1. **Паспорт Образовательной программы**

**«** Основы создания игр, ботов, сайтов с использованием языка Python **»**

|  |  |
| --- | --- |
| **Версия программы** | **1** |
| **Дата Версии** | 01.10.2020 |

1. **Сведения о Провайдере**

|  |  |  |
| --- | --- | --- |
| 1.1 | Провайдер | ГАУ "Технопарк в сфере высоких технологий "ИТ-парк" |
| 1.2 | Логотип образовательной организации | C:\Users\User\Documents\ПЦС ЧИТ\Logo1.jpg |
| 1.3 | Провайдер ИНН | 1655191213 |
| 1.4 | Ответственный за программу ФИО | Агалакова Ольга Геннадьевна |
| 1.5 | Ответственный должность | Заместитель руководителя по коммерческим вопросам ГАУ "Технопарк в сфере высоких технологий "ИТ-Парк" |
| 1.6 | Ответственный Телефон | +79270378425 |
| 1.7 | Ответственный Е-mail | O.Agalakova@tatar.ru |

1. **Основные Данные**

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Описание** |
| 2.1 | Название программы | Основы создания игр, ботов, сайтов с использованием языка Python |
| 2.2 | Ссылка на страницу программы | https://edu.center2035.ru/local/crw/course.php?id=28 |
| 2.3 | Формат обучения | Онлайн |
|  | Подтверждение от ОО наличия возможности реализации образовательной программы с применением электронного обучения и (или) дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа | ГАУ "Технопарк в сфере высоких технологий "ИТ-парк" подтверждает возможность реализации образовательной программы с применением электронного обучения и дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа на LMS СДО платформе ЦОК НТИ, ссылка на платформу  https://edu.center2035.ru/ |
| 2.4 | Уровень сложности | Продвинутый |
| 2.5 | Количество академических часов | **72** |
|  | Практикоориентированный характер образовательной программы: не менее 50 % трудоёмкости учебной деятельности отведено практическим занятиям и (или) выполнению практических заданий в режиме самостоятельной работы (кол-во академических часов) | 51% трудоёмкости 37 часов  31 час учебной деятельности отведен практическим занятиям,  6 часов отведено на выполнение самостоятельной работы |
| 2.6 | Стоимость обучения одного обучающегося по образовательной программе, а также предоставление ссылок на 3 (три) аналогичные образовательные программы иных организаций, осуществляющих обучение, для оценки объективности стоимости или обоснование уникальности представленной образовательной программы в случае отсутствия аналогичных образовательных программ на рынке образовательных услуг | 30 000 рублей  Предлагаемая стоимость обучения: 30 000 рублей  Аналогичные программы:  1) https://skillfactory.ru/python-fullstack-web  2) https://skillbox.ru/course/python/  3) https://productstar.ru/dev-web-course?utm\_source=partner&utm\_medium=cpa&utm\_campaign=checkroi  4) https://netology.ru/programs/python#/  https://geekbrains.ru/professions/python\_developer |
| 2.7 | Минимальное количество человек на курсе | 20 |
| 2.8 | Максимальное количество человек на курсе | 100 |
| 2.9 | Данные о количестве слушателей, ранее успешно прошедших обучение по образовательной программе | 240 |
| 2.10 | Формы аттестации | Тестирование по результатам прохождения модуля, решение практико-ориентированных кейсов. Итоговое тестирование - зачет |
|  | Указание на область реализации компетенций цифровой экономики, к которой в большей степени относится образовательная программа, в соответствии с Перечнем областей | Программирование и создание ИТ-продуктов |

1. **Аннотация программы**

Наиболее полное и содержательное описание программы, которое включает:

1) общую характеристику компетенций, качественное изменение которых осуществляется в результате обучения или которые формируются в результате освоения образовательной программы;

2) описание требований и рекомендаций для обучения по образовательной программе;

3) краткое описание результатов обучения в свободной форме, а также описание востребованности результатов обучения в профессиональной деятельности.

Ограничение по размеру: не менее 1000 символов -?

Образовательный курс «Основы создания игр, ботов, сайтов с использованием языка Python» направлен на прохождение пути от основ программирования (переменные, циклы, ветвления) до построения готовых программ.

Программа курса включает четыре модуля:

- Основы программирования. Переменные, условия, циклы

- Создание игр на Python

- Создание telegram-бота

- Создание первого сайта на Django

Продолжительность 72 ак.ч.

Успешное завершение обучения позволит слушателям получить следующие навыки:

• Разработка программ на языке программирования Python как в процедурном, так и в объектно-ориентированном стиле

• Разработка системных скриптов с применением стандартных и сторонних библиотек (виртуальное окружение virtualenv)

• Работа с файлами, базой данных MySQL и SQLite

• Знание основ синтаксиса языка SQL и проектирования базы данных

• Применение IDE и полезных приемов автопроверки кода

• Знание синтаксиса Python

• Разработка модульных программ и веб-приложений

• Работа с внешними API

• Основы создания ботов

• Основы создания игр

Форма обучения - онлайн, с применением электронного обучения и дистанционных образовательных технологий на образовательной платформе Moodle. После окончания обучения и успешного прохождения тестирования выдается удостоверение о повышении квалификации.

1. ШАБЛОН ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ (ДПО)

Титульный лист программы

Название организации

Дополнительная профессиональная программа повышения квалификации

«Основы создания игр, ботов, сайтов с использованием языка Python »

72 час.

**ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ**

**1.Цель программы**

Целью реализации программы «Основы создания игр, ботов, сайтов с использованием языка Python» является прохождение пути от основ программирования (переменные, циклы, ветвления) до построения готовых программ. Образовательный курс направлен на формирование и развитие у обучающегося компетенций в области цифровой экономики "Программирование и создание ИТ-продуктов "

**2.Планируемые результаты обучения:**

2.1.Знание (осведомленность в областях)

2.1.1. Основ программирования

2.1.2. Методики работы с базами данных

2.1.3. Алгоритмов создания собственных приложений

2.1.4. Особенностей разработки на языке Python

2.2. Умение (способность к деятельности)

2.2.1 . Писать чистый код

2.2.2 . Создавать программы на Python

2.2.3. Работать с базами данных

2.2.4. Тестировать код

2.2.5. Работать с сетью

2.3.Навыки (использование конкретных инструментов)

2.3.1. Разработка программ на языке программирования Python как в процедурном, так и в объектно-ориентированном стиле

2.3.2. Разработка системных скриптов с применением стандартных и сторонних библиотек (виртуальное окружение virtualenv)

2.3.3. Знание основ синтаксиса языка SQL и проектирования базы данных

2.3.4. Применение IDE и полезных приемов авто-проверки кода

2.3.5. Знание синтаксиса Python

2.3.6. Разработка модульных программ и веб-приложений

2.3.7. Работа с внешними API

2.3.8. Основы создания ботов

2.3.9. Основы создания игр

**3.Категория слушателей** (возможно заполнение не всех полей)

* 1. Лица, имеющие среднее профессиональное и/или высшее образование, и лиц, получающих среднее профессиональное и/или высшее образование
  2. Квалификация не требуется
  3. Наличие опыта профессиональной деятельности не требуется
  4. Пользователь ПК

**4.Учебный план программы «…..наименование программы….»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль** | **Всего, час** | **Виды учебных занятий** | | |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Основы программирования. Переменные, условия, циклы | 44 | 23 | 20 | 1 |
| 2 | Создание игр на Python | 8 | 4 | 3 | 1 |
| 3 | Создание telegram-бота | 8 | 4 | 3 | 1 |
| 4 | Создание первого сайта на Django | 10 | 4 | 5 | 1 |
| 5 | Итоговое тестирование | 2 |  |  | 2 |
|  | Итого по всей программе | 72 | 35 | 31 | 6 |
| **Итоговая аттестация** | |  | **Указывается вид (экзамен, зачёт, реферат и т.д.)** | | |
| Программирование на языке Python | | 2 | Зачет в форме выходного тестирования | | |

**5.Календарный план-график реализации образовательной** программы

(дата начала обучения – дата завершения обучения) в текущем календарном году, указания на периодичность набора групп (не менее 1 группы в месяц)

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование учебных модулей** | **Трудоёмкость (час)** | **Сроки обучения** |
| **1** | Основы программирования. Переменные, условия, циклы | 44 | 01.11-10.11.2020 |
| **2** | Создание игр на Python | 8 | 10.11-11.11.2020 |
| 3 | Создание telegram-бота | 8 | 11.11-12.11.2020 |
| 4 | Создание первого сайта на Django | 10 | 13.11-14.11.2020 |
| 5 | Итоговая аттестация (тестирование) | 2 | 15.11.2020 |
| **Всего:** | | 72 | 01.11-05.11.2020 |

**6.Учебно-тематический план программы «**  Основы создания игр, ботов, сайтов с использованием языка Python **»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль / Тема** | **Всего, час** | **Виды учебных занятий** | | | **Формы контроля** |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Основы программирования. Переменные, условия, циклы | 44 | 23 | 20 | 1 | Тестирование, кейсы, самостоятельная работа |
| 1.1 | Установка Python, вычисления и переменные | 3 | 2 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.2 | Строки, списки, кортежи, словари | 3 | 1 | 2 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.3 | Работа с SQL базами данных | 3 | 1 | 2 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.4 | Ядро языка Python | 3 | 2 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.5 | Стандартная библиотека | 3 | 2 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.6 | Система контроля версий и система управления задачами | 3 | 2 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.7 | Модуль «Черепашка» | 4 | 2 | 2 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.8 | Условия. IF - ELSE | 3 | 2 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.9 | Изучение циклов | 3 | 1 | 2 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.10 | Функции и модули | 2 | 1 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.11 | Классы и объекты | 3 | 1 | 2 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.12 | Встроенные функции и полезные модули | 5 | 3 | 2 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.13 | Структуры и перечисления. Интерфейсы. | 2 | 1 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 1.14 | Модуль «Tkinter» | 3 | 2 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
|  | Тестирование Модуль 1. | 1 |  |  | 1 | Итоговый тест по модулю 1  1 час - 10 вопросов (примеры тестов в п. 8.4) |
| 2 | Создание игр на Python | 8 | 4 | 3 | 1 | Тестирование, кейсы, самостоятельная работа |
| 2.1 | Первая игра на Python | 2 | 1 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 2.2 | Дорабатываем игру | 3 | 2 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
| 2.3 | Исправление ошибок | 2 | 1 | 1 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
|  | Тестирование Модуль 2. | 1 |  |  | 1 | Итоговый тест по модулю 2  1 час - 10 вопросов (примеры тестов в п. 8.4) |
| 3 | Создание telegram-бота | 8 | 4 | 3 | 1 | Тестирование, кейсы, самостоятельная работа |
| 3.1 | Создание telegram-бота | 8 | 4 | 3 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
|  | Тестирование Модуль 3. | 1 |  |  | 1 | Итоговый тест по модулю 3  1 час - 10 вопросов (примеры тестов в п. 8.4) |
| 4 | Создание первого сайта на Django | 10 | 4 | 5 | 1 | Тестирование, кейсы, самостоятельная работа |
| 4.1 | Создание первого сайта на Django | 10 | 4 | 5 |  | Разбор кейсов, практические занятия, изучение лекционного материала - присутствие СДО (п. 8.4) |
|  | Тестирование Модуль 4. | 1 |  |  | 1 | Итоговый тест по модулю 4  1 час - 10 вопросов (примеры тестов в п. 8.4) |
|  | Итоговый тест по всем модулям программы. | 2 |  |  | 2 | Итоговый тест по всей программе 2 часа - 20 вопросов  (примеры вопросов в п. 8.4) |

**7. Учебная (рабочая) программа повышения квалификации «**  Основы создания игр, ботов, сайтов с использованием языка Python  **»**

Модуль 1. Основы программирования. Переменные, условия, циклы (44 часа, из них: 23 часа лекций, 20 часов практических занятий, 1 час самостоятельной работы)

Тема 1.1. Установка Python, вычисления и переменные (3 часа, 2 часа лекций, 1 час практических занятий)

• Установка Python

• Сохранение Python-программ

• Вычисления в Python

• Переменные как ярлыки для данных

• Использование переменных

Тема 1.2. Строки, списки, кортежи, словари (3 часа, 1 час лекций, 2 часа практических занятий)

• Строки

• Списки мощнее строк

• Кортежи

• Словари в Python — не для поиска слов

Тема 1.3. Работа с SQL базами данных (3 часа, 1 час лекций, 2 часа практических занятий)

• Работа с SQL базами данных

Тема 1.4. Ядро языка Python (3 часа, 2 часа лекций, 1 час практических занятий)

• Синтаксис языка, управляющие конструкции

• Объектно-ориентированное программирование

Тема 1.5. Стандартная библиотека языка (3 часа, 2 часа лекций, 1 час практических занятий)

• Обзор стандартной библиотеки

Тема 1.6. Система контроля версий и система управления задачами (3 часа, 2 часа лекций, 1 час практических занятий)

• Система контроля версий и система управления задачами

Тема 1.7. Модуль «Черепашка» (4 часа, 2 часа лекций, 2 часа практических занятий)

• Использование модуля черепашки

Тема 1.8. Условия. IF - ELSE (3 часа, 2 часа лекций, 1 час практических занятий)

• Конструкция if

• Конструкция if-then-else

• Команды if и elif

• Объединение условий

Тема 1.9. Изучение циклов (3 часа, 1 час лекций, 2 часа практических занятий)

• Использование цикла for

• Цикл while

Тема 1.10. Функции и модули (2 часа, 1 час лекций, 1 час практических занятий)

• Применение функций

• Применение модулей

Тема 1.11. Классы и объекты (3 часа, 1 час лекций, 2 часа практических занятий)

• Разделяем сущности на классы

• Другие полезные свойства объектов и классов

• Инициализация объектов

Тема 1.12. Встроенные функции и полезные модули (5 часов, 3 часа лекций, 2 часа практических занятий)

• Использование встроенных функций

• Работа с файлами

Тема 1.13. Структуры и перечисления. Интерфейсы (2 часа, 1 час лекций, 1 час практических занятий)

• Структуры и перечисления. Интерфейсы

Тема 1.14. Модуль «Tkinter» (3 часа, 2 часа лекций, 1 час практических занятий)

• Создаем кнопку

• Создаем холст для рисования

• Рисование элементов

• Отображение текста

• Вывод изображений

• Создание простой анимации

1 час самостоятельной работы на межмодульный тест для закрепления полученных знаний и навыков - 10 вопросов (примеры вопросов в п. 8.4)

Модуль 2. Создание игр на Python (8 часов, из них: 4 часа лекций, 3 часа практических занятий, 1 час самостоятельной работы)

Тема 2.1. Первая игра на Python (2 часа, 1 час лекций, 1 час практических занятий)

• Прыгающий мяч

• Создаем игровой холст

• Создаем класс для мяча

• Добавим движение

Тема 2.2. Дорабатываем игру (3 часа, 2 часа лекций, 1 час практических занятий)

• Создаем ракетку

• Добавим возможность проигрыша

Тема 2.3. Исправление ошибок (2 часа, 1 час лекций, 1 час практических занятий)

• Исправляем недочеты

• Реализуем свои идеи

1 час самостоятельной работы на межмодульный тест для закрепления полученных знаний и навыков - 10 вопросов (примеры вопросов в п. 8.4)

Модуль 3. Создание telegram-бота (8 часов, из них: 4 часа лекций, 3 часа практических занятий, 1 час самостоятельной работы)

Тема 3.1. Создание telegram-бота (7 часов, 4 часа лекций, 3 часа практических занятий)

• Создание telegram-бота

1 час самостоятельной работы на межмодульный тест для закрепления полученных знаний и навыков - 10 вопросов (примеры вопросов в п. 8.4)

Модуль 4. Создание первого сайта на Django (10 часов, из них: 4 часа лекций, 5 часов практических занятий, 1 час самостоятельной работы)

Тема 4.1. Создание первого сайта на Django (4 часа лекций, 5 часов практических занятий)

• Создание первого сайта на Django

1 час самостоятельной работы на межмодульный тест для закрепления полученных знаний и навыков - 10 вопросов (примеры вопросов в п. 8.4)

Итоговый тест по всем модулям программы - 20 вопросов (примеры в п. 8.4)

**Описание практико-ориентированных заданий и кейсов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Номер темы/модуля** | **Наименование практического занятия** | **Описание** |
| 1 | 1.10 | Практическая работа №1 «Простейшие арифметические операции» | Написать функцию arithmetic, принимающую 3 аргумента: первые 2 - числа, третий - операция, которая должна быть произведена над ними. Если третий аргумент +, сложить их; если —, то вычесть; \* — умножить; / — разделить (первое на второе). В остальных случаях вернуть строку "Неизвестная операция". |
| 2 | 1.11 | Практическая работа №2 «Високосный год» | Написать функцию is\_year\_leap, принимающую 1 аргумент — год, и возвращающую True, если год високосный, и False иначе. |
| 3 | 1.12 | Практическая работа №3 «Квадрат» | Написать функцию square, принимающую 1 аргумент — сторону квадрата, и возвращающую 3 значения (с помощью кортежа): периметр квадрата, площадь квадрата и диагональ квадрата. |
| 4 | 1.13 | Практическая работа №4 «Времена года» | Написать функцию season, принимающую 1 аргумент — номер месяца (от 1 до 12), и возвращающую время года, которому этот месяц принадлежит (зима, весна, лето или осень). |
| 5 | 2.1 | Практическая работа №5 «Банковский вклад» | Пользователь делает вклад в размере a рублей сроком на years лет под 10% годовых (каждый год размер его вклада увеличивается на 10%. Эти деньги прибавляются к сумме вклада, и на них в следующем году тоже будут проценты).  Написать функцию bank, принимающая аргументы a и years, и возвращающую сумму, которая будет на счету пользователя. |
| 6 | 2.2 | Практическая работа №6 «Простые числа» | Написать функцию is\_prime, принимающую 1 аргумент — число от 0 до 1000, и возвращающую True, если оно простое, и False - иначе. |
| 7 | 3.1 | Практическая работа №7 «Правильная дата» | Написать функцию date, принимающую 3 аргумента — день, месяц и год. Вернуть True, если такая дата есть в нашем календаре, и False иначе. |
| 8 | 4.1 | Практическая работа №8 «XOR-шифрование» | Написать функцию XOR\_cipher, принимающая 2 аргумента: строку, которую нужно зашифровать, и ключ шифрования, которая возвращает строку, зашифрованную путем применения функции XOR (^) над символами строки с ключом. Написать также функцию XOR\_uncipher, которая по зашифрованной строке и ключу восстанавливает исходную строку. |

**8.Оценочные материалы по образовательной программе**

**8.1. Вопросы тестирования по модулям**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ модуля** | **Вопросы входного тестирования** | **Вопросы промежуточного тестирования** | **Вопросы итогового тестирования** |
| 1.2 | Алгоритм – это  1.1. описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов  1.2. протокол вычислительной сети  1.3. набор команд для компьютера | В чем разница между списками (list) и кортежами (tuple)? | В чем разница между списками (list) и кортежами (tuple)? |
| 1.2 | Алгоритм включает в себя ветвление, если  2.1. он представим в табличной форме  2.2. он включает в себя вспомогательный алгоритм  2.3. ход его выполнения зависит от истинности тех или иных условий | Как в питоне работает трёхместный (тернарный) оператор? | Как в питоне работает трёхместный (тернарный) оператор? |
| 1.2 | 3. Алгоритм называется циклическим, если  3.1. он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий  3.2. он представим в табличной форме  3.3. ход его выполнения зависит от истинности тех или иных условий | Что такое отрицательный индекс (negative index)? | Что такое отрицательный индекс (negative index)? |
| 1.1 | Алгоритм, записанный на «понятном» компьютеру языке программирования, называется  4.1. исполнителем алгоритмов  4.2. программой  4.3. протоколом алгоритма | Питон чувствителен к регистру? | Питон чувствителен к регистру? |
| 1.2 | Что из перечисленного не является языком программирования?  5.1. Python  5.2. HTML  5.3. Java | Как можно преобразовать строку (string) в нижний регистр (lowercase)? | Как можно преобразовать строку (string) в нижний регистр (lowercase)? |
| 1.4 | Логический тип данных можешь принимать следующие значения:  6.1. Истина или Ложь  6.2. Целые числа  6.3. Строки | Для чего нужен pass (pass statement) в питоне? | Для чего нужен pass (pass statement) в питоне? |
| 1.2 | Какие элементы данных могут изменять свое значение в ходе выполнения программы?  7.1. переменные  7.2. константы  7.3. комментарии | Что такое срез? | Что такое срез? |
| 1.1 | Как называется процесс перестановки элементов массива с целью упорядочивания их в соответствии с каким-либо критерием?  8.1. поиск  8.2. перебор  8.3. сортировка | Как пишутся комментарии в питоне? | Как пишутся комментарии в питоне? |
| 1.2 | В каком случае необходимо использовать цикл?  9.1. Вычисление суммы всех чисел от 1 до 100  9.2. Вычисление корня квадратного уравнения  9.3. суммы двух чисел, введенных с клавиатуры | Как вставить объект, чтобы он оказался под определенным индексом? | Как вставить объект, чтобы он оказался под определенным индексом? |
| 1.10 | Наименьшая единица измерения информации?  10.1. Бит.  10.2. Байт.  10.3. Кбайт. | Что нужно сделать, чтобы функция возвратила значение? | Что нужно сделать, чтобы функция возвратила значение? |
| 1.9 |  |  | Зачем нужны break и continue? |
| 1.2 |  |  | Как конвертировать список в строку? |
| 1.2 |  |  | Что такое словарь (dictionary)? |
| 1.4 |  |  | Что вам известно про операторы сравнения в питоне? |
| 1.1 |  |  | Какие типы данных поддерживаются в питоне? |
| 1.4 |  |  | Назовите и поясните два основных аспекта объектно-ориентированного программирования. |
| 1.11 |  |  | Поля, методы, атрибуты - дайте характеристику. |
| 1.11 |  |  | Напишите синтаксис создания класса в языке Python. |
| 1.11 |  |  | Какой синтаксис используется при обращении к атрибуту класса? |
| 1.11 |  |  | Чем методы класса отличаются от обычных функций? |
| 1.11 |  |  | Поясните роль параметра self. |
| 1.11 |  |  | Какой синтаксис используется при обращении к методу класса? |
| 1.11 |  |  | С какой целью создается метод init ? Напишите его синтаксис. |
| 1.11 |  |  | Объясните роль статических методов языка Python. Какие методы объявления статических методов вы знаете? |
| 1.11 |  |  | В чем заключается такой принцип ООП, как инкапсуляция? |
| 1.11 |  |  | Расскажите о методах создания закрытых атрибутов и способах доступа к ним. |
| 1.12 |  |  | С какой целью создаются свойства, и как происходит обращение к ним из клиентского кода? |
| 1.11 |  |  | Раскройте особенности одного из основных принципов ООП - наследования. Приведите синтаксис создания производного класса. |
| 1.2 |  |  | Приведите примеры преобразования типов в Python (списки, строки). |
| 1.12 |  |  | Опишите возможности применения вложенных списков в Python. |
| 1.9 |  |  | Какие циклы существуют в Python? |
| 1.9 |  |  | В каких случаях используется функция range в Python? |
| 1.9 |  |  | В каких случаях применяется цикл while в Python? |
| 1.13 |  |  | Как происходит обработка исключений в Python? |
| 1.14 |  |  | Что такое шаблон «Модель-вид-контроллер» (на примере модуля tkinter) |

**8.2.**  Формой аттестации является тестирование на выходе, определяющего оценку слушателя результата освоения образовательной программы. Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля (аттестации) - в 50 баллов. Для успешного прохождения курса и получения оценки «зачтено» необходимо набрать не менее 56 баллов. В остальных случаях предусматривается оценка «не зачтено». Оценка «зачтено» дифференцируется по 4-уровневой системе знаний в зависимости от количества набранных баллов.

Таблица оценки уровня знаний по итоговому тестированию

|  |  |
| --- | --- |
| Общее количество набранных баллов | Уровень знаний |
| 86-100 баллов | Зачтено - отлично |
| 71-85 баллов | Зачтено - хорошо |
| 56-70 баллов | Зачтено - удовлетворительно |
| 0-55 баллов | Не зачтено |

**.**

**8.3.**  Примеры контрольных заданий по модулям или всей образовательной программе:

1. Простейшие арифметические операции

Написать функцию arithmetic, принимающую 3 аргумента: первые 2 - числа, третий - операция, которая должна быть произведена над ними. Если третий аргумент +, сложить их; если —, то вычесть; \* — умножить; / — разделить (первое на второе). В остальных случаях вернуть строку "Неизвестная операция".

2. Високосный год

Написать функцию is\_year\_leap, принимающую 1 аргумент — год, и возвращающую True, если год високосный, и False иначе.

3. Квадрат

Написать функцию square, принимающую 1 аргумент — сторону квадрата, и возвращающую 3 значения (с помощью кортежа): периметр квадрата, площадь квадрата и диагональ квадрата.

4. Времена года

Написать функцию season, принимающую 1 аргумент — номер месяца (от 1 до 12), и возвращающую время года, которому этот месяц принадлежит (зима, весна, лето или осень).

5. Банковский вклад

Пользователь делает вклад в размере a рублей сроком на years лет под 10% годовых (каждый год размер его вклада увеличивается на 10%. Эти деньги прибавляются к сумме вклада, и на них в следующем году тоже будут проценты).

Написать функцию bank, принимающая аргументы a и years, и возвращающую сумму, которая будет на счету пользователя.

6. Простые числа

Написать функцию is\_prime, принимающую 1 аргумент — число от 0 до 1000, и возвращающую True, если оно простое, и False - иначе.

7. Правильная дата

Написать функцию date, принимающую 3 аргумента — день, месяц и год. Вернуть True, если такая дата есть в нашем календаре, и False иначе.

8. XOR-шифрование

Написать функцию XOR\_cipher, принимающая 2 аргумента: строку, которую нужно зашифровать, и ключ шифрования, которая возвращает строку, зашифрованную путем применения функции XOR (^) над символами строки с ключом. Написать также функцию XOR\_uncipher, которая по зашифрованной строке и ключу восстанавливает исходную строку. **.**

**8.4.**  **тесты и обучающие задачи (кейсы), иные практикоориентированные формы заданий** Входное тестирование:

Алгоритм – это

1.1. описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов

1.2. протокол вычислительной сети

1.3. набор команд для компьютера

Алгоритм включает в себя ветвление, если

2.1. он представим в табличной форме

2.2. он включает в себя вспомогательный алгоритм

2.3. ход его выполнения зависит от истинности тех или иных условий

3. Алгоритм называется циклическим, если

3.1. он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий

3.2. он представим в табличной форме

3.3. ход его выполнения зависит от истинности тех или иных условий

Алгоритм, записанный на «понятном» компьютеру языке программирования, называется

4.1. исполнителем алгоритмов

4.2. программой

4.3. протоколом алгоритма

Что из перечисленного не является языком программирования?

5.1. Python

5.2. HTML

5.3. Java

Логический тип данных можешь принимать следующие значения:

6.1. Истина или Ложь

6.2. Целые числа

6.3. Строки

Какие элементы данных могут изменять свое значение в ходе выполнения программы?

7.1. переменные

7.2. константы

7.3. комментарии

Как называется процесс перестановки элементов массива с целью упорядочивания их в соответствии с каким-либо критерием?

8.1. поиск

8.2. перебор

8.3. сортировка

В каком случае необходимо использовать цикл?

9.1. Вычисление суммы всех чисел от 1 до 100

9.2. Вычисление корня квадратного уравнения

9.3. суммы двух чисел, введенных с клавиатуры

Наименьшая единица измерения информации?

10.1. Бит.

10.2. Байт.

10.3. Кбайт.

Вопросы по промежуточному тестированию:

В чем разница между списками (list) и кортежами (tuple)?

Как в питоне работает трёхместный (тернарный) оператор?

Что такое отрицательный индекс (negative index)?

Питон чувствителен к регистру?

Примерные вопросы контрольных тестов при тестировании на выходе:

1. В чем разница между списками (list) и кортежами (tuple)?

2. Как в питоне работает трёхместный (тернарный) оператор?

3. Что такое отрицательный индекс (negative index)?

4. Питон чувствителен к регистру?

5. Как можно преобразовать строку (string) в нижний регистр (lowercase)?

6. Для чего нужен pass (pass statement) в питоне?

7. Что такое срез?

8. Как пишутся комментарии в питоне?

9. Как вставить объект, чтобы он оказался под определенным индексом?

10. Что нужно сделать, чтобы функция возвратила значение?

11. Зачем нужны break и continue?

12. Как конвертировать список в строку?

13. Что такое словарь (dictionary)?

14. Что вам известно про операторы сравнения в питоне?

15. Какие типы данных поддерживаются в питоне?

16. Назовите и поясните два основных аспекта объектно-ориентированного программирования.

17. Поля, методы, атрибуты - дайте характеристику.

18. Напишите синтаксис создания класса в языке Python.

19. Какой синтаксис используется при обращении к атрибуту класса?

20. Чем методы класса отличаются от обычных функций?

21. Поясните роль параметра self.

22. Какой синтаксис используется при обращении к методу класса?

23. С какой целью создается метод init ? Напишите его синтаксис.

24. Объясните роль статических методов языка Python. Какие методы объявления статических методов вы знаете?

25. В чем заключается такой принцип ООП, как инкапсуляция?

26. Расскажите о методах создания закрытых атрибутов и способах доступа к ним.

27. С какой целью создаются свойства, и как происходит обращение к ним из клиентского кода?

28. Раскройте особенности одного из основных принципов ООП - наследования. Приведите синтаксис создания производного класса.

29. Приведите примеры преобразования типов в Python (списки, строки).

30. Опишите возможности применения вложенных списков в Python.

31. Какие циклы существуют в Python?

32. В каких случаях используется функция range в Python?

33. В каких случаях применяется цикл while в Python?

34. Как происходит обработка исключений в Python?

35. Что такое шаблон «Модель-вид-контроллер» (на примере модуля tkinter)?

**.**

**8.5.**  Тестирование на выходе включает в себя вопросы, направленные как на контроль знаний, так и на проверку полученных навыков работы. Практическое задание представляет из себя задачу написания тестовой документации, а также применения навыков разработки такой документации. Результаты оцениваются по определённым заранее критериям.

При проведении тестирования с использованием единой информационной среды с электронными формами контроля и оценки у каждого слушателя есть три попытки на прохождение тестирования. Время на одну попытку - 120 минут.

Лицам, не прошедшим итоговую аттестацию, или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть программы повышения квалификации и (или) отчисленным из организации, выдается справка об обучении или о периоде обучения, по установленному образцу. **.**

**9.Организационно-педагогические условия реализации программы**

**9.1. Кадровое обеспечение программы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№**  **п/п** | **Фамилия, имя, отчество (при наличии)** | **Место основной работы и должность, ученая степень и ученое звание (при наличии)** | **Ссылки на веб-страницы с портфолио (при наличии)** | **Фото в формате jpeg** | **Отметка о полученном согласии на обработку персональных данных** |
| **1** | Мингалиев Раиль Рафаилович | ООО "ДН.РУ", инженер отдела разработки, совместительство – ГАУ «Технопарк в сфере высоких технологий ГАУ «ИТ-парк», ведущий консультант по обучению | https://drive.google.com/drive/folders/1Lloi-7FrnavxqouIrjvneOZSTrNY-BrO?usp=sharing |  | да |
| **2** | **Фамилия Имя Отчество** |  |  |  |  |

**9.2.Учебно-методическое обеспечение и информационное сопровождение**

|  |  |
| --- | --- |
| **Учебно-методические материалы** | |
| Методы, формы и технологии | Методические разработки,  материалы курса, учебная литература |
|  | Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с. |
|  | Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с. |
|  | Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с. |
|  | Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с. |
|  | Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с. |
|  | Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – 768 с. |
|  | Свейгарт, Эл. Автоматизация рутиных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямc, 2016. – 592 с. |
|  | Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O’Reilly»). |
|  | Любанович Билл Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.: – (Серия «Бестсепперы O’Reilly»). |
|  | Доусон М. Программируем на Python. – СПб.: Питер, 2014. – 416 с.  Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. – СПб.: БХВ-Петербург, 2012. – 704 с. |
|  | Пилгрим Марк. Погружение в Python 3 (Dive into Python 3 на русском) |
|  | Прохоренок Н.А. Самое необходимое. — СПб.: БХВ-Петербург, 2011. — 416 с. |

|  |  |
| --- | --- |
| **Информационное сопровождение** | |
| Электронные  образовательные ресурсы | Электронные  информационные ресурсы |
| Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: https://urait.ru/bcode/437489 (дата обращения: 13.02.2020) | Шелудько, В. М. Основы программирования на языке высокого уровня Python: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 146 c. – ISBN 978-5-9275-2649-9. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: http://www.iprbookshop.ru/87461.html (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей |
| Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 c. – ISBN 978-5-9275-2648-2. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: http://www.iprbookshop.ru/87530.html (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей |  |

**9.3.Материально-технические условия реализации программы**

|  |  |
| --- | --- |
| Вид занятий | Наименование оборудования,  программного обеспечения |
| Лекции с демонстрацией экрана | образовательная платформа Moodle |
| Лекции с обратной связью со слушателями | платформа Zoom |
| Практические занятия | платформа Moodle, IDE PyChram |

**III.Паспорт компетенций (Приложение 2)**

Описание перечня профессиональных компетенций, качественное изменение которых осуществляется в результате обучения.

Планируемые результаты обучения должны быть определены в виде знаний, умений, навыков и (или) опыта деятельности, характеризующих формирование/развитие компетенции(-й) в области цифровой экономики и представлены в виде Паспорта компетенций в машиночитаемом текстовом формате. Структура паспорта представлена в приложении.

ПАСПОРТ КОМПЕТЕНЦИИ

Основы создания игр, ботов, сайтов с использованием языка Python

Государственное автономное учреждение «Технопарк в сфере высоких технологий «ИТ-парк», осуществляющее образовательную деятельность на основании лицензии от 05.05.2015г. №6335, выданной Министерством образования и науки Республики Татарстан

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Наименование компетенции | | Знание инструментария | |
| 2. | Указание типа компетенции | общекультурная/  универсальная |  | |
| общепрофессиональная |  | |
| профессиональная | ПК-1 | |
| профессионально-специализированная |  | |
| 3. | Определение, содержание и основные сущностные характеристики компетенции | | Знание инструментария, умение работать с кодом. | |
| 4. | Дескриптор знаний, умений и навыков по уровням | | Уровни сформированности компетенции  обучающегося | Индикаторы |
|  | | Начальный уровень | Ограничены используемой IDE для редактирования текста. Использование односложных имен. |
|  | | Базовый уровень | Знает о некоторых альтернативах популярным стандартным инструментариям. Использует хорошие имена файлов, переменных, классов, методов и т.д. |
|  | | Продвинутый | Хорошие знания редакторов кода, отладчиков, различных IDE, использует горячие клавиши. Нет длинных функций и нестандартный код. |
|  | | Профессиональный | Написал и опубликовал свои инструментарии и макросы. Поток операций в коде естественный, нет глубокой вложенности условий или методов. |
| 5. | Характеристика взаимосвязи данной компетенции с другими компетенциями/ необходимость владения другими компетенциями для формирования данной компетенции | | Для дальнейшего повышения уровня овладения инструментарием необходимо освоение дополнительных компетенций по использованию возможностей интегрированной среды разработки | |
| 6. | Средства и технологии оценки | | Выполнение и проверка практического задания. Банк тестовых заданий. Контрольное тестирование | |
| 1. | Наименование компетенции | | Системное программирование | |
| 2. | Указание типа компетенции | общекультурная/  универсальная |  | |
| общепрофессиональная |  | |
| профессиональная | ПК-1 | |
| профессионально-специализированная |  | |
| 3. | Определение, содержание и основные сущностные характеристики компетенции | | Системное программирование | |
| 4. | Дескриптор знаний, умений и навыков по уровням | | Уровни сформированности компетенции  обучающегося | Индикаторы |
|  | | Начальный уровень | Не знает, что такое компилятор, линковщик или интерпретатор. |
|  | | Базовый уровень | Базовое понимание интерпретаторов. Понимает, что такое ассемблерный код и как работают программы на уровне железа. Небольшое понимание виртуальной памяти и пэйджинга. |
|  | | Продвинутый | Применяет способы синхронизации, может читать ассемблерный код. Понимает, как работают сети и сетевые протоколы. Работа в объектно-ориентированном стиле |
|  | | Профессиональный | Понимает, как работает весь "программный стэк": железо (CPU + Память + Кэш + Прерывания + микрокоды), двоичный код, интерпретация, адресация памяти. |
| 5. | Характеристика взаимосвязи данной компетенции с другими компетенциями/ необходимость владения другими компетенциями для формирования данной компетенции | | Для дальнейшего повышения уровня овладения системным программированием необходимо освоение дополнительных компетенций по использованию возможностей алгоритмов | |
| 6. | Средства и технологии оценки | | Выполнение и проверка практического задания. Банк тестовых заданий. Контрольное тестирование | |
| 1. | Наименование компетенции | | Разработка приложений | |
| 2. | Указание типа компетенции | общекультурная/  универсальная |  | |
| общепрофессиональная |  | |
| профессиональная | ОПК-1 | |
| профессионально-специализированная |  | |
| 3. | Определение, содержание и основные сущностные характеристики компетенции | | Разработка приложений | |
| 4. | Дескриптор знаний, умений и навыков по уровням | | Уровни сформированности компетенции  обучающегося | Индикаторы |
|  | | Начальный уровень | Осведомленность об основах изученного языка программирования. |
|  | | Базовый уровень | Проектирование и разработка графического интерфейса пользователя. Разработка первого приложения с использованием изученного языка программирования |
|  | | Продвинутый | Разработка приложения с использованием изученного языка программирования в объектно-ориентированном стиле |
|  | | Профессиональный | Освоение цифровой компетенции на профессиональном уровне. Проектировал и реализовывал несколько приложений, хорошо разбирается в ее сущностях и протоколах |
| 5. | Характеристика взаимосвязи данной компетенции с другими компетенциями/ необходимость владения другими компетенциями для формирования данной компетенции | | Для дальнейшего повышения уровня овладения инструментарием необходимо освоение дополнительных компетенций по использованию возможностей языка программирования | |
| 6. | Средства и технологии оценки | | Выполнение и проверка практического задания. Банк тестовых заданий. Контрольное тестирование | |

**VI.Иная информация о качестве и востребованности образовательной программы** (результаты профессионально-общественной аккредитации образовательной программы, включение в системы рейтингования, призовые места по результатам проведения конкурсов образовательных программ и др.) (при наличии)

нет

**V.Рекомендаций к программе от работодателей**: наличие не менее двух писем и/или подтверждения на цифровой платформе Государственной системы предоставления ПЦС от работодателей о рекомендации образовательной программы для реализации в рамках Государственной системы предоставления ПЦС на формирование у трудоспособного населения компетенций цифровой экономики с указанием востребованности результатов освоения программы в сфере деятельности соответствующих компаний и готовности к рассмотрению заявок наиболее успешно освоивших образовательную программу граждан на прохождение стажировки и (или) собеседования на предмет трудоустройства путем проставления отметки в профиле программы

Получены рекомендательные письма от организаций

ООО "АЙкью 300"

ИП Ярмолавичус Сергей Юрьевич

**VI.Указание на возможные сценарии профессиональной траектории граждан** по итогам освоения образовательной программы (в соответствии с приложением)

После прохождения курса и получения персонального цифрового сертификата возможны следующие сценарии профессиональной деятельности:

1) трудоустройство или переход в новую сферу деятельности, с целью повышения уровня дохода и расширения профессиональной деятельности, а также возможность регистрации в качестве самозанятого и/ или открыть собственное дело;

2) развитие компетенций в текущей сфере занятости, с целью освоения смежных профессиональных областей, повышения уровня дохода, расширения профессиональной деятельности, сохранения текущего рабочего места или смены работы.

**VII.Дополнительная информация**

Программу читаетквалифицированный преподаватель - Мингалиев Раиль Рафаилович, основное место работы ООО "ДН.РУ" - ведущий инженер отдела разработки; совместительство – ГАУ «Технопарк в сфере высоких технологий ГАУ «ИТ-парк», ведущий консультант по обучению, имеющий опыт преподавательской деятельности, верификации программного обеспечения; знания основных концепций и иной теоретической базы по данному курсу, владеющий методикой, разноуровневого личностно-ориентированного и интенсивного обучения.

**VIII.Приложенные Скан-копии**

Утвержденной рабочей программа (подпись, печать, в формате pdf)