособности, модификация компьютерного программного обеспечения.

**II. Описание трудовых функций, входящих в профессиональный стандарт (функциональная карта вида профессиональной  
деятельности):**

- Разработка и отладка программного кода.
- Проверка работоспособности и рефакторинг кода программного обеспечения.

### **III. Характеристика обобщенных трудовых функций.**

**Профстандарт: 06.001 «Программист», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 года N 424н**

**3.1. Обобщенная трудовая функция:** Разработка и отладка программного кода.

**Возможные наименования должностей:**

- Помощник программиста

**Требования к образованию и обучению.**

- Среднее профессиональное образование - программы подготовки квалифицированных рабочих (служащих);

**Требования к опыту практической работы** – нет.

**Особые условия допуска к работе** – нет.

**Другие характеристики** – нет.

**Наименование базовой группы, должности (профессии) или специальности:**

- Специалисты-техники по поддержке пользователей ИКТ
- Техник-программист
- Информатика и вычислительная техника

**3.1.1. Трудовая функция.** Формализация и алгоритмизация поставленных задач для разработки программного кода.

**Трудовые действия:**

- Составление формализованных описаний решений поставленных задач в соответствии с требованиями технического задания или внутренних документов организации
- Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или внутренних документов организации
- Проверка корректности алгоритмов решения поставленных задач
- Оценка и согласование сроков выполнения поставленных задач

**Необходимые умения:**

- Использовать методы и приемы формализации поставленных задач
- Использовать методы и приемы алгоритмизации поставленных задач
- Использовать программное обеспечение для графического отображения алгоритмов
- Применять алгоритмы решения типовых задач в соответствующих областях
- Осуществлять коммуникации с заинтересованными сторонами

**Необходимые знания:**

- Методы и приемы формализации поставленных задач
- Языки формализации функциональных спецификаций
- Методы и приемы алгоритмизации поставленных задач
- Нотации и программное обеспечение для графического отображения алгоритмов

- Алгоритмы решения типичных задач, области и способы их применения

### **3.1.2. Трудовая функция. Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных**

#### **Трудовые действия:**

- Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)
- Оптимизация программного кода с использованием специализированных программных средств
- Оценка и согласование сроков выполнения поставленных задач

#### **Необходимые умения:**

- Применять выбранные языки программирования для написания программного кода Использовать методы прогнозирования сбыта продукции и рынков
- Использовать выбранную среду программирования и средства системы управления базами данных
- Использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода
- Осуществлять коммуникации с заинтересованными сторонами

#### **Необходимые знания:**

- Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования
- Методологии разработки компьютерного программного обеспечения
- Методологии и технологии проектирования и использования баз данных
- Технологии программирования
- Особенности выбранной среды программирования и системы управления базами данных
- Компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними

### **3.1.3. Трудовая функция. Оформление программного кода в соответствии с установленными требованиями**

#### **Трудовые действия:**

- Приведение наименований переменных, функций, классов, структур данных и файлов в соответствие с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода
- Структурирование исходного программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода
- Комментирование и разметка программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода
- Форматирование исходного программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода
- Оформление технической документации на компьютерное программное обеспечение по заданному стандарту или шаблону

#### **Необходимые умения:**

- Применять нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода
- Применять инструментарий для создания и актуализации исходных текстов программ
- Применять заданные стандарты и шаблоны для составления и оформления технической документации
- Осуществлять коммуникации с заинтересованными сторонами

**Необходимые знания:**

- Инструментарий для создания и актуализации исходных текстов программ
- Методы повышения читаемости программного кода
- Системы кодировки символов, форматы хранения исходных текстов программ
- Нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода
- Основные стандарты оформления технической документации на компьютерное программное обеспечение

**3.1.4. Трудовая функция.** Работа с системой управления версиями программного кода

**Трудовые действия:**

- Регистрация изменений исходного текста программного кода в системе управления версиями
- Слияние, разделение и сравнение исходных текстов программного кода
- Сохранение сделанных изменений программного кода в соответствии с регламентом управления версиями

**Необходимые умения:**

- Использовать выбранную систему управления версиями
- Использовать вспомогательные инструментальные программные средства для обработки исходного текста программного кода
- Выполнять действия, соответствующие установленному регламенту используемой системы управления версиями
- Осуществлять коммуникации с заинтересованными сторонами

**Необходимые знания:**

- Возможности используемой системы управления версиями и вспомогательных инструментальных программных средств
- Установленный регламент использования системы управления версиями

**3.1.5. Трудовая функция.** Проверка и отладка программного кода

**Трудовые действия:**

- Анализ и проверка исходного программного кода
- Отладка программного кода на уровне программных модулей
- Отладка программного кода на уровне межмодульных взаимодействий и взаимодействий с окружением
- Оценка и согласование сроков выполнения поставленных задач

**Необходимые умения:**

- Выявлять ошибки в программном коде
- Применять методы и приемы отладки программного кода
- Интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов

- Применять современные компиляторы, отладчики и оптимизаторы программного кода
- Осуществлять коммуникации с заинтересованными сторонами

**Необходимые знания:**

- Методы и приемы отладки программного кода
- Типы и форматы сообщений об ошибках, предупреждений
- Способы использования технологических журналов, форматы и типы записей журналов
- Современные компиляторы, отладчики и оптимизаторы программного кода
- Сообщения о состоянии аппаратных средств

**3.2. Обобщенная трудовая функция.** Проверка работоспособности и рефакторинг кода программного обеспечения.

**Возможные наименования должностей:**

- Техник-программист

**Требования к образованию и обучению:**

- Среднее профессиональное образование - программы подготовки квалифицированных рабочих (служащих)

**Требования к опыту практической работы**

- Не менее трех месяцев в области разработки компьютерного программного обеспечения

**Особые условия допуска к работе** – нет.

**Другие характеристики** – нет.

**Наименование базовой группы, должности (профессии) или специальности:**

- Специалисты-техники по поддержке пользователей ИКТ
- Техник-программист
- Информатика и вычислительная техника

**3.2.1. Трудовая функция. Разработка процедур проверки работоспособности и измерения характеристик компьютерного программного обеспечения**

**Трудовые действия:**

- Разработка процедуры проверки работоспособности компьютерного программного обеспечения
- Разработка процедуры сбора диагностических данных проверки работоспособности компьютерного программного обеспечения
- Разработка процедуры измерения требуемых характеристик компьютерного программного обеспечения
- Оформление технической документации на компьютерное программное обеспечение по заданному стандарту или шаблону
- Оценка и согласование сроков выполнения поставленных задач

**Необходимые умения:**

- Писать программный код процедур проверки работоспособности компьютерного программного обеспечения на выбранном языке программирования

- Использовать выбранную среду программирования для разработки процедур проверки работоспособности компьютерного программного обеспечения
- Применять заданные стандарты и шаблоны для составления и оформления технической документации
- Осуществлять коммуникации с заинтересованными сторонами

**Необходимые знания:**

- Методы автоматической и автоматизированной проверки работоспособности компьютерного программного обеспечения
- Основные виды диагностических данных проверки работоспособности компьютерного программного обеспечения и способы их представления
- Языки, утилиты и среды программирования и средства пакетного выполнения процедур
- Типовые метрики компьютерного программного обеспечения
- Основные методы измерения и оценки характеристик компьютерного программного обеспечения
- Основные стандарты оформления технической документации на компьютерное программное обеспечение

### **3.2.2. Трудовая функция. Разработка тестовых наборов данных для проверки работоспособности компьютерного программного обеспечения**

**Трудовые действия:**

- Подготовка тестовых наборов данных в соответствии с выбранной методикой тестирования компьютерного программного обеспечения
- Оценка и согласование сроков выполнения поставленных задач

**Необходимые умения:**

- Разрабатывать и оформлять контрольные примеры для проверки работоспособности компьютерного программного обеспечения
- Разрабатывать процедуры генерации тестовых наборов данных с заданными характеристиками
- Подготавливать наборы данных, используемых в процессе проверки работоспособности компьютерного программного обеспечения
- Использовать командные средства разработки компьютерного программного обеспечения
- Осуществлять коммуникации с заинтересованными сторонами

**Необходимые знания:**

- Методы создания и документирования контрольных примеров и тестовых наборов данных
- Правила, алгоритмы и технологии создания тестовых наборов данных
- Требования к структуре и форматам хранения тестовых наборов данных

### **3.2.3. Трудовая функция. Проверка работоспособности компьютерного программного обеспечения**

**Трудовые действия:**

- Проверка работоспособности компьютерного программного обеспечения на основе разработанных тестовых наборов данных

- Оценка соответствия компьютерного программного обеспечения требуемым характеристикам
- Сбор и анализ полученных результатов проверки работоспособности компьютерного программного обеспечения
- Оценка и согласование сроков выполнения поставленных задач

**Необходимые умения:**

- Применять методы и средства проверки работоспособности компьютерного программного обеспечения
- Интерпретировать диагностические данные проверки работоспособности компьютерного программного обеспечения
- Анализировать значения полученных характеристик компьютерного программного обеспечения
- Документировать результаты проверки работоспособности компьютерного программного обеспечения
- Осуществлять коммуникации с заинтересованными сторонами

**Необходимые знания:**

- Методы и средства проверки работоспособности компьютерного программного обеспечения
- Среды проверки работоспособности и отладки компьютерного программного обеспечения
- Государственные стандарты испытания автоматизированных систем
- Руководящие документы по стандартизации требований к документам автоматизированных систем

### **3.2.4. Трудовая функция. Рефакторинг, оптимизация и инспекция программного кода**

**Трудовые действия:**

- Анализ программного кода на соответствие требованиям по читаемости и производительности
- Инспекция программного кода для поиска не обнаруженных на ранних стадиях разработки компьютерного программного обеспечения ошибок и критических мест
- Внесение изменений в программный код и проверка его работоспособности
- Оценка и согласование сроков выполнения поставленных задач

**Необходимые умения:**

- Применять методы, средства рефакторинга, оптимизации и инспекции программного кода
- Применять инструментальные средства коллективной работы над программным кодом
- Публиковать результаты рефакторинга, оптимизации и инспекции в коллективной базе знаний
- Использовать систему управления версиями для регистрации произведенных изменений
- Осуществлять коммуникации с заинтересованными сторонами

**Необходимые знания:**

- Методы и средства рефакторинга, оптимизации и инспекции программного кода
- Языки программирования и среды разработки
- Нормативно-технические документы (стандарты и регламенты), регламентирующие требования к программному коду, порядок отражения изменений в системе управления версиями, порядок отражения результатов рефакторинга, оптимизации и инспекции в коллективной базе знаний

–

### **3.2.5. Трудовая функция. Исправление дефектов программного кода, зафиксированных в базе данных дефектов**

#### **Трудовые действия:**

- Воспроизведение дефектов программного кода, зафиксированных в базе данных дефектов
- Установление причин возникновения дефектов программного кода
- Внесение изменений в программный код для устранения выявленных дефектов
- Оценка и согласование сроков выполнения поставленных задач

#### **Необходимые умения:**

- Применять методы и приемы отладки дефектного программного кода
- Интерпретировать сообщения, предупреждения, записи технологических журналов об ошибках, возникающих при выполнении дефектного кода
- Осуществлять коммуникации с заинтересованными сторонами

#### **Необходимые знания:**

- Методы и приемы отладки программного кода
- Типичные ошибки, возникающие при разработке компьютерного программного обеспечения, методы их диагностики и исправления

### **3.2.6. Трудовая функция. Осуществление сборки однородных программных модулей в программный проект**

#### **Трудовые действия:**

- Сборка однородных (одноязыковых) программных модулей в программный проект в средах разработки компьютерного программного обеспечения
- Проверка работоспособности собранного программного проекта
- Внесение изменений в процедуру сборки однородных (одноязыковых) программных модулей в программный проект
- Оценка и согласование сроков выполнения поставленных задач

#### **Необходимые умения:**

- Выполнять процедуры сборки однородных (одноязыковых) программных модулей в программный проект в средах разработки компьютерного программного обеспечения
- Производить настройки параметров программного проекта и осуществлять запуск процедур сборки
- Проводить проверку работоспособности программного проекта
- Документировать произведенные действия, выявленные проблемы и способы их устранения
- Создавать резервные копии программного проекта и данных, выполнять восстановление, обеспечивать целостность программного проекта и данных
- Осуществлять коммуникации с заинтересованными сторонами

#### **Необходимые знания:**

- Методы и средства сборки модулей в программный проект в средах разработки компьютерного программного обеспечения
- Возможности настройки программного проекта в средах разработки компьютерного программного обеспечения



- Интерфейсы взаимодействия внутренних модулей программного проекта
- Методы и средства проверки работоспособности программных проектов
- Языки, утилиты и среды программирования, средства пакетного выполнения процедур
- 

#### Аннотации модулей, входящих в образовательную программу

№ п/п	Трудоемкость (в часах)	Аннотации тем
1	24	<p><b>Тема 1. Сущность и область использования языка Python</b>  Python как объектно-ориентированный язык программирования, разработанный на языке С. История создания языка Python. Особенности, преимущества, сфера применения и перспективы. Спрос на специалистов Python. Инструкции и структура программы.</p> <p><b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– особенности, сферу применения языка Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul> <p><b>Тема 2. Ввод и вывод данных</b>  Установка оболочки программы. Запуск оболочки программы и инструкции языка Python. Исходные файлы Python. Доступ к документации. Функции для ввода данных и печати значений. Типы объектов языка Python. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций.  Первая программа на языке Python. Основы алгоритмизации.</p> <p><b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul> <p><b>Тема 3. Условия</b>  Типы данных. Преобразование типов. Различия типов данных.  Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not).  Синтаксис условной инструкции. Альтернативное выполнение. Вложенные условные инструкции. Операторы сравнения. Каскадные условные инструкции.</p>

№ п/п	Трудоемкость (в часах)	Аннотации тем
		<p>Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.</p> <p><b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul> <p><b>Тема 4. Вычисления</b></p> <p>Целочисленная арифметика. Действительные числа. Базовые функции (abs, round, int, math). Библиотека math.</p> <p><b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul> <p><b>Тема 5. Цикл for</b></p> <p>Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Цикл for. Функция range. Настройка функции print().</p> <p>Операторы управления циклом. Пример задач с использованием цикла for.</p> <p>Вложенные циклы. Циклы в циклах. Случайные циклы. Функция randrange. Функция random.</p> <p><b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul>

№ п/п	Трудоемкость (в часах)	Аннотации тем
		<p><b>Тема 6. Строки</b>  Строки. Доступ по индексу и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.  Срезы (slices). Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи).  Методы. Методы find и rfind. Метод replace. Метод count.  Примеры решения задач со строками.  <b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul> <p><b>Тема 7. Цикл while</b>  Цикл while. Инструкции управления циклом. Множественное присваивание.  Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления.  Примеры использования цикла while.  <b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul> <p><b>Тема 8. Списки</b>  Списки. Тип списков (list). Индексы. Обход списков. Операции со списками. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Удаление списка. Клонирование списка. Списочные параметры.  Методы split и join. Генераторы списков. Срезы списков.  Функция range. Примеры решения задач.</p>

№ п/п	Трудоемкость (в часах)	Аннотации тем
		<p><b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul> <p><b>Тема 9. Функции и рекурсия</b> Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Рекурсия. Вычисление факториала. Примеры решения задач.</p> <p><b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul> <p><b>Тема 10. Двумерные массивы</b> Обработка и вывод вложенных списков. Создание вложенных списков. Ввод двумерного массива. Пример обработки двумерного массива. Вложенные генераторы двумерных массивов. Примеры решения задач.</p> <p><b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul> <p><b>Тема 11. Множества</b> Множества. Множественный тип данных. Задание множеств. Описание множеств. Работа с элементами множеств.</p>

№ п/п	Трудоемкость (в часах)	Аннотации тем
		<p>Операции с множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству. Примеры решения задач.</p> <p><b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul> <p><b>Тема 12. Словари</b></p> <p>Словари. Тип словарь (dict). Когда нужно использовать словари. Создание словаря. Работа с элементами словаря. Перебор элементов словаря. Словарные операции. Словарные методы. Примеры решения задач.</p> <p><b>Необходимые умения:</b></p> <ul style="list-style-type: none"> <li>– работать в среде программирования на языке Python;</li> <li>– реализовывать построенные алгоритмы в виде программ на языке Python;</li> <li>– настраивать рабочую среду Python.</li> </ul> <p><b>Необходимые знания:</b></p> <ul style="list-style-type: none"> <li>– синтаксис языка Python, типы данных, функции, базовые конструкции;</li> <li>– инструментальные средства Python;</li> <li>– принципы объектно-ориентированного программирования.</li> </ul>
<b>ИТОГО</b>		<b>Общая трудоемкость образовательной программы – 24 часа</b>

**Разработчик:**

Руководитель образовательной программы

М.В. Леденёва